SOME DEVELOPMENTS OF SOFT COMPUTING METHODS FOR TSP UNDER UNCERTAIN OPTIMIZATION PARADIGMS

Thesis submitted to the

VIDYASAGAR UNIVERSITY

For the Award of Degree of

DOCTOR OF PHILOSOPHY

IN

SCIENCE

BY

SAMIR MAITY

Department of Computer Science

Vidyasagar University

Midnapore-721102, INDIA

SEPTEMBER, 2016

***D*edicated to the great philanthropist**

Jean-Paul Sartre

CERTIFICATE

This is to certify that the thesis entitled -**"SOME DEVELOPMENTS OF SOFT COMPUTING METHODS FOR TSP UNDER UNCERTAIN OPTIMIZATION PARADIGMS"** submitted by **Sri Samir Maity** for the award of degree of **DOCTOR OF PHILOSOPHY IN SCIENCE** to the **Vidyasagar University, Midnapore** is a record of bonafide research work carried out by him under our guidance and supervision. **Sri Maity** has worked in the Department of **Computer Science, Vidyasagar University** as per the regulations of this University.

In our opinion, this thesis is of the standard required for the award of the degree of **DOCTOR OF PHILOSOPHY IN SCIENCE**.

The results, embodied in this thesis, have not been submitted to any other University or Institution for the award of any degree or diploma.

Dr. Arindam Roy
Assistant Professor
Dept. of Computer Science and Applications,
Prabhat Kumar College,
Contai-721401,
INDIA

Prof. Manoranjan Maiti
Former Professor
Dept. of Applied Mathematics with Oceanology and Computer Programming,
Vidyasagar University,
Midnapore-721102,
INDIA

.

DECLARATION

I, Samir Maity, do hereby declare that, I have not submitted the results embodied in my thesis- **"SOME DEVELOPMENTS OF SOFT COMPUTING METHODS FOR TSP UNDER UNCERTAIN OPTIMIZATION PARADIGMS"** or a part of it for any degree/diploma or any other academic award anywhere before.

Samir Maity
(**Reg. No:366/Ph.D./Sc. 03-04-2013**)
Department of Computer Science,
Vidyasagar University
Midnapore, West Bengal, India

Date:

.

ACKNOWLEDGEMENT

This work never would have been started, much less finished, without two supervisors- Prof. Manoranjan Maiti and Dr. Arindam Roy. Despite my own uncertainties, they always expressed confidence that I could and would discover something interesting. Specially to spend the time with honorable Prof. Maiti is a memorable journey of my ordinary life. Personally to share and cheer the views and ides with him beyond the subjects born me again a new human being. I owe them many thanks for always ready accessibility, wise advice, excellent teaching, and principled advising.

A graceful thanks to the writers of some books in Bengali literature, critics and also from my best subjects History, Philosophy, Psychology and Sociology, that have given me ideas and breaks from the day to day life obstructions.

Lastly I confer special thanks from deep of my heart to all of my very closed neighbours.

Samir Maity
(Reg. No:366/Ph.D./Sc. 03-04-2013)
Department of Computer Science,
Vidyasagar University
Midnapore, West Bengal, India

Date:

.

PUBLICATION

The list of Published Papers:

1. S. Maity, A. Roy, M. Maiti, *A Modified Genetic Algorithm for solving uncertain Constrained Solid Travelling Salesman Problems,* **Computers & Industrial Engineering**, **(Elsevier), 83**(2015), 273-296, **(SCI),** Impact Factor 2.082.
2. S. Maity, A. Roy, M. Maiti, *An imprecise Multi-Objective Genetic Algorithm for uncertain Constrained Multi-Objective Solid Travelling Salesman Problem,* **Expert Systems With Applications**, **(Elsevier), 46**(2016), 196-223, **(SCI),** Impact Factor 2.986.
3. S. Maity, A. Roy, M. Maiti, *Constrained Solid Travelling Salesman Problem Solving by Rough GA Under Bi-Fuzzy Coefficients,* **Advances in Intelligent Systems and Computing 404**, (**Springer**), DOI 10.1007/978-81-322-2695-6-36, (2015).
4. M. Maiti, S. Maity, A. Roy, *An Improved Genetic Algorithm and Its Application in Constrained Solid TSP in Uncertain Environments*, **Proceedings in Mathematics** & **Statistics 125**, (**Springer**), DOI 10.1007/978-81-322-2301-6-14, (2016)
5. A. Roy, S. Maity, M. Maiti, *Constrained solid travelling salesman problem using Adaptive Genetic Algorithm in uncertain environment*, **IEEE**, ICIEV, Japan, DOI:10.1109/ICIEV.2015.7334044, (2016).

The list of Communicated Papers

1. Rough Genetic Algorithm for Constrained Solid TSP with Interval Valued Costs and Times.
   Communicated to Journal of *Fuzzy Information and Engineering* (**Elsevier**)
2. A Rough Multi-Objective Genetic Algorithm for uncertain Constrained Multi-Objective Solid Travelling Salesman Problem.
   Communicated to International Journal of *Discrete Applied Mathematics* (**Elsevier**)
3. An Intelligent Hybrid Algorithm for four Dimensional TSP (4DTSP).
   Communicated to International Journal of *Industrial Information Integration* (**Elsevier**)

4. A new Evolutionary Hybrid Algorithm for restricted 4- Dimensional TSP (r-4DTSP) in Uncertain Environment.
   Communicated to Journal of *Swarm and Evolutionary Computation* (**Elsevier**).

List of Acronyms

3DTSP	Three Dimensional Travelling Salesman Problem
4DTSP	Four Dimensional Travelling Salesman problem
ACO	Ant Colony Optimization
AGA	Adaptive Genetic Algorithm
BKS	Best Known Solution
BRCSTSP	Bi-random CSTSP
CCP	Chance Constraint Programming
CMOTSP	Constraint Multi-objective TSP
CMOSTSP	Constrained Multi-objective STSP
CSTSP	Constrained Solid Travelling Salesman Problem
CSTSPwR	CSTSP with Restricted Vehicles
Cr	Credibility
CV	Critical Value
CPU	Central Precessing Unit
CSTSP	Constraint Solid Travelling Salesman Problem
DM	Decision Maker
DMP	Decision Making Problem
Equ.	Equation
EVM	Expected Value Model
Ex-Tr	Expectation-Trust
FGA	Fuzzy Genetic Algorithm
FNLP	Fuzzy Non-Linear Programming
FRCSTSP	Fuzzy Rough CSTSP
Fu-Ra	Fuzzy Random
Fu-Ro	Fuzzy Rough
FV	Fuzzy Variable
GA	Genetic Algorithm
GRG	Generalized Reduced Gradient
HA	Hybrid Algorithm
HIA	Hybrid Intelligent Algorithm
iMOGA	imprecise Multi-Objective Genetic Algorithm
IGA	Improved Genetic Algorithm
LFN	Linear Fuzzy Number
MOGA	Multi-Objective Genetic Algorithm
MONLP	Multi-Objective Non-Linear Programming

MPSO	Modified Particle Swarm Optimization
NLP	Non-Linear Programming
Nes	Necessity
Nes - Nes	Necessity-Necessity
ODM	Optimistic Decision Maker
OR	Operations Research
PDM	Pessimistic Decision Maker
Pos	Possibility
Pos-Pos	Possibility-Possibility
PSO	Particle Swarm Optimization
rACO-GA	rough set based ACO-GA
r-4DTSP	restricted four dimensional TSP
R-MOGA	Rough-Multi Objective Genetic Algorithm
RaCSTSP	Random CSTSP
ReGA	Rough Extended GA
RFCSTSP	Random Fuzzy CSTSP
RoCSTSP	Rough CSTSP
RSGA	Rough Set based GA
SGA	Simple/Standard Genetic Algorithm
STSP	Solid Travelliing Salesman Problem
TSP	Travelling Salesman Problem
TP	Transportation Problem
TFN	Triangular Fuzzy Number
TrFN	Trapezoidal Fuzzy Number
VVH	Very Very High
VVHP	Very Very High Pheromone
VVL	Very Very Low
VVY	Very Very Young
VY	Very Young
Y	Young

List of Tables

List of Figures

Contents

Part I

Introduction and Methods/Techniques

Chapter 1

Introduction

Soft Computing (SC) is the fusion of methodologies that were designed to model and enable solutions to real world problems, which are otherwise too difficult to formulate, mathematically. SC is a consortium of methodologies that works synergistically and provides, in one form or another, flexible information processing capability for handling real-life ambiguous situations. Travelling salesman problem (TSP) is central to operations research and management science. It is now widely recognized that some of the most successful applications of operations research are encountered in TSP, most significantly in the airline industry where they underlay almost every aspect of strategic, tactical and operational planning. Still there have no state of art algorithm that exactly solve TSP in polynomial time. So in the present research, we develop different types of soft computing methods and present some real world TSP problem as solid travelling salesman problem, 4DTSP, etc., under stochastic as well as non stochastic uncertainties. The efficiency of the proposed algorithms are tested also by solving the standard problems taking statistical tests. These algorithms can be used to solve the problems in other areas such as network optimization, VLSI design, etc.

1.1 Soft Computing and Uncertainties

SC is a branch, in which, it is tried to build intelligent and wiser machines. Purity of thinking, machine intelligence, freedom to work, dimensions, complexity and fuzziness handling capability increase, as we go higher and higher in the hierarchy as shown in Fig.1.1. The final aim is to develop a computer or a machine which will work in a similar way as human beings can do, i.e. the wisdom of human beings can be replicated in computers in some artificial manner. If a tendency towards imprecision could be tolerated, then it should be possible to extend the scope of the applications even to those problems where the analytical and mathematical representations are readily available. The motivation for such an extension is the expected decrease in computational load and consequent increase of computation speeds that permit more robust system [73]. SC has three main branches: fuzzy systems, evolutionary computation, artificial neural computing, with the latter subsuming machine learning (ML) and probabilistic reasoning (PR), belief networks, chaos theory, parts of learning theory and wisdom based expert system (WES), etc.

Optimization is a subject that attempts to find best possible solution for a

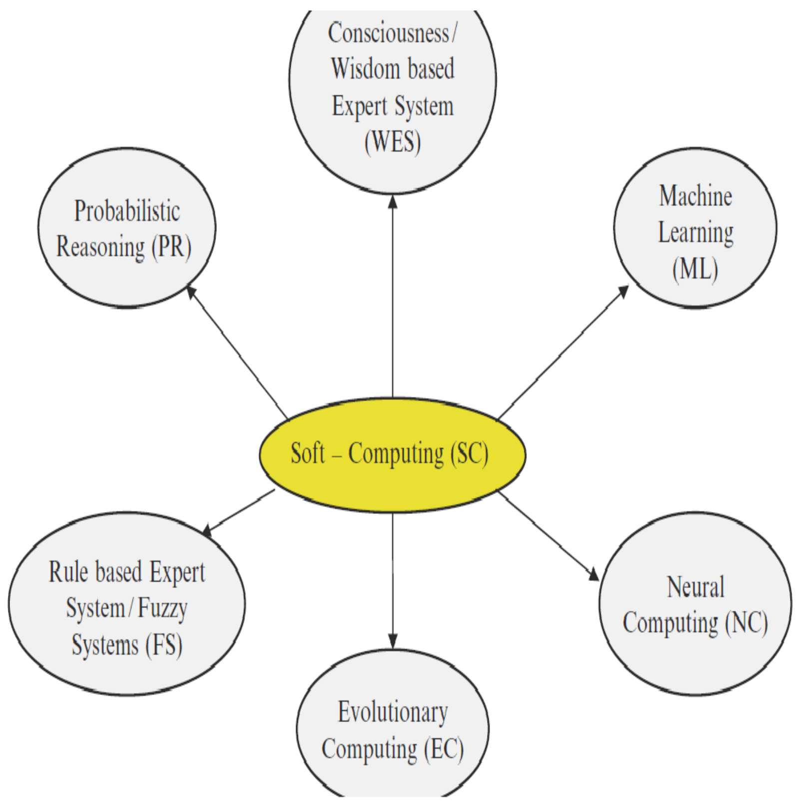


Figure 1.1: Componants of Soft Computing

problem. Optimization problems are quite common in computer science, whenever real-world applications are considered [150]. These problems are often impossible to resolve through an exact mathematical approach: this could be due to the applicability of an exact optimization method, or to a time consuming approach that do not satisfy application constraints. In such cases it could be necessary to perform a complete exploration of all the possible solutions in order to find the optimal one, unacceptable in terms of time and computational costs.

Uncertainty is universal dilemma. Uncertainty intrudes into plans for the future, interpretations of the past, and decision in the present. There are many kinds of uncertainty. The real world values are vague, fuzzy, confidence, ambiguity, inconsistent, incomplete, imprecise, general, anomalous, incongruent, ignorant and irrelevant. In the present study we use interval valued, fuzzy and rough data with their different combinations as various parameters in the optimization problem.

1.2 Some Components of Soft Computing

Fuzzy System

Fuzzy systems are a generalization of stiff Boolean logic. It uses fuzzy sets which are a generalization of crisp sets in classical set theory. In classical set theory, an object could just be either a member of set or not at all, in fuzzy set theory, a given object is said to be of a certain degree of membership to the set. Hence, in fuzzy sets, membership value of an object could be in the range 0 to 1, but in crisp set the membership value is always 0 or 1.

Artificial Neural Networks

Artificial neural networks (ANN), or simply neural networks, can be loosely defined as large sets of interconnected simple units which execute in parallel to perform a common global task. These units usually undergo a learning process which automatically updates network parameters in response to a possibly evolving input environment. The units are often highly simplified models of the biological neurons found in the animal brain.

Fuzzy Logic

The human beings deal with imprecise and uncertain information as we go about our day to day routines. This can be gleaned from the language we use which contains many qualitative and subjective words and phrases such as quite expensive, very young, or a little far, expensive, etc. In human information processing, approximate reasoning is used and tried to accommodate varying degrees of imprecision and uncertainty in the concepts and tokens of information that we deal with in fuzzy logic.

Many applications of fuzzy systems have been flourished. These applications include areas in industrial systems, intelligent control, decision support systems,

Table 1.1: Search Category Examples

Deterministic	Stochastic
Hill-Climbing	Random Mutation Hill- Climbing
Branch & Bound	Tabu Search
Depth First Search	Simulated Annealing
Breadth First Search	Genetic Algorithms
Best First Search	Monte Carlo Method
Greedy Algorithm	ACO

and consumer products. Fuzzy logic-based products now account for billions of US dollar business every year.

1.3 Biologically Inspired Methods

Biologically inspired methods is a general term pertaining to computing which is inspired by nature. Over the last thirty years many differing strategies have been developed, ranging from Artificial Neural Networks, Evolutionary Computation, Fuzzy Sets to Ant Colony Optimization, Genetic Algorithm and Swarm Optimization, etc. These differing algorithms have been applied to a number of complex problems, such as: signal and image processing, data visualization, data mining, and combinatorial optimization. Some of the deterministic examples listed in Table 1.1 attempt to limit the size of the search space by incorporating some domain specific information.

1.3.1 Evolutionary Computation

Evolutionary computation (EC) is a biologically inspired method of computation and has been applied to a wide variety of problems. The paradigm is inspired by the evolution exhibited by living organisms. It consists of a population of individuals (solutions for a problem) on which reproduction, recombination, mutation and selection are iteratively performed resulting in the survival of the fittest solution occurring in the population of solutions. The EC techniques were proposed in the late 1950s by a number of different researchers [57, 58, 38]. However the research area did not begin to gather much interest until the works by [54] proposing evolutionary programming, Holland [68] proposing genetic algorithms and Fogel [38] proposing evolutionary strategies were published. Each of these strategies developed independently and it was not until the early 1990s that a generic term would itself evolve: evolutionary computation. The field of evolutionary computation was proposed so as to unify efforts from each of the evolutionary based search techniques.

1.3.2 Genetic Algorithm

The genetic algorithm is another machine learning technique which derives its behaviour from an evolutionary biology metaphor. Genetic algorithms were formalised by Holland [68] in 1975 as a model of adaptation. In simple genetic algorithms, by Goldberg [61] randomly generated solution strings are formed into a population. The strings are decoded and then evaluated according to a fitness/objective function. Following this, individuals are selected to undergo reproduction to produce offspring (individuals for the next generation). The process of producing offspring consists of two operations. Firstly selected solution strings are recombined using a recombination operator i.e. crossover, where two or more parent solution strings provide elements of their string to generate a new solution. Secondly mutation is applied to the offspring. Following the generation of a complete population of offspring solution strings, the offspring population replaces the parent population. Each iteration of the process is called a generation. The genetic algorithm is usually run for a fixed number of generations, or until some criteria is met e.g.: no improvement in solutions fitness for a number of generations.

1.3.3 Ant Colony Optimization

One of the first behaviors studied by entomologists was the ability of ants to find the shortest path between their nest and a food source. From these studies and observations followed the first algorithmic models of the foraging behavior of ants developed by Marco Dorigo [41]. Collectively, algorithms that were developed as a result of studies of ant foraging behavior are referred to as instances of the ant colony optimization heuristic (ACO) [42, 43]. Section 2.1.7 illustrates brief discussion of ACO.

1.3.4 Particle Swarm Optimization

The particle swarm optimization (PSO) algorithm is a population-based search algorithm based on the simulation of the social behavior of birds within a flock. The initial intent of the particle swarm concept was to graphically simulate the graceful and unpredictable choreography of a bird flock [81], with the aim of discovering patterns that govern the ability of birds to fly synchronously. Here individuals referred to as particles, are flown through hyper-dimensional search space. Changes to the position of particles within the search space are based on the social-psychological tendency of individuals to emulate the success of other individuals. The changes to particle within the swarm are therefore influenced by the experience, or knowledge, of its neighbors.

1.4 Hybrid Intelligent Systems

In many cases, hybrid applications methods have proven to be effective in designing intelligent systems. As it was shown in recent years, fuzzy logic, neural networks and evolutionary computations are complementary methodologies in the design and implementation of intelligent systems. Each approach has its merits and drawbacks. To take advantage of the merits and eliminate their drawbacks, many ways of integrating these methodologies have been proposed by researchers during the past few years. These techniques include the integration of neural network and fuzzy logic techniques as well as the combination of these two technologies with evolutionary methods. The merging of ACO, GA and PSO can be realized in different directions, resulting in systems with different characteristics given in this thesis.

1.5 Combinatorial Optimization

The area of Combinatorial Optimization deals with algorithmic problems of the following flavour: Find a best object in a possibly large, but finite, space. As a subfield of mathematics, this area is relatively new, having been studied only in the last 100 years or so. Some of the problems that we will study, along with several problems arising in practice, are NP-hard, and so it is unlikely that we can design exact efficient algorithms for them. For such problems, we will study algorithms that are worst-case efficient, but that output solutions that can be sub-optimal. We will be able, however, to prove worst-case bounds to the ratio between the cost of optimal solutions and the cost of the solutions provided by our algorithms. Sub-optimal algorithms with provable guarantees about the quality of their output solutions are called approximation algorithms.

It turns out that the general TSP cannot have an efficient α-approximation algorithm for any α that is polynomial-time computable unless P = NP. This follows from a simple reduction from the well-known NP-complete problem called the Hamiltonian Cycle problem: the input here is a directed graph and we need to check if the graph contains a Hamiltonian cycle, a cycle in which each vertex of the graph appears exactly once.

1.6 Travelling Salesman Problems

The travelling salesman problem is stated as follows: given a number of cities with associated city to city distances, what is the shortest round trip tour that visits each city exactly once and returns to the start city [32]. The problem sounds quite simple, however as the number of cities in the problem increases so too does the number of permutations of valid tours e.g. for 5 cities 12, 7 cities 360 and for 9 cities 20160 possible permutations (for a 60 city problem it is possible that the number of permutations is of the same order of magnitude as the total number of atoms in the universe). Thus attempting to find the minimal distance tour in anything but very small problems is computationally expensive.

1.6.1 Historical Review of TSPs

The TSP has a long history, and this history can help in the understanding of the problem and in understanding why it remains a significant problem. The TSP on examination is firmly placed in the field of mathematics, specifically graph theory. It has influenced many differing problems in a wide range of areas: engineering, geography, transportation and computer science.

In graph theory, a Hamiltonian cycle is a path in an undirected graph which visits each node exactly once and also returns to the starting node. The Hamiltonian cycle problem can easily be extended to form an optimization problem. If the graph were to have weights on its edges, and suppose that the problem is to find a Hamiltonian cycle with the minimum weight, where the weight of a cycle is defined to be the sum of the weights on its edges, then this would be the travelling salesman problem. A complete history of the TSP is difficult to compile. The problem was originally known by a number of different names. The most important of these was the messenger problem (Karl Menger) [19, 67, 114].

George Dantzig, Ray Fulkerson and Selmer Johnson in their paper solution of a large-scale travelling salesman problem [31, 91] proposed a novel method for solving instances of the TSP using linear programming. Dantzig et al. [32], while working at the Rand Corporation, developed a technique to optimize solutions for combinatorial problems called the Simplex Algorithm. The cutting-plane method has been successfully applied to a wide range of problems in the combinatorial field [89]. The branch and bound technique was applied to the TSP by Little et al. [91] in 1963. Heuristic methods and further experiments with the cutting plane techniques made it possible to find optimal solutions for problems up to 100 cities in size [127]. The technique that was implemented was the cutting-plane method, as described in [91, 3, 4, 5]. This method found an optimal solution for a 15112 city TSP problem.

1.6.2 Real world examples of TSP

Practical examples of the TSP can be observed in transport, network routing and logistical problems. There are many reasons why people wish to solve the TSP. One reason is the abundance of day to day problems. Real life that was motivated to work on the TSP problem so as to reduce the costs for school bus routes in his district. One of the oldest reported of these was the attempt to solve problems in the agriculture and the construction industries by Mahalanobis [112] in the 1940s. In the electronics manufacturing field, component placement problems, robotic arm tour problems and similar manufacturing logistical problems are being addressed with techniques first developed for the TSP. One industrial example is the Printed Circuit Board problem which has been examined by Queyranne et al. [140, 141]. Lawler and others [89, 3, 4] have compiled a list of related real world problem instances, including call scheduling, delivery of meals on wheels, container movements in a port and warehouse automated fork-lift truck movements.

The Vehicle Routing Problem (VRP) [110] is typically bundled with the TSP. However it differs from the TSP in a number of different ways. The VRP is a combinatorial optimization problem that can be viewed as a combination of two well known NP- Hard problems - the TSP and the Bin Packing Problem. The Bin packing problem is stated as: objects of different sizes must be packed into a finite number of bins of specified capacity V, to minimize the number of bins used to pack all the objects [49]. The VRP is based on the problems associated with a fleet of vehicles supplying customers in different cities across a country. These vehicles each have a certain capacity and each customer has a certain set of requirements. The vehicles all operate from a depot(s). For each delivery to the customer there is a depot(s) and a distance (length, cost, time). The VRP sets a task to find the optimal vehicle routes (minimum distance or number of vehicles). All of the itineraries for the vehicles start and end at a depot, and each must be constructed so that each customer is visited once and by only one vehicle.

The Quadratic Assignment Problem (QAP) can also be considered a form of TSP problem [129]. The QAP consists of a set of n facilities and a set of n locations. For each pair of locations a distance is specified. For each pair of facilities a weight is specified (this might represent the amount of goods to be transported between the facilities). QAP is an optimization problem where weights and the distance for all the locations and facilities are minimize to find an optimal solution.

1.6.3 Complexity of TSP

The TSP is possible to think as a complete graph with n nodes where each edge of the graph is assigned a weight. These weights represent the distance or cost of moving from one node to another. The objective is to find a minimum distance Hamiltonian Cycle of the graph. From a combinatorial view point one might ask how many Hamiltonian Cycles must be examined in order to find a minimum cost circuit. Computing a possible tour of the graph, it is required to start at a particular node, from this node it is possible to visit any one of n-1 other nodes, and following the next move, any of n-2 other nodes, etc., the total number of circuits is therefore (n-1)!. It is this factorial growth that makes the task of solving the TSP immense even for modest n sized problems. An example of this immense size is that for a 20 city TSP problem the total number of possible routes is over 6x1016. This factorial growth makes using exhaustive search techniques impracticable for anything but the smallest of TSP problems. For example should it be possible always to compute a valid TSP tour in a millisecond, then with an 8 city TSP all possible tours could be computed in 2.52 seconds, a 16 city tour in just over 20 years and a 20 city tour in just less than 2 million years. This explosion in the number of potential tours has been one of the motivating factors that has driven the search for fast near optimal search algorithms.

Combinatorial optimizations problems including the TSP are generally classified in accordance with their relationship with the two complexity classes P and NP (Polynomial and Non Polynomial). The TSP is believed to be so called NPcomplete. Researchers in the 1960s accepted that there existed a difference between easy problems (see for example the Maxflow problem [154]) and hard problems like the TSP. This difference was the growth of the algorithms time consumption as the size of the problem increased. It is convention in the literature of complexity theory [75] to consider problems as yes/no questions.

Such problems are described as decision problems and the time estimate in determining an answer to the question is the deciding factor as to the problem being classed as *easy* or *hard*. The group of decision problems where the answer will be computed in polynomial time P (i.e. $O(n^k)$ where k is a constant) are termed easy and those that can not be answered in polynomial time NP (Non-deterministic Polynomial time) are classed as *hard* (e.g. O(2^n)). This then raises the question is P = NP? This is a question that has persisted for some time. It is widely believed that P$\neq$ NP, however this is not proven. It has been shown that a number of problems are in NP and equally it has been shown that a number of problems lie in P. However it is possible that a problem is in NP and in P but it has not been proven. With regard to the TSP, the question arises whether the TSP lies in P or in NP?

Lawler et al [89] state that a decision problem can be classed as NP should there exist a non-deterministic algorithm that solves the problem. Cormen et al [30] later stated that it is possible to *verify* that an algorithm belongs to the NP class if there exists a polynomial time algorithm that verifies that a solution is feasible. P class problems are therefore those that can be *solved* quickly and the NP class problems are those that can be *verified* quickly.

Benchmark problems

The TSP can be viewed as a generalised problem; there are a number of specialised TSP problems (see Figure 1.2 for Lawlers [89] illustration). The symmetric TSP is highlighted because it is this type of TSP that is experimented with in this thesis principally. A library of TSP data sets is maintained at the University of Heidelberg by Professor Gerhard Reinelt [143]. This library TSPLIB [162] of problems contains both problem data and also the best known solutions along with the tour and algorithm which generated the solution.

1.7 Some Different types of TSPs

Several types of TSP that are studied in the literature have been originated from various real life or potential applications. Let us first consider some of these variations that can be reformulated as a TSP using relatively simple transformations. These are TSPs with time windows [53], stochastic TSP [22], double TSP

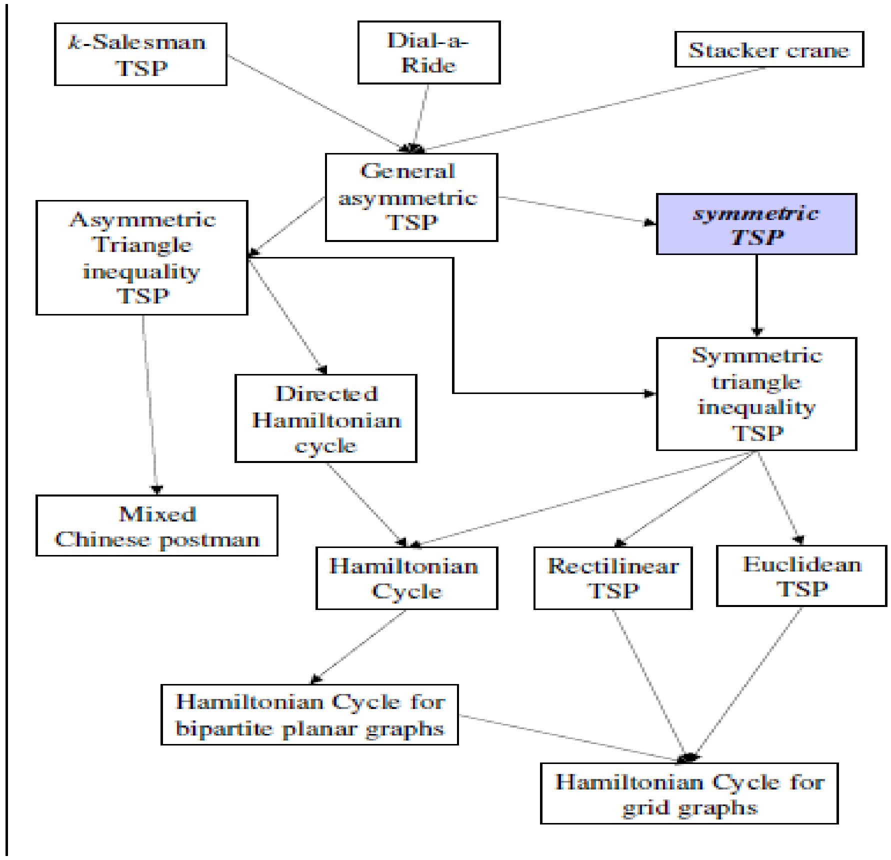


Figure 1.2: Graphical Representation of TSP

[137], asymmetric TSP [111, 116], TSP with precedence constraints [142, 120], etc.

1.7.1 Multi-dimensional TSPs

(a) Classical TSP(2DTSP)

In a classical two-dimensional TSP, a salesman has to travel N cities at minimum cost. In this tour, salesman starts from a city, visit all the cities exactly once and comes to the starting city using minimum cost. Let $c(i, j)$ be the cost for travelling from i-th city to j-th city. Then the problem can be mathematically formulated as:

$$\left.\begin{array}{r}\text{Minimize} \quad Z = \sum\limits_{i \neq j} c(i,j) x_{ij} \\ \text{subject to} \quad \sum\limits_{i=1}^{N} x_{ij} = 1 \ \text{ for } j = 1,2,...,N \\ \sum\limits_{j=1}^{N} x_{ij} = 1 \ \text{ for } i = 1,2,...,N \\ \sum\limits_{i \in S}^{N} \sum\limits_{j \in S}^{N} x_{ij} \leq |S| - 1, \forall S \subset Q \\ \sum\limits_{i=1}^{N} \sum\limits_{j=1}^{N} t(i,j) x_{ij} \leq t_{max} \\ where \ x_{ij} \in \{0,1\}, i,j = 1,2..,N.. \end{array}\right\} \tag{1.1}$$

where x_{ij} is the decision variable and $x_{ij} = 1$ if the salesman travels from city-i to city-j, otherwise $x_{ij} = 0$. Then the above 2DTSP reduces to

$$\left.\begin{array}{r}\text{determine a complete tour} \quad (x_1, x_2, ..., x_N, x_1) \\ \text{to minimize} \quad Z = \sum\limits_{i=1}^{N-1} c(x_i, x_{i+1}) + c(x_N, x_1) \\ where \ x_i \neq x_j, i,j = 1,2...,N. \end{array}\right\} \tag{1.2}$$

along with sub tour elimination criteria

$$\left.\begin{array}{r}\sum\limits_{i \in S}^{N} \sum\limits_{j \in S}^{N} x_{ij} \leq |S| - 1, \forall S \subset Q \\ where \ x_{ij} \in \{0,1\}, i,j = 1,2..,N.. \end{array}\right\} \tag{1.3}$$

Later on, this constraint Equ. 1.3 is not mentioned explicitly in the formulation different TSP models, assuming that it is automatically satisfied for a feasible solution.

1.7.2 Proposed Solid TSP(3DTSP)

(a) Proposed Solid TSP(3DTSP)

In a Solid TSP, a salesman has to travel N cities by choosing any one of the P types of conveyances available using minimum cost. risk/discomfort factors in travelling from one city to another using different vehicles are different. The salesman should choice such a path and conveyances. Let $c(i, j, k)$ be the cost for travelling from i-th city to j-th city using k-th type conveyance. Then the salesman has to determine a complete tour (x_1, x_2, ...,x_N, x_1) and corresponding conveyance types (v_1, v_2, ..., v_P) to be used for the tour, where $x_i \in \{1, 2, ..N\}$ for $i = 1, 2, ..., N$, $v_i \in \{1, 2, ..P\}$ for $i = 1, 2, ..., N$ and all x_i are distinct. Then the problem can be mathematically formulated as:

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one available corresponding conveyance in each step from the vehicle types $(v_1, v_2, ..., v_P)$ so as

$$\left.\begin{array}{c}\text{to minimize } \ Z = \sum_{i=1}^{N-1} c(x_i, x_{i+1}, v_i) + c(x_N, x_1, v_l), \\ where\ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\}\end{array}\right\} \tag{1.4}$$

(b) Solid TSP with restricted conveyances (3DTSPwR)

In real life, it is seen that in all stations, all types of conveyances may not available due to the geographical position of the station,weather conditions, etc. So it is more realistic, that restricted conveyances are available in different stations. Considering the availability of the conveyances, we design the STSP with restricted condition as below:

Let $c(i, j, k)$ be the cost for travelling from i-th city to j-th city using k-th type conveyance. Then the salesman has to determine a complete tour (x_1, x_2, ...,x_N, x_1) and corresponding conveyance types (v_1, v_2, ..., v_S) to be used for the tour, where $x_i \in \{1, 2, ..N\}$ for $i = 1, 2, ..., N$, $v_i \in \{1, 2, ..S\}$ for $i = 1, 2, ..., N$ and all x_is are distinct. Also $v_i \in \{1, 2, ..S\}$ provides maximum available S($\leq$ P) types of conveyances. Then the problem can be mathematically formulated as:

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one available corresponding conveyance in each step from the vehicle types $(v_1, v_2, ..., v_S)$ so as

$$\left.\begin{array}{c}\text{to minimize } \ Z = \sum_{i=1}^{N-1} c(x_i, x_{i+1}, v_i) + c(x_N, x_1, v_l), \\ where\ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{v_1, v_2.., v_S\}\end{array}\right\} \tag{1.5}$$

(c) STSP with risk/discomfort Constraints (CSTSP)

Let $c(i, j, k)$ be the cost for travelling from i-th city to j-th city using k-th type conveyance and $r(i, j, k)$ be the risk/discomfort factor in travelling from i-th city to j-th using k-th type conveyances. Then the salesman has to determine a complete tour (x_1, x_2, ...,x_N, x_1) and corresponding conveyance types (v_1, v_2, ..., v_P)

to be used for the tour, where $x_i \in \{1, 2, ..N\}$ for $i = 1, 2, ..., N$, $v_i \in \{1, 2, ..P\}$ for $i = 1, 2, ..., N$ and all x_i are distinct. Then the problem can be mathematically formulated as:

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one available corresponding conveyance in each step from the vehicle types $(v_1, v_2, ..., v_P)$ so as

$$\left.\begin{array}{c} \text{to minimize} \quad Z = \sum\limits_{i=1}^{N-1} c(x_i, x_{i+1}, v_i) + c(x_N, x_1, v_l), \\ \text{subject to} \quad \sum\limits_{i=1}^{N-1} r(x_i, x_{i+1}, v_i) + r(x_N, x_1, v_l) \leq r_{max}, \\ where \ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\} \end{array}\right\} \quad (1.6)$$

Where r_{max} is the maximum risk/discomfort factor that should be maintained by the salesman in the entire tour to avoid unwanted situation.

1.7.3 Proposed 4 Dimensional TSP(4DTSP)

(a) Four Dimensional TSP(4DTSP)

Let $c(i, j, r, k)$ and $t(i, j, r, k)$ be the cost and time respectively for travelling from i-th city to j-th city by the r-th route using k-th type conveyance. Then the salesman has to determine a complete tour (x_1, x_2, ...,x_N, x_1) and corresponding available route types (r_1, r_2, ..., r_s) with conveyance types (v_1, v_2, ..., v_p) to be used for the tour, where $x_i \in \{1, 2, ..N\}$ for $i = 1, 2, ..., N$, $r_i \in \{1, 2, ..s\}$ and $v_i \in \{1, 2, ..p\}$ for $i = 1, 2, ..., N$ and all x_i's are distinct. Then the problem can be mathematically formulated as:

$$\left.\begin{array}{c} \text{minimize} \quad Z = \sum\limits_{i=1}^{N-1} c(x_i, x_{i+1}, r_i, v_i) + c(x_N, x_1, r_l, v_l), \\ \text{subject to} \quad \sum\limits_{i=1}^{N-1} t(x_i, x_{i+1}, r_i, v_i) + t(x_N, x_1, r_l, v_l) \leq t_{max}, \\ where \ x_i \neq x_j, i, j = 1, 2...N, \quad r_i, r_l \in \{1, 2.., or\, s\}, \\ v_i, v_l \in \{1, 2.., or\, p\} \end{array}\right\} \quad (1.7)$$

(b) 4DTSP with Restricted Path and Time constraint

In real life, it is seen that in all stations, all types routes may not be available due to the geographical position of the station,weather conditions, etc. So it is more realistic, that restricted routes be considered to travel different stations. Let

$c(i, j, r, k)$ and $t(i, j, r, k)$ be the cost and time respectively for travelling from i-th city to j-th city by the r-th route using k-th type conveyance. Then the salesman has to determine a complete tour (x_1, x_2, ...,x_N, x_1) and corresponding available route types (r_{m1}, r_{m2}, ..., r_{ms}) with conveyance types (v_{q1}, v_{q2}, ..., v_{qp}) providing maximum available $s_1(\leq$ s) and $p_1(\leq$ p) types of routes and conveyances to be used for the tour, where $x_i \in \{1, 2, ..N\}$ for $i = 1, 2, ..., N$, $r_{mi} \in \{1, 2, ..s_1\}$ and $v_{qi} \in \{1, 2, ..p_1\}$ for $i = 1, 2, ..., N$ and all x_i's are distinct. Then the problem can be mathematically formulated as:

$$\left.\begin{array}{c}
\text{minimize} \quad Z = \sum_{i=1}^{N-1} c(x_i, x_{i+1}, r_{mi}, v_{qi}) + c(x_N, x_1, r_{ml}, v_{ql}), \\
\text{subject to} \quad \sum_{i=1}^{N-1} t(x_i, x_{i+1}, r_{mi}, v_{qi}) + t(x_N, x_1, r_{ml}, v_{ql}) \leq t_{max}, \\
where\ x_i \neq x_j, i, j = 1, 2...N, m = 1, 2, ...s_1\,, q = 1, 2, .., p_1, \\
r_{mi}, r_{ml} \in \{1, 2.., or\ s_1\}, v_{qi}, v_{ql} \in \{1, 2.., or\ p_1\},
\end{array}\right\} \quad (1.8)$$

1.7.4 Multi-TSPs

The Multiple Traveling Salesman Problem (mTSP) is a generalization of the Traveling Salesman Problem (TSP) in which more than one salesman is allowed. Given a set of cities, one depot (where m salesmen are located), and a cost metric, the objective of the mTSP is to determine a set of routes for m salesmen so as to minimize the total cost of the m routes. The cost metric can represent cost, distance, or time. The requirements on the set of routes are:

-All of the routes must start and end at the (same) depot.

- Each city must be visited exactly once by only one salesman.

The mTSP is a relaxation of the vehicle routing problem (VRP), if the vehicle capacity in the VRP is a sufficiently large value so as not to restrict the vehicle capacity, then the problem is the same as the mTSP. Therefore, all of the formulations and solution approaches for the VRP are valid for the mTSP. The mTSP is a generalization of the TSP, if the value of m is 1, then the mTSP problem is the same as the TSP. Therefore, all of the formulations and solution approaches for the mTSP are valid. Bektas [10] lists a number of variations on the mTSP.

1.7.5 Bottleneck TSPs

The Bottleneck traveling salesman problem (bottleneck TSP) is a problem in discrete or combinatorial optimization. It is stated as follows: Find the Hamiltonian cycle in a weighted graph which minimizes the weight of the most weighty edge of the cycle. The problem is known to be NP-hard. The decision problem version of this, ”for a given length x, is there a Hamiltonian cycle in a graph g with no edge longer than x?”, is NP-complete. In an asymmetric bottleneck TSP, there are cases where the weight from node A to B is different from the weight from B to A (e. g. travel time between two cities with a traffic jam in one direction). Euclidean bottleneck TSP, or planar bottleneck TSP, is the bottleneck TSP with the distance being the ordinary Euclidean distance. The problem still remains NP-hard, however many heuristics work better. If the graph is a metric space then there is an efficient approximation algorithm that finds a Hamiltonian cycle with maximum edge weight being no more than twice the optimum [130].

1.8 Historical Review of Uncertain TSPs

Traveling salesman problem is a fundamental combinatorial optimization model studied in the operations research community for nearly sixty years, yet there is surprisingly little literature that addresses uncertainty and multiple objectives in it. The traditional TSP studies mentioned above are all assumed in deterministic environment. However, in the real world, TSP situations are often in deterministic, some or all of the TSPs parameters are not known with certainty at the moment we have to make decision. With the great improvement of probability theory, the stochastic model has been widely used in many relevant TSPs to represent the indeterminacy, including the consideration of probability in the presence of customers Jaillet et al. [74], the demand level Bertsimas et al. [11], the travel time Kao et al. [77], and the service time at customers site Chang et al. [22], usually assuming a known distribution governs some of the problems parameters. Sepideh Fereidouni [152] used a fuzzy multi-objective linear programming. Chaudhuri et al. [26], used a Fuzzy multi-objective linear programming for TSP.

1.9 Review of Different Heuristic Methods for TSPs

D.B. Fogel implemented one of the first successful evolutionary optimisation approaches to the TSP which he described in an evolutionary approach to the travelling salesman problem [55] in 1988. In this paper he outlined an alternative to the genetic operators which Holland [126] proposed in 1987. Evolution pressure was provided by a single operator - mutation. This mutation operation was loosely modelled on L. J. Fogels Evolutionary Programming restricted to single state machines [55, 56].

The results reported by Fogel were that for 30, 50 and 75 city tours, his genetic algorithm found solutions which were better or at worst matched the previous best known tour lengths generated by Whitley et. al [169].

In last decades Majumder and Bhunia [111] formulated a TSP with asymmetric costs and imprecise travel times and solved using GA. Moon et al. [120] applied precedence constraints before visiting the nodes/cities in a TSP and solved using an improved GA. Xing et al. [170] presented a hybrid approach which combines an improved GA and optimization strategies for solving the asymmetric TSP (ATSP). Bai et al. [6] proposed a max-min ant colony optimization method for the solution of ATSPs bridging the gap between hybridization and theoretical analysis. Jula et al. [76] considered a routing problem with stochastic travel times and time windows estimating means and variances of arrival times at nodes and removing routes that are dominated by others. Chang et al. [59] solved a stochastic dynamic TSP with hard time windows following more or less same procedures of Jula et al. [76]. Chang and Mao [21] developed a modified ant algorithm to solve TSPTWs for minimum cost tour. Dong et al. [40] proposed a new hybrid algorithm, cooperative genetic ant system to solve TSP. Yuan et al. [176] proposed a new crossover operator called two-part chromosome crossover for solving the multiple travelling salesman problem (MTSP). Recently, Miranda- Bront [118] formulated and solved a time-dependent travelling salesman problem (TTSP).

Wang et al. [168] proposed an approximate method on sparse graph for TSP, Nagata et al. [123] developed a new GA for asymmetric TSP, Che et al. [27] considered genetic simulated annealing ant colony systems with PSO to solve TSP, Albanyrak et al. [2] developed a new mutation operator to solve TSP by GA, Xu et al. [175] solved multi-objective problem with power station operation, Elaoud et al. [47] proposed multiple crossover and mutation operators with dynamic selection scheme in MOGA for multi- objective TSP (MOTSP), Lust et al. [107, 108] presented two-phase Pareto local search (2PPLS) for bi objective TSP, Filippi et al. [52] considered a Pareto ϵ approximation named as ABE algorithm for MOTSP, Samanlioglu et al. [148] proposed weakly Pareto optimal solutions for symmetric MOTSP with memetic random-key GA, Zhou et al. [187] considered multi-objective estimation of distribution algorithm based on decomposition (MEDA/D) for some particular MOTSPs. Paquete et al. [128] analyze algorithmic components of stochastic local search algorithms for the multiobjective travelling salesman problem. A spanning tree concepy for For generation based evolutionary algorithms, normally most of the solutions of the parent population are replaced by children in each generation.

1.10 Motivation and Objectives of the Thesis

Motivation:

Soft Computing (SC) is a widely used technique in present research of the optimization. Now a days SC is used to design the complex real world problems. Again it is a part of artificial intelligence. Evolutionary computing techniques are a part of SC. GA, ACO and PSO are the most popular evolutionary approaches for designing and solving the complex optimization problems in present phenomena. Genetic algorithms are robust adaptive optimization techniques based on a biological paradigm. They perform efficient search on poorly-defined spaces by maintaining an ordered pool of strings that represent regions in the search space. Again it attempts to increase the effectiveness of the search techniques. GAs have already been applied to several difficult search problems. Similarly, ACO and PSO are the biologically inspired SC techniques used to solve the complex decision making problems. The hybridization of these methods is much effective for solving the problems. All these SC techniques elaborately are used mainly on continuous optimization but few methods are applied in discrete optimization problems also. So there are limited research works in discrete cases. Particularly for GA, a lot of well known operators are available to solve both continuous and discrete optimization problems. There is lot of scope of developing the different GA operators and also the hybridization of GA, ACO and PSO for the optimum solutions of NP-hard problems. **This prompted us to take up research works**

to bring the different variations in the GA, ACO and PSO operators and to make different combinations of GA, ACO and PSO to derive the near optimum solution of discrete NP-hard problems.

Normally two-dimensional TSPs are available in the literature. But, in real-life, three and four-dimensional (-3D and -4D) TSPs are in vogue. In 3DTSP, different conveyance available at different nodes are used by the salesman for minimum cost. In 4DTSP, in addition to availability of conveyances of the nodes, there are different paths for travel between the nodes. Though few researchers have considered 3DTSPs (Changder et al.,[23]) but till now, none formulated 4DTSPs. These TSPs have the wide application for medical representative, network routing, transport, logistical problems and electronic manufacturing field, etc. Again, these NP-hard problems can be formulated and solved in different imprecise (fuzzy, rough, etc.) environments. In these cases, the costs, distances, time, etc. of the system may be fuzzy, rough,, fuzzy-rough, etc.

So the above mentioned gaps and considerations motivated us to design different types of GA operators and to develop different hybridization of GA, ACo and PSO for the solution of the above mentioned TSPs. During the research period, it is observed that to solve the discrete optimization problems by SC techniques particularly GA, ACO and PSO, there is a lot of scope to design new operators with different uncertain parameters and new hybridization technique.

The available data of the travelling systems, such as costs, time, risk/discomfort and safety factors etc. are not always exact or precise but are uncertain or imprecise due to uncertainty in judgment, insufficient information, conditions of road, weather condition, etc. and uncertainty of availability of travelling vehicles also. **This motivated us to consider some innovative TSPs in uncertain environments like fuzzy, random, interval valued, rough, bi-fuzzy, bi-rough, bi-random, random-fuzzy, random-rough, fuzzy-random and fuzzy-rough etc.**

Objective of the Thesis:

The main objectives of the presented thesis are:

- **To formulate different types of operators of GA:**
  Some innovative and useful selection operators of GA such as probabilistic, fuzzy age based, extended fuzzy age based, rough age based, rough extended age based and rough set based pheromone classification for ACO

are developed. Again new crossover as Comparison crossover and adaptive crossover are modeled for GA. Many virgin mutation operators such as node oriented, p_m (probability of mutation) dependent and generation dependent mutations are developed to deal with solid TSPs.

- **To formulate hybridization of ACO-GA and ACO-PSO-GA:**
  To day in several cases, hybrid methods are effective in designing intelligent systems. In real world applications, such a fusion between different evolutionary approaches have always a concrete response to improve performance, to reduce computational burden, or to lower the total product/process cost. Here some needful combinations of rough set based pheromone updated ACO with GA and three well known evolutionary approaches ACO, PSO and GA are merged with modified version.

- **To formulate different types multi-objective GA:**
  Though several research works have been done about multi-objective GA, however there are some scopes of research in this field in particular to solve solid TSPs. As in every real world problem contains some uncertainty, the present investigation includes new improved of impreciseness of the multi-objective GA. Here we tried to introduce two different types of uncertainty i.e. fuzzy and rough in multi-objective GA, i.e. imprecise MOGA (iMOGA) and these are used to Rough MOGA (RMOGA) and solve solid TSPs with cost and time as two objectives along with a constraint.

- **To formulate different types of TSPs:**
  Here we have formulated some different types of TSPs models such as solid TSPs i.e. three dimensional TSPs (3DTSPs) where a traveler can choose a conveyance from different types of available vehicles to journey from one city to another city. Also we considered some constraints as time, cost, risk and safety, etc. in constraint solid TSP (CSTSP). Again a new model is designed such as constrained solid TSP with restricted conveyance (CSTSPwR). For the first time, four dimensional TSPs (4DTSPs) are modeled considering different paths from one city to another city, here several also vehicles are also available along each route.

- **To consider different types of uncertainties in TSPs:**
  Decision making with uncertainty is an emerging area. Though few re-

search works have been done on TSPs in fuzzy and random environments, however there are lots of scopes to do research in this area. In the present investigation, several uncertainties are considered for 3DTSPs and 4DTSPs such as fuzzy with possibility, necessity, expected value method and credibility approach, rough with expectation and trust measure, random with chance constraint programming approach, interval valued with different objective model, bi-fuzzy, bi-rough, bi-random, random-fuzzy, random-rough, fuzzy-random, fuzzy-rough environments with their different approaches. Again for the first time, trust measures are extended with five-point scale and seven point scale.

1.11 Organization of the Thesis

The proposed thesis has been divided into following four parts and eight Chapters.

Part-I: Introduction and Methods /Techniques

- **Chapter-1:** Introduction
- **Chapter-2:** Heuristic Computing Methods
- **Chapter-3:** Some Uncertainties Environment

Part-II: Single Objective Optimization by Single/Multi- Heuristic Methods

Chapter-4: Single Objective Optimization Using Single Heuristic Methods

- **Model 4.1:** An Improved Genetic Algorithm and Its Application in Constrained Solid TSP in Uncertain Environments
- **Model 4.2:** An Adaptive GA to Solve Constrained Solid Travelling Salesman Problem in Uncertain Environments
- **Model 4.3:** A Modified Genetic Algorithm for solving uncertain Constrained Solid Travelling Salesman Problems
- **Model 4.4:** Rough Genetic Algorithm for Constrained Solid TSP with Interval Valued Costs and Times

- **Model 4.5:** A Rough Extended GA for Solving Constrained Solid Travelling Salesman Problem Under Bi-Fuzzy Coefficients

Chapter-5: Single Objective Optimization Using Hybrid Heuristic Techniques

- **Model 5.1:** An Intelligent Hybrid Algorithm for four Dimensional TSP (4DTSP)
- **Model 5.2:** A new Evolutionary Hybrid Algorithm for restricted 4- Dimensional TSP (r-4DTSP) in Uncertain Environment

Part-III: Multi-Objective Optimization Using a Heuristic Method

Chapter-6: Multi-Objective Optimization Using a Heuristic Algorithm

- **Model 6.1:** An imprecise Multi-Objective Genetic Algorithm for uncertain Constrained Multi-Objective Solid Travelling Salesman Problem
- **Model 6.2:** A Rough Multi-Objective Genetic Algorithm for uncertain Constrained Multi-Objective Solid Travelling Salesman Problem

Part-IV: Summery and Future Research Scope

Chapter-7: Summery and Future Research Scope

Part-V: Bibliography and Index

Part-I: Introduction and Methods/ Techniques

In the first chapter, contain a brief introduction of the thesis. The general structure of the development of soft computing techniques, combinatorial optimization, TSP as NP hard problem, different uncertain environments and history of SC techniques to solve TSP in different hybrid uncertain environments have been discussed. In the second chapter, a brief over view about the heuristic computing are presented. In chapter-3, here some mathematical prerequisite of the uncertainty is presented.

Chapter-1
Introduction and Methods/Techniques

This chapter contains a brief introduction giving an overview of the development on soft computing methods with combinatorial optimization in different hybrid uncertain environments.

Chapter-2
Some Specific Heuristics

In this chapter, study the heuristics such as Genetic Algorithm (simple GA), Fast and Elitist Multi-Objective Genetic Algorithm, Non-dominated Sorting Genetic Algorithm, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) briefly with their merit and demerits. Here identified the lac of these algorithms to solve for particular discrete optimization problems. Also hybridization of two or more swarm heuristics with their literature review are given.

Chapter-3
Some Uncertain Environments

In this chapter, briefly discussed different uncertainty with their combinations as interval valued, random, fuzzy, rough, Bi-random, Bi-rough, Bi-fuzzy, Fuzzy-rough, rough-fuzzy, fuzzy-random, random-fuzzy, rough-random and random-rough variables. Here few proposed mathematical extension of the uncertainty variables are presented.

Part-II: Single Objective Optimization Using Single/Multi Heuristic Methods

Chapter-4
Single Objective Optimization using Single Heuristic Methods

In this chapter, presents the five model about the proposed Genetic algorithm different operators such as selection, crossover and mutation with introducing solid TSPs under crisp, fuzzy, random, random-fuzzy, fuzzy-random, bi-random, bi-rough environment are developed and solved by the proposed models.

Model-4.1: An Improved Genetic Algorithm and Its Application in Constrained Solid TSP in Uncertain Environments

In this investigation, a GA is to proposed to solved the solid TSPs under different uncertain environments. Here proposed an improved genetic algorithm (IGA) to solve Constrained Solid Travelling Salesman Problems (CSTSPs) in crisp, fuzzy, rough, and fuzzy-rough environments. The algorithm is model with a combination of probabilistic selection, cyclic crossover, and nodes-oriented random mutation. Here, CSTSPs in different uncertain environments have been designed and solved by the proposed algorithm. A CSTSP is usually a travelling salesman problem (TSP) where the salesman visits all cities using any one of the conveyances available at each city under a constraint say, safety constraint. Here a number of conveyances are used for travel from one city to another. . The salesman desires to maintain certain safety level always to travel from one city to another and a total safety for his entire tour. Costs and safety level factors for travelling between the cities are different. The requirement of minimum safety level is expressed in the form of a constraint. The safety factors are expressed by crisp, fuzzy, rough, and fuzzy-rough numbers. The problems are formulated as minimization problems of total cost subject to crisp, fuzzy, rough, or fuzzy-rough constraints. This problem is numerically illustrated with appropriate data values. Optimum results for the different problems are presented via IGA. Moreover, the problems from the TSPLIB (standard data set) are tested with the proposed algorithm with some statistical test.

Model-4.2: Constrained Solid Travelling Salesman Problem Using Adaptive Genetic Algorithm in Uncertain Environment

In this model, an Adaptive Genetic Algorithm (AGA) is developed to solve constrained solid travelling salesman problems (CSTSPs) in crisp, fuzzy and rough environments. In the proposed AGA, we model it with probabilistic selection and proposed a virgin adaptive crossover with random mutation. Present model, CSTSPs are illustrated numerically by some empirical data using this algorithm. In each environment, some sensitivity studies due to different risk/discomfort factors and other system parameters are presented.

Model-4.3: A Modified Genetic Algorithm for solving uncertain Constrained Solid Travelling Salesman Problems

The present investigation, design a Modified Genetic Algorithm (MGA) is developed to solve Constrained Solid Travelling Salesman Problems (CSTSPs) in crisp, fuzzy, random, random-fuzzy, fuzzy-random and bi-random environments. In the developed MGA, a probabilistic selection technique and a comparison crossover are used along with conventional random mutation. In CSTSP, along each route, there may be some risk/discomfort in reaching the destination and the salesman desires to have the total risk/discomfort for the entire tour less than a desired value. Here we model the CSTSP with traveling costs and route risk/discomfort factors as crisp, fuzzy, random, random-fuzzy, fuzzy-random and bi-random in nature. A number of benchmark problems from standard data set, TSPLIB are tested against the existing Genetic Algorithm (with Roulette Wheel Selection (RWS), cyclic crossover and random mutation) and the proposed algorithm and hence the efficiency of the new algorithm is established. In this model, CSTSPs are illustrated numerically by some empirical data using this algorithm. In each environment, some sensitivity studies due to different risk/discomfort factors and other system parameters are presented.

Model-4.4: A Rough Genetic Algorithm for Constrained Solid TSP with Interval Valued Costs and Times

This model presents a Rough Set based Genetic Algorithms (RSGAs) to solve constrained Solid Travelling Salesman Problems (CSTSPs) with restricted conveyances (CSTSPwR) having uncertain costs and times as interval values. In the proposed RSGAs, a rough set based age dependent selection technique and an age oriented min-point crossover are used along with three types of p_m- dependent random mutations. A number of benchmark problems from standard data set, TSPLIB are tested against the proposed algorithms and existing standard GA (SGA) and hence the efficiency of the new algorithms are established. Here CSTSP is a STSP with a constraint (say time constraint). We have modelled CSTSPwRs where some conveyances are not allowed to run in some particular routes. CSTSPwRs are formulated as constrained linear programming problems and solved by both proposed RSGAs and SGA. These are illustrated numerically by some empirical data and the results from the above methods are compared. Statistical significance of the proposed algorithms are demonstrated through statistical analysis using standard deviation (SD). Moreover, the non-parametric test, Friedman test is performed with the proposed algorithms. In addition, a Post Hoc paired comparison is applied and the out performance of the RSGAs are established.

Model-4.5: A Rough extended Genetic Algorithm for Solving Constrained Solid Travelling Salesman Problem Under Bi-Fuzzy Coefficients

In this model, a Rough extended Genetic Algorithm (ReGA) is proposed to solve constrained solid travelling salesman problems (CSTSPs) in crisp and bi-fuzzy coefficients. In the proposed ReGA, developed a rough set based selection (7-point scale) technique and comparison crossover with improved generation dependent mutation. The costs and risk/discomforts factors are in the form of crisp, bi-fuzzy in nature. Here CSTSPs are illustrated numerically by some standard test data from TSPLIB using ReGA. In each environment, some statistical significance studies due to different risk/discomfort factors and other system parameters are presented with some statistical test.

Chapter-5
Single objective Optimization Using hybrid heuristic Techniques

In this chapter two hybrid heuristics are developed solved proposed four dimensional TSPs under bi-fuzzy and bi-rough coefficients. The first model is the combinations of proposed ACO and GA with rough set based pheromone classifier. For the second model hybridize with another swarm intelligent approach PSO and formed a ACO-PSO-GA based model.

Model-5.1: An Intelligent Hybrid Algorithm for 4- Dimensional TSP

Present model described, a hybridized algorithmic approach to solve 4- dimensional Travelling Salesman Problem (4DTSP) where different paths with various number of conveyances are available to travel between two cities. The algorithm is a hybridization of rough set based ant colony optimization (rACO) with proposed genetic algorithm (GA). The initial solutions are produced by ACO which act as a selection operation of GA after it a GA is developed with a virgin extended rough set based selection (7-point scale), comparison crossover and generation dependent mutation. The said hybrid algorithm rough set based Ant Colony Optimization (rACO) with Genetic Algorithm (rACO-GA) is tested against some test functions and efficiency of the proposed algorithm is established. The 4DTSPs are formulated with crisp and bi-fuzzy costs. In each environment, some statistical significant studies due to different time constraint values and other system parameters are presented. The models are illustrated with some numerical data.

Model-5.2: A new Evolutionary Hybrid Algorithm for restricted 4-Dimensional TSP (r-4DTSP) in Uncertain Environment

In this model, we proposed an hybridized three known soft computing technique to solve a restricted 4- dimensional TSP (r-4DTSP). Here some restrictions on paths and conveyances are imposed. The developed hybrid methods combines the ant colony optimization (ACO) and swap operator based particle swarm optimization (PSO) with modified genetic algorithm (GA). The initial solutions are produced by ACO which used as swarm in PSO then a modified GA with virgin selection, comparison crossover and generation dependent mutation. The said hybrid algorithm (ACO-PSO- GA) is tested against some test functions and efficiency of the proposed algorithm is established. The r-4DTSPs are considered with crisp and bi-rough costs. In each environment, some statistical significant studies due to different time constraint values and other system parameters are presented. The models are illustrated with some numerical data.

Part-III: Multi Objective Optimization Using a Heuristic Methods

Chapter-6

Multi- objective optimization using heuristic algorithm

In this chapter contain two multi-objective GA with rough and fuzzy selection operators is developed and solve solid TSP with cost and time as objectives under different hybrid uncertainty.

Model-6.1: An imprecise Multi-Objective Genetic Algorithm for uncertain Constrained Multi-Objective Solid Travelling Salesman Problem

In this model, an imprecise Multi-Objective Genetic Algorithm (iMOGA) is developed to solve Constrained Multi-Objective Solid Travelling Salesman Problems (CMOSTSPs) in crisp, random, random-fuzzy, fuzzy-random and bi-random environments. In the proposed iMOGA, 3 - and 5 - level linguistic based fuzzy age oriented selection, probabilistic selection and an adaptive crossover are used along with a new generation dependent mutation. In each environment, some sensitivity studies due to different risk/discomfort factors and other system parameters are presented. To test the efficiency, combining same size single objective problems from standard TSPLIB, the results of such multi-objective problems are obtained by the proposed algorithm, simple MOGA (Roulette wheel selection, cyclic crossover and random mutation), NSGA-II, MOEA-D/ACO and compared. Moreover, a statistical analysis (Analysis of Variance) is carried out to show the supremacy of the proposed algorithm.

Model-6.2: A Rough Multi-Objective Genetic Algorithm for uncertain Constrained Multi-Objective Solid Travelling Salesman Problem

The present model proposed a Rough Multi-Objective Genetic Algorithm (R-MOGA) to solve Constrained Multi-Objective Solid Travelling Salesman Problems (CMOSTSPs) in rough, fuzzy rough and random rough environments. In the proposed R-MOGA, '3 - and 5 - level linguistic based rough age oriented selection', 'adaptive crossover' are used along with a improved generation dependent mutation. In CMOSTSP, along each route, there may be some risk/discomfort in reaching the destination and the salesman desires to have a total risk/discomfort for the entire tour less than a desired value. Here we model the CMOSTSP with travelling costs and times as two objectives and a constraint for route risk/discomfort factors. The costs, times and risk/discomfort are rough, fuzzy rough and random rough in nature. CMOSTSPs are illustrated numerically by some empirical data using this algorithm. To test the efficiency, combining same size single objective problems from standard TSPLIB, the results of the such multi-objective problems are obtained by the proposed algorithm, simple MOGA and NSGA-II compared. A statistical analysis (Analysis of Variance) is carried out to show the efficiency of the proposed algorithm.

Part-IV: Summary and Future Research Scope

Chapter-7
Summary and Future Research scope

In this chapter a short summary with a brief future research are discussed.

Part-V: Bibliography and Index

Chapter-8
Bibliography and Index

In this chapter Bibliography and Index are presented.

Chapter 2

Heuristic Computing Methods

2.1 Some Specific Heuristics

2.1.1 Introduction

Charles Darwinian evolution in 1859 is intrinsically a so bust search and optimization mechanism. Darwins principle Survival of the fittest captured the popular imagination. This principle can be used as a starting point in introducing evolutionary computation. Evolved biota demonstrates optimized complex behavior at each level: the cell, the organ, the individual and the population. Biological species have solved the problems of chaos, chance, nonlinear interactivities and temporality. These problems proved to be in equivalence with the classic methods of optimization. The evolutionary concept can be applied to problems where heuristic solutions are not present or which leads to unsatisfactory results. As a result, evolutionary algorithms are of recent interest, particularly for practical problems solving.

In this chapter some specific heuristics such as Evalutionary computation, Genetic Algorithm, Particle Swarm optimization and Ant Colony Optimization are described. Here, GAs are described both continuous and discrete and other two soft computing methods PSO and ACO only discrete optimization respectively. Some literature review with their hybridization are study here. Also multi-objective genetic algorithm and its reviews are present in this section.

2.1.2 Evolutionary Computation (EC)

Evolutionary computation (EC) techniques abstract these evolutionary principles into algorithms that may be used to search for optimal solutions to a problem. In a search algorithm, a number of possible solutions to a problem are available and the task is to find the best solution possible in a fixed amount of time. For a search space with only a small number of possible solutions, all the solutions can be examined in a reasonable amount of time and the optimal one found. This exhaustive search, however, quickly becomes impractical as the search space grows in size. Traditional search algorithms randomly sample or heuristically sample the search space one solution at a time in the hopes of finding the optimal solution. The key aspect distinguishing an evolutionary search algorithm from such traditional algorithms is that it is population-based. Through the adaptation of successive generations of a large number of individuals, an evolutionary algorithm performs an efficient directed search.

Evolutionary computing began by lifting ideas from biological evolutionary theory into computer science, and continues to look toward new biological research findings for inspiration. However, an over enthusiastic biology envy can only be to the detriment of both disciplines by masking the broader potential for two-way intellectual traffic of shared insights and analogizing from one another. Three fundamental features of biological evolution illustrate the range of potential intellectual flow between the two communities: particulate genes carry some subtle consequences for biological evolution that have not yet translated mainstream EC, the adaptive properties of the genetic code illustrate how both communities can contribute to a common understanding of appropriate evolutionary abstractions, finally, EC exploration of representational language seems pre-adapted to help biologists understand why life evolved a dichotomy of genotype and phenotype.

2.1.3 The Historical Development of EC

In the case of evolutionary computation, there are four historical paradigms that have served as the basis for much of the activity of the field: genetic algorithms by Holland [68], genetic programming by Koza [85, 86], evolutionary strategies [12], and evolutionary programming (Fogel et al. [54]). The basic differences between the paradigms lie in the nature of the representation schemes, the reproduction operators and selection methods.

2.1.4 Genetic Algorithm (GA)

The most popular technique in evolutionary computation research has been the genetic algorithm. In the traditional genetic algorithm, the representation used is a fixed-length bit string. Each position in the string is assumed to represent a particular feature of an individual, and the value stored in that position represents how that feature is expressed in the solution. There are many types of GA developed by the researchers such as Localized GA (LGA)[164], Adaptive GA (AGA) [124], Enhance GA [70], Efficient GA [48, 33], a novel GA [34], Elitist GA [13] etc, which are used to get the optimal solutions in different research areas. Usually, the string is evaluated as a collection of structural features of a solution that have little or no interactions. The analogy may be drawn directly to genes in biological organisms. Each gene represents an entity that is structurally independent of other genes.

The main reproduction operator used is bit-string crossover, in which two strings are used as parents and new individuals are formed by swapping a subsequence between the two strings. Another popular operator is bit-flipping mutation, in which a single bit in the string is flipped to form a new offspring string. A variety of other operators have also been developed, but are used less frequently (e.g., inversion, in which a sub sequence in the bit string is reversed). A primary distinction that may be made between the various operators is whether or not they introduce any new information into the population. Crossover, for example, does not while mutation does. All operators are also constrained to manipulate the string in a manner consistent with the structural interpretation of genes. For example, two genes at the same location on two strings may be swapped between parents, but not combined based on their values. Traditionally, individuals are selected to be parents probabilistically based upon their fitness values, and the offspring that are created replace the parents. For example, if N parents are selected, then N offspring are generated which replace the parents in the next generation. A GA for a particular problem must have the following six components.

(a) A genetic representation for potential solutions(**chromosomes**) to the problem

(b) A way to create an **initial population** of potential solutions (chromosomes).

(c) A way to **evaluate fitness** of each solution.

(d) An evolution function that plays the role of environment, rating solutions in term of their fitness, i.e., **selection process** for mating pool.

(e) Genetic operators- **crossover, mutation** that alter the composition of children

(f) Values of different parameters that the genetic algorithm uses (**Population size, probabilities of applying genetic operators etc**).

(i) GA for Continuous Optimization

(a) Chromosome representation: The concept of chromosome is normally used

in the GA to stand for a feasible solution to the problem. A chromosome has the form of a string of genes that can take on some value from a specified search space. The specific chromosome representation varies based on the particular problem properties and requirements. Normally, there are two types of chromosome representation – (i) the binary vector representation based on bits and (ii) the real number representation. Here real number representation scheme is used. Here, a 'K dimensional real vector' X=(x_1, x_2, ... x_K) is used to represent a solution, where x_1, x_2, ... x_K represent different decision variables of the problem.

(b) Initialization: A set of solutions (chromosomes) is called a population. N such solutions X_1, X_2, X_3, ... X_N are randomly generated from search space by random number generator such that each X_i satisfies the constraints of the problem. This solution set is taken as initial population and is the starting point for a GA to evolve to desired solutions. At this step, probability of crossover p_c and probability of mutation p_m are also initialized. These two parameters are used to select chromosomes from mating pool for genetic operations- crossover and mutation respectively.

(c) Fitness value: All the chromosomes in the population are evaluated using a fitness function. This fitness value is a measure of whether the chromosome is suited for the environment under consideration. Chromosomes with higher fitness will receive larger probabilities of inheritance in subsequent generations, while chromosomes with low fitness will more likely be eliminated. The selection of a good and accurate fitness function is thus a key to the success of solving any problem quickly. In this thesis, value of a objective function due to the solution X, is taken as fitness of X. Let it be f(X).

(d) Selection process to create mating pool: Selection in the GA is a scheme used to select some solutions from the population for mating pool. From this mating pool, pairs of individuals in the current generation are selected as parents to reproduce offspring. There are several selection schemes, such as roulette wheel selection, local selection, truncation selection, tournament selection, etc. Here, roulette wheel selection process is used in different cases. This process consist of following steps-

(i) Find total fitness of the population F=$\sum_{i=1}^{N} f(X_i)$

(ii) Calculate the probability of selection pr_i of each solution X_i by the formula $pr_i=f(X_i)/F$.

(iii) Calculate the cumulative probability qr_i for each solution X_i by the formula $qr_i=\sum_{j=0}^{i} pr_j$

(iv) Generate a random number 'r' from the range [0..1].

(v) If $r<qr_1$ then select X_1 otherwise select $X_i (2 \leq i \leq N)$ where $qr_{i-1} \leq r < qr_i$.

(vi) Repeat step (iv) and (v) N times to select N solutions from current population. Clearly one solution may be selected more than once.

(vii) Let us denote this selected solution set by $P^1(T)$.

(e) Crossover: Crossover is a key operator in the GA and is used to exchange the main characteristics of parent individuals and pass them on the children. It consist of two steps:

(i) Selection for crossover: For each solution of $P^1(T)$ generate a random number r from the range [0, 1]. If $r<p_c$ then the solution is taken for crossover, where p_c is the probability of crossover.

(ii) Crossover process: Crossover taken place on the selected solutions. For each pair of coupled solutions Y_1, Y_2 a random number c is generated from the range [0, 1] and Y_1, Y_2 are replaced by their offspring's Y_{11} and Y_{21} respectively where $Y_{11}=cY_1+(1-c)Y_2$, $Y_{21}=cY_2+(1-c)Y_1$, provided Y_{11}, Y_{21} satisfied the constraints of the problem.

(f) Mutation: The mutation operation is needed after the crossover operation to maintain population diversity and recover possible loss of some good characteristics. It is also consist of two steps:

(i) Selection for mutation: For each solution of $P^1(T)$ generate a random number r from the range [0, 1]. If r< p_m then the solution is taken for mutation, where p_m is the probability of mutation.

(ii) Mutation process: To mutate a solution X=$(x_1, x_2, ., x_K)$ select a random integer r in the range [1, K]. Then replace x_r by randomly generated value within the boundary of r^{th} component of X.

Following selection, crossover and mutation, the new population is ready for it's next iteration, i.e., $P^1(T)$ is taken as population of new generation. With these genetic operations a simple genetic algorithm takes the following form. In the algorithm T is iteration counter, P(T) is the population of potential solutions for iteration T, Evaluate(P(T)) evaluate fitness of each members of P(T).

Simple Genetic Algorithm (SGA)

1. Set iteration counter T=0.
2. Initialize probability of crossover p_c and probability of mutation p_m.
3. Initialize P(T).
4. Evaluate(P(T)).
5. Repeat
   a. Select N solutions from P(T), for mating pool using Roulette-wheel selection process. Let this set be $P(T)^1$.
   b. Select solutions from $P(T)^1$, for crossover depending on p_c .
   c. Made crossover on selected solutions for crossover to get population $P(T)^2$.
   d. Select solutions from $P(T)^2$, for mutation depending on p_m .
   e. Made mutation on selected solutions for mutation to get population $P(T+1)$.
   f. $T \leftarrow T+1$.
   g. Evaluate $P(T)$.
6. Until(Termination condition does not hold).
7. Output: Fittest solution(chromosome) of P(T).

(ii) GA for Discrete Optimization

(a) Representation: Here a complete tour on N cities represents a solution. So an N-dimensional integer vector $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$ is used to represent a solution, where $x_{i1}, x_{i2}, ..., x_{iN}$ represent N consecutive cities in a tour. For solid TSP another integer vector $V_k = (v_{k1}, v_{k2}, ..., v_{kN})$ is used to represent the conveyances types used travel between different cities. Here v_{kj} represents the conveyance (an integer) used to travel from city x_{ij} to $x_{i(j+1)}$ for j = 1, 2, . . .

,N-1 and v_{kN} represents the conveyance type used to travel from city x_{iN} to x_{i1}.

(b) Initialization: Population size number of such solutions $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$, i = 1, 2, . . . , pop size, are randomly generated by random number generator, such that each solution satisfies the constraints of the problem. A separate sub function check constraint S(X_i) is used for this purpose. For STSP another integer vector $V_k = (v_{k1}, v_{k2}, ..., v_{kN})$ is randomly generated corresponding to the solution X_i, to represent the conveyance types used to travel between different cities. So in that case (X_i, V_k) represent a solution.

Evaluation Process:

To find fitness of a solution $X_i(x_i, V_k)$ for STSP, the following two steps are used

- Calculate objective function value OBJ_i for the solution $X_i(x_i, V_k)$ for CSTSP.
- As the problems are minimization type take $MVAL - OBJ_i$ as fitness, FIT_i, of X_i, (X_i, V_k) for STSP, where MVAL is a sufficiently large value to make the fitness positive.

(c) Cyclic Crossover:

(i) Selection for crossover: For each solution of p(n) generate a random number r from the range [0, 1]. If r $< p_c$ then the solution is taken for crossover.

(ii) Crossover process: For simple TSP cyclic crossover process is used. The cyclic crossover focuses on subsets of cities that occupy the same subset of positions in both parents. Then these cities are copied from the first parent to the offspring (at the same positions), and the remaining positions are filled with the cities of the second parent. In this way, the position of each city is inherited from one of the two parents. However, many edges can be broken in the process, because the initial subset of cities is not necessarily located at consecutive positions in the parent tours. To illustrate the process let us consider a TSP consisting of nine cities and consider two parents PR_1, PR_2 as below:

PR_1:1 2 3 4 5 6 7 8 9

PR_2:3 4 5 1 2 9 8 7 6

Let CH_1, CH_2 be two children born after crossover. The mechanism of birth of CH_1, CH_2 using cycle crossover is explained with the help of the following steps:

Randomly generate an integer in the range [1 . . . 9]. Let it be 3. As PR_1[3] = 3, 3rd element of CH_1 is 3, i.e., CH_1[3] = 3. PR_2, is then searched to check for

the presence of element 3 and it has been found in the first position. Then first element of CH_1 is selected from the first element of PR_1, i.e., CH_1[1] = PR_1[1] = 1. PR_2, is again searched for the presence of element 1 and it has occurred at the fourth position. Thus fourth element of PR_1 has been copied as the fourth element of CH_1, i.e., CH_1[4] = PR_1[4] = 4. Similarly, following are obtained CH_1[2] = PR_1[2] = 2, CH_1[5] = PR_1[5] = 5.
This completes one cycle because element 5 is seen to be present at the third position of PR_2 and the corresponding third position element of PR_1 is element 3, which has already been selected as the starting element of the cycle. The remaining elements of CH_1 are selected directly from PR_2 as follows:
CH_1[6] = PR_2[6] = 9,CH_1[7] = PR_1[7] = 8
CH_1[8] = PR_2[8] = 7, CH_1[9] = PR_1[9] = 6

(d) Random Mutation

(i) Selection for mutation: For each solution of p(n) generate a random number r from the range [0, 1]. If r< p_m then the solution is taken for mutation.

(ii) Mutation Process: To mutate a solution X = $(x_1, x_2, ..., x_N)$ of TSP with T number of nodes, select T number of nodes randomly from the solution and just replace their places in the solution, i.e., if randomly two nodes x_i, x_j are selected then interchange x_i, x_j to get a child solution. The new solution, if satisfies the constraint of the problem, replaces the parent solution. For CSTSP to mutate a solution (X,V), where X = $(x_1, x_2, ..., x_N)$, at first an integer is randomly selected in the range [1, 2]. If 1 is selected then another two random integers i, j are selected in the range [1, N]. Then interchange x_i, x_j to get child solution. If the child solution satisfies the constraint of the problem then it replaces the parent solution.

2.1.5 Multi-Objective Genetic Algorithm (MOGA)

Genetic algorithms are robust search algorithms that use the operations of natural genetics to find the optimum through a search space. Recently, GAs have been used to solve several single and multi-objective decision making problems. In multi- objective optimization techniques (MOOTs), a Pareto Front (PF) is generated and an optimum solution set should be very close to the true PF. But, the above two goals are conflicting for the fixed number of functions, evaluations as the first property requires intensive search over a particular region of the search space and the second one for the uniform search of the whole region.

Thus MOOTs make a trade-off between exploration and exploitation. The first real implication of multi-objective evolutionary algorithm (vector evaluated GA or VEGA) was suggested by David Schaffer in 1984. Then Goldberg suggested to implement domination principle in evolutionary algorithm (EA). Realizing the potential of a good multi-objective evolutionary algorithm (MOEA) Deb [35] and Rubio et al. [146] which can be derived from Goldberg's suggestions, researchers developed different versions of MOEAs such as multi-objective GAs (MOGAs), Niched Pareto GAs (NPGAs) by Horn et al. [69], non-dominated sorting GAs (NSGAs) by Deb [36], hybrid scatter search like MOGA by Durillo et al. [46], decomposition -based MOAs like MOiA/D-DE [90], archive-based micro GAs like AMGA2 by Tiwari et al. [160], etc. In AMGA2, a modified definition of crowding distance for the generation of mating pool has been presented. Recently, an archived -based steady-state micro genetic algorithm (ASMiGA) has been developed with new environmental selection and mating selection strategies by Nag et al. [122]. These algorithms normally select solution from parent population for cross-over and mutation randomly. After these operations parent and child population are combined together and among them better solutions are selected for next iteration. Among non-elitist MOGAs Srinivas and Deb's NSGA (1995) is discussed here and is used to solve STSP. A fast and elitist MOGA is developed following Deb [36] and is used to solve the models also. This algorithm is named Fast and Elitist Multi-objective Genetic Algorithm (FEMOGA).

(i) Srinivas and Deb's NSGA

Major steps of this algorithm are discussed below-

(a) Generate randomly a population P of feasible solutions of size N of the optimization problem under consideration.

(b) Partition P into subsets $P_1, P_2, .., P_k$ such that every subset contains non-dominated solutions but every solution of P_i is not dominated by any solution of P_{i+1} for $i = 1, 2, k - 1$. For this purpose following steps are used:

1. Set subset counter $k = 1$.
2. Set solution counter $i = 1$ and set $P_k = \Phi$
3. For each solution $j \in P$ but, $j \neq i$, check if solution j dominates solution i. If yes go to step 5.

4. If more solutions are left in P, increase j by 1 and go to step 3; otherwise set $P_k = P_k U\{i\}$, here U stands for union.
5. Increase i by one. If $i \leq O(P)$; go to step 3, where $O(P)$ represents number of solutions in P.
6. $P = P - P_k$.
7. if $P \neq \Phi$ increase k by 1 and go to step 2.
8. $P_1, P_2, ...\ P_k$ are the required subsets.

(c) In this step fitness is assigned to every solution of P. The fitness assignment procedure begins from the first non-dominated set and successively proceeds to dominated sets. Any solution i of the first non-dominated set, P_1, is assigned a fitness, F_i, equal to N. To keep diversity among solutions, sharing function method is used font-wise. For this purpose following steps are used-

1. For each solution i in the font P_1, the normalized euclidean distance d_{ij} from another solution j in the same font is calculated by the formula $d_{ij} = \sqrt{\sum_{k=1}^{n}\left(\frac{x_k^i - x_k^j}{x_k^{max} - x_k^{min}}\right)^2}$, where n is the number of components in a solution vector and x_k^{min}, x_k^{max}, are the maximum and minimum values of the k^{th} component of the decision vector.
2. Calculate value of the sharing function σ_{share} by the formula $\sigma_{share} = 0.5/\sqrt[n]{q}$, where q is the number of optima and $q \ll N$.
3. Calculate niche count nc_i for i^{th} solution by the formula $nc_i = \sum_{j=1}^{|P_1|} sh(d_{ij})$ where $sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{share}}\right)^{\alpha} & \text{if } d_{ij} < \sigma_{share} \\ 0 & \text{otherwise} \end{cases}$
$|P_1|$ represents number of solutions in P_1 and α is a positive real number.
4. Reduce fitness of each solution by its niche count i.e., Set $F_i = \frac{F_i}{nc_i}$. This value is called shared fitness of i^{th} solution.

These four steps complete the share fitness assignment procedures of all the solutions in the first front P_1. In order to proceeds to the second font, we note the minimum fitness in the first font, P_1, and then assign fitness slightly smaller than this minimum shared fitness to every solution of P_2. This makes sure that no solution in the first font has a shared fitness value worse than the assigned fitness of any solution in the second font. Once again the sharing function method is applied to all the solution in the second font and the corresponding shared fitness values are computed. This procedure is continued until all solutions in all the fonts are assigned a shared fitness.

Using the above three major steps complete NSGA procedure takes the following form:

1. Generate randomly a population of feasible solution P of size N of the optimization problem under consideration.
2. Choose sharing parameter σ_{share} and a small positive number ξ, probability of crossover p_c and probability of mutation p_m and α.
3. Set $F_{min} = N + \xi$ and front counter $k = 1$.
4. Partition P into non-dominated disjoint subsets of solutions $P_1, P_2, \ldots, P_m$.
5. For each solution $q \in P_k$.
   (i) Assign fitness of q, $F(q) = F_{min} - \xi$
   (ii) Calculate niche count nc_q.
   (iii) Calculate shared fitness $F(q) = F(q)/nc_q$.
6. Set $F_{min} = min\{F(q) : q \in P_k\}$ and $k = k + 1$.
7. If $k \leq m$ go to step 5.
8. Taking shared fitness value as fitness of a solution, select solutions for mating pool from P using Roulette wheel selection process. Let this set be P^1
9. Select solution for crossover and mutation depending on probability of crossover p_c and probability of mutation p_m.
10. Made crossover and mutation on selected solutions and replace parent solutions by child solutions and let resultant set be P^2.
11. Set $P = P^2$ and if termination condition does not hold go to step 3.
12. Output P.
13. End Algorithm.

It can be easily proved that maximum time complexity at different steps of the above algorithm occurs at step-4 which is $O(MN^3)$, where M is the number of objectives. So overall time complexity of the algorithm is $O(MN^3)$. In the following section FEMOGA and it's procedures are described. Procedures of NSGA are same as common procedures of FEMOGA.

(ii) Fast and Elitist Multi-Objective Genetic Algorithm

This multi-objective genetic algorithm has the following two important components.

(a) Division of a population of solutions into subsets having non-dominated solutions: Consider a problem having M objectives and take a population P of feasible solutions of the problem of size N. We like to partition P into subsets $F_1, F_2, , F_k$, such that every subset contains non-dominated solutions, but every solution of F_i is not dominated by any solution of F_{i+1}, for $i = 1, 2, ..k - 1$. To do this for each solution, x, of P, calculate the following two entities.

(i) Number of solutions of P which dominate x, let it be n_x.

(ii) Set of solutions of P that are dominated by x. Let it be S_x.

The above two steps require $O(MN^2)$ computations. Clearly F_1 contains every solution x having $n_x = 0$. Now for each solution $x \in F_1$, visit every member y of S_x and decrease n_y by 1. In doing so if for any member y, $n_y = 0$, then $y \in F_2$. In this way F_2 is constructed. The above process is continued to every member of F_2 and thus F_3 is obtained. This process is continued until all subsets are identified. For each solution x in the second or higher level of non-dominated subsets, n_x can be at most $N - 1$. So each solution x will be visited at most $N - 1$ times before n_x becomes zero. At this point, the solution is assigned a subset and will never be visited again. Since there is at most $N - 1$ such solutions, the total complexity is $O(N^2)$. So overall complexity of this component is $O(MN^2)$.

(b) Determine distance of a solution from other solutions of a subset: To determine distance of a solution from other solutions of a subset following steps are followed:

(i) First sort the subset according to each objective function values in ascending order of magnitude.

(ii) For each objective function, the boundary solutions are assigned an infinite distance value (a large value).

(iii) All other intermediate solutions are assigned a distance value for the objective, equal to the absolute normalized difference in the objective values of two adjacent solutions.

(iv) This calculation is continued with other objective functions.

(v) The overall distance of a solution from others is calculated as the sum of individual distance values corresponding to each objective.

Since M independent sorting of at most N solutions (In case the subset contains all the solutions of the population) are involved, the above algorithm has $O(MNlogN)$ computational complexity.

Using the above two operations proposed multi-objective genetic algorithm takes the following form:

1. Set probability of crossover p_c and probability of mutation p_m.
2. Set iteration counter $T = 1$.
3. Generate initial population set of solution $P(T)$ of size N.
4. Select solution from $P(T)$ for crossover and mutation.
5. Made crossover and mutation on selected solution and get the child set $C(T)$.
6. Set $P_1 = P(T)UC(T)$ // Here U stands for union operation.
7. Divide P_1 into disjoint subsets having non-dominated solutions. Let these sets be $F_1, F_2, ..F_k$.
8. Select maximum integer n such that order of $P_2(= F_1UF_2U\ ...\ UF_n) \leq N$.
9. if $O(P_2) < N$ sort solutions of F_{n+1} in descending order of their distance from other solutions of the subset. Then select first $N - O(P_2)$ solutions from F_{n+1} and add with P_2, where $O(P_2)$ represents order of P_2.
10. Set $T = T + 1$ and $P(T) = P_2$.
11. If termination condition does not hold go to step-4.
12. Output: P(T)
13. End algorithm.

MOGAs that use non-dominated sorting and sharing are mainly criticized for their

- $O(MN^3)$ computational complexity
- non-elitism approach
- the need for specifying a sharing parameter to maintain diversity of solutions in the population.

In the above algorithm, these drawbacks are overcame. Since in the above algorithm computational complexity of step-7 is $O(MN^2)$, step-9 is $O(MNlogN)$ and other steps are $\leq O(N)$, so overall time complexity of the algorithm is $O(MN^2)$. Here selection of new population after crossover and mutation on old population, is done by creating a mating pool by combining the parent and offspring population and among them, best N solutions are taken as solutions of new population. By this way, elitism is introduced in the algorithm. When some solutions from a non-dominated set F_j (i.e., a subset of F_j) are selected for new population, those are accepted whose distance compared to others (which are not selected) are much i.e., isolated solutions are accepted. In this way taking some isolated solutions in the new population, diversity among the solutions is introduced in the algorithm, without using any sharing function. Since computational complexity of this algorithm $< O(MN^3)$ and elitism is introduced, this algorithm is named as FEMOGA. Time complexity of NSGA can be reduced to $O(MN^2)$ if step-4 of NSGA is done following step-7 of above FEMOGA, but need of sharing function in NSGA can not be removed. Different procedures of the above FEMOGA are discussed in the following section. Procedures for NSGA can easily be developed similarly.

(iii) Procedures of the proposed FEMOGA

(a) Representation: A 'K dimensional real vector' X=(x_1, x_2, ... x_K) is used to represent a solution, where x_1, x_2, ... x_K represent different decision variables of the problem such that constraints of the problem are satisfied.

(b) Initialization: N such solutions $X_1, X_2, X_3, ...X_N$ are randomly generated by random number generator from the search space such that each X_i satisfies the constraints of the problem. This solution set is taken as initial population P(1).

(c) Crossover:

(i) **Selection for crossover:** For each solution of $P(T)$ generate a random number r from the range [0,1]. If $r < p_c$ then the solution is taken for crossover.

(ii) **Crossover process:** Crossover taken place on the selected solutions. For each pair of coupled solutions Y_1, Y_2 a random number c is generated from the range [0,1] and offsprings Y_{11} and Y_{21} are calculated by $Y_{11} = cY_1 + (1-c)Y_2$, $Y_{21} = cY_2 + (1-c)Y_1$.

.(d) Mutation:

(i) **Selection for mutation:** For each solution of P(T) generate a random number r from the range [0..1]. If $r < p_m$ then the solution is taken for mutation.

(ii) **Mutation process:** To mutate a solution $X = (x_1, x_2, x_3, \; ... \; x_K)$ select a random integer r in the range [1, K]. Then replace x_r by randomly generated value within the boundary of r^{th} component of X.

(e) Division of $P(T)$ into disjoint subsets having non-dominated solutions: Following the discussions of the previous section the following algorithm is developed for this purpose-

```
For every x ∈ P(T) do
    Set S_x = Φ ,where Φ represents null set
    n_x = 0
    For every y ∈ P(T) do
       If x dominates y then
          S_x = S_x U{y}
       Else if y dominates x then
          n_x = n_x + 1
       End if
     End For
    If n_x = 0 then
       F_1 = F_1 U{x}
    End If
End For
Set i=1
While F_i ≠ Φ do
    F_{i+1} = Φ
    For every x ∈ F_i do
        For every y ∈ S_x do
          n_y = n_y − 1
          If n_y = 0 then
             F_{i+1} = F_{i+1} U{y}
          End If
       End For
```

End For
i=i+1
End While
Output:$F_1, F_2, ...F_{i-1}$.

(f) Determine distance of a solution of subset F from other solutions:
Following algorithm is used for this purpose-
Set n=number of solutions in F
For every $x \in F$ do
$x_{distance} = 0$
End For
For every objective m do
Sort F, in ascending order of magnitude of m^{th} objective.
$F[1] = F[n] = M$, where M is a big quantity.
For i=2 to n-1 do
$F[i]_{distance} = F[i]_{distance} + (F[i+1].objm - F[i-1].objm)/(f_m^{max} - f_m^{min})$
End For
End For
In the algorithm $F[i]$ represents i^{th} solution of F, $F[i].objm$ represent m^{th} objective value of $F[i]$. f_m^{max} and f_m^{min} represent the maximum and minimum values of m^{th} objective function.

2.1.6 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.

In past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods. Another reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. Particle swarm optimization has been used for approaches that can be used across a wide range of applications, as well as for specific applications focused on a specific requirement.

This is performed by particles in multidimensional space that have a position and a velocity. These particles are flying through hyperspace (i.e., n) and have two essential reasoning capabilities: their memory of their own best position and knowledge of the swarms best, best simply meaning the position with the smallest objective value. Members of a swarm communicate good positions to each other and adjust their own position and velocity based on these good positions.

Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another best value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbours of the particle. This location is called lbest. when a particle takes all the population as its topological neighbors, the best value is a global best and is called gbest.

Consider swarm of particles is flying through the parameter space and searching for optimum. Each particle is characterized by,

Position vector . . . .. x_i (t)

Velocity vector . . . . . . v_i (t)

During the process, each particle will have its individual knowledge pbest, i.e., its own best-so-far in the position and social knowledge gbest i.e., pbest of its best neighbor as

Performing the velocity and position update, using the formula given below,

$$\left\{ \begin{array}{l} V_i(t+1) = wV_i(t) + c_1 r_1 (X_{pbest}(t) - X_i(t)) + c_2 r_2 (X_{gbest}(t) - X_i(t)), \\ X_i(t+1) = X_i(t) + V_i(t+1) \end{array} \right\} \tag{2.1}$$

where α is the inertia weight that controls the exploration and exploitation of the search space. c_1 and c_2, the cognition and social components respectively are the acceleration constants which changes the velocity of a particle towards the pbest and gbest, rand is a random number between 0 and 1.

PSO utilizes several searching points like genetic algorithm (GA) and the searching points gradually get close to the optimal point using their pbests and the gbest. The first term of first equation RHS of Equ. 2.1 is corresponding to diversification in the search procedure. The second and third terms of that are corresponding to intensification in the search procedure. Namely, the method has a well balanced mechanism to utilize diversification and intensification in the search procedure efficiently. The original PSO can be applied to the only continuous problem. However, the method can be expanded to the discrete problem using discrete number position and its velocity easily.

Basic Flow of Particle Swarm Optimization:

The basic operation of PSO is given by,

- Step 1. Initialize the swarm from the solution space
- Step 2. Evaluate fitness of individual particles
- Step 3. Modify gbest, pbest and velocity
- Step 4. Move each particle to a new position
- Step 5. Goto step 2, and repeat until convergence or stopping condition is satisfied

The pseudo code of the procedure is as follows

```
For each particle
    Initialize particle
 END
 Do
    For each particle
        Calculate fitness value
        If the fitness value is better than the best fitness value
          (pbest) in history set current value as the new pbest
    End
    Choose the particle with the best fitness value of all the
     particles as the gbest
    For each particle
     Calculate particle velocity according equation 2.1
     Update particle position according equation 2.1
    End
```

While maximum iterations or minimum error is not attained

Particles velocities on each dimension are clamped to a maximum velocity V_{max}. If the sum of accelerations would cause the velocity on that dimension to exceed V_{max}, which is a parameter specified by the user. Then the velocity on that dimension is limited to V_{max}.

Applications of PSO

PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied. The various application areas of Particle Swarm Optimization include are Power Systems operations and control, NP-Hard combinatorial problems, Job Scheduling problems, Vehicle Routing Problems, Mobile Networking, Modelling optimized parameters, Batch process scheduling, Multi-objective optimization problems and Image processing, Pattern recognition problems and so on. Currently, several researchers are being carried out in the area of particle swarm optimization and hence the application area also increases tremendously.

2.1.7 Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) is a population-based, general search technique for the solution of difficult combinatorial problems, which is inspired by the pheromone trail laying behavior of real ant colonies. In ACO, a set of software agents called artificial ants search for good solutions to a given optimization problem. To apply ACO, the optimization problem is transformed into the problem of finding the best path on a weighted graph. The artificial ants (hereafter ants) incrementally build solutions by moving on the graph. The solution construction process is stochastic and is biased by a pheromone model, that is, a set of parameters associated with graph components (either nodes or edges) whose values are modified at run time by the ants.

Historical study of ACO

In the 40s and 50s of the 20th century, the French entomologist Pierre-Paul Grass observed that some species of termites react to what he called significant

Table 2.1: Development of various ACO Algorithms

ACO Algorithm	Authors	Year
Ant System	Dorigo, Maniezzo & Colomi	1991
Elitist AS	Dorigo	1992
Ant-Q	Gambardella & Dorigo	1995
Ant Colony System	Dorigo & Gambardella	1996
MMAS	Sttzle & Hoos	1996
Rank-based AS	Bullnheimer, Hartl & Strauss	1997
ANTS	Maniezzo	1998
Best-Worst AS	Cordn, et al.	2000
Hyper-cube ACO	Blum, Roli, Dorigo	2001

stimuli. He observed that the effects of these reactions can act as new significant stimuli for both the insect that produced them and for the other insects in the colony. Grass used the term stigmergy to describe this particular type of communication in which the workers are stimulated by the performance they have achieved.

ACO is a class of algorithms, whose first member, called Ant System, was initially proposed by Colorni, Dorigo and Maniezzo. The main underlying idea, loosely inspired by the behavior of real ants, is that of a parallel search over several constructive computational threads based on local problem data and on a dynamic memory structure containing information on the quality of previously obtained result. The collective behavior emerging from the interaction of the different search threads has proved effective in solving combinatorial optimization (CO) problems. Different ant colony optimization algorithms have been proposed. The original ant colony optimization algorithm is known as Ant System and was proposed in the early 90s. Since then, a number of other ACO algorithms were introduced. Table 2.1 gives a list of successful variants of Ant Colony Optimization Algorithms. Also Table 2.2 gives several applications of ACO.

Characteristics of Ant Colony Optimization:

The characteristics of ant colony optimization are as follows:

- **Natural algorithm** since it is based on the behavior of real ants in establishing paths from their colony to source of food and back.
- **Parallel and distributed** since it concerns a population of agents moving simultaneously, independently and without a supervisor
- **Cooperative** since each agent chooses a path on the basis of the information, pheromone trails laid by the other agents, which have previously se-

lected the same path. This cooperative behavior is also auto catalytic, i.e., it provides a positive feedback, since the probability of choosing a path increases with the number of agents that previously chose that path.

- **Versatile** that it can be applied to similar versions of the same problem; for example, there is a straightforward extension from the traveling salesman problem (TSP) to the asymmetric traveling salesman problem (ATSP).
- **Robust** that it can be applied with minimal changes to other combinatorial optimization problems such as quadratic assignment problem (QAP) and the jobshop scheduling problem (JSP).

Ant System:

Ant System is the first ACO algorithm proposed in the literature.Ant System applied to traveling Salesman problem is discussed here. Its main characteristic is that, at each iteration, the pheromone values are updated by all the m ants that have built a solution in the iteration itself. The pheromone ij associated with the edge joining cities i and j, is updated as follows:

$$\tau_{ij} = (1-\rho).\tau_{ij} + \sum \Delta\tau_{ij}^{k} \} \tag{2.2}$$

where ρ is the evaporation rate, m is the number of ants, and $\Delta\tau_{ij}^{k}$ is the quantity of pheromone laid on edge (i, j) by ant k:

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k \quad if\, ant\, k\, used\, edge(i,j)\, in\, its\, tour, \\ 0,\ otherwise \end{cases} \tag{2.3}$$

Where Q is a constant, and Lk is the length of the tour constructed by ant k.

In the construction of a solution, ants select the following city to be visited through a stochastic mechanism. When ant k is in city i and has so far constructed the partial solution s^p probability of going to city j is given by:

$$p_{ij}^{k} = \begin{cases} \frac{\tau_{ij}^{\alpha}.\eta_{ij}^{\beta}}{\sum \tau_{ij}^{\alpha}.\eta_{ij}^{\beta}} \quad if\, c_{ij} \in\ N(s^p), \\ 0,\ otherwise \end{cases} \tag{2.4}$$

where $N(s^p)$ is the set of feasible components; that is, edges (i, 1) where 1 is the city not yet visited by the ant k. The parameters α and β control the relative importance of the pheromone versus the heuristic information η_{ij}, which is given by,

$$\eta_{ij} = \left\{ \frac{1}{d_{ij}} \right. \tag{2.5}$$

where d_{ij} is the distance between cities i and j.

Ant Colony System (ACS)

The most interesting contribution of ACS is the introduction of a local pheromone update in addition to the pheromone update performed at the end of the construction process (called offline pheromone update). The local pheromone update is performed by all the ants after each construction step. Each ant applies it only to the last edge traversed:

$$\tau_{ij} = (1 - \phi).\tau_{ij} + \phi\tau_0 \tag{2.6}$$

where $\phi \in (0, 1)$is the pheromone decay coefficient, and τ_0 is the initial value of the pheromone.

The main goal of the local update is to diversify the search performed by subsequent ants during an iteration: by decreasing the pheromone concentration on the traversed edges, ants encourage subsequent ants to choose other edges and, hence, to produce different solutions. This makes it less likely that several ants produce identical solutions during one iteration. The offline pheromone update is applied at the end of each iteration by only one ant, which can be either the iteration-best or the best-so-far. However, the update formula is:

$$\tau_{ij} = \begin{cases} (1 - \rho).\tau_{ij} + \rho.\Delta\tau_{ij} & if\ (i, j)\ belongs\ to\ best\ tour, \\ \tau_{ij},\ otherwise \end{cases} \tag{2.7}$$

where $\Delta\tau_{ij} = \frac{1}{L_{best}}$, where L_{best} can be either L_{ib} or L_{bs}. L_{best} is the length of the tour of the best ant. This may be (subject to the algorithm designer decision) either the best tour found in the current iteration-best, L_{ib} -or the best solution found since the start of the algorithm-so-far, L_{best} or a combination of both.

Another important difference between ACS and AS is in the decision rule used by the ants during the construction process. In ACS, the so-called pseudo random proportional rule is used: the probability for an ant to move from city i to city j depends on a random variable q uniformly distributed over [0, 1], and a parameter q_0; if q $\leq q_0$, then j = $\text{argmax}_{cij \in N(s^p)}\{\tau_{il}\eta_{il}^{\beta}\}$ otherwise Equ. 2.3 is used.

Basic Flow of ACO

The basic operational flow in Ant Colony Optimization is as follows

- Step1. Represent the solution space by a construction graph
- Step 2. Initialize ACO parameters
- Step 3. Generate random solutions from each ants random walk
- Step 4. Update pheromone intensities
- Step 5. Goto Step 3, and repeat until convergence or a stopping condition is satisfied

The step by step procedure to solve combinatorial optimization problems using ACO in a nutshell is:

- Represent the problem in the form of sets of components and transitions or by means of a weighted graph that is travelled by the ants to build solutions.
- Appropriately define the meaning of the pheromone trails, i.e., the type of decision they bias. This is a crucial step in the implementation of an ACO algorithm. A good definition of the pheromone trails is not a trivial task and it typically requires insight into the problem being solved.
- Appropriately define the heuristic preference to each decision that an ant has to take while constructing a solution, i.e., define the heuristic information associated to each component or transition. Notice that heuristic information is crucial for good performance if local search algorithms are not available or can not be applied.
- If possible, implement an efficient local search algorithm for the problem under consideration, because the results of many ACO applications to NP-hard combinatorial optimization problems show that the best performance is achieved when coupling ACO with local optimizers.
- Choose a specific ACO algorithm and apply it to the problem being solved, taking the previous aspects into consideration.

Table 2.2: Applications of ACO algorithms

Problem Type	Problem name	Authors	Year
Routing	TSP	Dorigo et al.	1991, 1996
	Multi-objective TSP	Ariyasingha et al.	2015
	Evacuation path optimization	Liu & Zhang	2016
	Dynamic location routing	Gao et al.	2016
	Vehicle routing	Dorigo & Gambardella	1997
		Stotzile& Hods	1997,2000
		Gambardella et al.	1999
		Reimann et al.	2004
Assignment	Sequential Ordering	Gambardella & Dorigo	2000
	Order batching	Cheng et al.	2015
	Quadratic Assignment	Stutzle & Hobbs	2000
		Maeinezzo	1999
Scheduling	Course Timetabling	Socha et al	2002,2003
	Graph Coloring	Costa & Hertz	1997
	Project Scheduling	Merkle et al	2002
	Total Weighted Tardiness	Den Bestern et al	2000
	Total Weighted Tardiness	Merkle & Midderdonf	2000
	Open Shop	Blum	2005
	Grid scheduling	Tiwari & Vidyarthi	2016
Subset	Set Covering	Lessing et al	2004
	j-Cardinality Tree	Blum & Blesa	2005
	Multiple Knapsack	Leguizamon & Michlewicz	1999
	Maximum Clique	Fenet & Solnon	2003
Other	Constraint Stratification	Solnon	2000,2002
	Classification Rules	Parpenlie et al	2002
	Bayesian Network	& Campos et al	2002
	Protein Folding	Shymgelska	2005
	Protein-Ligand Docking	Korb et al	2006
	High Dimensional Design	Borrotti et al.	2016
	Software Design	Tawosi et al.	2015
	Graph Clustering	Moradi et al.	2015
	Traffic Control	Dias & Machado et al.	2014
	Inventory Control	Nia & Far et al.	2014
	Pattern Recognition	Liu et al.	2015

2.2 Some Hybrid Heuristics

2.2.1 Introduction

Traditional methods of optimization are not robust to dynamic changes in the environment and they require a complete restart for providing a solution. In contrary, evolutionary computation can be used to adapt solutions to changing circumstances. Hybridization of evolutionary algorithms is getting popular due to their capabilities in handling several real world problems involving complexity, noisy environment, imprecision, uncertainty and vagueness. Usually grouped under the term evolutionary computation or evolutionary algorithms, we find the domains of genetic algorithms [68], evolution strategies [144, 151], evolutionary programming [54], and genetic programming [85, 86]. They all share a common conceptual base of simulating the evolution of individual structures via processes of selection, mutation, and reproduction.

For several problems, a simple Evolutionary algorithm might not be good enough to find the desired solution. As reported in the literature, there are several types of problems where a direct evolutionary algorithm could fail to obtain a convenient (optimal) solution [95, 106, 157, 161]. This clearly paves way to the need for hybridization of evolutionary algorithms with other optimization algorithms, machine learning techniques, heuristics etc. Some of the possible reasons for hybridization are as follows [155]:

- To improve the performance of the evolutionary algorithm (example: speed of convergence)
- To improve the quality of the solutions obtained by the evolutionary algorithm
- To incorporate the evolutionary algorithm as part of a larger system

From a problem solving perspective, it is difficult to formulate a universal optimization algorithm that could solve all the problems. Hybridization may be the key to solve practical problems. Evolutionary algorithms may be hybridized by using operators from other algorithms (or algorithms themselves) or by incorporating domain-specific knowledge. Adaptive evolutionary algorithms have been built for inducing exploitation/exploration relationships that avoid the premature convergence problem and optimize the final results.

As reported in the literature, several techniques and heuristics/meta heuristics have been used to improve the general efficiency of the evolutionary algorithm. Some of most used hybrid architectures are summarized as follows:

- Hybridization between an evolutionary algorithm and another evolutionary algorithm (example: a genetic programming technique is used to improve the performance of a genetic algorithm)
- Neural network assisted evolutionary algorithms
- Fuzzy logic assisted evolutionary algorithm
- Particle swarm optimization (PSO) assisted evolutionary algorithm
- Ant colony optimization (ACO) assisted evolutionary algorithm
- Bacterial foraging optimization assisted evolutionary algorithm
- Hybridization between evolutionary algorithm and conventional optimization techniques
- Hybridization between evolutionary algorithm and other heuristics (such as local search, tabu search, simulated annealing, hill climbing, dynamic programming, greedy random adaptive search procedure, etc).

The integration of different learning and adaptation techniques, to overcome individual limitations and achieve synergetic effects through hybridization or fusion of these techniques, has in recent years contributed to a large number of new hybrid evolutionary systems. Most of these approaches, however, follow an ad hoc design methodology, further justified by success in certain application domains. Due to the lack of a common framework, it remains often difficult to compare the various hybrid systems conceptually and evaluate their performance comparatively. There are several ways to hybridize a conventional evolutionary algorithm for solving optimization problems.

Tan et al. [158] proposed a two-phase hybrid evolutionary classification technique to extract classification rules that can be used in clinical practice for better understanding and prevention of unwanted medical events. In the first phase, a hybrid evolutionary algorithm is used to confine the search space by evolving a pool of good candidate rules. Zmuda et al.[190] proposed an hybrid evolutionary learning scheme for synthesizing multiclass pattern recognition systems. A considerable effort is spent for developing complex features that serve as inputs to a simple classifier back end. The nonlinear features are created using a combination of genetic programming [85, 86] to synthesize arithmetic expressions, genetic algorithms [68] to select a viable set of expressions, and evolutionary programming [55, 56] to optimize parameters within the expressions. PSO incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior, from which the idea is emerged [29, 81, 82]. The main steps of the hybrid approach are depicted below [153]:

- Initialize EA and PSO subsystems.
- Execute EA and PSO simultaneously.
- Memorize the best solution as the final solution and stop if the best individual in one of the two subsystems satisfies the termination criterion.
- Perform the hybrid process if generations could be divided exactly by the designated number of iterations N. Select P individuals from both sub systems randomly according to their fitness and exchange.

A hybrid technique combining GA and PSO called genetic swarm optimization (GSO) was proposed by Grimaldi et al. [63] for solving an electromagnetic optimization problem. The method consists of a strong co-operation of GA and PSO, since it maintains the integration of the two techniques for the entire run. In each iteration, the population is divided into two parts and they are evolved with the two techniques, respectively.

Tseng and Liang [161] proposed a hybrid approach that combines (ACO), the genetic algorithm (GA) and a Local Search (LS) method. The algorithm is applied for solving the Quadratic Assignment Problem (QAP). Instead of starting from a population that consists of randomly generated chromosomes, GA has an initial population constructed by ACO in order to provide a good start. Pheromone acts as a feedback mechanism from GA phase to ACO phase. When GA phase reaches the termination criterion, control is transferred back to ACO phase. Then ACO utilizes pheromone updated by GA phase to explore solution space and produces a promising population for the next run of GA phase. The local search method is applied to improve the solutions obtained by ACO and GA. Another hybrid approach for the same problem were proposed by Vasquez and Whitley [165] where GA is combined with Tabu Search. Ahuja et al. [1]

used a greedy genetic algorithm. Recently Renato et al. [145] prosed an hybrid evolutionary approach for Traveling Car Renter Problem. Prodhon et al. [138] proposed an hybrid approaches for periodic location-routing problem. Ma et al. [109] described a Hybrid biogeography-based evolutionary algorithms. Koc et al. [83] approached a hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. A Hybrid evolutionary fuzzy learning scheme in the applications of TSPs is proposed by Feng et al [51]. Psychas et al. [131] proposed a hybrid technique for multi-objective TSP.

2.2.2 ACO-GA

This part is presented in Chapter 5 for Model 5.1.

2.2.3 ACO-PSO-GA

This part is presented in Chapter 5 in for Model 5.2.

Chapter 3

Some Uncertain Environments

3.1 Crisp Set Theory

Crisp Set: By crisp one mean dichotomous, that is, yes or no type rather than more-or-less type. In conventional dual logic, for instance, a statement can be true or false and nothing in between. In set theory, an element can either belongs to a set or not; and in optimization, a solution is either feasible or not. A classical set, X, is defined by crisp boundaries, i.e., there is no uncertainty in the prescription of the elements of the set. Normally it is defined as a well defined collection of elements or objects, $x \in X$, where X may be countable or uncountable.

Convex Set: A subset $S \subset \Re^n$ is said to be convex, if for any two points x_1, x_2 in S, the line segment joining the points x_1 and x_2 is also contained in S. In other words, a subset $S \subset \Re^n$ is convex, if and only if

$$x_1, x_2 \in S \Rightarrow \lambda x_1 + (1-\lambda)x_2 \in S; \quad 0 \leq \lambda \leq 1.$$

Convex Combination: Given a set of vectors $\{x_1, x_2, \cdots, x_n\}$, a linear combination $x = \lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_n x_n$ is called a convex combination of the given vectors, if $\lambda_1, \lambda_2, \cdots, \lambda_n \geq 0$ and $\sum\limits_{i=1}^{n} \lambda_i = 1$.

Convex function: The function $f : S \rightarrow \Re$ is said to be convex if for any $x_1, x_2 \in S$ and $0 \leq \lambda \leq 1$, implies that

$$f\{(1-\lambda)x_1 + \lambda x_2\} \leq (1-\lambda)f(x_1) + \lambda f(x_2).$$

The Graphical representation of Convex Function is depicted in Figure 3.1.

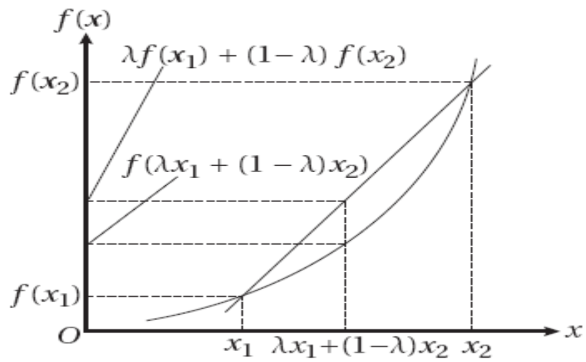


Figure 3.1: Graphical representation of Convex Function

Quasi-convex function: The function $f(x)$ is said to be quasi-convex if for any $x_1, x_2 \in S$ and $0 \leq \lambda \leq 1$,

$$f((1-\lambda)x_1 + \lambda x_2) \leq \text{max } (f(x_1), f(x_2)).$$

It is noted that a convex function is also quasi-convex since

$$f((1-\lambda)x_1 + \lambda x_2) < (1-\lambda)f(x_1) + \lambda f(x_2) < \text{max } (f(x_1), f(x_2)).$$

Pseudo-convex function: The function $f(x)$ is said to be pseudo-convex function if for any $x_1, x_2 \in S$, $f(x_2) \geq f(x_1)$ implies that $(x_2 - x_1)^T \nabla f(x_1) \geq 0$.

The definition of convex functions can be modified for concave functions by replacing $' \leq'$ by $' \geq'$. Correspondingly, the definition of quasi-convex functions becomes appropriate for quasi-concave functions by the exchange of $' \leq'$ to $' \geq'$ and ‘max’to ‘min’. In the definition of pseudo-convex functions, $' \geq'$ is replaced by $' \leq'$ to get the definition for pseudo-concave functions.

3.2 Interval Arithmetic

An order pair of brackets defines an interval $A = [a_L, a_R] = \{a : a_L \leq a \leq a_R\}$ where a_L and a_R are respectively left and right limits of A. Throughout this section lower case letters denote real numbers and upper case letter denote closed intervals.

To represent an unknown number as an approximation plus/minus an error bound, the midpoint $\check{A}$ and with of an interval $\mathbf{A}$ are respectively introduced as

$\check{A} \equiv mid(x) = \frac{a_L + a_R}{2}$, and wid(A)=$a_R - a_L$.

Hence **A** can be represented as

Á=[$\check{A}$, wid(**A**)]=$\langle$ a_c, $a_w\rangle$.

Definition 3.1 Let $* \in \{+, -, ., /\}$ be a binary operation on the set of positive real numbers. If A and B are closed intervals then $A*B = \{a*b : a \in A, b \in B\}$ defines a binary operation on the set of closed intervals [119]. In the case of division, it is assumed that $0 \notin B$. The operations on intervals used here may be explicitly calculated from the above definition as

$$\begin{aligned} A + B &= [a_L, a_R] + [b_L, b_R] = [a_L + b_L, a_R + b_R] \\ A - B &= [a_L, a_R] - [b_L, b_R] = [a_L - b_R, a_R - b_L] \\ A.B &= [a_L, a_R].[b_L, b_R] = [min\{a_Lb_L, a_Lb_R, a_Rb_L, \qquad (3.1) \\ &\quad a_Rb_R\}, max\{a_Lb_L, a_Lb_R, a_Rb_L, a_Rb_R\}] \\ \frac{A}{B} &= \frac{[a_L, a_R]}{[b_L, b_R]} = [a_L, a_R].[\frac{1}{b_R}, \frac{1}{a_R}], \qquad (3.2) \end{aligned}$$

where $0 \notin B$

$$kA = \begin{cases} [ka_L, ka_R], \text{ for } k \geq 0 \\ (ka_R, ka_L), \text{ for } k < 0 \\ \text{where } k \text{ is a real number.} \end{cases}$$

Order relations between intervals:

Here, the order relations which represent the decision-maker's preference between interval costs are defined for minimization problems. Let the uncertain costs for two alternatives be represented by intervals A and B respectively. It is assumed that the cost of each alternative is known only to lie to the corresponding interval. The order relation by the left and right limits of interval is defined in Definition 3.2.

Definition 3.2 The order relation $\leq_{LR}$ between $A = [a_L, a_R]$ and $B = [b_L, b_R]$ is defined as

$$A \leq_{LR} B \text{ iff } a_L \leq b_L \text{ and } a_R \leq b_R$$
$$A <_{LR} B \text{ iff } A \leq_{LR} B \text{ and } a_R \neq b_R$$

The order relation $\leq_{LR}$ represents the DM's performance for the alternative with the lower minimum cost, that is, if $A \leq_{LR} B$, then A is preferred to B.

The operations on intervals used in this thesis may be explicitly calculated for two interval numbers, A=[a_L,a_R], B=[b_L,b_R] and $\acute{A}$=$\langle$ a_c,$a_w\rangle$, $\acute{B}$=$\langle$ b_c,$b_w\rangle$ from above definition as:

$$\left.\begin{array}{c}\acute{A}+\acute{B}=\langle\, a_c,\, a_w\rangle+\langle\, b_c,\, b_w\rangle=\langle\, a_c+b_c,\, a_w+b_w\rangle,\\ =[ka_R,\, ka_L],\text{ for k} < 0,\\ k\acute{A}=k\langle\, a_c,a_w\rangle=\langle\, ka_c,|k|a_w\rangle, where\, k\, is\, a\, real\, number\end{array}\right\}\quad(3.3)$$

Let the uncertain costs from two alternatives be represented by two closed intervals $\acute{A}$=$\langle$ a_c,$a_w\rangle$, $\acute{B}$=$\langle$ b_c,$b_w\rangle$ respectively. It also assumed that the cost of each alternative lies in the corresponding interval. These two intervals $\acute{A}$ and $\acute{B}$ any be of the following three types:

Type I: Both the intervals are disjoint.

Type II: Intervals are partially overlapping.

Type III: One interval is contained in the other.

In optimistic decision making, the decision maker (DM) expects the lowest cost ignoring the uncertainty. According to Majumder et al. [111] the order relations of the interval numbers for minimization problems in case of optimistic decision making are as follows:

Definition 3.3 (Majumder et al. [111]) Let us define the order relation $\leq_{omin}$ between A=$[a_L, a_R]$ and B=$[b_L, b_R]$ as

$$\left.\begin{array}{c}A \leq ominB \leftrightarrow a_L \leq b_L\\ A \leq_{omin} B \leftrightarrow A \leq_{omin} B \wedge A \neq B.\end{array}\right\}\quad(3.4)$$

Pessimistic decision making

For pessimistic decision making, the DM expects the minimum cost for minimization problems according to the principle "Less uncertainty is better than more uncertainty". According to Karmakar et al. [78] and Majumder et al. [111], the order relations of interval numbers for minimization problems in case of pessimistic decision making are as follows:

Definition 3.4 (Majumder et al. [111]) Let us define the order relation $\leq_{pmin}$ between $\acute{A}$=$\langle$ a_c,$a_w\rangle$ and $\acute{B}$=$\langle$ b_c,$b_w\rangle$ as

$$\left.\begin{array}{c}\acute{A} \leq_{pmin} \acute{B} \leftrightarrow a_c \leq b_c\text{ for Type I and Type II intervals}\\ \acute{A} \leq_{pmin} \acute{B} \leftrightarrow (a_c \leq b_c)\wedge(a_w < b_w)\text{ for Type III intervals.}\end{array}\right\}\quad(3.5)$$

Table 3.1: Probability Distribution

Discrete distribution	Continuous distribution
Discrete uniform distribution	Uniform (or rectangular) distribution
Binomial distribution	Normal (or Gaussian) distribution
Geometric distribution	Gamma distribution
Multimodular distribution	Exponential distribution
Poisson distribution	Laplace distribution
Hypergeometric distribution	Weibull distribution
Negative binomial	Rayleigh distribution
or Pascal's distribution	Beta distribution

However, for Type III intervals with $(a_c \leq$ b $_c) \wedge (a_w < b_w)$, the pessimistic decision cannot be taken. Here, the optimistic decision is to be considered.
Remark 3.2.1: Now as the interval valued objectives are not well defined, so we use common features of arithmetic mean(AM) and geometric mean(GM) as follows:
Let $A = [a_L, a_R]$ be a common interval for a particular objective function. Since we know that for a minimization problem,

$$\left. AM \geq GM \quad \Rightarrow \frac{m_1 * a_L + m_2 * a_R}{m_1 + m_2} \geq (a_L^{m_1} * a_R^{m_2})^{\frac{1}{m_1+m_2}} \right\} \tag{3.6}$$

determine the minimum of the objective function $(a_L^{m_1} * a_R^{m_2})^{\frac{1}{m_1+m_2}}$, here m_1 and m_2 are the given weights.

3.3 Probability Distribution

There are several types of probability distributions for describing various types of discrete and continuous random variables. Some of common distributions are shown in Table-3.1. In any physical problem, one chooses a particular type of probability distribution depending on (i) the nature of the problem, (ii) the underlying assumptions associated with the distribution of the parameters, (iii) the shape of the graph between the probability density function $f(x)$ (or distribution function $F(x)$) and x obtained after plotting the available data and (iv) the convenience and simplicity afforded by the distribution. In this thesis, only Normal distributions have been used.
Normal Distribution
The best known and most widely used probability distribution is the Normal

distribution. The density function of the normal distribution is a bell-shaped symmetrical curve about mean and its probability density function with parameters m and $\sigma(>0)$ is defined as:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} exp\left\{\frac{-(x-m)^2}{2\sigma^2}\right\}$$

where $-\infty < x < \infty$, mean = m and variance = σ^2.

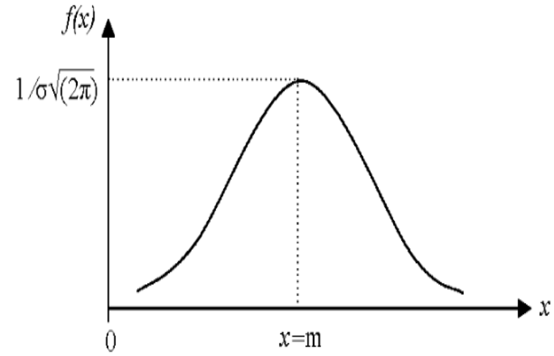


Figure 3.2: Graphical representation Normal Distrubition

The notation N(m,σ) is usually used to represent a normal distribution with mean m and standard deviation σ and its density function is a bell-shaped symmetrical curve about m (cf., Fig 3.2).

3.4 Fuzzy Set Theory

The notion of fuzzy set has been introduced by Lotfi Zadeh [178] in order to formalize the concept of gradedness in class membership, in connection with the representation of human knowledge. It was developed to define and solve the complex system with sources of uncertainty or imprecision which are non-statistical in nature. The term **FUZZY** was proposed by Prof. L, A. Zadeh in 1962 (Zadeh [177]). A short delineation of the fuzzy set theory is given below.

Definition 3.5 α **- Cut of a fuzzy number**

A α- cut of a fuzzy number $\tilde{A}$ in X is denoted by A_α and is defined as the following crisp set (cf. Fig 3.5):

$$A_\alpha = \{x : \mu_{\tilde{A}}(x) \geq \alpha, x \in X\} \text{ where } \alpha \in [0, 1]$$

A_α is a non-empty bounded closed interval contained in X and it can be denoted by $A_\alpha = [A_L(\alpha), A_R(\alpha)]$. $A_L(\alpha)$ and $A_R(\alpha)$ are the lower and upper bounds of the closed interval respectively. Fig 3.5 shows a fuzzy number $\tilde{A}$ with α-cuts $A_{\alpha_1} = [A_L(\alpha_1), A_R(\alpha_1)], A_{\alpha_2} = [A_L(\alpha_2), A_R(\alpha_2)]$. It is seen that if $\alpha_2 \geq \alpha_1$ then $A_L(\alpha_2) \geq A_L(\alpha_1)$ and $A_R(\alpha_1) \geq A_R(\alpha_2)$.

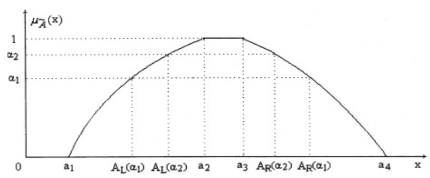


Figure 3.3: α-cut of a general fuzzy number

Definition 3.6 Fuzzy number (FN)

A fuzzy number is a special class of a fuzzy sets. Different definitions and properties of fuzzy numbers are encountered in the literature but they all agree on that a fuzzy number represents the conception of "a set of real numbers close to a", where 'a' is the number being fuzzyfied.

A fuzzy number is a fuzzy set in the universe of discourse X that is both convex and normal. Figure 3.5 shows a fuzzy number $\tilde{A}$ of the universe of discourse X that is both convex and normal. The term "fuzzy number" is used to handle imprecise numerical quantities. For example, shortage cost of a commodity is about 5 \$. A general definition of a fuzzy number according to Dubois and Prade [44] is a real fuzzy number $\tilde{A}$ described as a fuzzy subset on the real line $\Re$ whose membership function $\mu_{\tilde{A}}(x)$ is

(i) a continuous mapping from $\Re$ to the closed interval [0,1],

(ii) constant on $(-\infty, a_1] : \mu_{\tilde{A}}(x) = 0, \forall x \in (-\infty, a_1]$,

(iii) strictly increasing on $[a_1, a_2]$: e.g., $\mu_{\tilde{A}}(x) = f(x), \forall x \in [a_1, a_2]$ where $f(x)$ is a strictly increasing function of x,

(iv) constant on $[a_2, a_3]$: e.g., $\mu_{\tilde{A}}(x) = 1, \forall x \in [a_2, a_3]$,

(v) strictly decreasing on $[a_3, a_4]$, e.g., $\mu_{\tilde{A}}(x) = g(x), \forall x \in [a_3, a_4]$ where $g(x)$ is a strictly decreasing function of x,

(vi) constant on $[a_4, \infty)$: e.g., $\mu_{\tilde{A}}(x) = 0, \forall x \in [a_4, \infty)$.

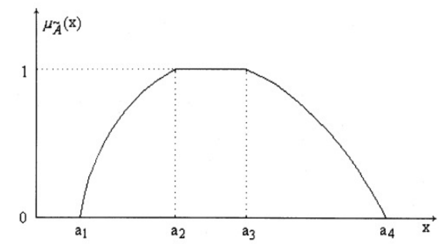


Figure 3.4: Membership function of a General Fuzzy number

A general shape of a fuzzy number following the above definition may be shown pictorially as in Figure 3.6. Here, a_1, a_2, a_3 and a_4 are real numbers. A fuzzy number $\widetilde{A}$ in X is said to be discrete or continuous according as its membership function $\mu_{\widetilde{A}}(x)$ is discrete or continuous. Triangular Fuzzy Number, Trapezoidal Fuzzy Number, Parabolic Fuzzy Number and Parabolic Flat Fuzzy Number are a special class of continuous fuzzy numbers.

Definition 3.7 Linear Fuzzy Number (LFN)
A LFN $\widetilde{A}$ is specified by two parameters (a_1, a_2) and is defined by its continuous membership function $\mu_{\widetilde{A}}(x) : X \to [0, 1]$ as follows (cf. Figure 3.5):

$$\mu_{\widetilde{A}}(x) = \begin{cases} 1 & \text{if } x \leq a_1 \\ \dfrac{a_2 - x}{a_2 - a_1} & \text{if } a_1 \leq x \leq a_2 \\ 0 & \text{if } x \geq a_2 \end{cases}$$

Definition 3.8 Triangular Fuzzy Number (TFN)
A TFN $\widetilde{A}$ is specified by the triplet (a_1, a_2, a_3) and is defined by its continuous membership function $\mu_{\widetilde{A}}(x) : X \to [0, 1]$ as follows (cf. Fig 3.6):

$$\mu_{\widetilde{A}}(x) = \begin{cases} \dfrac{x - a_1}{a_2 - a_1} & \text{if } a_1 \leq x \leq a_2 \\ \dfrac{a_3 - x}{a_3 - a_2} & \text{if } a_2 \leq x \leq a_3 \\ 0 & \text{otherwise} \end{cases}$$

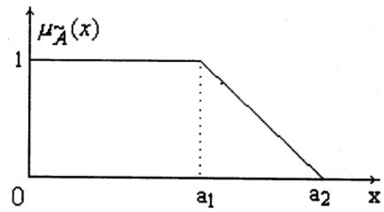


Figure 3.5: Membership function of a LFN

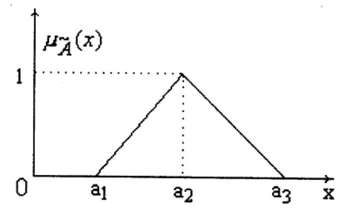


Figure 3.6: Triangular Fuzzy Number (TFN)

Fuzzy Possibility and Necessity Approach:

Let $\tilde{a}$ and $\tilde{b}$ be two fuzzy numbers with membership functions $\mu_{\tilde{a}}(x)$ and $\mu_{\tilde{b}}(x)$ respectively. Then according to Zadeh [183],

$$pos(\tilde{a}*\tilde{b}) = sup\{min(\mu_{\tilde{a}}(x), \mu_{\tilde{b}}(y)), x, y \in \Re, x*y\}, nes(\tilde{a}*\tilde{b}) = 1 - pos\overline{(\tilde{a}*\tilde{b})} \tag{3.7}$$

where the abbreviation pos represents possibility, nes represents necessity $*$ is any one of the relations $>, <, =, \leq, \geq$ and $\Re$ represents set of real numbers.

A TFN $\tilde{a} = (a_1, a_2, a_3)$ (cf. Fig. A1) has seven parameters a_1, a_{11}, a_{12}, a_2, a_{21}, a_{22}, a_3 where $a_1 < a_{11} < a_{12} < a_2 < a_{21} < a_{22} < a_3$ and is characterized by the membership function $\mu_{\tilde{a}}$, given by

$$\mu_{\tilde{a}}(x) = \begin{cases} \frac{x-a_1}{a_{11}-a_1} & \text{for } a_1 \leq x \leq a_{11} \\ \frac{x-a_{11}}{a_{12}-a_{11}} & \text{for } a_{11} < x \leq a_{12} \\ \frac{x-a_{12}}{a_2-a_{12}} & \text{for } a_{12} < x \leq a_{21} \\ \frac{x-a_{21}}{a_{22}-a_{21}} & \text{for } a_{21} < x \leq a_{22} \\ \frac{x-a_{22}}{a_3-a_{22}} & \text{for } a_{22} < x \leq a_3 \\ 0 & \text{otherwise.} \end{cases} \tag{3.8}$$

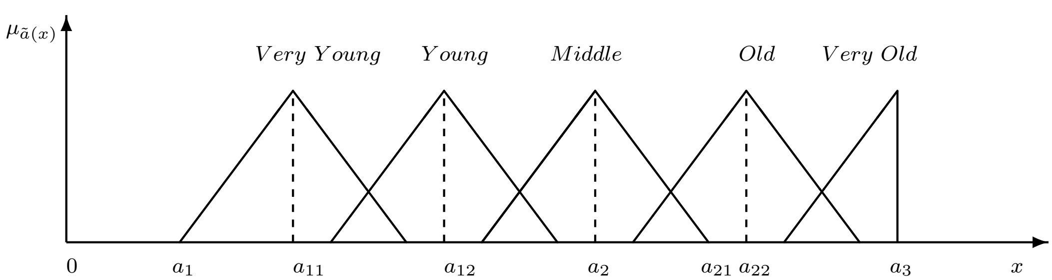


Fig-A1: Extended age distribution of Triangular Fuzzy Number as linguistic variable

3.4.1 Expected Value of a Fuzzy Variable:

Based on the credibility measure, Liu [105] have been presented the expected value operator of a fuzzy variable as follows.

Definition 3.9 Let $\tilde{X}$ be a normalized fuzzy variable the expected value of the fuzzy variable $\tilde{X}$ is defined by

$$E[\tilde{X}] = \int_{0}^{\infty} Cr(\tilde{X} \geq r)dr - \int_{-\infty}^{0} Cr(\tilde{X} \leq r)dr \tag{3.9}$$

When the right hand side of (3.9) is of form $\infty - \infty$, the expected value is not defined. Also, the expected value operation has been proved to be linear for bounded fuzzy variables, i.e., for any two bounded fuzzy variables $\tilde{X}$ and $\tilde{Y}$, we have $E[\,a\tilde{X} + b\tilde{Y}\,] = aE[\tilde{X}] + bE[\tilde{Y}]$ for any real numbers a and b.

Lemma 3.1 (Liu [105])The expected value of triangular fuzzy variable $\tilde{A} = (a_1, a_2, a_3)$ is defined as

$$\begin{aligned} E[\tilde{A}] &= \frac{1}{2}[(1-\rho)a_1 + a_2 + \rho a_3] & (3.10) \\ &= \frac{1}{4}[a_1 + 2a_2 + a_3], \text{taking } \rho = 0.5 & (3.11) \end{aligned}$$

3.4.2 Graded Mean and Modified Graded Mean:

Graded Mean (Chen and Hasieh [28]) Integration Representation method is based on the integral value of graded mean α-level(cut) of generalized fuzzy number. For a fuzzy number $\tilde{A}$ the graded mean integration representation of A is denoted and defined as

$$P(A) = \int_0^1 \alpha \left[\frac{A_\alpha^L + A_\alpha^R}{2}\right] d\alpha / \int_0^1 \alpha d\alpha \tag{3.12}$$

where A_α^L, A_α^R is the α-cut of $\tilde{A}$. For example graded mean of a TrFN $\tilde{A} = (a_1, a_2, a_3, a_4)$ is $\frac{1}{6}[a_1 + a_2 + a_3 + a_4]$. Here, equal weightage has been given to the lower and upper bounds of the α-level of the fuzzy number. But the weightage may depends on the decision maker's preference or attitude. So, the modified graded mean α-level value of the fuzzy number $\tilde{A}$ is $[k\alpha^L + (1-k)A_\alpha^R]$, where $k \in [0; 1]$ is called the decision makers attitude or optimism parameter. The value of k closer to 0 implies that the decision maker is more pessimistic while the value of k closer to 1 means that the decision maker is more optimistic. Therefore, the modified form of the above graded mean integration representation is

$$P(A) = \int_0^1 \alpha \left[\frac{kA_\alpha^L + (1-k)A_\alpha^R}{2}\right] d\alpha / \int_0^1 \alpha d\alpha \qquad (3.13)$$

where A_α^L, A_α^R is the α-cut of $\tilde{A}$. For example modified graded mean of a TrFN $\tilde{A} = (a_1, a_2, a_3, a_4)$ is $\frac{1}{3}[k(a_1 + 2a_2) + (1-k)(2a_3 + a_4)]$.

3.4.3 Possibility/Necessity/ Credibility in Fuzzy Environments

Considering the degree of membership $\mu_F(u)$ of an element u in a fuzzy set F, defined on a referential U, one can find in the literature, three interpretations of this degree (Dubois and Prade [45]).

Degree of similarity: According to degree of similarity, $\mu_F(u)$ is the degree of proximity of u to prototype elements of F. Historically, this is the oldest semantics of membership grades since Bellman *et al.*[8].

Degree of preference: According to degree of preference, F represents a set of more or less preferred objects (or values of a decision variable x) and $\mu_F(u)$ represents an intensity of preference in favor of object u, or the feasibility of selecting u as a value of x. Fuzzy sets then represent criteria or flexible constraints. This view is the one later put forward by Bellman and Zadeh [9]; it has given birth to an abundant literature on fuzzy optimization, especially fuzzy linear programming and decision analysis.

Degree of uncertainty: This interpretation was proposed by Zadeh [182] when he introduced the possibility theory and developed his theory of approximate reasoning (Zadeh [182]). $\mu_F(u)$ is then the degree of possibility that a parameter x has value u, given that all that is known about it is that ”x is F”. Then the values encompasses by the support of the membership functions are mutually exclusive, and the membership degrees rank these values in terms of their respective plausibility. Set functions called possibility and necessity measures can be derived so as to rank-order events in terms of unsurprising-ness and acceptance respectively.

Let $\Re$ represents the set of real numbers and $\tilde{A}$ and $\tilde{B}$ be two fuzzy numbers with membership functions $\mu_{\tilde{A}}$ and $\mu_{\tilde{B}}$ respectively. Then taking degree of uncertainty as the semantics of fuzzy number, according to Zadeh [182], Dubois and Prade [44], Liu and Iwamura [92, 97]:

$$\text{Pos }(\tilde{A} \star \tilde{B}) = \sup\{\min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)), x, y \in \Re, x \star y\} \tag{3.14}$$

where the abbreviation Pos represent possibility and $\star$ is any one of the relations $>, <, =, \leq, \geq$. Analogously if $\tilde{B}$ is a crisp number, say b, then

$$\text{Pos }(\tilde{A} \star b) = \sup\{\mu_{\tilde{A}}(x), x \in R, x \star b\} \tag{3.15}$$

On the other hand necessity measure of an event $\tilde{A} \star \tilde{B}$ is a dual of possibility measure. The grade of necessity of an event is the grade of impossibility of the opposite event and is defined as:

$$\text{Nes}(\tilde{A} \star \tilde{B}) = 1 - \text{Pos }(\overline{\tilde{A} \star \tilde{B}}) \tag{3.16}$$

where the abbreviation Nes represents necessity measure and $\overline{\tilde{A} \star \tilde{B}}$ represents complement of the event $\tilde{A} \star \tilde{B}$.
If $\tilde{A}, \tilde{B} \in \Re$ and $\tilde{C} = f(\tilde{A}, \tilde{B})$ where $f : \Re \times \Re \to \Re$ be a binary operation then according to Fuzzy Extension Principle (Zadeh [180], Dubois and Prade [44], membership function $\mu_{\tilde{C}}$ of $\tilde{C}$ is given by

$$\mu_{\tilde{C}}(z) = \sup\ \{\min\ (\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)), x, y \in \Re, \text{ and } z = f(x, y), \forall z \in \Re\} \tag{3.17}$$

Credibility measure is defined as

$$\text{Cr }(\tilde{A} \star \tilde{B}) = \frac{1}{2}\Big(\text{Pos }(\tilde{A} \star \tilde{B}) + \text{ Nec }(\tilde{A} \star \tilde{B})\Big) \tag{3.18}$$

3.4.4 Different Approaches of TFN

Let $\widetilde{A} = (a_1, a_2, a_3)$ and $\widetilde{B} = (b_1, b_2, b_3)$ are two TFNs. From the definition (3.14) and the possibility measure of $(\widetilde{A} \leq \widetilde{B})$ for membership function is as follows

$$Pos_{\mu}(\widetilde{A} \leq \widetilde{B}) = \begin{cases} 1, & a_2 \leq b_2; \\ \frac{b_3 - a_1}{b_3 - b_2 + a_2 - a_1}, & a_2 > b_2, a_1 < b_3; \\ 0, & a_1 \geq b_3. \end{cases}$$

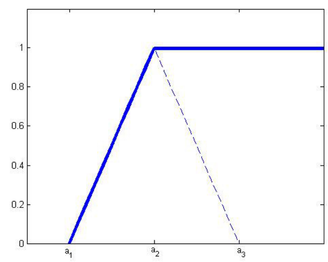


Figure 3.7: Membership of TFN $Pos_\mu(\widetilde{A} \leq x)$

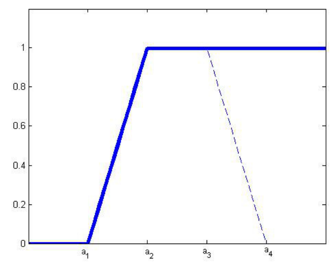


Figure 3.8: Membership of TrFN $Pos_\mu(\widetilde{A} \leq x)$

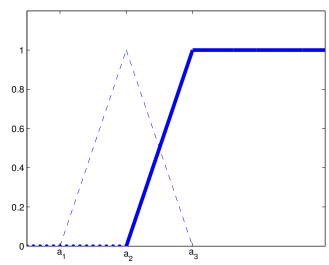


Figure 3.9: Membership of TFN $Nec_\mu(\widetilde{A} \leq x)$

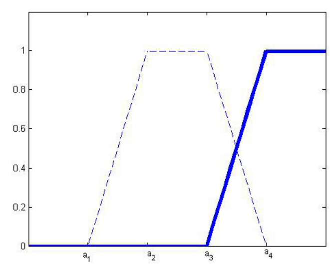


Figure 3.10: Membership of TFN $Nec_\mu(\widetilde{A} \leq x)$

In particular

$$Pos_\mu(\widetilde{A} \leq x) = \begin{cases} 1, & x \geq a_2; \\ \frac{x-a_1}{a_2-a_1}, & a_1 \leq x \leq a_2; \\ 0, & x \leq a_1. \end{cases}$$

Now by definition, the necessity and measure of $(\widetilde{A} \leq x)$ are as follows (depicted in Figure3.9)

$$Nec_\mu(\widetilde{A} \leq x) = \begin{cases} 0, & x \leq a_2; \\ \frac{x-a_2}{a_3-a_2}, & a_2 \leq x \leq a_3; \\ 1, & x \geq a_3. \end{cases}$$

By definition, the credibility measure of $(\widetilde{A} \leq x)$ are as follows (depicted in Figure 3.11)

$$Cr_\mu(\widetilde{A}^I \leq x) = \begin{cases} 0, & x \leq a_1; \\ \frac{x-a_1}{2(a_2-a_1)}, & a_1 \leq x \leq a_2; \\ \frac{x-2a_2+a_3}{2(a_3-a_2)}, & a_2 \leq x \leq a_3; \\ 1, & x \geq a_3. \end{cases}$$

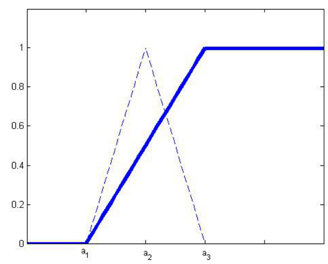


Figure 3.11: Membership of TFN $Cr_\mu(\widetilde{A} \leq x)$

3.5 Rough Set Theory

Let U be a universe. Slowinski and Vanderpooten [156] extended the equivalence relation to more general case and proposed a binary similarity relation that has not symmetry and transitivity but reflexivity.

The similarity class of x, denoted by $R(x)$ and $R^{-1}(x)$, are the set of objects which are similar to x,

$$R(x) = \{y \in U | y \simeq x\}, R^{-1}(x) = \{y \in U | x \simeq y\}$$

Then the lower and the upper approximations of a set are given by the following definition. Let U be a universe, and X a set representing a concept. Then its lower and upper approximation are defined by

$$\underline{X} = \{x \in U | R^{-1}(x) \subset X, \overline{X} = \bigcup_{x \in X} R(x)$$

The collection of all sets having the same lower and upper approximations is called a rough set, denoted by $(\underline{X}, \overline{X})$. Let Λ be a non empty set, A a σ algebra of subsets of Λ, Δ an element in A, and Π a set function satisfying the four axioms. Then $(\Lambda, \Delta, \kappa, \Pi)$ is called a rough space. Let a rough variable ξ is a measurable function from the rough space $(\Lambda, \Delta, \kappa, \Pi)$ to the set of real numbers. i.e. for every Borel set B of $\Re$, $\{\lambda \in \Lambda | \xi(\lambda) \in B\} \in \kappa$. Let $(\Lambda, \Delta, \kappa, \Pi)$ be a rough space. Then the upper and lower trust of an event A is defined by

$$\overline{Tr} = \frac{\Pi\{A\}}{\Pi\{\delta\}}, \underline{Tr} = \frac{\Pi\{A \cap \Delta\}}{\Pi\{\Delta\}}$$

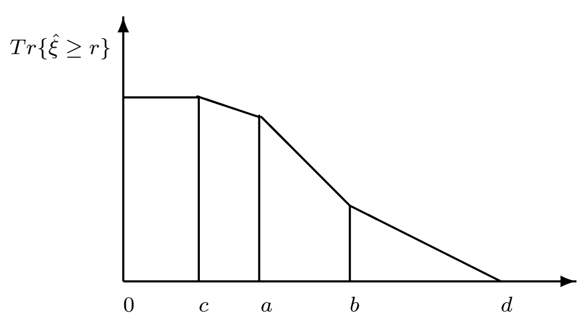
and the trust of the event A is defined by

$$Tr\{A\} = \frac{1}{2}(\underline{Tr}\{A\} + \overline{Tr}\{A\}) \tag{3.19}$$

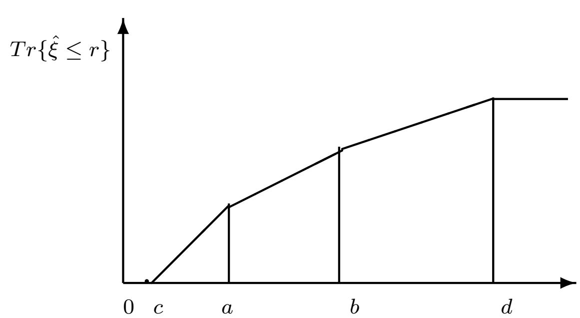
When enough information about the measure Π is not available, it may be treated as the Lebesgue measure. Then we can get the trust measure of the rough event $\hat{\xi} \geq r$ as $Tr\{\hat{\xi} \geq r\}$ and its function curves Figures 3.5.1 and 3.5.2 are presented below where r is a crisp number, $\hat{\xi}$ is a rough variable given by $\hat{\xi}$=([a,b],[c,d]), $0 \leq c \leq a \leq b \leq d$.

$$Tr\{\hat{\xi} \geq r\} = \begin{cases} 0 & for \quad d \leq r \\ \frac{(d-r)}{2(d-c)} & \text{for } b \leq r \leq d \\ \frac{1}{2}(\frac{(d-r)}{(d-c)} + \frac{(b-r)}{(b-a)}) & \text{for } a \leq r \leq b \\ \frac{1}{2}(\frac{(d-r)}{(d-c)} + 1) & \text{for } c \leq r \leq a \\ 1 & \text{for } r \leq c. \end{cases} \tag{3.20}$$

$$Tr\{\hat{\xi} \leq r\} = \begin{cases} 0 & for \quad r \leq c \\ \frac{(r-c)}{2(d-c)} & \text{for } c \leq r \leq a \\ \frac{1}{2}(\frac{(r-c)}{(d-c)} + \frac{(r-a)}{(b-a)}) & \text{for } a \leq r \leq b \\ \frac{1}{2}(\frac{(r-c)}{(d-c)} + 1) & \text{for } b \leq r \leq d \\ 1 & \text{for } d \leq r. \end{cases} \tag{3.21}$$



$Fig.3.5.1 : Tr\{\hat{\xi} \geq r\}$ function curve.



$Fig.3.5.2 : Tr\{\hat{\xi} \leq r\}$ function curve.

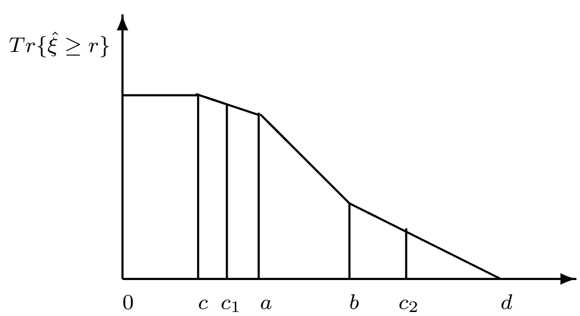
3.5.1 Extension of Trust Measure

Here introduce a new mathematical extension on the rough intervals. We consider a modification/ refinement of the rough intervals. Here we consider in Eqs. 3.22 -3.23 five sub-intervals on the rough intervals (Fig. 3.5.3).

If the interval is divided in more regions, then the trust values of $\hat{\xi}$ is a rough variable given by $\hat{\xi}$=([a,b],[c,d]), $0 \leq c \leq c_1 \leq a \leq b \leq c_2 \leq d$ are given as:

$$Tr\{\hat{\xi} \geq r\} = \begin{cases} 0 & for \quad d \leq r \\ \frac{(d-r)}{3(d-c)} & \text{for } c_2 \leq r \leq d \\ \frac{(d-r)}{3(d-c)} + \frac{(c_2-r)}{3(c_2-c_1)} & \text{for } b \leq r \leq c_2 \\ \frac{1}{3}(\frac{(d-r)}{(d-c)} + \frac{(c_2-r)}{(c_2-c_1)} + \frac{(b-r)}{(b-a)}) & \text{for } a \leq r \leq b \\ \frac{1}{3}(\frac{(d-r)}{(d-c)} + \frac{(c_2-r)}{(c_2-c_1)} + 1) & \text{for } c_1 \leq r \leq a \\ \frac{1}{3}(\frac{(d-r)}{(d-c)} + 2) & \text{for } c \leq r \leq c_1 \\ 1 & \text{for } r \leq c. \end{cases} \tag{3.22}$$

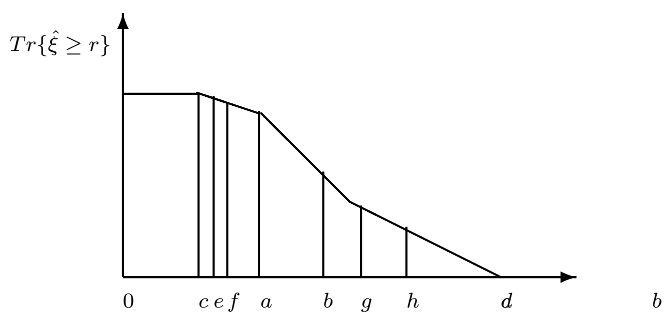
$$Tr\{\hat{\xi} \leq r\} = \begin{cases} 0 & for \quad r \leq c \\ \frac{(r-c)}{3(d-c)} & \text{for } c \leq r \leq c_1 \\ \frac{1}{3}(\frac{(r-c)}{(d-c)} + \frac{(r-c_1)}{(c_2-c_1)}) & \text{for } c_1 \leq r \leq a \\ \frac{1}{3}(\frac{(r-c)}{(d-c)} + \frac{(r-c_1)}{(c_2-c_1)} + \frac{(r-a)}{(b-a)}) & \text{for } a \leq r \leq b \\ \frac{1}{3}(\frac{(r-c)}{(d-c)} + \frac{(r-c_1)}{(c_2-c_1)} + 1) & \text{for } b \leq r \leq c_2 \\ \frac{1}{3}(\frac{(r-c)}{(d-c)} + 2) & \text{for } c_2 \leq r \leq d \\ 1 & \text{for } d \leq r. \end{cases} \tag{3.23}$$



$Fig.3.5.3$: $Tr\{\hat{\xi} \geq r\}$ function curve.

Here the trust measure for 7-point scale of the rough event $\hat{\xi} \geq r$, $Tr\{\hat{\xi} \geq r\}$ and its function curve (Fig 3.5.4) is presented, where r is a crisp number, $\hat{\xi}$ is a rough variable given by $\hat{\xi}$=([a,b],[c,d]), $0 \le c \le e \le f \le a \le b \le g \le h \le d$.

$$Tr\{\hat{\xi} \ge r\} = \begin{cases} 0 & for \quad d \le r \\ \frac{(d-r)}{4(d-c)} & \text{for } h \le r \le d \\ \frac{(d-r)}{4(d-c)} + \frac{(h-r)}{4(h-e)} & \text{for } g \le r \le h \\ \frac{1}{4}(\frac{(d-r)}{(d-c)} + \frac{(h-r)}{(h-e)} + \frac{(g-r)}{(g-f)}) & \text{for } b \le r \le g \\ \frac{1}{4}(\frac{(d-r)}{(d-c)} + \frac{(h-r)}{(h-e)} + \frac{(g-r)}{(g-e)} + \frac{(b-r)}{(b-e)}) & \text{for } a \le r \le b \\ \frac{1}{4}(\frac{(d-r)}{(d-c)} + \frac{(h-r)}{(h-e)} + \frac{(g-r)}{(g-e)} + 1) & \text{for } f \le r \le a \\ \frac{1}{4}(\frac{(d-r)}{(d-c)} + \frac{(h-r)}{(h-e)} + 2) & \text{for } e \le r \le f \\ \frac{1}{4}(\frac{(d-r)}{(d-c)} + 3) & \text{for } c \le r \le e \\ 1 & \text{for } r \le c. \end{cases} \quad (3.24)$$



$Fig.3.5.4$: $Tr\{\hat{\xi} \ge r\}$ function curve.

3.6 Bi-Fuzzy Set Theory

Generally speaking, a level-2 fuzzy set is a fuzzy set in which the elements are also fuzzy sets, and the bi-fuzzy variable is a fuzzy variable with fuzzy parameters. Level-2 fuzzy sets were originally presented by Zadeh [179]. Such sets are fuzzy sets whose elements themselves are ordinary fuzzy sets. They are very useful in circumstances where it is difficult to determine some elements for a fuzzy set.

Definition 3.10 In Mendel [115], a type-2 fuzzy set, denoted $\tilde{A}$, is characterized by a type-2 membership function $\mu_{\tilde{A}}(x, u)$, where $x \in X$ and $u \in J), x \subseteq [0, 1]$, i.e.,

$$\tilde{V} = \{\tilde{V}, \mu_{\tilde{V}}(\tilde{V})) \mid \forall x \in \tilde{\Gamma}(U) : \mu_{\tilde{V}} > 0\} \quad (3.25)$$

where each ordinary fuzzy set $\tilde{V}$ is defined by

$$\tilde{V} = \{(x, \mu_{\tilde{V}}(x)) \mid \forall x \in U : \mu_{\tilde{V}} > 0\} \tag{3.26}$$

For convenience, the membership grades $\mu_{\tilde{V}}(\tilde{V})$ of the fuzzy sets $.\tilde{V} \in \tilde{\Gamma}(U)$ are called 'outer-layer' membership grades, whereas the membership grades $\mu_{\tilde{V}}(\tilde{x})$ of the elements $x \in U$ are called inner-layermembership grades. Since level-2 fuzzy sets are still fuzzy sets, their mathematical behavior is defined by the fuzzy set operators. Normally speaking, a Fu-Fu variable ξ is a fuzzy variable under fuzzy environment.

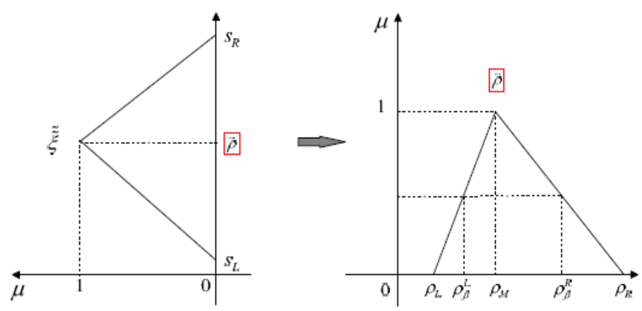


Figure 3.12: Triangular Bi-fuzzy variable

Example 3.1 $\tilde{\tilde{\xi}} = (s_L, \tilde{\xi}, s_R)$ with $\rho = (\rho_L, \rho_M, \rho_R)$ is called Fu-Fu variable,(cf. Fig. 3.12), if the outer-layer and inner-layer membership functions are as follows

$$\mu_{\tilde{\tilde{\xi}}}(x) = \begin{cases} \left(\dfrac{x - s_L}{\tilde{\rho} - s_L}\right) & \text{if } s_L \le x \le \tilde{\rho} \\ 0 & \text{otherwise} \\ \left(\dfrac{s_R - x}{s_R - \tilde{\rho}}\right) & \text{if } \tilde{\rho} \le x \le s_R \end{cases}$$

and $$\mu_{\tilde{\rho}}(x) = \begin{cases} \left(\dfrac{x' - \rho_L}{\rho_M - \rho_L}\right) & \text{if } \rho_L \le x' \le \rho_M \\ 0 & \text{otherwise} \\ \left(\dfrac{\rho_R - x'}{\rho_R - \rho_M}\right) & \text{if } \rho_M \le x' \le \rho_R \end{cases}$$

where $\tilde{\rho}$ is the center of $\tilde{\xi}$, which is a triangular fuzzy variable, $s_L \in R$ and $s_R \in R$ are the smallest possible value and the largest possible value of $\tilde{\tilde{\xi}}$, $s_L \in R$,

$s_M \in R$ and $s_R \in R$ are the the smallest possible value, the most promising value and the largest possible value of $\tilde{\rho}$, respectively.

Lemma 3.2 The expected value for the bi-fuzzy variable $\tilde{\tilde{c}} = (\tilde{c} - l_1, \tilde{c}, \tilde{c} + r_1)$ with $\tilde{c} = (c - l_2, c, c + r_2)$ we obtain that

$$E[\tilde{\tilde{c}}] = c + \frac{(r_1 + r_2) - (l_1 + l_2)}{4} \quad (3.27)$$

Proof: Let $\tilde{\tilde{c}} = (\tilde{c} - l_1, \tilde{c}, \tilde{c} + r_1)$, where $\tilde{c} = (c - l_2, c, c + r_2)$. Therefore

$$\begin{aligned} E(\tilde{\tilde{c}}) &= \frac{E(\tilde{c} - l_1) + 2E(\tilde{c}) + E(\tilde{c} + r_1)}{4} \text{ (UsingLemma} - 3.2) \\ &= \frac{E(\tilde{c}) - l_1 + 2E(\tilde{c}) + E(\tilde{c}) + r_1}{4} \\ &= \frac{4E(\tilde{c}) - l_1 + r_1}{4} \\ &= E(\tilde{c}) + \frac{r_1 - l_1}{4} \\ &= c + \frac{r_2 - l_2}{4} + \frac{r_1 - l_1}{4} \\ &= c + \frac{(r_1 + r_2) - (l_1 + l_2)}{4} \end{aligned}$$

Particular case: When $l_2 = 0 = r_2 \Rightarrow \tilde{\tilde{c}} = \tilde{c} \Rightarrow E(\tilde{\tilde{c}}) = c + \frac{r_1 - l_1}{4}$

Theorem 3.1 (Zhou [171]) If $\alpha^e_{rj1}, \alpha^e_{rj2}$, $\beta^e_{rj1}, \beta^e_{rj2}$ are left and right spreads of $\tilde{\tilde{e}}_{rj}(\theta)$ and $\tilde{e}_{rj}(\theta)$, $\alpha^b_{r1}, \alpha^b_{r2}$, $\beta^b_{r1}, \beta^b_{r2}$ are left and right spreads of $\tilde{\tilde{b}}_r(\theta)$ and $\tilde{b}_r(\theta)$, $r = 1, 2, \cdots, p, j = 1, 2, \cdots, n$, the basis function $L, R : [0, 1] \to [0, 1]$ are monotone decreasing continuous function, and it satisfies $L(1) = R(1) = 0, L(0) = R(0) = 1$ and the LR fuzzy variable is specified as the triangular fuzzy variable and $R^{-1}(\theta_i) = 1 - \theta_i$, $R^{-1}(\eta_i) = 1 - \eta_i$. For any $j = 1, 2, \cdots, n$, and if $\tilde{\tilde{e}}_{rj}(\theta)$ and $\tilde{\tilde{b}}_r(\theta)$ are independent fuzzy variables. Then

$$Pos\{\theta | Pos\{\tilde{\tilde{e}}^T_{rj}(\theta)x \leq \tilde{\tilde{b}}_r(\theta)\} \geq \theta_r\} \geq \eta_r$$

is equivalent to

$$R^{-1}(\theta_r)\beta^b_{r1} + L^{-1}(\theta_r)\alpha^{eT}_{r1}x - e^T_r x + b_r + L^{-1}(\eta_r)(\alpha^{eT}_{r2}x + \beta^b_{r2}) \geq 0$$

.

Theorem 3.2 (Zhou [171]) Assume that the Fu-Fu variable $\tilde{e}_{ij}$ and $\tilde{b}_r$ is as same as the assumption in Theorem -3.1, $i = 1, 2, \cdots, m, j = 1, 2, \cdots, n$. For confidence level $\delta_i, \gamma_i \in [0, 1], i = 1, 2, \cdots, m$. Then

$$Nes\{\delta | Nes\{\tilde{\tilde{e}}_{rj}(\delta)^T x \leq \tilde{\tilde{b}}_r(\delta)\} \geq \delta_r \geq\}\gamma_r$$

is equivalent to

$$b_r - e_r^T x - L^{-1}(1-\gamma_r)(\alpha_{r2}^b + \beta_{r2}^{eT} x) - L^{-1}(1-\delta_r)\alpha_{r1}^b - R^{-1}(\delta_r)\beta_{r2}^{eT} x \geq 0$$

3.7 Bi-Random Variables

Bi-random variable, which is proposed by Peng and Liu [136], is a mathematical tool to describe two-fold random phenomena. An n-dimensional bi-random vector ξ is a map from the probability space (Ω, A, Pr) to a collection of n-dimensional random vectors such that Pr$\{\xi(\omega) \in B\}$ is a measurable function with respect to ω for any Borel set B of the real space R^n.

Definition 3.11 (Peng and Liu [136]): Let ξ = ($\xi_1, \xi_2, ...\xi_n$) be a bi-random vector defined on Ω and g:R^n $\rightarrow$ R is Borel measurable function. Then the primitive chance of a bi-random event characterized by g(ξ) $\leq$ 0 is a function from [0,1] to [0,1], defined as
$Ch\{g(\xi) \leq 0\}(\alpha) = sup_{\beta\in[0,1]}\{\beta | Pr\{\omega \in \Omega | Pr\{g(\xi) \leq 0\} \geq \beta\} \geq \alpha\}$,
where α is a prescribed probability level. The value of primitive chance at α is called α-chance.

Definition 3.12 Let ξ = ($\xi_1, \xi_2, ...\xi_n$) be a bi-random vector, and g:R^n $\rightarrow$ R be Borel measurable function. Then the equilibrium chance of bi-random event g(ξ) $\leq$ 0 is defined as
Ch$^e\{g(\xi) \leq 0\} = sup_{\alpha\in[0,1]}\{\alpha \wedge Pr\{\omega \in \Omega | Pr\{g(\omega)) \leq 0\} \geq \alpha\}\}$.

Definition 3.13 (Bi-random efficient solution at α_i levels): Suppose a feasible solution x* of a LPP satisfies
Ch$^e\{f_i(x^*, \xi) \leq \bar{f}_i(x^*)\} \geq \alpha_i$, i=1, 2...., m
where confidence levels $\alpha_i \in [0, 1]$. Also x* is said to be a bi-random efficient solution at α_i -levels to the problem iff there exists no other feasible solution x such that

$\text{Ch}^e\{f_i(x,\xi) \leq \bar{f}_i(x)\} \geq \alpha_i$, i=1, 2...., m
and $\bar{f}_i(x) \leq \bar{f}_i(x^*)$ for all i and $\bar{f}_j(x) \leq \bar{f}_j(x^*)$ for at least one j$\in \{1, 2, ..., m\}$.

Theorem 3.3 Let ξ = ($\xi_1, \xi_2, ...\xi_n$) be a bi-random vector, and g:R^n $\rightarrow$ R be Borel measurable function, and $\alpha \in$ [0, 1] such as given below
$Ch^e\{g(\xi) \leq 0\} \geq \alpha \Leftrightarrow Pr\{\omega \in \Omega | Pr\{g(\xi(\omega)) \leq 0\} \geq \alpha\} \geq \alpha\}$.

Remark 3.3.1: If ξ degenerates to a random vector,
Pr$\{\omega \in \Omega | Pr\{g(\xi(\omega)) \leq 0\} \geq \alpha\}$ implies Pr$\{g(\xi(\omega)) \leq 0\} \geq \alpha$,
then above equation is equivalent to Pr$\{g(\xi(\omega)) \leq 0\}$, which is a probability measure.

Remark 3.3.2: Bi-random variables $\xi_1, \xi_2, ..., \xi_n$, which are defined on the probability space ($\Omega, \mathcal{A}$, Pr) are said to be independent if $\xi_1(\omega), \xi_2(\omega), ..., \xi_n(\omega)$ are independent random variables for all $\omega \in \Omega$.

Lemma 3.3 Assume that bi-random vector $\tilde{\tilde{c}}_i(\omega)$=($\tilde{\tilde{c}}_{i1}(\omega)$, $\tilde{\tilde{c}}_{i2}(\omega)$,..., $\tilde{\tilde{c}}_{in}(\omega))^T$ follows normal distribution with mean vector $\tilde{c}_i(\omega)$ and positive definite covariance matrix V^c_i, denoted by $\tilde{\tilde{c}}_i(\omega) \sim N(\tilde{c}_i(\omega), {V^c}_i)$, $\tilde{c}_i(\omega)$ is a normal random vector with mean vector μ^c_i and positive definite covariance matrix V^{nc}_i, written as $\tilde{c}_i(\omega) \sim N(\mu^c_i, V^{nc}_i)$. If $\tilde{\tilde{c}}_{i1}(\omega)$, $\tilde{\tilde{c}}_{i2}(\omega)$,..., $\tilde{\tilde{c}}_{in}(\omega)$ are independent bi-random variables, then Pr$\{\omega | Pr(\tilde{\tilde{c}}_i(\omega)^T x \leq \bar{f}_i) \geq \alpha_i\} \geq \alpha_i$ holds iff

$$\mu^{cT}_i x + \Phi^{-1}(\alpha_i)\sqrt{(x^T V^c_i x)} + \Phi^{-1}(\alpha_i)\sqrt{(x^T V^{nc}_i x)} \leq \bar{f}_i,$$

where Φ is the standardized normal distribution.

Lemma 3.4 (Xu and Tao[175]): Assume that bi-random vector $\tilde{\tilde{a}}_r(\omega)$ = ($\tilde{\tilde{a}}_{r1}(\omega)$, $\tilde{\tilde{a}}_{r2}(\omega)$,..., $\tilde{\tilde{a}}_{rn}(\omega))^T$ follows normal distribution with mean vector $\tilde{a}_r(\omega)$ and positive definite covariance matrix V^a_r, denoted by $\tilde{\tilde{a}}_r(\omega) \sim N(\tilde{a}_r(\omega), {V^a}_r)$, $\tilde{a}_r(\omega)$ is a normal random variable, written as $\tilde{a}_r(\omega) \sim N(\mu^a_r, V^{na}_r)$. Bi-random variable $\tilde{\tilde{b}}_r(\omega)$ follows normal distribution with mean value $\tilde{b}_r(\omega)$ and variance $(\sigma^b_r)^2$, denoted by $\tilde{\tilde{b}}_r(\omega) \sim N(\tilde{b}_r(\omega), (\sigma^b_r)^2)$, where $\tilde{b}_r(\omega)$ is normally distributed random variable, written as $\tilde{b}_r(\omega) \sim N(\mu^b_r, (\sigma^{nb}_r)^2)$. If $\tilde{\tilde{a}}_{r1}(\omega)$, $\tilde{\tilde{a}}_{r2}(\omega)$,..., $\tilde{\tilde{a}}_{rn}(\omega), \tilde{\tilde{b}}_r(\omega)$ are independent bi-random variables, then

$$\text{Ch}^e\{\tilde{\tilde{a}}_r^T x \leq \tilde{\tilde{b}}_r\} \geq \beta_r \text{ holds iff,}$$

$$\mu^{aT}_r x + \Phi^{-1}(\beta_r)\sqrt{(x^T V^a_i x + (\sigma^b_r)^2))} + \Phi^{-1}(\beta_r)\sqrt{(x^T V^{na}_r x + (\sigma^{nb}_r)^2))} \leq \mu^b_r$$

where Φ is the standardized normal distribution and β_r are predetermined confidence levels.

3.8 Bi-Rough Set Theory

A bi-rough variable is a function ξ from a rough space $(\Lambda, \Delta, \kappa, \Pi)$ to the set of rough variables such that Tr$\{\xi(\lambda) \in B\}$ is a measurable function of λ for any Borel set B of $\Re$.

Theorem 3.4 Liu [103] Assume that ξ is a bi-rough variable, and B is a Borel set of $\Re$. Then the trust Tr$\{\xi(\lambda) \in B\}$ is a rough variable.

Theorem 3.5 Liu [103] Assume that ξ is a bi-rough variable, and B is a Borel set of $\Re$.if the expected value E$[\xi(\lambda)]$ is a finite for each λ, then E$[\xi(\lambda)]$ is a rough variable.

Theorem 3.6 Liu [103] Let ξ be a bi-rough variable. Then its expected value is defined by

E$[\xi]$=$\int_0^{+\alpha} Tr\{\lambda \in \Lambda | E[\xi(\lambda)] \geq r\}dr$ - $\int_{-\alpha}^{0} Tr\{\lambda \in \Lambda | E[\xi(\lambda)] \leq r\}dr$

provided that at least one of the two integrals is finite.

3.9 Fuzzy-Random (Fu-Ra) Set Theory

In this section, we give basic concepts of fuzzy random theory. According to Liu[103]. Let ξ be a fuzzy random variable with membership function μ. then the possibility, necessity, and credibility of a fuzzy event $\{\xi \geq r\}$ can be defined by

Pos$\{\xi \geq r\} = sup\mu_{\mu \geq r}(\mu)$,

Nes$\{\xi \geq r\} = 1 - sup\mu_{\mu < r}(\mu)$,

Cr$\{\xi \geq r\} = \frac{1}{2}(Pos\{\xi \geq r\} + Nes\{\xi \geq r\})$. A fuzzy random variable ξ is functions from the probability space (Ω, A, Pr) to the set of fuzzy variable such that Pos$\{\xi(\omega) \in B\}$ is measurable function of ω for any Borel set B of R.

Theorem 3.7 Peng and Liu [135] Assume that $\tilde{\hat{c}}_{ij}$ is LR fuzzy random variable, for any $\omega \in \Omega$. Then the membership function of $\tilde{\hat{c}}_{ij}(\omega)$ is

$$\mu_{\tilde{\hat{c}}_{ij}(\omega)}(t) = \begin{cases} L(\frac{c_{ij}(\omega)-t}{\alpha^c_{ij}}) & \text{if } c_{ij}(\omega) \geq t, \alpha^c_{ij} \geq 0, \\ R(\frac{t-c_{ij}(\omega)}{\beta^c_{ij}}) & \text{if } c_{ij}(\omega) \leq t, \beta^c_{lj} \geq 0 \end{cases} \tag{3.28}$$

where the random vector $(c_{ij}(\omega))_{n\times 1}=(c_{i1}(\omega), c_{i2}(\omega), c_{i3}(\omega), ...c_{in}(\omega))^T$ is normally distributed, the mean vector is $d_i{}^c$, the covariance matrix is $V_i{}^c$, denoted by $(c_{ij}(\omega))_{n*1} \sim N(d_i{}^c, V_i{}^c)$, α_{ij}^c, and α_{ij}^c are the left and right spread of $\tilde{\hat{c}}_{ij}(\omega)$, i=1,2...,m, j=1,2..,n, the reference function L,R:[0,1]→ [0,1] satisfies that L(1)= R(1)=0, L(0)=R(0)=1, and it is monotone function. Also for two LR-type fuzzy numbers $\tilde{M}$, $\tilde{N}$ such as $\tilde{M}$=(m, $\alpha, \beta)_{LR}$, $\tilde{N}$=(n, $\gamma, \delta)_{LR}$ then

(m, $\alpha, \beta)_{LR}$+ (n,$\gamma, \delta)_{LR}$ =(m+n, $\alpha + \gamma, \beta + \delta)_{LR}$.

Thus $Pr\{\omega | Pos\{\{\tilde{\hat{c}}_{ij}(\omega)^T \leq f_i\} \geq \delta_i\} \geq \gamma_i$ is equivalent to

$$R^{-1}(\delta_i)\beta_i^{cT}x + d_i^{cT}x + \phi^{-1}(1-\gamma_i)\sqrt{(x^T V_i^c x)} \leq f_i, i = 1, 2, ..., m$$

where ϕ are standard normally distributed, $\delta_i, \gamma_i \in [0, 1]$ are predetermined confidence levels.

Theorem 3.8 Assume that $\tilde{\hat{e}}_{rj}$ and $\tilde{\hat{b}}_r$ are LR fuzzy random variable, for any $\omega \in \Omega$. Then the membership function of $\tilde{\hat{e}}_{rj}(\omega)$ and $\tilde{\hat{b}}_r(\omega)$ are

$$\mu_{\tilde{\hat{e}}_{rj}(\omega)}(t) = \begin{cases} L(\frac{e_{rj}(\omega)-t}{\alpha_{rj}^e}) & \text{if } e_{rj}(\omega) \geq t, \alpha_{rj}^e \geq 0, \\ R(\frac{t-e_{rj}(\omega)}{\beta_{rj}^e}) & \text{if } e_{rj}(\omega) \leq t, \beta_{rj}^e \geq 0 \end{cases} \tag{3.29}$$

$$\mu_{\tilde{\hat{e}}_{rj}(\omega)}(t) = \begin{cases} L(\frac{b_r(\omega)-t}{\alpha_r^b}) & \text{if } b_r(\omega) \geq t, \alpha_r^b \geq 0, \\ R(\frac{t-b_r(\omega)}{\beta_r^b}) & \text{if } b_r(\omega) \leq t, \beta_r^b \geq 0 \end{cases} \tag{3.30}$$

where the random vector $(e_{rj}(\omega))_{n\times 1}=(e_{r1}(\omega), e_{r2}(\omega), e_{r3}(\omega), ...e_{rn}(\omega))^T \sim N(d_r^e, V_r^e)$, $b_r(\omega) \sim N(d_r^b, (\sigma_r^b)^2)$, α_{rj}^e, β_{rj}^e are left and right spread of $\tilde{\hat{e}}_{rj}(\omega)$, α_r^b, β_r^b are left and right spread of $\tilde{\hat{b}}_r(\omega)$, r=1,2,..p, j=1,2,...,n, the reference function L,R:[0,1]→ [0,1] satisfies that L(1)=R(1)=0, L(0)=R(0)=1, and it is monotone function.

Then $Pr\{\omega | Pos\{\{\tilde{\hat{e}}_r(\omega)^T \leq \tilde{\hat{b}}_r(\omega)^T\} \geq \theta_r\} \geq \eta_r$ is equivalent to

$$R^{-1}(\theta_r)\beta_r^b + L^{-1}(\theta_r)\alpha_r^{cT}x - (d_r^{eT}x - d_r^b) - \phi^{-1}(\eta_r)\sqrt{(x^T V_r^e x + (\sigma_r^b)^2)} \geq 0.$$

Theorem 3.9 Assume that the fuzzy random variable $\tilde{\hat{c}}_{ij}$ is as same as the assumption in Theorem 3.7, i=1,2..m, j=1,2..,n. For the confidence level $\delta_i, \gamma_i \in$ [0,1], i=1,2,...,m, we have

$Pr\{\omega | Nes\{\tilde{\hat{c}}_i(\omega)^T x \leq f_i\} \geq \delta_i\} \geq \gamma_i \Leftrightarrow d_i^{cT}x - L^{-1}(1-\delta_i)\alpha_i^{cT}x + \phi^{-1}(1-\gamma_i)\sqrt{(x^T V_i^c x)} \leq f_i$

Theorem 3.10 Assume that the fuzzy random variable $\tilde{\hat{e}}_{rj}$ and $\tilde{\hat{b}}_r$ are as same as the assumption in Theorem 3.8, j=1,2..n, r=1,2..,p. Then for the certain confidence level $\theta_r, \eta_r \in$ [0,1], r=1,2,...,p, we have
$Pr\{\omega | Nes\{\tilde{\hat{e}}_r(\omega)^T x \leq \tilde{\hat{b}}_r(\omega)\} \geq \theta_r\} \geq \eta_r$
$\Leftrightarrow \phi^{-1}(1-\eta_r)\sqrt{(x^T V_r^e x + (\sigma_r^b)^2)} - L^{-1}(1-\theta_r)\alpha_r^b x - R^{-1}(\theta_r)\beta_r^{eT} x + (d_r^b - d_r^{eT} x) + \geq 0.$

3.10 Fuzzy-Rough (Fu-Ro) Set Theory

In this section, we will state some basic concepts, theorems and lemmas on fuzzy rough theory by Xu and Zhou [173]. These results are crucial for the remainder of this investigation.

Definition 3.14 Xu and Zhou [173] proposed some definitions and discussed some important properties of fuzzy rough variables. Let U be a universe, and X a set representing a concept. Then its lower approximation is defined by

$$\underline{X} = \{x \in U \mid R^{-1}(x) \subset X\} \tag{3.31}$$

and the upper approximation is defined by

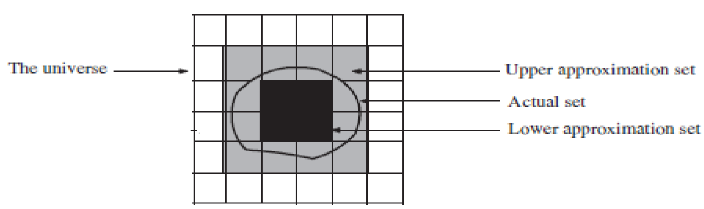


Figure 3.13: A Rough Set

$$\overline{X} = \bigcup_{x \in X} R(x) \tag{3.32}$$

where R is the similarity relationship on U.

Definition 3.15 The collection of all sets having the same lower and upper approximations is called a rough set, denoted by $(\underline{X}, \overline{X})$. The figure of a rough set is depicted in Figure 3.13.

Example 3.2 Let us consider on the continuous set in the one dimension real space R. There are still some vague sets which cannot be directly fixed and need to be described by the rough approximation. For example, set R be the universe, a similarity relation is defined as $a \simeq b$ if and only if $|a - b| \leq 10$. We have that for the set $[20, 50]$, its lower approximation $\underline{[20, 50]}$ = [30, 40] and its upper approximation $\overline{[20, 50]} = [10, 60]$. Then the upper and lower approximation of the set $[20, 50]$ make up a rough set $([30, 40], [10, 60])$ which is the collection of all sets having the same lower approximation $[30, 40]$ and upper approximation [10,60].

Definition 3.16 A fuzzy rough variable ξ is a fuzzy variable with uncertain parameter $\rho \in X$, where X is approximated by $(\underline{X}, \overline{X})$ according to the similarity relation R, namely, $\underline{X} \subseteq X \subseteq \overline{X}$.
For convenience, we usually denote $\rho \vdash (\underline{X}, \overline{X})_R$ expressing that ρ is in some set A which is approximated by $(\underline{X}, \overline{X})$ according to the similarity relation R, namely, $\underline{X} \subseteq A \subseteq \overline{X}$.

Example 3.3 Let's consider the LR fuzzy variable ξ with the following membership function, $\mu_\xi(x) = \begin{cases} L\left(\dfrac{\rho - x}{\alpha}\right) & \text{if } \rho - \alpha < x < \rho \\ 1 & \text{if } x = \rho \\ L\left(\dfrac{x - \rho}{\beta}\right) & \text{if } \rho < x < \rho + \beta \end{cases}$
If $\rho \vdash ([1, 2], [0, 3])_R$, then ξ is called a fuzzy rough variable.

Lemma 3.5 (Xu and Zhou [173]) Assume that ξ and η are the introduction of variables with finite expected values. Then for any real numbers a and b, we have

$$E[a\xi + b\eta] \ = \ aE[\xi] + bE[\eta] \tag{3.33}$$

Theorem 3.11 (Xu and Zhou [174]) Let ξ be a LR Fu-Ro variable with the membership function of fuzzy variable ξ has the following form

$$\mu_\xi(x) = \begin{cases} L(\frac{x - \bar{z})}{\alpha}), & \bar{z} - \alpha < x \leq \bar{z} \\ 1, & x = \bar{z} \\ R(\frac{x - z}{\beta}), & \bar{z} < x \leq \bar{z} + \beta \end{cases} \tag{3.34}$$

where $\bar{z}$ is a rough variable and $\bar{z} = ([z_2, z_3], [z_1, z_4]), a < z_1 < z_2 < z_3 < z_4$. And here we just consider the situation when the reference function $L(x) = R(x) = 1 - x$, then this LR fuzzy rough variable is triangular type, and the left and right spread $\alpha, \beta > 0$. Then the expected value of ξ is

$$E[\xi] = \frac{1}{4}(z_1 + z_2 + z_3 + z_4 + \alpha + \beta) \tag{3.35}$$

Lemma 3.6 Let $\bar{\tilde{\xi}}$ be a LR fuzzy rough variable with the membership function of fuzzy variable, $\bar{\tilde{\xi}}$ has the following form

$$\mu_{\bar{\tilde{\xi}}}(x) = \begin{cases} L\left(\frac{\tilde{m}_1 - x}{\alpha}\right), & \text{if } \tilde{m}_1 - \alpha \le x \le \tilde{m}_1 \\ 1, & \text{if } \tilde{m}_1 \le x \le \tilde{m}_2 \\ R\left(\frac{x - \tilde{m}_2}{\beta}\right), & \text{if } \tilde{m}_2 \le x \le \tilde{m}_2 + \beta \end{cases} \tag{3.36}$$

where $\tilde{m}_1$ and $\tilde{m}_2$ are rough variables, as follows: $\tilde{m}_1 = ([q_2, q_3], [q_1, q_4])$, $0 < q_1 \le q_2 < q_3 \le q_4$ and $\tilde{m}_2 = ([p_2, p_3], [p_1, p_4])$, $0 < p_1 \le p_2 < p_3 \le p_4$. Then, the expected value of $\bar{\tilde{\xi}}$ is:

$$E[\bar{\tilde{\xi}}] = \frac{q_1 + q_2 + q_3 + q_4 + p_1 + p_2 + p_3 + p_4}{8} + \frac{\beta}{2}\int_0^1 R(t)dt - \frac{\alpha}{2}\int_0^1 L(t)dt$$

Lemma 3.7 Let $\tilde{a} = (\alpha_1, m_1, \bar{m}_1, \beta_1)_{LR}$ and $\tilde{b} = (\alpha_2, m_2', \bar{m}_2', \beta_2)_{LR}$ be two L-R type fuzzy numbers with continuous membership function. For a given confidence level $\alpha \in [0, 1]$, if

$$Pos\left\{\tilde{a}_i \ge \tilde{b}\right\} \ge \eta$$

then we have:

$$m_1 + \beta R^{-1}\left(\eta\right) \ge m_2 - \alpha_2 R^{-1}\left(\eta\right) \tag{3.37}$$

Proof: Let $\tilde{a} = (\alpha_1, m_1, \bar{m}_1, \beta_1)$ and $\tilde{b} = (\alpha_2, m_2, \bar{m}_2, \beta_2)$ be two LR-type fuzzy numbers. Then $\tilde{\lambda} = \tilde{a} - \tilde{b} = (\alpha_1 + b_1, m_1 - \bar{m}_2, \bar{m}_1 - m_2, \alpha_2 + \beta_1)_{LR}$ is a LR type fuzzy number, the possibility of the fuzzy event, $Pos(\tilde{\lambda} \ge 0)$ can be expressed as

$$Pos(\tilde{\lambda} \geq 0) = \begin{cases} 1, & r < n_1 < n_1 - \alpha \\ R(\frac{r-n_2}{\beta}), & n_2 < r < n_2 + \beta \\ 1, & r > n_2 + \beta \end{cases} \tag{3.38}$$

where $\alpha = \alpha_1 + \beta_1, n_1 = m_1 - \bar{m}_2, n_2 = \bar{m}_1 - m_2, \beta = \alpha_2 + \beta_1$. Let us consider $Pos(\tilde{\lambda} \geq 0) \geq \eta$. Then

$$\begin{aligned} R(\frac{-n_2}{\beta}) \geq \eta \;\; &\Rightarrow \;\; \frac{-n_2}{\beta} \geq R^{-1}(\eta) \\ &\Rightarrow \;\; (\alpha_2 + \beta_1)R^{-1}\bigg(\eta\bigg) \geq -(\bar{m}_1 - m_2) \\ &\Rightarrow \;\; m_1 + \beta_1 R^{-1}\bigg(\eta\bigg) \geq m_2 - \alpha_2 R^{-1}\bigg(\eta\bigg) \end{aligned}$$

Thus, with the above Lemma-3.7, a fuzzy linear constraint can be written in its deterministic form.

Theorem 3.12 (Xu et al. [173]) Assume that $\hat{\tilde{c}}_{ij}$ is a fuzzy rough variable, for any $\lambda \in \Lambda$, the fuzzy variable $\tilde{c}_{ij}(\lambda)$ is characterized by the following membership function

$$\mu_{\hat{c}_{ij}(\lambda)}(t) = \begin{cases} L(\frac{c_{ij}(\lambda)x-t}{\gamma^c_{ij}}) & \text{if } c_{ij}(\lambda)x \geq t, \gamma^c_{ij} \geq 0, \\ R(\frac{t-c_{ij}(\lambda)x}{\delta^c_{ij}}) & \text{if } c_{ij}(\lambda)x \leq t, \delta^c_{lj} \geq 0 \end{cases} \tag{3.39}$$

where $\gamma^c_{ij}, \delta^c_{ij}$ are positive numbers expressing the left and right spread of $\tilde{c}_{ij}(\lambda)$, reference function L, R :[0, 1] $\rightarrow$ [0, 1] with L(1)=R(1)=0, and L(0)=R(0)=1 are non-increasing, continuous functions.
$(c_{ij}(\lambda))_{n\times 1}$=$(c_{i1}(\lambda), c_{i2}(\lambda), ..., c_{in}(\lambda))^T$ is a rough vector.

It follows that $c_i(\lambda)^T x$=([a,b] [c,d]) where c$\leq$ a,b$\leq$ b is a rough variable and characterized by the trust measure in Equ. 3.24. Then we have

Tr$\{\lambda | Pos\{\hat{\tilde{c}}_{ij}(\lambda)^T \geq f_i\} \geq \beta_i\} \geq \alpha_i$ if and only if

$$\Leftrightarrow \begin{cases} f_i \leq d - 2\alpha_i(d-c) + R^{-1}(\beta_i)\gamma^{cT}_i x, & \text{if } b \leq w \leq d \\ f_i \leq \frac{d(b-a)+b(d-c)-2\alpha_i(d-c)(b-c)}{d-c+b-c} + R^{-1}(\beta_i)\gamma^{cT}_i x & \text{if } a \leq< b \\ f_i \leq d - (d-c)(2\alpha_i - 1) + R^{-1}(\beta_i)\gamma^{cT}_i x & \text{if } c \leq w \leq a \\ f_i \leq c + R^{-1}(\beta_i)\gamma^{cT}_i x & \text{if } w \leq c \end{cases} \tag{3.40}$$

where $\beta_i, \alpha_i, w = f_i - R^{-1}(\beta_i)\delta_i^{cT} x$ are predetermined confidence levels.

Lemma 3.8 (Xu et al. [173]) Let $\tilde{m}$ and $\tilde{n}$ between independent fuzzy numbers with continuous membership functions. For a given confidence level $\alpha \in [0,1]$ then Pos $\{\tilde{m} \geq \tilde{n}\} \geq \alpha \Longleftrightarrow$ $\mathrm{m}_\alpha^R \geq n_\alpha^L$, where m_α^L, m_α^R and n_α^L, n_α^R are left and right side extreme points of the $\alpha-$level sets [m_α^L, m_α^R] and [n_α^L, n_α^R] of $\tilde{m}$ and $\tilde{n}$, respectively, and Pos$\{\tilde{m} \geq \tilde{n}\}$ means the degree of possibility that $\tilde{m}$ is greater than or equal to $\tilde{n}$.

3.11 Random-Fuzzy (Ra-Fu) Variables

Definition 3.17 (Possibility space (Liu, [103]) Let Θ be a nonempty set, and $P(\Theta)$ be the power set of Θ. For each $A \in P(\Theta)$, there is a nonnegative number $Pos\{A\}$, called its possibility, such that
1. $Pos\{\phi\} = 0, Pos\{\Theta\} = 1$; and
2. $Pos\{\bigcup_k A_k\} = sup_k Pos\{A_k\}$ for any arbitrary collection A_k in $P(\Theta)$.
The triplet $(\Theta, P(\Theta), Pos)$ is called a possibility space, and the function Pos is referred to as a possibility measure. Then, a random fuzzy variable is firstly defined by Liu [103] as a function from a possibility space to a collection of random variables.

Definition 3.18 (Random fuzzy variable (Liu [103]) A random fuzzy variable is defined as a function from the possibility space $(\Theta, P(\Theta), Pos)$ to the set of random variables. An example of random fuzzy variables are given by Liu [103] as follows:

Definition 3.19 (Membership function of a random fuzzy variable (Liu, [103]) Let $\bar{\tilde{\xi}}$ be a random fuzzy variable on the possibility space $(\Theta, P(\Theta), Pos)$. Then its membership function is derived from the possibility measure Pos by

$$\mu(\bar{\eta}) = Pos\{\theta \in \Theta | \bar{\tilde{\xi}}(\theta) = \bar{\eta}\}, \bar{\eta} \in \Gamma \tag{3.41}$$

Definition 3.20 (Random fuzzy variable (Katagiri et al., [80]) Let Γ be a collection of random variables. Then, a random fuzzy variable $\bar{\tilde{C}}$ is defined by its membership function

$$\mu_{\bar{\tilde{C}}} : \Gamma \to [0,1] \tag{3.42}$$

Example 3.4 Assume that $\bar{\eta}_1, \bar{\eta}_2, \cdots, \bar{\eta}_m$ are random variables and $u_1, u_2, \cdots, u_m$ are real numbers in $[0, 1]$ such that $max\{u_1, u_2, \cdots, u_m\} = 1$. Then $\tilde{\bar{\eta}}$ is a random fuzzy variable and its membership function is expressed as

$$\tilde{\bar{\xi}}(\bar{\gamma}) = \begin{cases} u_1, & \text{if} \bar{\gamma} = \bar{\eta}_1, \\ u_2, & \text{if} \bar{\gamma} = \bar{\eta}_2 \\ \cdots \\ u_m, & \text{if} \bar{\gamma} = \bar{\eta}_m \end{cases} \tag{3.43}$$

Theorem 3.13 Katagiri et al.[80]

$$Pos\{Prob\{\tilde{\bar{C}}_l x \leq f_l\} \geq \hat{\theta}_l^{obj}\} \geq \hat{h}_l^{obj}, l = 1, 2, \cdots, k \text{ and}$$
$$Nec\{Prob\{\tilde{\bar{C}}_l x \leq f_l\} \geq \hat{\theta}_i^{obj}\} \geq \hat{h}_i^{obj}, l = 1, 2, \cdots, k$$

is equivalently transformed into the condition

$$\sum_{j=1}^{n}\{m_{lj}^c - L^*(\hat{h}_l^{obj})\alpha_{lj}^c\}x_j + \Phi^{-1}(\hat{\theta}_l^{obj})\sqrt{x^t V_l^c x} \leq f_l$$
$$\sum_{j=1}^{n}\{m_{lj}^c + L^*(1 - \hat{h}_l^{obj})\beta_{lj}^c\}x_j + \Phi^{-1}(\hat{\theta}_l^{obj})\sqrt{x^t V_l^c x} \leq f_l$$

where α, β's are spreads, θ and h's are desired amount of satisfaction of probability constraints and possibility/ necessity constraints chosen by DM.

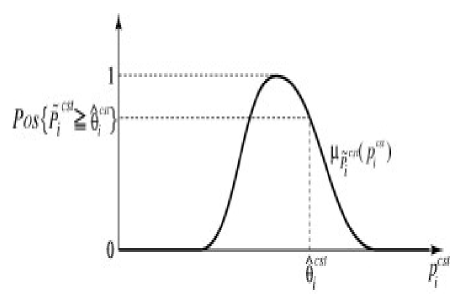


Figure 3.14: Degree of possibility $Pos\{Prob\{\tilde{\bar{C}}_l x \leq f_l\} \geq \hat{\theta}_i^{obj}\} \geq \hat{h}_i^{obj}$

Theorem 3.14 Katagiri et al.[80]

$$Pos\{Prob\{\tilde{\bar{A}}x \leq \tilde{\bar{B}}\} \geq \hat{\theta}_i^{cst}\} \geq \hat{h}_i^{cst}, i = 1, 2, \cdots, r \quad \text{and}$$
$$Nec\{Prob\{\tilde{\bar{A}}x \leq \tilde{\bar{B}}\} \geq \hat{\theta}_i^{cst}\} \geq \hat{h}_i^{cst}, i = 1, 2, \cdots, r$$

are equivalently transformed into the conditions

$$\sum_{j=1}^{n}\{m_{ij}^a - L^*(\hat{h}_i^{cst})\alpha_{ij}^a\}x_j + \Phi^{-1}(\hat{\theta}_i^{cst})\sqrt{x^t V_i^a x + (\sigma_i^b)^2}$$
$$\leq m_i^b + L^*(\hat{h}_i^{cst})\beta_{ij}^b \text{ and}$$
$$\sum_{j=1}^{n}\{m_{ij}^a + L^*(1-\hat{h}_i^{cst})\beta_{ij}^b\}x_j + \Phi^{-1}(\hat{\theta}_i^{cst})\sqrt{x^t V_i^a x + (\sigma_i^b)^2}$$
$$\leq m_i^b - L^*(1-\hat{h}^{cst})\alpha_{ij}^b$$

Theorem 3.15 (Xu, Zhou [174]) If $\tilde{a}_r$, $\tilde{b}_r$ is triangular LR fuzzy variables, then the following expression are equivalent

$$Pos\left\{\sum_{j=1}^{n}\tilde{a}_r^T x \leq \tilde{b}_r\right\} \geq \theta_r$$
$$\Leftrightarrow \quad b_r - \theta_r\alpha_r^b \geq a_r^T x + (1-\theta_r)\beta_r^{aT}x,\ r = 1, 2, \cdots p.$$

3.12 Random-Rough (Ra-Ro) Variables

A random rough variable was initialized by Liu [100] as a rough variable defined on the universal set of random variables, or a rough variable taking random variable values.

Definition 3.21 (Liu [100]) A random rough variable is a function ξ from a rough space $(\Lambda, \Delta, \kappa, \Pi)$ to the set of random variables such that Pr$\{\xi(\lambda) \in B\}$ is a measurable function of λ for any Borel set B of $\Re$.

Theorem 3.16 (Liu [100]) Assume that ξ is a random rough variable, and B is a Borel set of $\Re$. Then the probability Pr$\{\xi(\lambda) \in B\}$ is a rough variable.

Theorem 3.17 Let ξ be a random rough variable. If the expected value E[$\xi(\lambda)$] is finite for each λ, then E[$\xi(\lambda)$] is a rough variable.

Definition 3.22 (Liu[100]) An n-dimensional random rough vector is a function ξ from a rough space $(\Lambda, \Delta, \kappa, \Pi)$ to the set of n-dimensional random vectors such that Pr$\{\xi(\lambda) \in B\}$ is a measurable function of λ for any Borel set B of $\Re^n$.

Theorem 3.18 (Liu[100]) If $(xi_1, xi_2, , xi_n)$ is a random rough vector, then $xi_1, xi_2,$ $, xi_n$ are random rough variables. Conversely, if $xi_1, xi_2, , xi_n)$ are random rough variables, and for each $\lambda \in \Lambda$, the random variables $\xi_1(\lambda), \xi_2(\lambda), ...\xi_n(\lambda)$ are independent, then $xi_1, xi_2, , xi_n)$ is a random rough vector.

Theorem 3.19 Let ξ be an n-dimensional random rough vector, and f : $\Re \rightarrow \Re$a measurable function. Then f(ξ) is a random rough variable.

Definition 3.23 (Liu[100], Random Rough Arithmetic on Single Space) Let f : $\Re \rightarrow \Re$ be a measurable function, and $xi_1, xi_2, , xi_n)$ random rough variables defined on the rough space $(\Lambda, \Delta, \kappa, \Pi)$. Then xi = f($xi_1, xi_2, , xi_n$) is a random rough variable defined by $\xi(\lambda)$ = f$\xi_1(\lambda), \xi_2(\lambda), ...\xi_n(\lambda), \forall\lambda \in \lambda$.

Definition 3.24 (Liu[100], Random Rough Arithmetic on Different Spaces) Let f : $\Re \rightarrow \Re$ be a measurable function, and ξ_i random rough variables defined on $(\Lambda_i, \Delta_i, \kappa_i, \Pi_i)$, i = 1, 2, , n, respectively. Then ξ = f($xi_1, xi_2, , xi_n$) is a random rough variable defined on the product rough space $(\Lambda, \Delta, \kappa, \Pi)$ as ($\lambda_1, \lambda_2, ..\lambda_n$) = f($\xi_1(\lambda_1), \xi_2(\lambda_2), , \xi_n(\lambda_n)$) for all ($\lambda_1, \lambda_2, ..., \lambda_n$) $\in \Lambda$.

Theorem 3.20 (Liu[100]) Let ξ be a random rough variable, and B a Borel set of $\Re$. For any given $\alpha^* \in (0, 1]$, write $\beta^* = Ch\{\xi \in B\}(\alpha^*)$. Then we have Tr$\{\lambda \in \Lambda | Pr\{\xi(\lambda) \in B\} \geq \beta^* \geq \alpha^*$

Theorem 3.21 (Liu[100]) Assume that ξ and η are random rough variables with finite expected values. Then for any real numbers a and b, we have E[a$\xi + b\eta] = aE[\xi] + bE[\eta]$.

Theorem 3.22 (Liu[100]) If ξ is a random rough variable with finite expected value, a and b are real numbers, then V$[a\xi + b] = a^2V[\xi]$.

3.13 Optimization Models in Different Environments

3.13.1 Single Objective Random Model

A stochastic linear programming can be stated as follows:

$$\left.\begin{array}{r}\text{Minimize} \quad f(X)=\sum\limits_{j=1}^{n}\hat{c}_j x_j \\ \text{subject to} \quad \sum\limits_{j=1}^{n}\hat{a}_{ij}x_j \leq \hat{b}_i \ \text{ for } i=1,2,...,m \\ x_j \geq 0, \ \text{ for } j=1,2,...,n\end{array}\right\} \tag{3.44}$$

where $\hat{c}_j$, $\hat{a}_{ij}$, and $\hat{b}_i$ are random variables with known probability distribution.

3.13.2 Equivalent Crisp Model by Chance Constraint

The chance - constrained programming technique [25] can be used to solve the above problem. In this method, Equ. 3.44 is stated as follows:

$$\left.\begin{array}{r}\text{Minimize} \quad f(X)=\sum\limits_{j=1}^{n}\hat{c}_j x_j \\ \text{subject to} \quad P[\sum\limits_{j=1}^{n}\hat{a}_{ij}x_j \leq \hat{b}_i] \geq p_i \ \text{ for } i=1,2,...,m \\ x_j \geq 0, \ \text{ for } j=1,2,...,n\end{array}\right\} \tag{3.45}$$

where p_i's are specified probabilities. For simplicity, we assume that the design variables x_j are deterministic variables. We shall further assume that all the random variables are normally distributed with known mean and standard deviations. So the objective function f(X) will also be a normally distributed random variable. Then mean and variance of f(X) are given by

$$\left.\begin{array}{c}\overline{f}(X)=\sum\limits_{j=1}^{n}\overline{c}_j x_j, \quad \overline{c}_j=E(c_j) \\ var(f)=X^T V X\end{array}\right\} \tag{3.46}$$

where $E(c_j)$ is the mean value of c_j and the matrix V is the covariance matrix of c_j.

Thus the stochastic linear programming problem of Equ. 3.45 can be stated as an equivalent deterministic nonlinear programming problem as given below

$$\left.\begin{array}{rl} \text{Minimize} & F(X) = k_1 * \sum_{j=1}^{n} \bar{c}_j x_j + k_2 * \sqrt{X^T V X}, \quad k_1, k_2 \geq 0 \\ \text{subject to} & \bar{h}_i + s_i * \sqrt{(var(h_i))} \leq 0, \quad i = 1, 2, ..., m \\ & x_j \geq 0, \quad j = 1, 2, ..., n \\ & h_i = \sum_{j=1}^{n} a_{ij} x_j - b_i, \end{array}\right\} \tag{3.47}$$

where k_1 and k_2 are constants indicating the weights of mean and variance functions. Here h_i is a new random variable. The mean of h_i is given by $\bar{h}_i = \sum_{j=1}^{n} \bar{a}_{ij} x_j - \bar{b}_i$.

3.13.3 Single Objective Bi-fuzzy Model

Let us consider the following single-objective decision making model with Bi-fuzzy coefficients:

$$\begin{cases} Max & f(x, \xi) \\ s.t & \begin{cases} g_r(x, \xi) \leq 0, r = 1, 2, \cdots, p \\ x \in X \end{cases} \end{cases} \tag{3.48}$$

where x is a n-dimensional decision vector, $\xi = (\xi_1, \xi_2, \xi_3, \cdots, \xi_n)$ is a Bi-fuzzy vector, $f(x, \xi)$ are objective functions, $i = 1, 2, \cdots, m$. Because of the existence of Bi-fuzzy vector ξ, problem (3.48) is not well-defined. That is, the meaning of maximizing $f(x, \xi)$ is not clear and constraints $g_r(x, \xi) \leq 0, r = 1, 2, \cdots, p$ do not define a deterministic feasible set.

3.13.4 Equivalent Crisp Model of Bi-fuzzy Model

For the single-objective model (3.48) with Bi-fuzzy parameters, we cannot deal with it directly, we should use some tools to make it have mathematical meaning, we then can solve it. In this subsection, we employ the expected value operator to transform the fuzzy rough model into Bi-fuzzy EVM i.e. crisp model.

Based on the definition of the expected value of Bi-fuzzy events $f(x, \xi)$ and $g_r(x, \xi)$, the general model for Fu-Ro EVM is proposed as follows,

$$\begin{cases} Max \ \ \mathrm{E}\Big[f(x,\xi)\Big] \\ s.t \quad \begin{cases} E\Big[g_r(x,\xi)\Big] \leq 0, r = 1, 2, \cdots, p \\ x \in X \end{cases} \end{cases} \tag{3.49}$$

where x is n-dimensional decision vector and ξ is n-dimensional Bi-fuzzy variable.

3.13.5 Multi-objective Bi-random Model

Now a general equilibrium chance-constrained multi objective programming model with bi-random parameters can be formulated as

$$\left.\begin{array}{ll} & \text{minimize}\{\bar{f}_1, \bar{f}_2, ...\bar{f}_m\} \\ \text{subject to} & Ch^e\{f_i(x,\xi) \leq \bar{f}_i\} \geq \alpha_i, i = 1, 2, ...m \\ & Ch^e\{g_r(x,\xi) \leq 0\} \geq \beta_r, r = 1, 2, ...p \\ & x \in D \end{array}\right\} \tag{3.50}$$

where x is an n-dimensional decision vector, ξ is a m-dimensional bi-random vector, D is a fixed set that is usually determined by a finite number of inequalities involving functions of x, and f_i and g_r are (m + n)-dimensional real-valued continuous functions, α_i and β_r are predetermined confidence levels, i = 1, 2,..., m, r = 1, 2,..., p.

3.13.6 Equivalent Crisp of multi-objective Bi-random Model

It follows from the definition (3.11) that Equ.(3.50) formulated as given below

$$\left.\begin{array}{ll} & \text{minimize}\{\bar{f}_1, \bar{f}_2, ...\bar{f}_m\} \\ \text{s.t} & Ch^e\{\tilde{\tilde{c}}_i^T x \leq \bar{f}_i\} \geq \alpha_i, i = 1, 2, ...m \\ & Ch^e\{\tilde{\tilde{a}}_r^T x \leq \tilde{\tilde{b}}_r\} \geq \beta_r, i = 1, 2, ...p \\ & x \in D \end{array}\right\} \tag{3.51}$$

where $\tilde{\tilde{c}}_i(\omega)$=($\tilde{\tilde{c}}_{i1}(\omega)$, $\tilde{\tilde{c}}_{i2}(\omega)$,..., $\tilde{\tilde{c}}_{in}(\omega))^T$, $\tilde{\tilde{a}}_r(\omega)$=($\tilde{\tilde{a}}_{r1}(\omega)$, $\tilde{\tilde{a}}_{r2}(\omega)$,..., $\tilde{\tilde{a}}_{rn}(\omega))^T$ are bi-random vectors, $\tilde{\tilde{b}}_r$ are bi-random variables, i=1, 2,..., m, r=1, 2,..., p.

It follows from Theorem 3.3 that Equ.(3.51) can be rewritten as

$$\left.\begin{array}{ll} & \text{minimize}\{\bar{f}_1, \bar{f}_2, ...\bar{f}_m\} \\ \text{s.t} & Pr\{\omega \in \Omega | Pr\{\tilde{\tilde{c}}_i^T(\omega)x \leq \bar{f}_i\} \geq \alpha_i, i = 1, 2, ...m \\ & Pr\{\omega \in \Omega | Pr\{\tilde{\tilde{a}}_r^T(\omega)x \leq \tilde{\tilde{b}}_r\} \geq \beta_r, r = 1, 2, ...p \\ & x \in D \end{array}\right\} \tag{3.52}$$

Now using Lemma 3.3 and 3.4, then Equ.(3.52) represents as

$$\left.\begin{array}{c} \text{minimize}\{\bar{f}_1, \bar{f}_2, ...\bar{f}_m\} \\ \text{s.t } \mu_i^{cT}x + \Phi^{-1}(\alpha_i)\sqrt{(x^T V_i^c x)} + \Phi^{-1}(\alpha_i)\sqrt{(x^T V_i^{nc} x)} \leq \bar{f}_i, i = 1, 2, ...m \\ \mu_r^{aT}x + \Phi^{-1}(\beta_r)\sqrt{(x^T V_i^a x + (\sigma_r^b)^2)} + \Phi^{-1}(\beta_r)\sqrt{(x^T V_r^{na} x + (\sigma_r^{nb})^2))} \leq \mu_r^b, \\ r = 1, 2, ...p, x \in D. \end{array}\right\} \tag{3.53}$$

where Φ is the standardized normal distribution and α_i, β_r are predetermined confidence levels.

3.13.7 Multi-objective Bi-rough Model

Consider the following multi objective programming problem with Bi-rough coefficients

min $\{f_1(x,\xi), f_2(x,\xi), ..., f_m(x,\xi)\}$
s.t. $g_k(x,\xi) \leq 0$, k=1,2,.., p.

where x is a n-dimensional vector, $\xi = (\xi_1, \xi_2, ..., \xi_n)$ is a Bi- rough vector, $f_i(x,\xi)$ are objective functions, i=1,2.., m and $g_k(x,\xi)$ are constraint functions, k=1,2,.., p. Now the above model not well defined as the existence of Bi- rough vector ξ. The Bi- rough chance constrained multi objective programming (BiR-CCMOP) [103] model was proposed as follows :

$$\left.\begin{array}{ll} & \text{minimize}\{\bar{f}_1, \bar{f}_2, ...\bar{f}_m\} \\ \text{subject to} & Ch\{f_i(x,\xi) \leq f_i\}(\alpha_i) \geq \beta_i, i = 1, 2, ...m \\ & Ch\{g_k(x,\xi) \leq 0\}(\eta_k) \geq \sigma_k, k = 1, 2, ...p \\ & x \in D \end{array}\right\} \tag{3.54}$$

where Ch is the chance measure of the Bi- rough events and $\alpha_i, \beta_i, \eta_k$ and σ_k are predetermined confidence levels, i=1,2,..m, k=1,2,...p.

3.13.8 Equivalent Crisp of multi-objective Bi-rough Model

Also the chance written as given

$$Ch\{f_i(x,\xi) \le \bar{f}_i\}(\alpha_i) \ge \beta_i \Leftrightarrow Ex\{\lambda|Tr\{\{f_i(x,\xi) \le \bar{f}_i\} \ge \alpha_i\} \ge \beta_i$$
$$Ch\{g_r(x,\xi) \le 0\}(\eta_k) \ge \sigma_r \Leftrightarrow Ex\{\lambda|Tr\{\{g_r(x,\xi) \le 0\} \ge \eta_k\} \ge \sigma_k$$

Now the above equation of BiRCCMOP can written as

$$\left.\begin{array}{ll} & \text{minimize}\{f_1, f_2, ..., f_m\} \\ \text{subject to} & Ex\{\lambda|Tr\{\hat{\hat{c}}_i(\lambda)^T x \le f_i\} \ge \beta_i\} \ge \alpha_i, i = 1, 2, ...m \\ & Ex\{\lambda|Tr\{\hat{\hat{e}}_i(\lambda)^T x \le \hat{\hat{b}}_k\} \ge \sigma_k\} \ge \eta_k, k = 1, 2, ...p \\ & x \ge 0 \end{array}\right\} \quad (3.55)$$

where $\alpha_i, \beta_i, \eta_k, \gamma_k \in [0,1]$ are the predetermined confidence levels, c$_i(\lambda)$ is a rough variable as ($[\xi - p,\ \xi + q]$,$[\xi - r,\ \xi + s]$) where p< q < r < s are any real numbers and ξ is a rough variable ([a,b], [c,d]) and e$_i(\lambda)$ - b$_i(\lambda)$ also rough variable as ($[\xi_1 - p_1,\ \xi_1 + q_1]$,$[\xi_1 - r_1,\ \xi_1 + s_1]$) where $p_1 < q_1 < r_1 < s_1$ are any real numbers and ξ_1 is a rough variable $([a_1, b_1], [c_1, d])$. Ex$\{.\}$ denotes expectation of the rough events in $\{.\}$, and Tr$\{.\}$ denotes the trust measure of the events in $\{.\}$. Here the above model known as Ex-Tr constrained multi objective programming model.

Thus the above model transformed as minimized $\{f_1, f_2, ..., f_m\}$ for objective functions

$$f_i = \begin{cases} u - r + 2\alpha(s + r), & \text{if } u - r \le f_i \le u - p \\ \frac{u(p+q+r+s) - r(q+p) - p(s+r) + 2\alpha(s+r)(q+p)}{p+q+r+s} & \text{if } u - p \le f_i \le u + q \\ u - r + (2\alpha - 1)(s + r) & \text{if } u + q \le f_i \le u + s \end{cases} \quad (3.56)$$

and for constraint as

$$w \ge \begin{cases} u_1 - r_1 + 2\eta(s_1 + r_1), & \text{if } u_1 - r_1 \le w \le u_1 - p_1 \\ \frac{u_1(p_1+q_1+r_1+s_1) - r_1(q_1+p_1) - p_1(s_1+r_1) + 2\eta(s_1+r_1)(q_1+p_1)}{p_1+q_1+r_1+s_1} & \text{if } u_1 - p_1 \le w \le u_1 + q_1 \\ u_1 - r_1 + (2\eta - 1)(s_1 + r_1) & \text{if } u_1 + q_1 \le w \le u_1 + s_1 \end{cases} \quad (3.57)$$

where w be given crisp values, E[ξ] = u = $\frac{a+b+c+d}{4}$, E[ξ_1] = u$_1$ = $\frac{a_1+b_1+c_1+d_1}{4}$.

3.13.9 Multi-objective Fu-Ra Model

Here the general fuzzy random chance-constrained decision making model as follows.

$$\left.\begin{array}{c}\text{minimize}\{\bar{f}_1, \bar{f}_2, ...\bar{f}_m\} \\ \text{s.t. } Ch\{f_i(x,\xi) \leq \bar{f}_i\}(\gamma_i) \geq \delta_i, i = 1,2,...m \\ Ch\{g_r(x,\xi) \leq 0\}(\eta_r) \geq \theta_r, r = 1,2,...p \\ x \in X\end{array}\right\} \tag{3.58}$$

where Ch is the chance measure of the fuzzy random events, $\gamma_i, \delta_i, \eta_r, \theta_r$ are the predetermined confidence levels, f_i and x_i are the decision variables, i=1,2... m. Also the chance written as given

$$Ch\{f_i(x,\xi) \leq \bar{f}_i\}(\gamma_i) \geq \delta_i \Leftrightarrow Pr\{\omega|Pos\{\{f_i(x,\xi) \leq \bar{f}_i\} \geq \delta_i\} \geq \gamma_i$$
$$Ch\{g_r(x,\xi) \leq 0\}(\eta_r) \geq \theta_r \Leftrightarrow Pr\{\omega|Pos\{\{g_r(x,\xi) \leq 0\} \geq \theta_r\} \geq \eta_r$$

3.13.10 Equivalent Crisp of multi-objective Fu-Ra Model

Now the above optimization model Equ.(3.58) converted as below in Probability Possibility form

$$\left.\begin{array}{c}\text{minimize}\{\bar{f}_1, \bar{f}_2, ...\bar{f}_m\} \\ \text{s.t. } Pr\{\omega|Pos\{\{f_i(x,\xi) \leq \bar{f}_i\} \geq \delta_i\} \geq \gamma_i, i = 1,2,...m \\ Pr\{\omega|Pos\{\{g_r(x,\xi) \leq 0\} \geq \theta_r\} \geq \eta_r, r = 1,2,...p \\ x \in X\end{array}\right\} \tag{3.59}$$

and in the Probability Necessity form

$$\left.\begin{array}{c}\text{minimize}\{\bar{f}_1, \bar{f}_2, ...\bar{f}_m\} \\ \text{s.t. } Pr\{\omega|Nes\{\{f_i(x,\xi) \leq \bar{f}_i\} \geq \delta_i\} \geq \gamma_i, i = 1,2,...m \\ Pr\{\omega|Nes\{\{g_r(x,\xi) \leq 0\} \geq \theta_r\} \geq \eta_r, r = 1,2,...p \\ x \in X\end{array}\right\} \tag{3.60}$$

where $\gamma_i, \delta_i, \eta_r, \theta_r \in [0,1]$ are the predetermined confidence levels.

3.13.11 Single Objective Fu-Ro Model

Let us consider the following single-objective decision making model with fuzzy rough coefficients:

$$\left\{\begin{array}{ll} Max & f(x,\xi) \\ s.t & \left\{\begin{array}{l} g_r(x,\xi) \leq 0, r = 1,2,\cdots,p \\ x \in X \end{array}\right. \end{array}\right. \tag{3.61}$$

where x is a n-dimensional decision vector, $\xi = (\xi_1, \xi_2, \xi_3, \cdots, \xi_n)$ is a Fu-Ro vector, $f(x, \xi)$ are objective functions, $i = 1, 2, ..., m$. Because of the existence of Fu-Ro vector ξ, problem (3.61) is not well-defined. That is, the meaning of maximizing $f(x, \xi)$ is not clear and constraints $g_r(x, \xi) \leq 0, r = 1, 2, \cdots, p$ do not define a deterministic feasible set.

3.13.12 Equivalent Crisp Model of Fu-Ro Model

For the single-objective model (3.61) with Fu-Ro parameters, we cannot deal with it directly, we should use some tools to make it have mathematical meaning, we then can solve it. In this subsection, we employ the expected value operator to transform the fuzzy rough model into Fu-Ro EVM i.e. crisp model.
Based on the definition of the expected value of fuzzy rough events $f(x, \xi)$ and $g_r(x, \xi)$, the general model for Fu-Ro EVM is proposed as follows,

$$\begin{cases} Max \quad \mathrm{E}\Big[f(x, \xi)\Big] \\ s.t \qquad \begin{cases} E\Big[g_r(x, \xi)\Big] \leq 0, r = 1, 2, \cdots, p \\ x \in X \end{cases} \end{cases} \tag{3.62}$$

where x is n-dimensional decision vector and ξ is n-dimensional fuzzy rough variable.

3.13.13 Multi-objective Fu-Ro Model

Consider the following multi objective programming problem with fuzzy rough coefficients

min $\{f_1(x, \xi), f_2(x, \xi), ..., f_m(x, \xi)\}$
s.t. g$_k(x, \xi) \leq 0$, k=1,2,.., p.

where x is a n-dimensional vector, $\xi = (\xi_1, \xi_2, ..., \xi_n)$ is a fuzzy rough vector, $f_i(x, \xi)$ are objective functions, i=1,2.., m and g$_k(x, \xi)$ are constraint functions, k=1,2,.., p. Now the above model not well defined as the existence of fuzzy rough vector ξ. The fuzzy rough chance constrained multi objective programming (FR-

CCMOP) [32] model was proposed as follows :

$$\left.\begin{array}{lc} & \text{minimize}\{\bar{f}_1, \bar{f}_2, ...\bar{f}_m\} \\ \text{subject to} & Ch\{f_i(x,\xi) \le f_i\}(\alpha_i) \ge \beta_i, i = 1,2,...m \\ & Ch\{g_k(x,\xi) \le 0\}(\eta_k) \ge \sigma_k, k = 1,2,...p \\ & x \in D \end{array}\right\} \quad (3.63)$$

where Ch is the chance measure of the fuzzy rough events and $\alpha_i, \beta_i, \eta_k$ and σ_k are predetermined confidence levels, i=1,2,..m, k=1,2,...p. Also the chance written as given

$$Ch\{f_i(x,\xi) \le \bar{f}_i\}(\alpha_i) \ge \beta_i \Leftrightarrow Tr\{\lambda|Pos\{\{f_i(x,\xi) \le \bar{f}_i\} \ge \alpha_i\} \ge \beta_i$$
$$Ch\{g_r(x,\xi) \le 0\}(\eta_k) \ge \sigma_r \Leftrightarrow Tr\{\lambda|Pos\{\{g_r(x,\xi) \le 0\} \ge \eta_k\} \ge \sigma_k$$

3.13.14 Equivalent Crisp of multi-objective Fu-Ro Model

Now the above equation of FRCCMOP can written as Xu et al. [173]

$$\left.\begin{array}{lc} & \text{minimize}\{f_1, f_2, ..., f_m\} \\ \text{subject to} & Tr\{\lambda|Pos\{\hat{\tilde{c}}_i(\lambda)^T x \le f_i\} \ge \beta_i\} \ge \alpha_i, i = 1,2,...m \\ & Tr\{\lambda|Pos\{\hat{\tilde{e}}_i(\lambda)^T x \le \hat{\tilde{b}}_k\} \ge \sigma_k\} \ge \eta_k, k = 1,2,...p \\ & x \ge 0 \end{array}\right\} \quad (3.64)$$

where $\alpha_i, \beta_i, \eta_k, \gamma_k \in [0,1]$ are the predetermined confidence levels, c$_i(\lambda)$ is a rough variable as ([a, b],[c, d]) and e$_i(\lambda)$ - b$_i(\lambda)$ also rough variable as ([a_1, b_1],[c_1, d_1]). Pos$\{.\}$ denotes possibility of the fuzzy events in $\{.\}$, and Tr$\{.\}$ denotes the trust measure of the events in $\{.\}$. Here the above model known as Tr-Pos constrained multi objective programming model Xu et al. [173].
Thus the above model transformed as minimized $\{f_1, f_2, ..., f_m\}$ for objective functions

$$f_i = \begin{cases} c + 2\alpha_i(d-c) - L^{-1}(\beta_i)\gamma_i^{cT} x, & \text{if } c \le S \le a \\ \frac{c(b-a)+a(d-c)+2\alpha_i(d-c)(b-a)}{d-c+b-a} - L^{-1}(\beta_i)\gamma_i^{cT} x & \text{if } a \le S \le b \\ c + (d-c)(2\alpha_i - 1) - L^{-1}(\beta_i)\gamma_i^{cT} x & \text{if } b \le S \le d \\ d - L^{-1}(\beta_i)\gamma_i^{cT} x & \text{if } S \ge d \end{cases} \quad (3.65)$$

and for constraint as

$$W \geq \begin{cases} c_1 + 2(d_1 - c_1)\eta_k, & \text{if } c_1 \leq W \leq a_1 \\ \frac{c_1(b_1-a_1)+a_1(d_1-c_1)+2\eta_k(d_1-c_1)(b_1-a_1)}{d_1-c_1+b_1-a_1} & \text{if } a_1 \leq W \leq b_1 \\ c_1 + (d_1 - c_1)(2\eta_k - 1) & \text{if } b_1 \leq W \leq d_1 \\ d_1 & \text{if } d_1 \leq W \end{cases} \tag{3.66}$$

where W=$R^{-1}(\sigma_k)\delta_k^b + L^{-1}(\sigma_k)\gamma_k^{eT}x, S = f_i + L^{-1}(\beta_i)\delta_i^{cT}x$.

3.13.15 Single Objective Ra-Fu Model

Consider the following random fuzzy multi-objective linear programming problems formulated as

$$\begin{cases} \min\limits_{x} & \tilde{\bar{C}}_l x, l = 1, 2, \cdots, k \\ s.t & \begin{cases} \tilde{\bar{A}}_i(x, \widehat{\xi}) \leq \tilde{\bar{B}}, i = 1, 2, \cdots, r \\ x \geq 0 \end{cases} \end{cases} \tag{3.67}$$

where x is an n dimensional decision variable column vector.
When we formulate multi objective programming problems as stochastic programming (Birge and Louveaux [15] and Infanger, [72]), one of the most basic approaches is to assume that $\tilde{\bar{c}} = (\tilde{\bar{c}}_{111}, \tilde{\bar{c}}_{112}, \cdots, \tilde{\bar{c}}_{11n}; \cdots, \tilde{\bar{c}}_{n11}, \tilde{\bar{c}}_{n12}, \cdots, \tilde{\bar{c}}_{nmk})$ is a random variable vector which has multivariate Gaussian random distribution.

3.13.16 Equivalence Crisp of Ra-Fu Model

In this investigation, we assume that the mean of $\tilde{\bar{c}}_l$ is represented with an L-L fuzzy number $\mu_{\tilde{\bar{c}}_l}$ characterized by the membership function, is given by

$$\mu_{\tilde{\bar{M}}_l^c}(\tau) = \begin{cases} L\left(\frac{m_{lj}^c - \tau}{\alpha_{lj}}\right) & \text{for } m_{lj}^c \geq \tau \\ L\left(\frac{\tau - m_{lj}^c}{\alpha_{lj}}\right) & \text{for } m_{lj}^c < \tau \end{cases}$$

where the shape functions L is a nonnegative continuous function satisfying the following conditions:
a. $L(t)$ is non increasing for any $t > 0$.
b. $L(0) = 1$.

c. $L(t) = L(-t)$ for any $t \in R$.
d. There exists a t_0^L such that $L(t) = 0$ for any t larger than t_0^L.
The parameters m_{lj}^c, α_{lj}^c and β_{lj}^c are real constant values, and the values of α_{lj}^c and β_{lj}^c represent left and right spreads of the fuzzy number $\tilde{\tilde{M}}_{lj}^c$. The Fig 3.17. illustrates an example of the membership function $\mu_{\tilde{\tilde{M}}_l^c}(\tau)$.

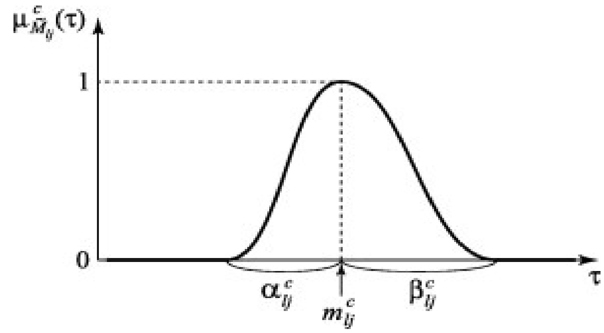


Figure 3.15: An example of the membership function $\mu_{\tilde{\tilde{M}}_l^c}(\tau)$

Recently, Katagiri et al.[80] have developed random fuzzy multi-objective linear programming: Optimization of possibilistic value at risk (pVaR). Using possibility and necessity approach the above problem (3.67) can be expressed as

$$\begin{cases} \min\limits_{x} \quad f_l, \quad l = 1, 2, \cdots, k \\ s.t \quad \begin{cases} Pos\{Prob\{\tilde{\bar{C}}_l x \leq f_l\} \geq \hat{\theta}_l^{obj}\} \geq \hat{h}_l^{obj} \\ Nec\{Prob\{\tilde{\bar{C}}_l x \leq f_l\} \geq \hat{\theta}_i^{obj}\} \geq \hat{h}_i^{obj} \\ Pos\{Prob\{\tilde{\bar{A}} x \leq \tilde{\bar{B}}\} \geq \hat{\theta}_i^{cst}\} \geq \hat{h}_i^{cst} \\ Nec\{Prob\{\tilde{\bar{A}} x \leq \tilde{\bar{B}}\} \geq \hat{\theta}_i^{cst}\} \geq \hat{h}_i^{cst} \\ x \geq 0, l = 1, 2, \cdots, k, i = 1, 2, \cdots, r \end{cases} \end{cases} \quad (3.68)$$

Using Theorem-3.13 and Theorem-3.14, the above problem (3.68) can be written as

$$\begin{cases} \min_x \quad f_l, \quad l = 1, 2, \cdots, k \\ s.t \quad \begin{cases} \sum_{j=1}^{n} \{m_{lj}^c - L^*(\hat{h}_l^{obj})\alpha_{lj}^c\} x_j + \Phi^{-1}(\hat{\theta}_l^{obj})\sqrt{x^t V_l^c x} \leq f_l \\ \sum_{j=1}^{n} \{m_{lj}^c - L^*(1 - \hat{h}_l^{obj})\beta_{lj}^c\} x_j + \Phi^{-1}(\hat{\theta}_l^{obj})\sqrt{x^t V_l^c x} \leq f_l \\ \sum_{j=1}^{n} \{m_{ij}^c - L^*(\hat{h}_i^{obj})\alpha_{ij}^a\} x_j + \Phi^{-1}(\hat{\theta}_i^{cst}) \\ \sqrt{x^t V_l^c x + (\sigma_i^a)^2} \leq m_i^b + L^*(\hat{h}_i^{cst})\beta_{ij}^b \\ \sum_{j=1}^{n} \{m_{ij}^c - L^*(1 - \hat{h}_i^{obj})\alpha_{ij}^b\} x_j + \Phi^{-1}(\hat{\theta}^{cst}) \\ \sqrt{x^t V_i^c x + (\sigma_i^b)^2} \leq m_i^b - L^*(1 - \hat{h}^{cst})\alpha_{ij}^b \\ x \geq 0, i = 1, 2, \cdots, r \end{cases} \end{cases} \tag{3.69}$$

3.13.17 Multi-objective Ra-Ro Model

Consider the following multi objective programming problem with random rough coefficients

$$\min \{f_1(x,\xi), f_2(x,\xi), ..., f_m(x,\xi)\}$$
$$\text{s.t. } g_k(x,\xi) \leq 0, \text{ k=1,2,.., p.}$$

where x is a n-dimensional decision vector, $\xi = (\xi_1, \xi_2, ..., \xi_n)$ is a random rough vector, $f_i(x,\xi)$ are objective functions, i=1,2.., m and $g_k(x,\xi)$ are constraint functions, k=1,2,.., p.

Now the above model not well defined as the existence of random rough vector ξ. The random rough chance constrained multi objective programming (RRCC-MOP) (Liu, [100]) model was proposed as follows :

$$\left.\begin{array}{lc} & \text{minimize}\{\bar{f}_1, \bar{f}_2, ...\bar{f}_m\} \\ \text{subject to} & Ch\{f_i(x,\xi) \leq f_i\}(\alpha_i) \geq \beta_i, i = 1, 2, ...m \\ & Ch\{g_k(x,\xi) \leq 0\}(\eta_k) \geq \gamma_k, k = 1, 2, ...p \\ & x \in D \end{array}\right\} \tag{3.70}$$

Where $\alpha_i, \beta_i, \eta_k$ and σ_k are predetermined confidence levels. i=1,2,..m, k=1,2,...p. Also the chance written as given

$$Ch\{f_i(x,\xi) \leq \bar{f}_i\}(\alpha_i) \geq \beta_i \Leftrightarrow Tr\{\lambda | Pr\{\{f_i(x,\xi) \leq \bar{f}_i\} \geq \alpha_i\} \geq \beta_i$$
$$Ch\{g_k(x,\xi) \leq 0\}(\eta_k) \geq \gamma_k \Leftrightarrow Tr\{\lambda | Pr\{\{g_k(x,\xi) \leq 0\} \geq \eta_k\} \geq \gamma_k$$

3.13.18 Equivalence Crisp of Multi-objective Ra-Ro Model

Now the above equation of RRCCMOP can written as Xu et al. ([172])

$$\left.\begin{array}{ll} & \text{minimize}\{f_1, f_2, ..., f_m\} \\ \text{subject to} & Tr\{\lambda|Pr\{\hat{\bar{c}}_i(\lambda)^T x \leq f_i\} \geq \beta_i\} \geq \alpha_i, i = 1, 2, ...m \\ & Tr\{\lambda|Pr\{\hat{\bar{e}}_k(\lambda)^T x \leq \hat{\bar{b}}_k\} \geq \gamma_k\} \geq \eta_k, k = 1, 2, ...p \\ & x \geq 0 \end{array}\right\} \quad (3.71)$$

where $\alpha_i, \beta_i, \eta_k, \gamma_k \in [0, 1]$ are the predetermined confidence levels, $\hat{\bar{c}}_i(\lambda)$ is assume that $\hat{\bar{c}}_i(\lambda) \sim N(c_i(\lambda), V_i^c)$ variate where $c_i(\lambda)$ is rough variable as ([a,b],[c,d]) and $\hat{\bar{e}}_k(\lambda) \sim N(e_k(\lambda), V_k^e)$ variate, $\hat{\bar{b}}_k(\lambda) \sim N(b_k(\lambda), (\sigma_k^b)^2)$ variate where $e_k(\lambda)$ - $b_k(\lambda)$ is a rough variable ($[a_1, b_1], [c_1, d_1]$). Pr$\{.\}$ denotes probability of the random events in $\{.\}$, and Tr$\{.\}$ denotes the trust measure of the rough events in $\{.\}$. Here the above model known as Tr-Pr constrained multi objective programming model.

Thus the above model transformed as $minimize\{f_1, f_2, ..., f_m\}$

$$f_i = \begin{cases} c + 2\alpha_i(d - c) + \phi^{-1}(\beta_i)\sqrt{x^T V_i^c x}, & \text{if } c \leq W \leq a \\ \frac{c(b-a)+a(d-c)+2\alpha_i(d-c)(b-a)}{d-c+b-a} + \phi^{-1}(\beta_i)\sqrt{x^T V_i^c x} & \text{if } a \leq W \leq b \\ c + (d - c)(2\alpha_i - 1) + \phi^{-1}(\beta_i)\sqrt{x^T V_i^c x} & \text{if } b \leq W \leq d \\ d + \phi^{-1}(\beta_i)\sqrt{x^T V_i^c x} & \text{if } W \geq d \end{cases} \quad (3.72)$$

$$s.t. \quad M \geq \begin{cases} c_1 + 2(d_1 - c_1)\eta_k, & \text{if } c_1 \leq M \leq a_1 \\ \frac{c_1(b_1-a_1)+a_1(d_1-c_1)+2\eta_k(d_1-c_1)(b_1-a_1)}{d_1-c_1+b_1-a_1} & \text{if } a_1 \leq M \leq b_1 \\ c_1 + (d_1 - c_1)(2\eta_k - 1) & \text{if } b_1 \leq M \leq d_1 \\ d_1 & \text{if } d_1 \leq M \end{cases} \quad (3.73)$$

where M= - $\phi^{-1}(\gamma_k)\sqrt{x^T V_k^e x + (\sigma_k^b)^2}, W = f_i - \phi^{-1}(\beta_i)\sqrt{x^T V_i^c x}$.

.

Part II

Single Objective Optimization by Single/Multi-Heuristic Methods

Chapter 4

Single Objective Optimization Using Single Heuristic Methods

4.1 Introduction

In this chapter, modifications and new operators that have been developed for genetic algorithms are applied to TSP are presented. These modifications are made for a number of reasons for example to improve the quality of end results or to reduce the computation time. Researchers have adopted a number of different approaches to achieve the above goals. In this chapter, the presented approaches, are: an improved GA (IGA), an adaptive GA (AGA), a modified GA (MGA), a rough GA (RGA) and a rough extended GA (ReGA) developed to solve solid TSP. Here modified probabilistic, rough age based, rough extended age based selections, Comparison crossover and problem (generation) dependent mutations are developed. Also the solid TSP models with different constraints as safety, risk/discomfort and time constraints etc. are formulated. Again solid TSPs are studied in crisp, fuzzy, interval values, rough, bi-fuzzy, bi-rough, bi-random, random, fuzzy-rough, fuzzy-random, random-fuzzy and random-rough environments. This chapter concludes with an examination of how these modifications of GAs are effective for proposed solid TSPs through some statistical tests such as ANOVA, Friedman test and Post hoc comparison, etc. are presented.

4.2 Model-4.1: An Improved Genetic Algorithm and Its Application in Constrained Solid TSP in Uncertain Environments[1]

In this investigation, we propose an improved genetic algorithm (IGA) to solve Constrained Solid Travelling Salesman Problems (CSTSPs) in crisp, fuzzy, rough, and fuzzy-rough environments. IGA is a combination of proposed probabilistic selection, cyclic crossover, and nodes-oriented random mutation. Here, CSTSPs in different uncertain environments have been designed and solved by the proposed algorithm. In the present problem, there are some risks of travelling between the cities through different conveyances. The salesman desires to maintain certain safety level always to travel from one city to another and a total safety for his entire tour. Costs and safety level factors for travelling between the cities are different. The requirement of minimum safety level is expressed in the form of a constraint. The safety factors are expressed by crisp, fuzzy, rough, and fuzzy-rough numbers. The models are formulated as minimization problems of total cost subject to crisp, fuzzy, rough, or fuzzy-rough constraints. The model is numerically illustrated with appropriate data values. Optimum results for the different models are presented via IGA. Moreover, the problems from the TSPLIB (standard data set) are tested with the proposed algorithm. Some statistical tests are performed to established the effectiveness of the proposed IGA.

4.2.1 Proposed IGA

(a) Representation:

Here a complete tour on N cities represents a solution. So an N-dimensional integer vector $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$ is used to represent a solution, where $x_{i1}, x_{i2}, ..., x_{iN}$ represent N consecutive cities in a tour. For solid TSP another integer vector $V_k = (v_{k1}, v_{k2}, ..., v_{kN})$ is used to represent the conveyance types used for travel between different cities. Here v_{kj} represents the conveyance (an integer) used to travel from city x_{ij} to $x_{i(j+1)}$ for j = 1, 2, . . . ,N 1 and v_{kN} represents the conveyance type used to travel from city x_{iN} to x_{i1}.

[1]This portion is published in **Facets of Uncertainties and Applications**, Springer Proceedings in Mathematics & Statistics 125, DOI 10.1007/978-81-322-2301-6_14, with title *An Improved Genetic Algorithm and Its Application in Constrained Solid TSP in Uncertain Environments.*

(b) Initialization:

Population size number of such solutions $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$, i = 1, 2, . . . , pop size, are randomly generated by random number generator, such that each solution satisfies the constraints of the problem. A separate sub function, s(X_i) checks the constraint. For STSP, another integer vector $V_k = (v_{k1}, v_{k2}, ..., v_{kN})$ is randomly generated corresponding to the solution X_i, to represent the conveyance types used to travel between different cities. So in that case (X_i, V_k) represents a solution.

(c) Probabilistic Selection:

For minimum cost objective, it is better to choose that population which is in the neighborhood of the minimum solution of the entire solution space. So we get the convergence rate much high. From the initial population, choose the best fitted population for TSP. It is chosen as the most minimum fitness value (say f_{min}). To form the matting pool, we use the **Boltzmann-Probability** of the each chromosome from the initial population.

Let, $p_B = e^{((f_{min} - f(X_i))/T)}$,

where $T = T_0(1-a)^k$, k=(1+100*(g/G)), g=current generation number, G= maximum generation, T_0= rand[5,100], a=rand[0,1], f(X_i) means the chromosome corresponding to X_i, i=chromosome number.

(d) Procedure of Selection:

input : Max-gen (G), Probability of selection (p_s), pop - size.
output : Matting pool.

```
begin
  for (i=1 to G)
        for (j=1 to pop - size)
          r=rand[0,1];
          T0= rand[5,100];
          a=rand[0,1];
          k=(1+100*(i/G));
          T=T0(1-a)^k;
          pB=e^((fmin-f(Xj))/T);
              if (r < pB )
                    choose the current chromosome;
                    j++;
              else
```

Select the corresponding chromosome of f_{min};
j++;
end for
end for
end

(e) Cyclic Crossover: which already discussed in section 2.1.4(ii)(c).

(f) Nodes Oriented Mutation:

(i) Selection for mutation: For each solution of p(n), generate a random number r from the range [0, 1]. If $r < p_m$ then the solution is taken for mutation.

(ii) Mutation Process: To mutate a solution $X = (x_1, x_2, ..., x_N)$ of TSP with T number of nodes, select T number of nodes randomly from the solution and replace their places in the solution, i.e., if randomly two nodes x_i, x_j are selected, then interchange x_i, x_j to get a child solution. The new solution, if it satisfies the constraint of the problem, replaces the parent solution. For CSTSP to mutate a solution (X,V), where $X = (x_1, x_2, ..., x_N)$,$V = (v_1, v_2, ..., v_p)$, at first an integer is randomly selected in the range [1, 2]. If 1 is selected, then another two random integers i, j are selected in the range [1, N]. Then interchange x_i, x_j to get the child solution. If 2 is selected, then another two random integers i and j are selected in the range [1,N] and [1, P] respectively. The value of v_p is replaced by j to get a child solution. If the child solution satisfies the constraint of the problem, then it replaces the parent solution.

Algorithm of IGA:

1 . Begin

2 . Initialize max generation (s_0), population size (pop size), p_c, p_m.

3 . Randomly generate initial population p(n)

4 . Evaluate initial population p(n) (i.e. fitness of the objective function from p(n))

5 . While n $\leq s_0$ do

  a . n = n + 1

  b . Probabilistic Selection

  c . Cyclic Crossover

d . Random Mutation

e . Evaluate p(n)

6 . Update

7 . End While

8 . Print optimum result

9 . End

(g) Complexity Analysis :

Genetic Algorithms are not chaotic, they are stochastic. The complexity depends on the genetic operators, their implementation (which may have a very significant effect on overall complexity), the representation of the individuals and the population, and obviously on the fitness function. Given the usual choices, a Genetic Algorithm, complexity is O(s_0(mn + mn + m)) with s_0 the number of generations, m the population size and n the size of the individuals. Therefore the complexity is on the order of O($s_0$nm)).

The genetic algorithms with cyclic crossover operators have time complexity O(s_0mn^2). The n^2 factor is due to the fact that all repair procedures need to scan all the possible pairs of cities and the complexity of the algorithm is O(n^2).

4.2.2 Mathematical Formulation and Its crisp equivalence

Model 4.1A: Constrained Solid TSP

The mathematical expression is already given in section 1.7.2(a).

Model 4.1B: CSTSP in Fuzzy Environment (FCSTSP)

If costs and safety factors and limit are fuzzy numbers, i.e, $\tilde{c}(i,j,k)$, $\tilde{s}(i,j,k)$ and $\tilde{s}_{min}$ respectively, then the TSP problem given by Equ. 1.4 reduces to:
Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one available conveyance in each step from the vehicle types $(v_1, v_2, ..., v_P)$

$$\left.\begin{array}{ll}\text{to minimize} & Z = \sum_{i=1}^{N-1} \tilde{c}(x_i, x_{i+1}, v_i) + \tilde{c}(x_N, x_1, v_l) \\ \text{subject to} & \sum_{i=1}^{N-1} \tilde{s}(x_i, x_{i+1}, v_i) + \tilde{s}(x_N, x_1, v_l) \geq \tilde{s}_{min} \\ \multicolumn{2}{l}{where\ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or\ P\}}\end{array}\right\} \quad (4.1)$$

Possibilistic (Optimistic) Approach:
The above Equ. 4.1 is converted as given below:
Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one of the available conveyances in each step from the vehicle types $(v_1, v_2, ..., v_P)$

$$\left.\begin{array}{c} \text{to minimize} \quad F \\ \text{subject to} \quad Pos(\sum_{i=1}^{N-1} \tilde{c}(x_i, x_{i+1}, v_i) + \tilde{c}(x_N, x_1, v_l) < F) \geq \alpha_3 \\ Pos(\sum_{i=1}^{N-1} \tilde{s}(x_i, x_{i+1}, v_i) + \tilde{s}(x_N, x_1, v_l) \geq \tilde{s}_{min}) \geq \beta_3 \\ where \ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or\ P\} \end{array}\right\} \quad (4.2)$$

where α_3, β_3 are predefined levels of possibility which are entirely determined by the salesman. If we consider the fuzzy numbers as TFNs,
$\tilde{c}(i, j, k) = (c(i, j, k)_1, c(i, j, k)_2, c(i, j, k)_3)$, $\tilde{S}(i, j, k) = (s(i, j, k)_1, s(i, j, k)_2, s(i, j, k)_3)$, $\tilde{s}_{min} = (s_1, s_2, s_3)$.

$$\text{where } F_j = \sum_{i=1}^{N-1} c(x_i, x_{i+1}, v_i)_j + c(x_N, x_1, v_l)_j, \ \ j = 1, 2, 3.$$

$$\text{and } \ S_j = \sum_{i=1}^{N-1} s(x_i, x_{i+1}, v_i)_j + s(x_N, x_1, v_l)_j, \ \ j = 1, 2, 3.$$

where $x_i \neq x_j, i, j = 1, 2...N.$ $v_i, v_l \in \{1, 2.., or\ P\}$
The objective function in Equ. 4.2 changed to

$$\left.\begin{array}{c} \text{minimize} \quad F_1 + \alpha_3(F_2 - F_1) \\ \text{subject to} \quad \dfrac{S_3 - s_1}{S_3 - S_2 + s_2 - s_1} \geq \beta_3 \end{array}\right\} \quad (4.3)$$

Here α_3, β_3 are predefined possibility levels.

Necessity (Pessimistic) Approaches:
Similarly, converting the fuzzy expression of Equ. 4.1 in pessimistic sense, we

get as follows: Using necessity measure, we have

$$\left.\begin{array}{c} \text{minimize} \quad F \\ \text{subject to} \quad Nes(\sum_{i=1}^{N-1} \tilde{c}(x_i, x_{i+1}, v_i) + \tilde{c}(x_N, x_1, v_l) < F) \geq \alpha_4 \\ Nes(\sum_{i=1}^{N-1} \tilde{s}(x_i, x_{i+1}, v_i) + \tilde{s}(x_N, x_1, v_l) \geq \tilde{s}_{min}) \geq \beta_4 \\ where\ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\} \end{array}\right\} \quad (4.4)$$

where α_4, β_4 are predefined levels of necessity which are entirely determined by the salesman. Then the above problem can be reduced accordingly as:
Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one of the available conveyance in each step from the vehicle types $(v_1, v_2, ..., v_P)$ so as

$$\left.\begin{array}{c} \text{to minimize} \quad F \\ \text{subject to} \quad \dfrac{F_3 - F}{F_3 - F_2} \leq 1 - \alpha_4 \\ \dfrac{s_3 - S_1}{S_2 - S_1 + s_3 - s_2} \leq 1 - \beta_4 \end{array}\right\} \quad (4.5)$$

The objective function is changed to

$$\left.\begin{array}{c} \text{to minimize} \quad F_3 - (1 - \alpha_4)(F_3 - F_2) \\ \text{subject to} \quad \dfrac{s_3 - S_1}{S_2 - S_1 + s_3 - s_2} \leq 1 - \beta_4 \end{array}\right\} \quad (4.6)$$

Here α_4 and β_4 are predefined necessity levels.

Model 4.1C: CSTSP under rough environments(RCSTSP):

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one of the available conveyances in each step from the vehicle types $(v_1, v_2, ..., v_P)$

$$\left.\begin{array}{c} \text{to minimize} \quad Z = \sum_{i=1}^{N-1} \hat{c}(x_i, x_{i+1}, v_i) + \hat{c}(x_N, x_1, v_l) \\ \text{subject to} \quad \sum_{i=1}^{N-1} \hat{s}(x_i, x_{i+1}, v_i) + \hat{s}(x_N, x_1, v_l) \geq \hat{s}_{min} \\ where\ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\} \end{array}\right\} \quad (4.7)$$

The above model can be converted as below: Here, the expected value of $\hat{C}$, $E(\hat{C}) = \frac{c_1+c_2+c_3+c_4}{4}$ is used.

Table 4.1: Test TSPLIB problems by IGA

Instance	Problem Size	Best Solution	IGA	Iteration	GA	Iteration
bays29	29×29	2020	2020	349	2020	571
bayg29	29×29	1610	1610	256	1610	480
fri26	26×26	937	937	202	937	368
dantzig42	42×42	699	699	245	699	986

Model 4.1D: CSTSP under fuzzy-rough environment:
Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one available conveyances in each step from the vehicle types $(v_1, v_2, ..., v_P)$

$$\left.\begin{array}{ll} \text{to minimize} & Z = \sum_{i=1}^{N-1} \tilde{\tilde{c}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{c}}(x_N, x_1, v_l) \\ \text{subject to} & \sum_{i=1}^{N-1} \tilde{\tilde{s}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{s}}(x_N, x_1, v_l) \geq \tilde{\tilde{s}}_{min} \\ where\ x_i \neq x_j, i, j = 1, 2...N, & v_i, v_l \in \{1, 2.., or P\} \end{array}\right\} \quad (4.8)$$

Crisp Equivalent Model:
To get the crisp equivalent by using Possibility-Expectation on the model given in Equ. 4.8, we have:
to minimize $\frac{(c_1+c_2+c_3+c_4)}{4} + \frac{\rho R_1-(1-\rho)L_1}{2}$
s.t. $\frac{(s_1+s_2+s_3+s_4)}{4} + \frac{\rho R_2-(1-\rho)L_2}{2} \geq \frac{(s_{min1}+s_{min2}+s_{min3}+c_{min4})}{4} + \frac{\rho R_3-(1-\rho)L_3}{2}$

4.2.3 Numerical Experiments

The proposed IGA is used for the standard TSP from the TSPLIB[162] and the results are compared with the simple GA (RW selection, Cyclic crossover and Random Mutation) in number of iteration the result shows the efficiency of the proposed algorithm (Table 4.1). Here p_c=0.34, p_m=0.3 and Pop-size=30, Max-Gen=400.

4.2.4 Statistical Test

Here, we study the best, worst and average results with standard deviation and percentage error of the standard TSP from TSPLIB [162] under of 25 individual run by IGA. The Table 4.11 given the results.

Table 4.2: Input Data: Crisp cost in CSTSP (Model 4.1A)

i/j	1	2	3	4	5
1	∞	15, 16, 17	18, 19, 20	12, 13, 14	20, 21, 22
2	27, 28, 29	∞	20, 21, 22	48, 49, 50	35, 36, 37
3	42, 43, 44	28, 29, 30	∞	30, 31, 32	25, 26, 27
4	38, 39, 40	30, 31, 32	8, 9,10	∞	20, 21, 22
5	66, 67, 68	22, 23, 24	35, 36, 37	30, 31, 32	∞

Table 4.3: Input Data: Crisp safety values in CSTSP (Model 4.1A)

i/j	1	2	3	4	5
1	∞	.3, .4, .5	.5, .6, .7	.2, .3, .4	.1, .2, .3
2	.6, .7, .8	∞	.2, .3, .4	.5, .6, .7	.3, .4, .5
3	.2, .3, .4	.3, .4, .5	∞	.2, .3, .4	.1, .2, .3
4	.6, .7, .8	.4, .3, .2	.6, .7, .8	∞	.3, .4, .5
5	.8, .7, .6	.3, .2, .1	.6, .5, .4	.4, .5, .6	∞

Table 4.4: Input Data: FCSTSP (Model 4.1B)

i/j	1	2	3	4	5
1	∞	(14, 15, 16) (15, 16, 17) (16, 17, 18)	(17, 18, 19) (18, 19, 20) (19, 20, 21)	(11, 12, 13) (12, 13, 14) (13, 14, 15)	(19, 20, 21) (20, 21, 22 (21, 22, 23)
2	(26, 27, 28) (27, 28, 29) (28, 29, 30)	∞	(19, 20, 21) (20, 21, 22) (21, 22, 23)	(47, 48, 49) (48, 49, 50) (49, 50, 51)	(34, 35, 36) (35, 36, 37) (36, 37, 38)
3	(41, 42, 43) (42, 43, 44) (43, 44, 45)	(27, 28, 29) (28, 29, 30) (29, 30, 31)	∞	(29, 30, 31) (30, 31, 32) (31, 32, 33)	(24, 25, 26) (25, 26, 27) (26, 27 ,28)
4	(37, 38, 39) (38, 39, 40) (39, 40, 41)	(29, 30, 31) (30, 31, 32) (31, 32, 33)	(7, 8, 9) (8, 9, 10) (9, 10, 11)	∞	(19, 20, 21) (20, 21, 22) (21, 22, 23)
5	(65, 66, 67) (66, 67, 68) (35, 36, 37)	(21, 22, 23) (67, 68, 69) (36, 37, 38)	(21, 22, 23) (22, 23, 24) (29, 30, 31)	(34, 35, 36) (23, 24, 25) (30, 31, 32)	∞

Table 4.5: Input Data: Fuzzy safety in FCSTSP (Model 4.1B)

i/j	1	2	3	4	5
1	∞	(.2, .3, .4) (.3, .4, .5) (.4, .5, .6)	(.4, .5, .6) (.5, .6, .7) (.6, .7, .8)	(.1, .2 .3) (.2, .3, .4) (.3, .4, .5)	(.3, .4, .5) (.4, .5, .6) (.5, .6, .7)
2	(.5, .6, .7) (.6, .7, .8) (.7, .8, .9)	∞	(.1, .2 .3) (.2, .3, .4) (.3, .4, .5)	(.4, .5, .6) (.5, .6, .7) (.6, .7, .8)	(.2, .3, .4) (.3, .4, .5) (.4, .5, .6)
3	(.1, .2 .3) (.2, .3, .4) (.3, .4, .5)	(.2, .3, .4) (.3, .4, .5) (.4, .5, .6)	∞	(.1, .2 .3) (.2, .3, .4) (.3, .4, .5)	(.1, .2 .3) (.2, .3, .4) (.3, .4, .5)
4	(.5, .6, .7) (.6, .7, .8) (.7, .8, .9)	(.3, .2, .1) (.4, .3, .2) (.5, .4, .3)	(.5, .6, .7) (.6, .7, .8) (.7, .8, .9)	∞	(.2, .3, .4) (.3, .4, .5) (.4, .5, .6)
5	(.9, .8, .7) (.8, .7, .6) (.7, .6, .5)	(.4, .3, .2) (.3, .2, .1) (.2, .1, .0)	(.7, .6, .5) (.6, .5, .4) (.4, .5, .6)	(.3, .4, .5) (.5, .4, .3) (.5, .6, .7)	∞

Table 4.6: Input Data: Rough costs in RCSTSP (Model 4.1C)

i/j	1	2	3	4	5
1	∞	([14, 15][13, 16]) ([15, 16][14, 17]) ([16, 17][15, 18])	([17, 18][16, 19]) ([18, 19][17, 20]) ([19, 20][18, 21])	([10, 11][9, 12]) ([11, 12][10, 13]) ([12, 13][11, 14])	([18, 19,][17, 20]) ([19, 20,][18, 21]) ([20, 21][19, 22])
2	([25, 26][24, 27]) ([26, 27][25, 28]) ([27, 28][26, 29])	∞	([18, 19][17, 20]) ([19, 20][18, 21]) ([20, 21][19, 22])	([46, 47][45, 48]) ([47, 48][46, 49]) ([48, 49][47, 50])	([33, 34][32, 35]) ([34, 35][33, 36]) ([35, 36][]34, 37)
3	([40, 41][39, 42]) ([41, 42][40, 43]) ([42, 43][41, 44])	([26, 27][25, 28]) ([27, 28][26, 29]) ([28, 29][27, 30])	∞	([28, 29][27, 30]) ([29, 30][28, 31]) ([30, 31][29, 32])	([23,24][22, 25]) ([24, 25][23, 26]) ([25,26][24, 27])
4	([36, 37][35, 38]) ([37, 38][36, 39]) ([30, 31][29, 32])	([28, 29][27, 30]) ([29, 30][28, 31]) ([8, 9][7, 10])	([6, 7][5, 8]) ([7, 8][6, 9]) ([19, 20,][18, 21])	∞	([18, 19,][17, 20]) ([38, 39][37, 40]) ([20, 21][19, 22])
5	([64, 65][63, 66]) ([65, 66][64, 67]) ([66, 67][65, 68])	([20, 21][19, 22]) ([21, 22][20, 23]) ([22, 23][21, 24])	([33, 34][32, 35]) ([34, 35][33, 36]) ([35, 36][]34, 37)	([28, 29][27, 30]) ([29, 30][28, 31]) ([30, 31][29, 32])	∞

Table 4.7: Input Data: Rough safety in CSTSP (Model 4.1C)

i/j	1	2	3	4	5
1	∞	([.1, .2][.01, .3]) ([.2, .3][.1, .4]) ([.3, .4][.2, .5])	([.3, .4][.2, .5]) ([.4, .5][.3, .6]) ([.5, .6][.4, .7])	([.2, .3][.1, .4]) ([.3, .4][.2, .5]) ([.4, .5][.3, .6])	([.5, .6][.4, .7]) ([.6, .7][.5, .8]) ([.7, .8][.6, .9])
	([.5, .6][.4, .7]) ([.6, .7][.5, .8]) ([.7, .8][.6, .9])	∞	([.1, .2][.0, .3]) ([.2, .3][.1, .4]) ([.3, .4][.2, .5])	([.3, .4][.2, .5]) ([.4, .5][.3, .6]) ([.5, .6][.4, .7])	([.2, .3][.1, .4]) ([.3, .4][.2, .5]) ([.4, .5][.3, .6])
3	([.1, .2][.01, .3]) ([.2, .3][.1, .4]) ([.3, .4][.2, .5])	([.2, .3][.1, .4]) ([.3, .4][.2, .5]) ([.4, .5][.3, .6])	∞	([.1, .2][.0, .3]) ([.2, .3][.1, .4]) ([.3, .4][.2, .5])	([.2, .3][.1, .4]) ([.3, .4][.2, .5]) ([.4, .5][.3, .6])
4	([.4, .5][.3, .6]) ([.5, .6][.4, .7]) ([.6, .7][.5, .8])	([.2, .3][.1, .4]) ([.3 .4][.2, .5]) ([.4, .5][.3, .6])	([.5, .6][.4, .7]) ([.6, .7][.5, .8]) ([.7, .8][.6, .9])	∞	([.2, .3][.1, .4]) ([.3, .4][.2, .5]) ([.4, .5][.3, .6])
5	([.7, .8][.6, .9]) ([.6, .7][.5, .8]) ([.5, .6][.4, .7])	([.4, .5][.3, .6]) ([.3, .4][.2, .5]) ([.2, .3][.1, .4])	([.6, .7][.5, .8]) ([.5, .6][.4, .7]) ([.4, .5][.3, .6])	([.2, .3][.1, .4]) ([.3, .4][.2, .5]) ([.4, .5][.3, .6])	∞

Table 4.8: Input Data: Fuzzy-rough costs in CSTSP (Model 4.1D)

i/j	1	2	3	4	5
1	∞	([14, 15][13, 16]) (14.5-5, 14.5, 14.5+5) ([15, 16][14, 17]) (15.5-5, 15.5, 15.5+5) ([16, 17][15, 18]) (16.5-5, 16.5, 16.5+5)	([17, 18][16, 19]) (17.5-5, 17.5, 17.5+5) ([18, 19][17, 20]) (18.5-5, 18.5, 18.5+5) ([19, 20][18, 21]) (19.5-5, 19.5, 19.5+5)	([10, 11][9, 12]) (10.5-5, 10.5, 10.5+5) ([11, 12][10, 13]) (11.5-5, 11.5, 11.5+5) ([12, 13][11, 14]) (12.5-5, 12.5, 12.5+5)	([18, 19,][17, 20]) (18.5-5, 18.5, 18.5+5) ([19, 20,][18, 21]) (19.5-5, 19.5, 19.5+5) ([20, 21][19, 22]) (20.5-5, 20.5, 20.5+5)
2	([25, 26][24, 27]) (25.2-5, 25.5, 25.5+5) ([26, 27][25, 28]) (26.5-5, 26.5, 26.5+5) ([27, 28][26, 29]) (27.5-5, 27.5, 27.5+5)	∞	([18, 19][17, 20]) (18.5-5, 18.5, 18.5+5) ([19, 20][18, 21]) (19.5-5, 19.5, 19.5+5) ([20, 21][19, 22]) (20.5-5, 20.5, 20.5+5)	([46, 47][45, 48]) (46.5-5, 46.5, 46.5+5) ([47, 48][46, 49]) (47.5-5, 47.5, 47.5+5) ([48, 49][47, 50]) (48.5-5, 48.5, 48.5+5)	([33, 34][32, 35]) (33.5-5, 33.5, 33.5+5) ([34, 35][33, 36]) (34.5-5, 34.5, 34.5+5) ([35, 36][34, 37]) (35.5-5, 35.5, 35.5+5)
3	[40, 41][39, 42]) (40.5-5, 40.5, 40.5+5) ([41, 42][40, 43]) (41.5-5, 41.5, 41.5+5) ([42, 43][41, 44]) (42.5-5, 42.5, 42.5+5)	([26, 27][25, 28]) (26.5-5, 26.5, 26.5+5) ([27, 28][26, 29]) (27.5-5, 27.5, 27.5+5) ([28, 29][27, 30]) (28.5-5, 28.5, 28.5+5)	∞	([28, 29][27, 30]) (28.5-5, 28.5, 28.5+5) ([29, 30][28, 31]) (29.5-5, 29.5, 29.5+5) ([30, 31][29, 32]) (30.5-5, 30.5, 30.5+5)	([23,24][22, 25]) (23.5-5, 23.5, 23.5+5) ([24, 25][23, 26]) (24.5-5, 24.5, 24.5+5) ([25,26][24, 27]) (25.5-5, 25.5, 25.5+5)
4	([36, 37][35, 38]) (36.5-5, 36.5, 36.5+5) ([37, 38][36, 39]) (37.5-5, 37.5, 37.5+5) ([38, 39][37, 40]) (38.5-5, 38.5, 38.5+5)	([28, 29][27, 30]) (28.5-5, 28.5, 28.5+5) ([29, 30][28, 31]) (29.5-5, 29.5, 29.5+5) ([30, 31][29, 32]) (30.5-5, 30.5, 30.5+5)	([6, 7][5, 8]) (6.5-5, 6.5, 6.5+5) ([7, 8][6, 9]) (7.5-5, 7.5, 7.5+5) ([8, 9][7, 10]) (8.5-5, 8.5, 8.5+5)	∞	([18, 19][17, 20]) (18.5-5, 18.5, 18.5+5) ([19, 20][18, 21]) (19.5-5, 19.5, 19.5+5) ([20, 21][19, 22]) (20.5-5, 20.5, 20.5+5)
5	([64, 65][63, 66]) (64.5-5, 64.5, 64.5+5) ([65, 66][64, 67]) (65.5-5, 65.5, 65.5+5) ([66, 67][65, 68]) (66.5-5, 66.5, 66.5+5)	([20, 21][19, 22]) (20.5-5, 20.5, 20.5+5) ([21, 22][20, 23]) (21.5-5, 21.5, 21.5+5) ([22, 23][21, 24]) (22.5-5, 22.5, 22.5+5)	([33, 34][32, 35]) (33.5-5, 33.5, 33.5+5) ([34, 35][33, 36]) (34.5-5, 34.5, 34.5+5) ([35, 36][34, 37]) (35.5-5, 35.5, 35.5+5)	([28, 29][27, 30]) (28.5-5, 28.5, 28.5+5) ([29, 30][28, 31]) (29.5-5, 29.5, 29.5+5) ([30, 31][29, 32]) (30.5-5, 30.5, 30.5+5)	∞

Table 4.9: Input Data: Fuzzy-rough safety in CSTSP (Model 4.1D)

i/j	1	2	3	4	5
1	∞	([.1, .2][.01, .3]) (.15-.1, .15, .15+.1) ([.2, .3][.1, .4]) (.25-.1, .25, .25+.1) ([.3, .4][.2, .5]) (.35-.1, .35, .35+.1)	([.3, .4][.2, .5]) (.35-.1, .35, .35+.1) ([.4, .5][.3, .6]) (.45-.1, .45, .45+.1) ([.5, .6][.4, .7]) (.55-.1, .55, .55+.1)	([.2, .3][.1, .4]) (.25-.1, .25, .25+.1) ([.3, .4][.2, .5]) (.35-.1, .35, .35+.1) ([.4, .5][.3, .6]) (.45-.1, .45, .45+.1)	([.5, .6][.4, .7]) (.55-.1, .55, .55+.1) ([.6, .7][.5, .8]) (.65-.1, .65, .65+.1) ([.7, .8][.6, .9]) (.75-.1, .75, .75+.1)
2	([.5, .6][.4, .7]) (.55-.1, .55, .55+.1) ([.6, .7][.5, .8]) (.65-.1, .65, .65+.1) ([.7, .8][.6, .9]) (.75-.1, .75, .75+.1)	∞	([.1, .2][.01, .3]) (.15-.1, .15, .15+.1) ([.2, .3][.1, .4]) (.25-.1, .25, .25+.1) ([.3, .4][.2, .5]) (.35-.1, .35, .35+.1)	([.3, .4][.2, .5]) (.35-.1, .35, .35+.1) ([.4, .5][.3, .6]) (.45-.1, .45, .45+.1) ([.5, .6][.4, .7]) (.55-.1, .55, .55+.1)	([.2, .3][.1, .4]) (.25-.1, .25, .25+.1) ([.3, .4][.2, .5]) (.35-.1, .35, .35+.1) ([.4, .5][.3, .6]) (.45-.1, .45, .45+.1)
3	([.3, .4][.2, .5]) (.35-.1, .35, .35+.1) ([.4, .5][.3, .6]) (.45-.1, .45, .45+.1) ([.5, .6][.4, .7]) (.55-.1, .55, .55+.1)	([.2, .3][.1, .4]) (.25-.1, .25, .25+.1) ([.3, .4][.2, .5]) (.35-.1, .35, .35+.1) ([.4, .5][.3, .6]) (.45-.1, .45, .45+.1)	∞	([.1, .2][.01, .3]) (.15-.1, .15, .15+.1) ([.2, .3][.1, .4]) (.25-.1, .25, .25+.1) ([.3, .4][.2, .5]) (.35-.1, .35, .35+.1)	([.2, .3][.1, .4]) (.25-.1, .25, .25+.1) ([.3, .4][.2, .5]) (.35-.1, .35, .35+.1) ([.4, .5][.3, .6]) (.45-.1, .45, .45+.1)
4	([.4, .5][.3, .6]) (.45-.1, .45, .45+.1) ([.5, .6][.4, .7]) (.55-.1, .55, .55+.1) ([.6, .7][.5, .8]) (.65-.1, .65, .65+.1)	([.2, .3][.1, .4]) (.25-.1, .25, .25+.1) ([.3, .4][.2, .5]) (.35-.1, .35, .35+.1) ([.4, .5][.3, .6]) (.45-.1, .45, .45+.1)	([.5, .6][.4, .7]) (.55-.1, .55, .55+.1) ([.6, .7][.5, .8]) (.65-.1, .65, .65+.1) ([.7, .8][.6, .9]) (.75-.1, .75, .75+.1)	∞	([.2, .3][.1, .4]) (.25-.1, .25, .25+.1) ([.3, .4][.2, .5]) (.35-.1, .35, .35+.1) ([.4, .5][.3, .6]) (.45-.1, .45, .45+.1)
5	([.7, .8][.6, .9]) (.75-.1, .75, .75+.1) ([.6, .7][.5, .8]) (.65-.1, .65, .65+.1) ([.5, .6][.4, .7]) (.55-.1, .55, .55+.1)	([.4, .5][.3, .6]) (.45-.1, .45, .45+.1) ([.3, .4][.2, .5]) (.35-.1, .35, .35+.1) ([.2, .3][.1, .4]) (.25-.1, .25, .25+.1)	([.6, .7][.5, .8]) (.65-.1, .65, .65+.1) ([.5, .6][.4, .7]) (.55-.1, .55, .55+.1) ([.4, .5][.3, .6]) (.45-.1, .45, .45+.1)	([.2, .3][.1, .4]) (.25-.1, .25, .25+.1) ([.3, .4][.2, .5]) (.35-.1, .35, .35+.1) ([.4, .5][.3, .6]) (.45-.1, .45, .45+.1)	∞

Table 4.10: Results of different models by IGA

Model	Path	Costs	S_{min}
(Model 4.1A) Crisp CSTSP	(1,1)(4,2)(3,2)(5,3)(2,3)	100	2
(Model 4.1B) Fuzzy CSTSP	(1,2)(4,1)(3,1)(5,1)(2,3)	(110, 114, 118)	(1.9, 2.2, 2.5)
(Model 4.1C) Rough CSTSP	(1,1)(4,1)(3,2)(5,1)(2,2)	([87, 92][82, 97])	([2.1, 2.6][1.6, 3.1])
(Model 4.1D) Fuzzy-Rough CSTSP	(1,3)(4,2)(3,3)(5,3)(2,1)	([90, 95][85, 100]) (66.5, 91.5,116.5)	[1.8, 2.3][1.3, 2.8] (1.55, 2.05, 2.45)

Table 4.11: Statistical Test for IGA

Instances	Best	Worst	Average	SD	Error
bays29	2020	2047	2028.4	1.32	0.72
bayg29	1610	1629	1617.25	0.97	0.02
fr26	937	954	939.75	0.76	1.64
dantzig42	699	721	707.25	0.81	1.02

4.2.5 Discussion

The developed algorithm IGA is compared with simple GA for standard TSP problem and the results shown in Table 4.1. In every test problem, the proposed algorithm gives better result with respect to number of iterations.

In Table 4.2, crisp costs for the classical CSTSP with three conveyances are given. Here we consider a 5×5 crisp matrix for the CSTSP. In Table 4.3 we have given the individual safety values of the corresponding conveyances. In Table 4.4, the fuzzy cost values of FCSTSP are given, in Table 4.5 the fuzzy costs of the safety values are presented. In this fuzzy environment, FCSTSP is solved by IGA and the results obtained via different conveyances which are shown in Table 4.10.

Similarly we construct the rough costs and safety values for RCSTSPs which are shown in Table 4.6 and 4.7 respectively. For the fuzzy-rough environments, we present the costs and safety values as fuzzy-rough data in the Tables 4.8 and 4.9 for FRCSTSP respectively. The final Table 4.10 gives the results of the above given matrices with imprecise costs and safety values for the imprecise TSPs solved by the proposed IGA.

4.3 Model-4.2: An Adaptive Genetic Algorithm for CSTSP under Uncertain Environments [2]

In this model, an Adaptive Genetic Algorithm (AGA) is developed to solve the constrained solid traveling salesman problems (CSTSPs) in crisp, fuzzy and rough environments. In the developed AGA, we model it with probabilistic selection technique and proposed adaptive crossover with random mutation. Here CSTSPs with costs and risk/discomforts values are in the form of crisp,fuzzy and rough in nature. Also CSTSPs are illustrated numerically by some empirical data using this algorithm. In each environment, some sensitivity studies due to different risks/discomforts values and other system parameters are presented.

4.3.1 Proposed AGA

The proposed AGA and its procedures are presented below:

(i) Representation:
Here a complete tour of N cities represents a solution. So an N dimensional integer vector $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$ and $Y_i = (v_{i1}, v_{i2}, ..., v_{iP})$ are used to represent a solution, where $x_{i1}, x_{i2}, ..., x_{iN}$ represent N consecutive cities in a tour and $Y_i = (v_{i1}, v_{i2}, ..., v_{iP})$ represents the available conveyances. Populations of such solutions $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$, and $Y_i = (v_{i1}, v_{i2}, ..., v_{iP})$ i = 1, 2, ..., N, are randomly generated by random number generator.

(ii) Probabilistic Selection Technique:
It is described in section 4.2.1(c).

(iii) Adaptive Crossover:
At first we select two individuals (parents) from the matting pool randomly (say P_{r1} and P_{r2}). Let these are P_{r1}: $a_1, a_2,..., a_N$, $(v_1, v_2,..., v_p)$ and P_{r2}: $s_1, s_2,..., s_N$, $(v_1, v_2,..., v_p)$. Here $(a_1, a_2,..., a_N)$ and $(s_1, s_2,..., s_N)$ are nodes within (1, 2, 3,..., N), these are numbers of cities. Then we choose a city randomly from 1 to N, say $a_i = s_k$(i=1, 2, ..., N), k=(1, 2,..., N). We modify the first parents by placing a_i or s_k in the first place of P_{r1} and P_{r2}. Now the modified parents are given by P_{r1}: $a_i, a_1, a_2,.., a_{i-1}, a_{i+1},....a_N$, $(v_1, v_2,..., v_p)$, P_{r2}: $s_k, s_1, s_2,., s_{k-1}, s_{k+1},....., s_N$, $(v_1, v_2,..., v_p)$.To get the first child (Ch_1), placing a_i in the first place of Ch_1, we compare the travelling costs between the nodes respectively, $A(a_i, a_1)$ and $A(a_i,$

[2]This portion is published in **Informatics, Electronics & Vision (ICIEV), Japan, 2015 International Conference on** , IEEE, 1-6,Print ISBN:9781467369015, DOI:10.1109ICIEV.2015.7334044, with title *Constrained solid travelling salesman problem using Adaptive Genetic Algorithm in uncertain environment.*

s_1), the cost between the two node a_i to a_1, and a_i, s_1. Minimum cost path be selected for Ch_1. The procedure is as follows:

if ($A(a_i, a_1) < A(a_i, s_1)$)
  concatenate a_1 in Ch_1.
else
  concatenate s_1 in Ch_1.

Ch_1: a_i, s_1(say).

Repeating this process, we get the first child Ch_1: a_i, s_1, a_1,....., a_N (say) (v_1, v_2,..., v_p).

Now for the second child we modify the first parents by placing a_i or s_k at the end of P_{r1} and P_{r2}.The modified parents are given by P_{r1}: a_1, a_2,.., a_{i-1}, a_{i+1},....a_N, a_i, (v_1, v_2,..., v_p), and P_{r2}: s_1, s_2,., s_{p-1}, s_{p+1},....., s_N, s_k, (v_1, v_2,..., v_p). Following the previous procedure, we find the second child as Ch_2: a_i, s_N, a_N,....., a_1 (say) (v_1, v_2,..., v_p). Since the above crossover is done only the by exchange of the nodes(cities) but vehicle/conveyance are not changed, so here we modify them with just only 20% of the number of vehicles. So finally the offspring's are Ch_1: a_i, s_1, a_1,....., a_N (say) (v_5, v_4,..., v_p)(say) and Ch_2: a_i, s_N, a_N,....., a_1 (say) (v_3, v_1,..., v_p)(say). Hence by the above mechanism, using two modified parents(P_{r1}) and (P_{r2}), the two children (offspring) Ch_1 and Ch_2 are found. In every step of crossover, generate the children's.

(iv) Random Mutation:

It is disused in section 2.1.4(ii)(c).

Thus the above proposed algorithm is as follows:

Algorithm of AGA:

1. Begin.
2. Randomly generate initial population $p(t)$.
3. Evaluate initial population $p(t)$.
4. Determine maximum generation number s_0, population size(pop size), probability of crossover(p_c) and probability of mutation (p_m).
5. While t<= s_0 do
6. t=t+1.
7. Selection Operation.
   (a) Determine the Boltzmann Probability(p_B).
   (b) Select the matting pool based on p_B.
8. Crossover Operation
   (a) Select the parents using p_c.

(b) For each pair of parents do
(c) Modify the parents.
(d) Generate off springs from modified parents using the operations presented in section 4.3.1(ii).
(e) End do.

9. Mutation Operation.
(a) Select the off springs for mutation based on P_m.
(b) Randomly choose any two node.
(c) Exchange the place of these nodes.
(d) Store the new off springs into offspring set.

10. Store the local optimum and near optimum solutions.

11. End while.

12. Store the global optimum and near optimum results.

13. End Algorithm.

4.3.2 Mathematical Formulation and Its crisp equivalence

Model 4.2A: Solid TSP with Risk Constraint in Crisp Environment

In a Solid TSP, a salesman has to travel N cities by choosing any one of the P types of conveyances available at the cities. Risks/discomforts factors in travelling from one city to another using different vehicles are different. The salesman should choice such a path and conveyances such that a maximum risk/discomfort levels not exceed and the total travel cost is minimum for the entire tour. Let $c(i, j, k)$ be the cost and $r(i, j, k)$ be the risk/discomfort value in travelling from i-th city to j-th using k-th type conveyances. Then the salesman has to determine a complete tour (x_1, x_2, ...,x_N, x_1) and corresponding conveyance types (v_1, v_2, ..., v_P) to be used for the tour, where $x_i \in \{1, 2, ..N\}$ for $i = 1, 2, ..., N$, $v_i \in \{1, 2, ..P\}$ for $i = 1, 2, ..., N$ and all x_i are distinct. Then the problem can be mathematically formulated as:

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one available corresponding conveyance in each step from the vehicle types $(v_1, v_2, ..., v_P)$ so as

$$\left.\begin{array}{r}\text{to minimize } \; Z = \sum_{i=1}^{N-1} c(x_i, x_{i+1}, v_i) + c(x_N, x_1, v_l), \\ \text{to minimize } \; \text{subject to } \; \sum_{i=1}^{N-1} r(x_i, x_{i+1}, v_i) + r(x_N, x_1, v_l) \leq r_{max}, \\ where \; x_i \neq x_j, i, j = 1, 2...N, \;\; v_i, v_l \in \{1, 2.., or P\}\end{array}\right\} \quad (4.9)$$

where r_{max} is the allowable maximum risk/discomfort value that should be maintained by the salesman in the entire tour.

Model 4.2B: CSTSP in Fuzzy Environment (FCSTSP)

In the above section of the Equ. 4.9, if costs and risk/discomfort values are fuzzy numbers, i.e, $\tilde{c}(i, j, k)$ and $\tilde{r}(i, j, k)$ respectively, risk/discomfort limit r_{max} is also fuzzy number $\tilde{r}_{max}$.

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one available corresponding conveyances in each step from the vehicle types $(v_1, v_2, ..., v_P)$
Using Possibility Measure,

$$\left.\begin{array}{r}\text{to minimize } F, \\ \text{subject to } \; Pos(\sum_{i=1}^{N-1} \tilde{c}(x_i, x_{i+1}, v_i) + \tilde{c}(x_N, x_1, v_l) < F) \geq \alpha_3 \\ Pos(\sum_{i=1}^{N-1} \tilde{r}(x_i, x_{i+1}, v_i) + \tilde{r}(x_N, x_1, v_l) \leq \tilde{r}_{max}) \geq \beta_3 \\ where \; x_i \neq x_j, i, j = 1, 2...N, \;\; v_i, v_l \in \{1, 2.., or P\}\end{array}\right\} \quad (4.10)$$

Similarly, using necessity measure, we have

$$\left.\begin{array}{r}\text{minimize } \; F \\ \text{subject to } \; Nes(\sum_{i=1}^{N-1} \tilde{c}(x_i, x_{i+1}, v_i) + \tilde{c}(x_N, x_1, v_l) < F) \geq \alpha_4 \\ Nes(\sum_{i=1}^{N-1} \tilde{r}(x_i, x_{i+1}, v_i) + \tilde{r}(x_N, x_1, v_l) \leq \tilde{r}_{max}) \geq \beta_4 \\ where \; x_i \neq x_j, i, j = 1, 2...N, \;\; v_i, v_l \in \{1, 2.., or P\}\end{array}\right\} \quad (4.11)$$

where α_3, β_3, γ_3 and α_4, β_4, γ_4 are predefined levels of possibility and necessity respectively which are entirely determined by the salesman. If we consider the

TFN then,

$$\begin{aligned}\tilde{c}(i,j,k) &= (c(i,j,k)_1, c(i,j,k)_2, c(i,j,k)_3),\\ \tilde{t}(i,j,k) &= (t(i,j,k)_1, t(i,j,k)_2, t(i,j,k)_3),\\ \tilde{r}(i,j,k) &= (r(i,j,k)_1, r(i,j,k)_2, r(i,j,k)_3),\\ \tilde{r}_{max} &= (r_1, r_2, r_3).\end{aligned}$$

Then the above problems can be reduced to crisp ones accordingly as given in section 3.4.4:
Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one available corresponding conveyance in each step from the vehicle types $(v_1, v_2, ..., v_P)$ so as
Using possibility measure,

$$\left.\begin{array}{c}\text{to minimize } \; F\\ \text{subject to } \; \dfrac{F-F_1}{F_2-F_1} \geq \alpha_3\\ \dfrac{r_3-R_1}{r_3-r_2+R_2-R_1} \geq \beta_3\end{array}\right\} \tag{4.12}$$

Using necessity measure, It is represented as

$$\left.\begin{array}{c}\text{minimize } \; F\\ \text{subject to } \; \dfrac{F_3-F}{F_3-F_2} \leq 1-\alpha_4\\ \dfrac{R_3-r_1}{r_2-r_1+R_3-R_2} \leq 1-\beta_4\end{array}\right\} \tag{4.13}$$

$$\text{where } \; F_j = \sum_{i=1}^{N-1} c(x_i, x_{i+1}, v_i)_j + c(x_N, x_1, v_l)_j, \;\; j = 1,2,3.$$

$$\text{and } \; R_j = \sum_{i=1}^{N-1} r(x_i, x_{i+1}, v_i)_j + r(x_N, x_1, v_l)_j, \;\; j = 1,2,3.$$

where $x_i \neq x_j, i, j = 1, 2...N.\;\; v_i, v_l \in \{1, 2.., or P\}$
The objective function in Equ. 4.12 and Equ. 4.13 are respectively changed to

$$\left.\begin{array}{c}\text{minimize } \; F_1 + \alpha_3(F_2 - F_1)\\ \text{subject to } \; \dfrac{r_3-R_1}{r_3-r_2+R_2-R_1} \geq \beta_3\end{array}\right\} \tag{4.14}$$

and

$$\left.\begin{array}{rl}\text{minimize} & F_3-(1-\alpha_4)(F_3-F_2)\\ \text{subject to} & \dfrac{R_3-r_1}{r_2-r_1+R_3-R_2}\leq 1-\beta_4\end{array}\right\}\tag{4.15}$$

Model 4.2C: CSTSP in Rough Environment (RCSTSP)

In the section 4.9, if costs and risk/discomfort values are rough numbers, i.e, $\hat{c}(i,j,k)$ and $\hat{r}(i,j,k)$ respectively, risk/discomfort limit r_{max} is also fuzzy number $\hat{r}_{max}$, then the above problem in Equ.4.9 reduces to:

$$\left.\begin{array}{c}\text{Minimize}\ \ \hat{Z}=\hat{C}(x,v)\\ \text{subject to}\ \hat{R}(x,v)\leq\tilde{R}_{max}\end{array}\right\}\tag{4.16}$$

where $\hat{C}=([a,b],[c,d]),\hat{R}=([R_1],[R_2],[R_3,R_4])$,
$\hat{R}_{max}=([R_{max1},R_{max2}],[R_{max3},R_{max3},R_{max4}])$ are rough variables.
The above rough model Equ. 4.16 is reformed as given below

$$\left.\begin{array}{c}\text{Minimize}\ Z_1\\ \text{subject to}\ Tr\{\hat{C}(x,v)\leq Z_1\}\geq\alpha\\ Tr\{\hat{R}(x,v)\leq\tilde{R}_{max}\}\geq\beta\end{array}\right\}\tag{4.17}$$

Here α, β are confidence values of the trust levels and Z_1 is crisp values. Tr represents the trust measure.

The above rough model can also be converted to a deterministic one with the help of lemma given in section 3.5.
$E(\hat{C})=\frac{1}{4}(a+b+c+d)$ and $E(\hat{R})=\frac{1}{4}(R_1+R_2+R_3+R_4)$.
Then converted crisp model is

$$\left.\begin{array}{rl}\text{minimize} & \frac{a+b+c+d}{4}\\ \text{subject to} & \dfrac{R_1+R_2+R_3+R_4}{4}\leq\dfrac{R_{max1}+R_{max2}+R_{max3}+R_{max4}}{4}\end{array}\right\}\tag{4.18}$$

Table 4.12: Test TSPLIB Problems by AGA

Instances	Problem Size	Optimum Result	AGA Cost	AGA Iteration	GA Cost	GA Iteration
fri26	26	937	937	**78**	937	269
bays29	29	2020	2020	**61**	2020	451
bayg29	29	1610	1610	**66**	1610	378
dantzig42	42	699	699	**152**	699	612
eil51	51	426	426	**98**	426	341
berlin52	52	7542	7542	1**45**	7542	526
st70	70	675	675	**165**	675	813
eil76	76	538	538	**124**	538	457
pr76	76	108159	108159	**165**	108159	410
rat99	99	1211	1211	**147**	1211	328
kroa100	100	21282	21282	**276**	21282	285

4.3.3 Numerical Experiments

Testing for AGA:

To judge the effectiveness and feasibility of developed algorithm AGA, we have applied it on the standard TSP problems from TSPLIB [162]. Table 4.12 gives the results along with the simple GA in terms of total cost and GA iterations.

Input data for Model 4.2A:

Here, for a CSTSP, where we consider three types of conveyances. The cost matrix for the CSTSP and corresponding risk/discomfort matrix are presented in Table 4.13.

For AGA, we have taken maximum generation=2000, p_c=0.34, p_m=0.43 .

Input Data for Model 4.2B:

Here we take the cost and risk/discomfort values as fuzzy for the FCSTSP in Equ. 4.12 and Equ. 4.13. Also we consider three types of conveyances. The fuzzy cost matrix for the FCSTSP and corresponding fuzzy risk/discomfort matrix are given in Table 4.15.

Input Data for Rough costs (Model 4.2C)

Here we take the cost and risk/discomfort values as rough values for the RCSTSP. Also we consider three types of conveyances. Assume that $\hat{C}$ is a rough number which are travelling costs and $\hat{R}$ is the risk/discomfort values with rough maximum level $\hat{R}_{max}$. The rough cost matrix for the RCSTSP and corresponding rough risk/discomfort matrix are presented in Table 4.17.

Table 4.13: Input Data: Crisp CSTSP (Model 4.2A)

	Crisp Cost Matrix(10 *10) With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	35,36,27	18,39,30	20,33,34	30,21,62	6,23,8	15,36,47	27,38,19	40,31,42	20,31,42
2	35,26,17	∞	40,21,32	18,29,10	35,26,37	40,31,22	40,31,59	33,42,59	18,37,20	24,16,18
3	38,30,29	17,58,34	∞	12,25,14	42,25,46	35,36,34	19,11,8	32,33,25	30,19,41	30,22,33
4	28,20,11	10,22,14	17,8,29	∞	30,19,24	25,16,27	21,31,33	35,36,17	12,23,34	27,48,39
5	17,15,9	42,23,34	35,36,37	20,31,43	∞	30,21,42	45,16,27	30,31,13	19,10,8	28,26,7
6	15,6,7	30,21,29	5,26,28	8,9,12	28,29,40	∞	33,42,24	40,31,22	32,23,35	30,41,32
7	38,39,30	25,54,26	30,38,26	22,43,24	37,58,39	40,21,45	∞	10,41,13	32,33,35	20,15,26
8	40,41,23	25,6,17	32,53,45	40,21,42	35,36,47	25,16,5	40,22,43	∞	22,53,24	37,37,39
9	40,11,33	40,39,36	3,36,37	25,34,29	20,32,21	22,33,25	7,38,39	32,33,14	∞	28,19,26
10	18,27,29	30,21,32	28,19,30	20,31,22	11,33,22	32,12,34	37,28,39	40,41,33	30,51,33	∞
	Crisp Risks/Discomforts Matrix(10 *10) With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	.69,.68,.75	.84,.63,.7	.82,.7,.71	.72,.8,.42	.96,.79,.93	.87,.66,.55	.74,.42,.81	.41,.7,.59	.81,.7,.59
2	.67,.76,.84	∞	.61,.8,.7	.83,.73,.92	.67,.76,.65	.41,.71,.79	.41,.71,.43	.69,.6,.42	.83,.64,.81	.77,.85,.3
3	.63,.71,.73	.83,.44,.67	∞	.89,.76,.86	.59,.76,.55	.66,.65,.67	.83,.91,.94	.69,.68,.76	.71,.82,.6	.71,.79,.68
4	.73,.81,.9	.9,.78,.86	.84,.93,.72	∞	.71,.82,.77	.77,.86,.75	.81,.71,.69	.66,.65,.84	.89,.79,.77	.74,.53,.43
5	.84,.86,.92	.59,.78,.67	.66,.65,.64	.82,.71,.59	∞	.71,.81,.59	.57,.85,.74	.71,.7,.88	.82,.91,.93	.74,.75,.93
6	.85,.84,.93	.7,.8,.71	.95,.74,.72	.92,.91,.89	.73,.72,.61	∞	.69,.59,.77	.61,.71,.79	.69,.78,.66	.71,.6,.69
7	.63,.62,.71	.77,.47,.76	.71,.63,.76	.79,.59,.77	.66,.43,.62	.6,.79,.55	∞	.9,.6,.87	.69,.68,.66	.81,.87,.76
8	.61,.6,.78	.76,.95,.84	.69,.47,.56	.61,.81,.6	.67,.66,.55	.6,.85,.95	.61,.8,.59	∞	.79,.48,.77	.64,.64,.62
9	.61,.91,.71	.61,.62,.65	.97,.65,.64	.76,.77,.72	.81,.69,.73	.79,.68,.76	.94,.66,.63	.69,.68,.87	∞	.73,.82,.75
10	.83,.74,.72	.71,.8,.69	.73,.83,.72	.8,.69,.78	.89,.67,.78	.7,.9,.71	.64,.74,.22	.61,.59,.68	.71,.5,.67	∞

Table 4.14: Results of CSTSP in Crisp (Model 4.2A)

Algorithm	Path(Vehicle)	Cost	Risk achieved	R_{max}
AGA	1(1)-10(1)-5(2)-4(1)-2(1)-9(1)-3(2)-7(1)-8(3)-6(2)	107.00	8.71	8.75
AGA	9(1)-7(1)-8(1)-6(2)-1(1)-3(1)-4(1)-2(1)-10(1)-5(3)	131.00	8.50	8.75
AGA	2(2)-10(1)-9(1)-6(2)-4(2)-3(3)-7(2)-5(2)-1(3)	141.00	8.50	8.75
AGA	7(1)-8(1)-6(2)-1(1)-10(3)-5(2)-4(3)-2(3)-9(1)-3(2)	144.00	8.19	8.75
GA	2(2)-9(1)-3(3)-7(3)-8(1)-6(2)-1(3)-10(2)-5(2)-4(1)	190.00	8.73	8.75
AGA	5(2)-4(3)-2(1)-9(3)-8(1)-6(3)-1(1)-3(2)-7(1)-10(1)	151.00	8.25	8.25
AGA	2(2)-10(3)-9(1)-3(1)-7(1)-8(1)-6(2)-1(1)-5(2)-4(1)	165.00	7.99	8.00
AGA	7(1)-592)-4(1)-2(3)-9(2)-3(3)-8(1)-6(2)-1(2)-10(1)	240.00	7.25	7.25

Table 4.15: Input Data: FCSTSP (Model 4.2B)

	Fuzzy Cost Matrix(10 × 10) With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	32,35,36 36,37,39 26,28,29	17,19,20 38,39,42 26,30,31	17,21,22 31,33,34 33,35,36	29,30,31 20,21,23 60,62,63	5,7,10 22,23,25 6,8,9	15,16,18 35,33,37 46,47,48	25,28,29 37,39,43 16,19,20	39,41,42 26,31,33 41,42,43	20,22,23 30,31,34 42,43,45
2	34,35,38 22,26,27 14,17,19	∞	40,41,44 18,21,22 27,32,33	16,18,19 28,29,32 6,10,12	32,35,37 25,26,27 34,37,38	39,40,41 30,31,32 21,23,26	39,40,42 29,30,32 57,59,60	30,33,34 41,42,45 58,59,62	17,19,22 36,37,38 17,20,21	23,24,26 13,16,17 17,18,20
3	36,38,39 29,30,32 28,29,32	16,17,20 54,58,60 31,34,35	∞	10,12,13 24,25,26 12,14,17	40,42,45 23,25,26 45,46,48	33,35,36 34,36,39 33,34,35	17,19,20 11,11,12 5,8,10	30,32,33 30,33,34 24,25,27	28,30,31 18,19,21 40,41,44	29,30,31 19,22,23 32,33,35
4	27,28,30 18,20,21 9,10,12	9,10,11 19,22,23 12,14,15	16,18,20 7,9,10 27,29,30	∞	29,30,33 17,19,20 23,24,25	23,25,26 15,16,18 25,27,28	19,21,22 30,31,32 30,33,34	33,35,36 32,36,38 16,17,18	10,12,13 20,23,24 32,34,35	24,27,29 47,48,49 37,39,40
5	16,18,19 14,15,18 6,8,9	41,42,44 21,23,24 32,34,37	34,35,37 35,36,37 33,38,39	17,20,21 12,13,14 40,43,44	∞	29,30,31 20,21,23 40,41,42	42,45,46 14,16,18 25,27,27	27,30,31 30,31,32 12,13,16	18,19,22 8,10,11 7,8,9	26,28,29 25,26,27 25,27,28

6	13,15,16 5,6,8 5,7,8	26,29,30 20,21,23 27,29,30	4,4,6 25,26,27 27,28,30	6,8,9 7,9,11 10,12,13	26,28,29 26,29,30 38,39,41	∞	31,33,34 40,43,44 23,24,26	39,40,42 30,31,31 20,22,23	30,32,33 22,23,24 35,35,36	28,30,31 40,41,42 30,32,34
7	36,37,39 37,39,40 28,30,32	23,25,26 53,53,55 25,26,27	27,30,32 37,38,39 24,26,27	21,22,24 40,43,44 23,24,25	35,37,38 56,58,60 37,39,40	38,40,41 20,21,22 43,45,46	∞	7,10,11 40,43,44 11,13,14	31,33,34 33,34,35 34,36,37	19,20,22 13,15,16 25,26,28
8	39,41,42 41,42,43 20,23,24	24,26,28 5,6,7 16,17,18	30,32,33 52,53,54 43,45,46	38,40,42 19,21,22 40,42,43	34,35,37 34,36,37 46,47,48	23,25,26 15,16,18 4,5,6	39,40,42 19,21,22 41,43,44	∞	20,22,23 52,53,54 23,24,27	35,37,38 35,36,38 39,40,41
9	38,40,41 10,11,13 31,32,33	39,41,42 38,39,40 34,36,37	4,6,9 34,36,37 36,37,39	23,25,26 33,34,36 28,29,30	20,21,23 31,32,33 20,21,22	22,23,25 31,33,34 23,25,26	5,7,8 36,38,39 38,39,41	30,32,33 32,33,34, 11,13,15	∞	27,28,30 18,19,20 24,26,27
10	15,17,18 25,26,28 25,29,30	28,30,31 20,21,22 31,32,34	26,28,29 18,19,20 28,30,32	18,20,21 29,31,32 21,22,24	9,11,12 32,33,34 20,22,24	30,32,34 10,12,13 33,34,35	35,38,39 26,28,29 38,39,41	40,41,43 41,42,43 30,33,34	29,31,32 51,52,54 30,32,33	∞
	Fuzzy Risks/Discomforts Matrix(10 $\times$10) With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	.7,.65,.63 .69,.67,.66 .75,.72,.7	.85,.81,.78 .59,.57,.56 .65,.63,.6	.8,.78,.77 .7,.67,.65 .69,.71,.7	.75,.72,.71 .8,.81,.83 .37,.32,.29	.9,.87,.85 .78,.74,.72 .89,.84,.81	.85,.83,.8 .64,.61,.59 .51,.5,.47	.7,.67,.61 .6,.58,.54 .79,.75,.73	.54,.5,.47 .65,.6,.58 .54,.5,.48	.79,.75,.7 .61,.58,.54 .48,.42,.41
2	.58,.55,.5 .7,.66,.61 .8,.75,.71	∞	.56,.41,.47 .76,.71,.69 .68,.61,.59	.78,.77,.71 .67,.62,.6 .9,.85,.82	.65,.61,.59 .75,.68,.65 .6,.58,.5	.53,.5,.47 .68,.64,.61 .7,.65,.62	.59,.52,.48 .69,.63,.6 .31,.26,.2	.7,.63,.59 .51,.45,.4 .32,.34,.19	.75,.7,.68 .6,.57,.53 .7,.69,.62	.69,.64,.61 .8,.76,.71 .81,.76,.7
3	.55,.51,.48 .6,.56,.53 .61,.58,.56	.72,.69,.62 .38,.31,.26 .6,.58,.51	∞	.81,.76,.7 .71,.68,.66 .8,.76,.71	.51,.46,.4 .7,.64,.61 .48,.44,.4	.59,.55,.52 .61,.58,.56 .62,.6,.57	.8,.75,.71 .9,.86,.81 .89,.86,.81	.65,.6,.59 .64,.6,.58 .68,.65,.61	.58,.55,.51 .8,.76,.71 .55,.5,.48	.67,.61,.58 .76,.71,.68 .64,.6,.57
4	.69,.64,.62 .78,.75,.71 .85,.83,.8	.86,.81,.79 .76,.71,.69 .81,.78,.74	.79,.75,.72 .9,.85,.82 .7,.64,.6	∞	.65,.63,.6 .76,.72,.7 .78,.71,.69	.69,.65,.62 .78,.75,.71 .68,.67,.65	.78,.74,.71 .68,.65,.61 .6,.54,.5	.6,.56,.52 .59,.58,.56 .79,.76,.72	.85,.82,.8 .78,.74,.71 .71,.69,.64	.68,.63,.59 .5,.45,.41 .6,.54,.5
6	.8,.75,.71 .81,.79,.76 .88,.85,.81	.65,.63,.6 .75,.72,.7 .66,.61,.59	.85,.82,.78 .7,.68,.62 .65,.62,.6	.88,.84,.79 .87,.84,.8 .85,.81,.78	.7,.67,.63 .6,.58,.55 .58,.54,.49	∞	.64,.6,.58 .55,.51,.46 .7,.68,.65	.55,.52,.48 .65,.63,.6 .76,.71,.68	.68,.61,.58 .73,.7,.68 .62,.58,.55	.65,.61,.58 .55,.52,.48 .65,.62,.6
7	.58,.54,.49 .56,.52,.48 .65,.62,.58	.65,.63,.6 .44,.38,.33 .71,.65,.6	.64,.6,.58 .6,.58,.55 .67,.64,.6	.7,.68,.65 .55,.51,.45 .71,.68,.64	.56,.54,.51 .38,.32,.28 .55,.53,.51	.55,.51,.46 .75,.71,.68 .52,.47,.4	∞	.85,.81,.78 .55,.54,.51 .75,.76,.72	.65,.61,.59 .58,.54,.5 .65,.61,.58	.78,.74,.69 .71,.68,.64 .65,.62,.58
8	.56,.52,.49 .54,.52,.51 .5,.43,.4	.7,.68,.65 .9,.88,.84 .8,.81,.78	.64,.6,.58 .41,.38,.37 .51,.45,.4	.56,.52,.5 .76,.74,.7 .56,.52,.49	.62,.58,.53 .62,.57,.55 .52,.48,.45	.55,.52,.48 .8,.77,.7 .88,.83,.8	.55,.54,.51 .78,.72,.7 .54,.53,.5	∞	.78,.76,.73 .43,.4,.36 .73,.7,.68	.58,.56,.51 .6,.54,.5 .58,.54,.49
9	.56,.51,.48 .88,.85,.81 .68,.65,.51	.58,.52,.5 .59,.57,.56 .58,.55,.53	.9,.85,.82 .62,.61,.58 .6,.54,.5	.7,.68,.64 .74,.7,.67 .68,.52,.58	.78,.75,.71 .65,.61,.58 .74,.7,.68	.74,.7,.68 .64,.61,.59 .67,.64,.6	.85,.81,.8 .62,.6,.57 .58,.54,.49	.62,.6,.58 .65,.61,.6 .79,.75,.72	∞	.69,.65,.63 .78,.73,.7 .72,.7,.68
10	.78,.71,.69 .7,.67,.64 .69,.64,.6	.66,.61,.58 .77,.74,.7 .78,.76,.71	.69,.65,.62 .8,.76,.74 .68,.65,.63	.74,.7,.68 .65,.6,.57 .76,.71,.68	.83,.78,.75 .62,.58,.56 .75,.71,.66	.65,.61,.58 .87,.83,.78 .68,.64,.59	.59,.54,.5 .68,.64,.61 .59,.55,.51	.55,.52,.47 .52,.48,.54 .64,.6,.58	.64,.59,.58 .45,.41,.37 .61,.59,.58	∞

Table 4.16: Optimum Results of FCSTSP (Model 4.2B)

α	β	Algorithm	DM	Path(Vehicle)	Obj Value	Fuzzy Cost	Risk Value	R_{max}
0.95	0.8	AGA	ODM	10(1)-5(2)-9(2)-1(1)-6(1)-4(2)-3(3)-7(1)-8(2)-2(3)	99	80,100,115	8.72,8.31,7.93	9.25.,9,8.5
			PDM	10(1)-5(2)-9(2)-1(1)-6(1)-4(2)-3(3)-7(1)-8(2)-2(3)	114.35	80,100,115	8.72,8.31,7.93	9.25.,9,8.5
		AGA	ODM	5(3)-9(1)-3(3)-7(3)-8(2)-2(1)-10(2)-6(1)-4(3)-1(1)	126.15	110,127,142	8.33,8.08,7.73	9.25.,9,8.5
			PDM	7(3)-8(2)-4(3)-6(2)-1(2)-5(1)-9(3)-10(2)-3(2)	139.3	126,140,154	8.21,7.97,7.65	9.25.,9,8.5
		AGA	ODM	8(3)-6(2)-1(1)-4(1)-2(2)-10(1)-5(3)-9(1)-3(3)-7(3)	103.25	84,104,118	8.42,8.09,7.74	8.75,8.5,8.25
			PDM	1(3)-5(3)-9(1)-3(3)-7(3)-8(2)-2(2)-10(2)-6(2)-4(3)	119.5	101,120,135	8.43,8.2,7.84	8.75,8.5,8.25
		GA	ODM	4(3)-1(1)-5(3)-9(1)-3(3)-7(3)-8(3)-2(2)-10(2)-6(3)	126.15	110,127,141	8.31,7.98,7.61	8.75,8.5,8.25
			PDM	5(3)-8(3)-2(1)-4(2)-9(1)-3(3)-7(2)-10(2)-6(2)-1(2)	138.1	121,139,156	8.16,7.93,7.59	8.75,8.5,8.25
.8	.9	AGA	ODM	6(3)-4(3)-1(2)-5(3)-9(1)-3(2)-7(3)-8(3)-2(2)-10(2)	125.25	111,126,141	8.42,8.19,7.83	8.5.8,7.75
			PDM	10(2)-2(1)-4(3)-1(2)-5(3)-9(3)-8(2)-612)-3(3)-7(1)	138.35	126,139,156	8.21,7.96,7.54	8.5.8,7.75

Table 4.17: Input Data: RCSTSP (Model 4.2C)

	Rough Cost Matrix(10 $\times$10) for RCSTSP With Three Conveyances				
i/j	1	2	3	4	5
1	∞	([32,33],[30,36]) ([36,37],[35,39]) ([26,28],[24,29])	([17,19],[15,20]) ([38,39],[36,42]) ([26,30],[25,31])	([17,21],[16,22]) ([31,33],[30,34]) ([(33,35],[31,36])	([29,30],[27,31]) ([(20,21],[17,23]) ([60,62],[58,63])]
2	([34,35],[33,38]) ([22,26],[21,27]) ([14,17],[18,19])	∞	([40,41],[38,44]) ([18,21],[16,22]) ([(27,32],[26,33])	([16,18],[15,19]) ([28,29],[27,32]) ([6,10],[5,12])	([(32,35],[30,37]) ([25,26],[24,27]) ([34,37],[33,38])
3	([36,38],[35,39]) ([29,30],[28,32]) ([28,29],[26,32])	([16,17],[15,20]) ([54,58],[53,60]) ([(31,34],[30,35])	∞	([10,12],[9,13]) ([24,25],[23,26]) ([(12,14],[11,17])	([40,42],[38,45]) ([23,25],[22,26]) ([(45,46],[44,48])
4	([27,28],[25,30]) ([18,20],[17,21]) ([9,10],[8,12])	([9,10],[8,11]) ([19,22],[18,23]) ([12,14],[11,15])	([16,18],[14,20]) ([7,9],[6,10]) ([27,29],[25,30])	∞	([(29,30],[28,33]) ([17,19],[15,20]) ([23,24],[22,25])
5	([16,18],[15,19]) ([14,15],[12,18]) ([6,8],[4,9])	([41,42],[40,44]) ([21,23],[20,24]) ([32,34],[31,37])	([34,35],[32,37]) ([35,36],[33,37]) ([33,38],[31,39])	([17,20],[16,21]) ([12,13],[10,14]) ([40,43],[39,44])	∞
6	[(13,15],[11,16]) ([5,6],[4,8]) ([5,7],[4,8)	([26,29],[25,30]) ([20,21],[19,23]) ([27,29],[26,30])	([7,9],[6,10]) ([25,26],[24,27]) ([27,28],[26,30])	([6,8],[5,9]),1.3]) ([7,9],[6,11]) ([10,12],[9,13])	([26,28],[24,29]) ([26,29],[25,30]) ([38,39],[37,41])
7	([36,37],[35,39]) ([37,39],[36,40]) ([(28,30],[27,32])	([23,25],[22,26]) ([53,54],[51,55]) ([25,26],[24,27])	([27,30],[26,32]) ([37,38],[36,39]) ([24,26],[23,27])	((21,22],[20,24]) ([40,43],[39,44]) ([23,24],[22,25])	([35,37],[34,38]) ([56,58],[55,60]) ([37,39],[36,40])
8	([39,41],[38,42] ([41,42],[40,43]) ([20,23],[19,24])	([24,26],[23,28]) ([5,6],[4,7]) ([16,17],[15,18])	([30,32],[29,33]) ([52,53],[50,54]) ([43,45],[42,46])	([38,40],[37,42]) ([19,21],[18,22]) ([40,42],[38,43])	([34,35],[33,37]) ([34,36],[33,37]) ([46,47],[44,48])
9	([38,40],[37,41]) ([10,11],[9,13]) ([31,32],[30,33])	([39,41],[38,42]) ([38,39],[37,40]) ([34,36],33,37])	([4,6],[3,9]) ([34,36],[33,37]) ([36,37],[35,39])	([23,25],[22,26]) ([33,34],[32,36]) ([28,29],[27,30])	([20,21],[18,23]) ([31,32],[29,33]) ([20,21],[19,22])
10	([15,17],[14,18]) ([25,26],[24,28]) ([25,26],[24,29])	([28,30],[27,31]) ([20,21],[19,22]) ([31,32],[28,34])	([26,28],[25,29]) ([18,19],[17,20]) ([28,30],[27,32])	([18,20],[17,21]) ([29,31],[28,32]) ([21,22],[20,24])	([9,11],[8,12]) ([32,33],[31,34]) ([20,22],[19,24])
	Rough Cost Matrix(10 $\times$10) for RCSTSP With Three Conveyances				
i/j	6	7	8	9	10
1	([5,7],[4,10]) ([22,23],[21,25]) ([6,8],[5,9])	([15,16],[14,18]) ([34,35],[33,37]) ([46,47],[45,48])	([25,28],[23,29]) ([37,39],[36,43]) ([16,17],[15,20])	([39,41],[38,42]) ([26,31],[25,33]) ([41,42],[40,43])	([20,22],[19,23]) ([30,31],[29,34]) ([42,43],[41,45])
2	([39,40],[38,41]) ([30,31],[29,32]) ([21,23],[20,26])	([39,40],[38,42]) ([29,30],[28,32]) ([57,59],[55,60])	([30,33],[28,34]) ([41,42],[40,45]) ([58,59],[57,62])	([17,19],[16,22]) ([36,37],[35,38]) ([17,20],[16,21])	([23,24],[22,26]) ([13,16],[12,17]) ([17,18],[15,20])
3	([33,35],[32,36]) ([34,36],[33,39]) ([33,34],[31,35])	([17,19],[16,20]) ([11,12],[10,13]) ([5,8],[4,10])	([30,32],[28,33]) ([30,33],[29,34]) ([24,25],[23,27])	[(28,30],[27,31]) ([18,19],[17,21]) ([40,41],[39,44])	([29,30],[28,31]) ([19,22],[18,23]) ([32,33],[31,35])
4	([23,25],[22,26]) ([15,16],[14,18]) ([25,27],[24,28])	([19,21],[18,22]) ([30,31],[29,32]) ([30,33],[29,34])	([33,35],[32,36]) ([32,36],[31,38]) ([16,17],[15,18])	([10,12],[9,13]) ([20,23],[19,24]) ([32,34],[31,35])	([24,27],[22,29]) ([47,48],[45,49]) ([37,39],[36,40])
5	([29,30],[28,31]) ([20,21],[19,23]) ([40,41],[39,42])	([42,45],[41,46]) ([14,16],[12,18]) ([25,27],[24,27])	([27,30],[26,31]) ([30,31],[29,32]) ([12,13],[14,16])	([18,19],[17,22]) ([8,10],[7,11]) ([7,8],[6,9])	([26,28],[25,29]) ([25,26],[24,27]) ([25,27],[24,28])
6	∞	([31,33],[30,34]) ([40,43],[38,44]) ([23,24],[22,26])	([39,40],[38,42]) ([30,31],[27,31]) ([20,22],[18,23])	([30,32],[29,33]) ([22,23],[21,24]) ([35,36],[34,37])	[28,30],[27,31]) ([40,41],[38,42]) ([30,32],[29,34])
7	([38,40],[37,41]) ([20,21],[19,22]) ([43,45],[42,46])	∞	([7,10],[6,11]) ([40,43],[39,44]) ([11,13],[10,14])	([31,33],[29,34]) ([33,34],[31,35]) ([34,36],[33,37])	([19,20],[18,22]) ([13,15],[12,16]) ([25,26],[24,28])
8	([23,25],[22,26]) ([15,16],[13,18]) ([4,5],[3,6])	([39,40],[38,42]) ([19,21],[18,22]) ([41,43],[40,45])	∞	([20,22],[19,23]) ([52,53],[50,54]) ([23,24],[22,27])	([35,37],[34,38]) ([35,36],[34,38]) ([39,40],[38,41])
9	([22,23],[20,25]) ([31,33],[28,34]) ([23,25],[22,26])	([5,7],[4,8]) ([36,38],[33,39]) ([38,39],[37,41])	([30,32],[29,33]) ([32,33],[31,34]) ([11,13],[10,15])	∞	[(27,28],[26,30]) ([18,19],[17,20]) ([24,26],[22,27])

10	([30,32],[29,34]) ([10,12],[9,13]) ([33,34],[32,35])	([35,38],[33,39]) ([26,28],[27,29]) ([38,39],[37,41])	([40,41],[39,43]) ([41,42],[40,43]) ([30,33],[27,34])	([29,31],[28,32]) ([51,52],[50,54]) ([30,32],[26,33])	∞
	Rough Risks/Discomforts Matrix(10 × 10) for RCSTSP With Three Conveyances				
i/j	1	2	3	4	5
1	∞	([.69,.7],[.65,.71]) ([.36,.37],[.35,.39]) ([.26,.28],[.25,.29])	([.7,.71],[.68,.73]) ([.38,.39],[.37,.42]) ([.26,.30],[.25,.31])	([.73,.74],[.72,.76]) ([.31,.33],[.3,.34]) ([.33,.35],[.31,.36)	([.31,.32],[.29,.33]) ([.20,.21],[.19,.23]) ([.60,.62],[.59,.63])
2	([.34,.35],[.32,.38]) ([.22,.26],[.21,.27]) ([.14,.17],[.13,.19])	∞	([.40,.41],[.38,.44]) [[(.18,.21],[.17,.22]) ([.27,.32],[.26,.33])	([.16,.18],[.15,.19]) ([.28,.29],[.27,.32]) ([.06,.10],[.0.05,.12])	([.32,.35],[.31,.37]) ([.25,.26],[.24,.27]) ([.34,.37],[0.33,.38])
3	([.36,.38],[0.35,.39]) ([.29,.3],[.28,.32]) ([.28,.29],[.27,.32])	([.16,.17],[.15,.2]) ([.54,.58],[.55,.61]) ([.31,.34],[.3,.35])	∞	([.10,.12],[0.9,.13]) ([.24,.25],[.23,.26]) ([.12,.14],[.1,.17])	[(.40,.42],[.39,.45]) ([.23,.25],[.22,.26]) ([.45,.46],[.44,.48])
4	([.27,.28],[.26,.30]) [(.18,.20],[.17,.21]) ([.09,.1],[.08,.12])	([.09,.1],[.08,.11]) ([.19,.22],[.18,.23]) ([.12,.14],[.11,.15])	([.16,.18],[.15,.2]) ([.07,.09],[.06,.10]) ([.27,.29],[.26,.3])	∞	([.29,.3],[.8,.33])] ([.17,.19],[.16,.2]) ([.23,.24],[.22,.25])
5	([.16,.18],[.15,.19]) ([14.,.15],[.13,.18]) ([.06,.08],[.05,.09])	([.41,.42],[.4,.44]) ([.21,.23],[.2,.24]) ([.32,.34],[31,.37])	([.34,.35],[.33,.37]) ([.35,.36],[.33,.37]) ([.33,.38],[.32,.39])	([.17,.2],[.16,.21]) ([.12,.13],[.11,.14]) ([.4,.43],[.39,.44])	∞
6	([.13,.15],[.12,.16]) ([.05,.06],[.04,.08]) ([.05,.07],[0.4,.08])	([.26,.29],[.25,.30]) ([.2,.21],[.19,.23]) ([.27,.29],[.26,.30])	([.4,.41],[.39,.44]) ([.25,.26],[.24,.27]) ([.27,.28],[.25,.3])	([.06,.08],[.05,.09]) ([.07,.09],[.06,.11]) ([.1,.12],[.09,.13])	([.26,.28],[.25,.29]) ([.26,.29],[.25,.3]) ([.38,.39],[.37,.41])
7	([.36,.37],[.35,.39]) ([.37,.39],[.36,.4]) ([.28,.3],[.27,.32])	([.23,.25],[.22,.26]) ([.53,.54],[.52,.55]) ([.25,.26],[.24,.27])	([.27,.3],[.26,.32]) ([.37,.38],[.36,.39]) ([.24,.26],[.25,.27])	([.21,.22],[.2,.24]) ([.4,.43],[.39,.44]) ([.23,.24],[.22,.25])	([.35,.37],[.34,.38]) ([.56,.58],[.55,.6]) ([.37,.39],[.36,.4])
8	([.39,.41],[.4,.42]) ([.41,.42],[.4,.43]) ([.2,.23],[.21,.24])	([.24,.26],[.23,.28]) ([.05,.06],[.04,.07]) ([.16,.17],[.15,.18])	([.3,.32],[.29,.33]) ([.52,.53],[.51,.54]) ([.43,.45],[.41,.46])	([.38,.4],[.37,.42]) ([.19,.21],[.17,.22]) ([.4,.42],[.39,.43])	([.34,.35],[.33,.37]) ([.34,.36],[.33,.37]) ([.46,.47],[.44,.48])
9	([.38,.4],[.37,.41]) ([.1,.11],[.09,.13]) ([.31,.32],[.3,.33])	([.39,.41],[.38,.42]) ([.38,.39],[.37,.4]) ([.34,.36],[.33,.37])	([.04,.06],[.03,.09]) ([.34,.36],[.33,.37]) ([.36,.37],[.35,.39])	([.23,.25],[.22,.26]) ([.33,.34],[.31,.36]) ([.28,.29],[.27,.3])	([.2,.21],[.19,.23]) ([.31,.32],[.3,.33]) ([.2,.21],[.19,.22])
10	([.15,.17],[.14,.18]) ([.25,.26],[.24,.28]) ([.25,.29],[.24,.3])	([.28,.3],[.27,.31]) ([.2,.21],[.19,.22]) ([.31,.32],[.3,.34])	([.26,.28],[.25,.29]) ([.18,.19],[.17,.2]) ([.28,.3],[.27,.32])	([.18,.2],[.17,.21]) ([.29,.31],[.28,.32]) ([.21,.22],[.2,.24])	([.09,.11],[.08,.12]) ([.32,.33],[.31,.34]) ([.2,.22],[.19,.24])
	Rough Risks/Discomforts Matrix(10 ×10) for RCSTSP With Three Conveyances				
i/j	6	7	8	9	10
1	([.05,.07],[.04,.10]) ([.22,.23],[.21,.25]) ([.06,.08],[.05,.09])	([.15,.16],[.14,.18]) ([.35,.33],[.31,.37]) ([.46,.47],[.45,.48])	([.25,.28],[.23,.29]) ([.37,.39],[.36,.43]) ([.16,.19],[.15,.2])	([.39,.41],[.38,.42]) ([.26,.31],[.25,.33]) ([.41,.42],[.4,.43])	([.2,.22],[.19,.23]) ([.3,.31],[.29,.34]) ([.42,.43],[.4,.45])
2	([.39,.4],[.38,.41]) ([.3,.31],[.29,.32]) ([.21,.23],[.2,.26])	([.39,.4],[.37,.42]) ([.29,.3],[.27,.32]) ([.57,.59],[.55,.6])	([.3,.33],[.29,.34]) ([.41,.42],[.4,.45]) ([.58,.59],[.57,.62])	([.17,.19],[.16,.22]) ([.36,.37],[.35,.38]) ([.17,.2],[.16,.21])	([.23,.24],[.22,.26]) ([.13,.16],[.15,.17]) ([.17,.18],[.19,.2])
3	([.33,.35],[.32,.36]) ([.34,.36],[.33,.39]) ([.33,.34],[.32,.35])	([.17,.19],[.16,.2]) ([.11,.12],[.1,.13]) ([.05,.08],[.04,.1])	([.3,.32],[.29,.33]) ([.3,.33],[.29,.34]) ([.24,.25],[.23,.27])	([.28,.3],[.27,.31]) ([.18,.19],[.17,.21]) ([.4,.41],[.39,.44])	([.29,.3],[.28,.31]) ([.19,.22],[.18,.23]) ([.32,.33],[.31,.35])]
4	([.23,.25],[.22,.26]) ([.15,.16],[.14,.18]) ([.25,.27],[.24,.28])	([.19,.21],[.18,.22]) ([.3,.31],[.29,.32]) ([.3,.33],[.29,.34])	([.33,.35],[.32,.36]) ([.32,.36],[.31,.38]) ([.16,.17],[.15,.18])	([.1,.12],[.09,.13]) ([.2,.23],[.19,.24]) ([.32,.34],[.31,.35])	([.24,.27],[.22,.29]) ([.47,.48],[.45,.49]) ([.37,.39],[.36,.4])
5	([.29,.3],[.28,.31]) ([.2,.21],[.19,.23]) ([.4,.41],[.39,.42])	([.42,.45],[.41,.46]) ([.14,.16],[.13,.18]) ([.25,.27],[.23,.28])	([.27,.3],[.25,.31]) ([.3,.31],[.28,.32]) ([.12,.13],[.11,.16])	([.18,.19],[.17,.22]) ([.08,.1],[.07,.11]) ([.07,.08],[.06,.09])	([.26,.28],[.24,.29]) ([.25,.26],[.23,.27]) ([.25,.27],[.23,.28])
6	∞	([.31,.33],[.3,.34]) ([.4,.43],[.39,.44]) ([.23,.24],[.22,.26])	([.39,.4],[.37,.42]) ([.3,.31],[.29,.31]) ([.2,.22],[.19,.23])	([.3,.32],[.29,.33]) ([.22,.23],[.21,.24]) ([.35,.36],[.33,.37])	([.28,.3],[.27,.31]) ([.4,.41],[.39,.42]) ([.3,.32],[.29,.34])
7	([.38,.4],[.37,.41]) ([.2,.21],[.19,.22]) ([.43,.45],[.41,.46])	∞	([.07,.1],[.06,.11]) ([.4,.43],[.39,.44]) ([.11,.13],[.1,.14])	([.31,.33],[.28,.34]) ([.33,.34],[.31,.35]) ([.34,.36],[.33,.37])	([.19,.2],[.18,.22]) ([.13,.15],[.12,.16]) ([.25,.26],[.23,.28])
8	([.23,.25],[.21,.26]) ([.15,.16],[.14,.18]) ([.04,.05],[.03,.06])	([.39,.4],[.38,.42]) ([.19,.21],[.18,.22]) ([.41,.43],[.4,.44])	∞	([.2,.22],[.19,.23]) ([.52,.53],[.5,.54]) ([.23,.24],[.22,.27])	([.35,.37],[.34,.38]) ([.35,.36],[.33,.38]) ([.39,.4],[.38,.41])
9	([.22,.23],[.2,.25]) ([.31,.33],[.3,.34]) ([.23,.25],[.21,.26])	([.05,.07],[.04,.08]) ([.36,.38],[.35,.39]) ([.38,.39],[.37,.41])	([.3,.32],[.29,.33]) ([.32,.33],[.3,.34]) ([.11,.13],[.1,.15])	∞	([.27,.28],[.26,.3]) ([.18,.19],[.17,.2]) ([.24,.26],[.23,.27])
10	([.3,.32],[.29,.34]) ([.1,.12],[.09,.13]) ([.33,.34],[.31,.35])	([.35,.38],[.33,.39]) ([.26,.28],[.25,.29]) ([.38,.39],[.37,.41])	([.4,.41],[.38,.43]) ([.41,.42],[.39,.43]) ([.3,.33],[.29,.34])	([.29,.31],[.28,.32]) ([.51,.52],[.5,.54]) ([.3,.32],[.28,.33])	∞

Table 4.18: Results of RCSTSP (Model 4.2C)

Algorithm	Path(Vehicle)	Costs	R_{max}
AGA	3(1)-10(3)-2(3)-9(3)-7(3)-8(1)-6(2)-1(1)-5(2)-4(3)	152.68	8.5
	5(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-6(2)-1(3)-7(3)	156.52	8.5
	10(2)-1(1)-9(2)-5(1)-7(2)-3(1)-4(2)-2(3)-8(1)-6(1)	156.61	6.75
GA	6(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-5(2)-1(3)-7(3)	172.21	6.0
AGA	6(2)-8(1)-7(1)-1(1)-10(2)-5(3)-2(3)-3(1)-4(2)-9(3)	175.29	6.75
	4(3)-7(1)-8(1)-6(2)-1(2)-5(1)-9(2)-2(1)-10(1)-3(1)	194.96	6.5
	3(1)-4(3)-2(3)-10(2)-5(3)-9(3)-8(1)-6(1)-1(3)-7(2)	215.21	6.0

Table 4.19: Statistical Test for AGA

Instances	Best	Worst	Average	SD	Error
fri26	937	964	940.2	0.912	0.01
bays29	2020	2046	2027.8	1.35	1.52
bayg29	1610	1639	1620.35	1.61	1.65
dantzig42	699	736	704.5	0.32	0.98
eil51	426	445	429.15	1.03	0.76
berlin52	7542	7576	7549.45	1.32	2.04
st70	675	689	683.5	1.75	2.03
eil76	538	563	552.75	1.43	1.93
pr76	108159	110342	108567.45	3.78	2.87
rat99	1211	1229	1218.35	0.75	1.46
kroa100	21282	21763	21347.76	2.34	0.98

4.3.4 Statistical Test

Here, we study the best, worst and average results with standard deviation and percentage error of the standard TSP from TSPLIB [162] under of 20 individual run by proposed AGA. The Table 4.19 given the results.

4.3.5 Discussion

To validate the feasibility and effectiveness of the proposed algorithm, we have applied the proposed AGA on some standard TSP problem taken from TSPLIB [162]. The proposed algorithm was implemented in C++ with following parameters as 100 chromosomes, 2000 iterations in maximum. Table 4.12 shows the comparison between AGA and SGA for the some standard TSP problems. It is seen that the number of iterations is less in AGA than the simple GA. Here also, AGA performs better than the SGA.

For a two-dimensional CTSP, we take a single conveyance in CSTSP and the corresponding crisp costs and risk/discomfort matrices are given in Table 4.13 by $(10 \times 10 \times 1)$ matrices. The CTSP is solved by both AGA and SGA and the results are presented in Table 4.14. It is observed that CTSP without risk constraint gives the lowest minimum cost. Here as the maximum allowable risk value decreases, the total cost increases. This as per expectation. Moreover, GA gives more cost than the AGA for the allowable risk value.

Again, we form a CSTSP with the three conveyances i.e. $(10 \times 10 \times 3)$ costs and risk/discomfort matrices are presented in Table 4.13 for **Model 4.2A**. Along each route, the corresponding conveyance is in parentheses. Next the optimum results of CSTSP are given in Table 4.14. Here also as total risk/discomfort goes down, the corresponding travelling cost increases.

A $(10 \times 10 \times 3)$ FCSTSP is presented in Table 4.15 where both costs and risk/discomfort values along with the targeted total risk/discomfort are triangular fuzzy numbers for **Model 4.2B**. The optimum results in both optimistic and pessimistic senses with different possibility and necessity levels are presented in Table 4.16. As expected, optimistic model fetches less travelling cost than the pessimistic model.

In Table 4.17, the costs and risk/discomfort values for the same size CSTSP are rough data for **Model 4.2C**. Last Table 4.18 shows the results of RCSTSP in rough environment. In all cases, the near-optimum solutions and optimum solution are available. Also AGA gives better results than the SGA.

4.4 Model-4.3: A Modified Genetic Algorithm for solving Uncertain CSTSPs [3]

In this investigation, a Modified Genetic Algorithm (MGA) is developed to solve constrained solid travelling salesman problems (CSTSPs) in crisp, fuzzy, random, random-fuzzy, fuzzy-random and bi-random environments. In the proposed MGA, for the, a probabilistic selection technique and a comparison crossover are used along with conventional random mutation. Here we model the CSTSP with travelling costs and route risk/discomfort factors as crisp, fuzzy, random, random-fuzzy, fuzzy-random and bi-random in nature. A number of benchmark problems from standard data set, TSPLIB [162] are tested against the SGA and the proposed MGA, hence the efficiency of the proposed algorithm is established. In this investigation, CSTSPs are illustrated numerically by some empirical data using this algorithm. In each environment, some sensitivity studies due to different risk/discomfort factors and other system parameters are presented.

4.4.1 Proposed MGA

Here using the probabilistic selection (Boltzmann Probability), comparison crossover and p_m dependent random mutation operators, we develop modified GA (MGA). The proposed MGA and its procedures are presented below

i. Representation: Here a complete tour of N cities represents a solution. So an N dimensional integer vector X_i = (x_{i1}, x_{i2}, ..., x_{iN}) and Y_i = (v_{i1}, v_{i2}, ..., v_{iP}) are used to represent a solution, where x_{i1}, x_{i2}, ..., x_{iN} represent N consecutive cities in a tour and Y_i = (v_{i1}, v_{i2}, ..., v_{iP}) represents the corresponding conveyances. Populations with the solutions X_i = (x_{i1}, x_{i2}, ..., x_{iN}), and Y_i = (v_{i1}, v_{i2}, ..., v_{iP}) i = 1, 2, ..., N, are randomly generated by random number generator.

ii. Probabilistic Selection:

a. Probability of Selection Parameter (p_s):

Here we introduce a predefined value say probability of selection parameter (p_s). For each solution of X_i, generate a random number r from the range [0,1]. If $r < p_s$ then the corresponding chromosome is stored at matting pool.

[3]This portion is published in **Computers** & **Industrial Engineering, Elsevier**, 83 (2015), 273-296, http://dx.doi.org/10.1016/j.cie.2015.02.023, with title *A Modified Genetic Algorithm for solving uncertain Constrained Solid Travelling Salesman Problems.*

b. Boltzmann-Probability: It is described in section 4.2.1(c)

Here $p_B = e^{((f_{min}-f(X_i))/T)}$, $T=T_0(1\text{-}a)^k$, k=(1+100*(g/G)), g(current) and G(maximum)generation, T_0= rand[5,100], a=rand[0,1], X_i means the chromosome corresponding to X_i, i=chromosome number.

c. Pseudo code of Selection:

```
input : Max-gen (G), Probability of selection (p_s), pop - size.
output : Matting pool.
begin
  for (i=1 to G)
   for (j=1 to pop - size)
      r=rand[0,1];
      T_0= rand[5,100];
      a=rand[0,1];
      k=(1+100*(i/G));
      T=T_0(1-a)^k;
      p_B=e^((f_min-f(X_j))/T);
         if (r < p_s )
           {
             choose the current chromosome;
             j++;
           }
         else if (r < p_B)
           {
             select X_j;
             j++;
           }
         else
           {
             Select the corresponding chromosome of f_min;
             j++;
           }
   end for
  end for
end
```

iii. Comparison Crossover: It is described in the section 4.3.1(iii).

Pseudo code of Crossover:

```
input: Matting Pool, p_c, Tptal number of node (N).
output: Offspring (child).
begin
    for ( j=1; j<=N; j++) // N= total number of nodes.
        if (c (a_i, a_1) < c (a_i, s_1)) // i ∈ {1, 2, ..., N}
            {
             if (a_1 exist in Ch_1)
                 {
                   j++;
                   compare next node from P_r1;
                 }
             else
                 {
                   concatenate a_1 in Ch_1;
                   j++;
                 }
            }
        else
            {
            if (s_1 exist in Ch_1)
                 {
                   j++;
                   compare next node from P_r2;
                 }
             else
                 {
                   concatenate s_1 in Ch_1;
                   j++;
                 }
            }
    end for
end
```

During every comparison, concatenate a node such that the travel path satisfies the TSP conditions. Firstly in every comparison, check if the node already exists in the child, then the cost of the next node in modified parents will be considered i.e. repetition of the nodes are not allowed. Secondly comparison will occur until every node of the modified parents are checked i.e. every node must exist in the child.

iv. p_m dependent Random Mutation:

a. Selection for mutation: For each solution of $p(t)$, generate a random number r from the range [0,1]. If $r < p_m$ then the solution is taken for mutation.

b. Mutation process: At first determined the total number of mutated node (T). To mutate a solution $X = (x_1, x_2, ..., x_N)$, number of mutated node T= p_m* N, N=total number of nodes.

c. Pseudo code of Mutation:

```
input: pop_size, (p_m) and total number of nodes (N).
output: Mutated offspring (child).
begin
Determine T= p_m*N // total number of mutated node
  for i=0 to pop_size
  r=rand(0,1)
  if( r< p_m){
      Select chromosome depending p_m
          for j=1 to T
            Randomly select two different nodes between [1,N];
             Swap the nodes;
          end for
        }
  end for
end
```

Procedure of MGA:

procedure name: Modified Genetic Algorithm (MGA).

input: Max Gen (S_0), Population Size (pop_size), Probability of Selection (p_s), Probability of Crossover (p_c), Probability of Mutation (p_m), Problem Data (cost and risk matrices).

output: The optimum and near optimum solutions.

1. **Start**
2. Set initial generation t ← 0.
3. (Initialization) Randomly generate initial population $p(t)$ where X_i, i=1,2...,pop$_{-size}$ are the chromosomes, N numbers of node in each chromosome represent a solution/path of the TSP.
4. Evaluate the fitness of each solution of the initial population $p(t)$.
5. Check the condition **while** (t≤ S_0) **do** upto step 21.
6. Update the generation t ← t+1.
7. Selection Procedure.
8. Determine the Boltzmann Probability (p_B) of each chromosome
9. Create the matting pool based on p_s and p_B.
10. Crossover Procedure.
11. Select the parents using p_c from matting pool.
12. According to Subsection 4.4.1.(iii) perform the crossover
13. Modified the parents.
14. Generate off springs and replace the parents.
15. Repeat the Step 11 to Step 14 depend on p_c.
16. Mutation Procedure done according the Subsection 3.iv.c.
17. Select the off springs for mutation based on p_m.
18. Exchange the place of these nodes;
19. Store the new off springs into offspring set.
20. Compare the fitness and Store the local, near optimum.
21. Repeat the Step 5 to Step 21.
22. (Optimum Solution) Store the optimum and near optimums
23. **Stop**.

4.4.2 Mathematical Formulation and Its crisp equivalence

STSP with risk/discomfort Constraints (CSTSP):

Model 4.3A: This model is same as given in section 4.3.2.

CSTSP in Fuzzy Environment (FCSTSP):

Model 4.3A1: This model given in section 4.3.2.

Deterministic form of Model 4.3A1: Possibility and Necessity Approaches
Deterministic forms due to possibilistic and necessity approaches are given in Equs. 4.14 and 4.15 respectively.

Deterministic form of Model 4.3A1: GMIV approach:
Again the Model 4.3A1 defined in Equ.4.1 can be converted, using the section 3.4.2. Applying GMIV method on FCSTSP, redesigned crisp model is given below:
Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one of the available conveyances in each step from the vehicle types $(v_1, v_2, ..., v_P)$

$$\left.\begin{array}{rl} \text{to minimize Z} &= \frac{1}{6}[F_1 - 4F_2 + F_3] \\ \text{subject to} & \frac{1}{6}[R_1 - 4R_2 + R_3] \leq \frac{1}{6}[r_1 - 4r_2 + r_3] \end{array}\right\} \tag{4.19}$$

Deterministic form of Model 4.3A1: Credibility Approach:
Now for the model defined in section 4.3.2, crisp form according to credibility measure given in Equ. 3.18. is :
Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one of the available conveyances in each step from the vehicle types $(v_1, v_2, ..., v_P)$

$$\left.\begin{array}{r} \text{to minimize} \quad F \\ \text{subject to} \quad Cr(\sum_{i=1}^{N-1} \tilde{c}(x_i, x_{i+1}, v_i) + \tilde{c}(x_N, x_1, v_l) < F) \\ Cr(\sum_{i=1}^{N-1} \tilde{r}(x_i, x_{i+1}, v_i) + \tilde{r}(x_N, x_1, v_l)) \leq Cr(\tilde{r}_{max}) \\ where\ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\} \end{array}\right\} \tag{4.20}$$

Using Equ. 3.18, the above Equ. 4.20 is transformed as

$$\left.\begin{array}{r} \text{to minimize} \quad F \\ \text{subject to} \quad \frac{F - F_1}{2(F_2 - F_1)} \geq \alpha_5 \ \ if \ \ F_1 \leq F \leq F_2 \\ \frac{F - 2F_2 + F_3}{2(F_3 - F_2)} \geq \alpha_5 \ \ if \ \ F_2 \leq F \leq F_3 \\ \frac{r_1 - R_1}{2(R_2 - R_1 - r_2 + r_1)} \geq \alpha_6 \ \ if \ \ \alpha_6 > 0.5 \\ \frac{2(r_2 - R_2) + R_3 - r_3}{2(R_3 - R_2 - r_3 + r_2)} \leq \alpha_6 \ \ if \ \ \alpha_6 \leq 0.5 \end{array}\right\} \tag{4.21}$$

Here α_5, α_6 are predefined confidence levels and F be crisp values given by the salesman.
Thus above equation can be written as

$$\left.\begin{array}{rl} \text{to minimize} & F_1 + 2\alpha_5(F_2 - F_1) \;\; if \;\; F_1 \le F \le F_2 \\ \text{subject to} & \dfrac{r_1 - R_1}{2(R_2 - R_1 - r_2 + r_1)} \ge \alpha_6 \;\; if \;\; \alpha_6 > 0.5 \end{array}\right\} \tag{4.22}$$

and

$$\left.\begin{array}{rl} \text{to minimize} & 2F_2 + F_3 + 2\alpha_5(F_3 - F_2) \;\; if \;\; F_2 \le F \le F_3 \\ \text{subject to} & \dfrac{2(r_2 - R_2) + R_3 - r_3}{2(R_3 - R_2 - r_3 + r_2)} \le \alpha_6 \;\; if \;\; \alpha_6 \le 0.5 \end{array}\right\} \tag{4.23}$$

Deterministic form of Model 4.3A1: EVM Approach:
Again we rewrite the model 4.3.2 according to another crisp conversion form given in section 3.9 known as expected value model.
Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one of teh available corresponding conveyances in each step from the vehicle types $(v_1, v_2, ..., v_P)$

$$\left.\begin{array}{l} \text{to minimize} \;\; Z = E(\sum_{i=1}^{N-1} \tilde{c}(x_i, x_{i+1}, v_i) + \tilde{c}(x_N, x_1, v_l)) \\ \text{subject to} \;\; E(\sum_{i=1}^{N-1} \tilde{r}(x_i, x_{i+1}, v_i) + \tilde{r}(x_N, x_1, v_l)) \le E(\tilde{r}_{max}) \\ where \; x_i \ne x_j, i, j = 1, 2...N, \;\; v_i, v_l \in \{1, 2.., or\ P\} \end{array}\right\} \tag{4.24}$$

Thus the above form is written as

$$\left.\begin{array}{rl} \text{to minimize} & \frac{1}{4}(F_1 + 2F_2 + F_3) \\ \text{subject to} & \frac{1}{4}(R_1 + 2R_2 + R_3) \le \frac{1}{4}(r_1 + 2r_2 + r_3) \end{array}\right\} \tag{4.25}$$

Model 4.3A2: CSTSP in Random Environment (RaCSTSP)

In the problem formulation section 4.3.2, if costs and risk/discomfort factors are random parameters, i.e, $\hat{c}(i, j, k)$ and $\hat{r}(i, j, k)$ respectively and maximum risk/discomfort limit r_{max} also a random variable $\hat{r}_{max}$ then the Equ. 4.9 reduces to:

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using suitable one among the available conveyances in each step from the vehicle types $(v_1, v_2, ..., v_P)$ so as

$$\left.\begin{array}{c}\text{to minimize} \quad Z = \sum_{i=1}^{N-1} \hat{c}(x_i, x_{i+1}, v_i) + \hat{c}(x_N, x_1, v_l) \\ \text{subject to} \quad \sum_{i=1}^{N-1} \hat{r}(x_i, x_{i+1}, v_i) + \hat{r}(x_N, x_1, v_l) \leq \hat{r}_{max} \\ where\ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\}\end{array}\right\} \quad (4.26)$$

Now using Chance-constrained programming technique according to section 3.13.2, the above model reduces to:

$$\left.\begin{array}{c}\text{to minimize} \quad Z = \sum_{i=1}^{N-1} \hat{c}(x_i, x_{i+1}, v_i) + \hat{c}(x_N, x_1, v_l) \\ \text{subject to} \quad P[\sum_{i=1}^{N-1} \hat{r}(x_i, x_{i+1}, v_i) + \hat{r}(x_N, x_1, v_l) \leq \hat{r}_{max}] \geq p_i, \\ where\ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\}. \\ \text{Here } p_i \text{ are crisp values.}\end{array}\right\} \quad (4.27)$$

Also we consider all random variables are normal variate. Then the objective function also a normal variate. Thus the mean and variance are given by

$$\left.\begin{array}{c}\bar{Z} = E[\sum_{i=1}^{N-1} \hat{c}(x_i, x_{i+1}, v_i) + \hat{c}(x_N, x_1, v_l)] \\ Var(Z) = X^T V X \\ \text{where V is the covariance matrix of } c_j.\end{array}\right\} \quad (4.28)$$

Thus the above constrained stochastic problem given by Equ. 4.27 finally reduce as to:

$$\left.\begin{array}{c}\text{Minimize F(X)} \quad = k_1 * E[\sum_{i=1}^{N-1} \hat{c}(x_i, x_{i+1}, v_i) + \hat{c}(x_N, x_1, v_l)] \\ +k_2 * \sqrt{(X^T V X)}, \\ \text{subject to} \quad \bar{h}_i + s_i\sqrt{(Var(h_i)} \leq 0, \quad i = 1, 2, ...n \\ x_j \geq 0, j = 1, 2, ..., n \\ where\ x_i \neq x_j, \ i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\}.\end{array}\right\} \quad (4.29)$$

Here $\bar{h}_i = E[\sum_{i=1}^{N-1} \hat{r}(x_i, x_{i+1}, v_i) + \hat{r}(x_N, x_1, v_l)] - \bar{r}_{max}, k_1, k_2 \geq 0$

where k_1 and k_2 are constants indicating the weights of mean and variance functions, s_i is the tabulated value of the normal distribution.

Model 4.3A3: CSTSP in Random-Fuzzy Environment (RFCSTSP):

In the problem section 4.3.2, if costs and risk/discomfort factors are random-fuzzy parameters, i.e, $\hat{\tilde{c}}(i,j,k)$ and $\hat{\tilde{r}}(i,j,k)$ respectively and maximum risk/discomfort limit r_{max} also is a random-fuzzy data $\hat{\tilde{r}}_{max}$, then the Equ. 4.9 reduces to:

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using a suitable one among the available conveyances in each step from the vehicle types $(v_1, v_2, ..., v_P)$ so as

$$\left.\begin{aligned} &\text{to minimize} \quad Z = \sum_{i=1}^{N-1} \hat{\tilde{c}}(x_i, x_{i+1}, v_i) + \hat{\tilde{c}}(x_N, x_1, v_l) \\ &\text{subject to} \quad \sum_{i=1}^{N-1} \hat{\tilde{r}}(x_i, x_{i+1}, v_i) + \hat{\tilde{r}}(x_N, x_1, v_l) \leq \hat{\tilde{r}}_{max} \\ &where\ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\}. \end{aligned}\right\} \tag{4.30}$$

Above Equ. 4.30 can be reformulated as given below where the objective function
$\sum_{i=1}^{N-1} \hat{\tilde{c}}(x_i, x_{i+1}, v_i) + \hat{\tilde{c}}(x_N, x_1, v_l) \leq F_1$, F_1 is a crisp values.

Using section 3.11, the Equ. 4.30 can be defined as possibilistic and necessity chance constraint forms

$$\left.\begin{aligned} &\text{minimize} \quad F_1 \\ &Pos\{Prob\{\sum_{i=1}^{N-1} \hat{\tilde{c}}(x_i, x_{i+1}, v_i) + \hat{\tilde{c}}(x_N, x_1, v_l) \leq F_1\} \geq \hat{\theta}^{obj}\} \geq \hat{h}^{obj} \\ &Nes\{Prob\{\sum_{i=1}^{N-1} \hat{\tilde{c}}(x_i, x_{i+1}, v_i) + \hat{\tilde{c}}(x_N, x_1, v_l) \leq F_1\} \geq \hat{\theta}^{obj}\} \geq \hat{h}^{obj} \\ \text{s.t.} \quad &Pos\{Prob\{\sum_{i=1}^{N-1} \hat{\tilde{r}}(x_i, x_{i+1}, v_i) + \hat{\tilde{r}}(x_N, x_1, v_l) \leq \hat{\tilde{r}}_{max}\} \geq \hat{\theta}^{cst}\} \geq \hat{h}^{cst} \\ &Nes\{Prob\{\sum_{i=1}^{N-1} \hat{\tilde{r}}(x_i, x_{i+1}, v_i) + \hat{\tilde{r}}(x_N, x_1, v_l) \leq \hat{\tilde{r}}_{max}\} \geq \hat{\theta}^{cst}\} \geq \hat{h}^{cst} \\ &where\ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\}. \end{aligned}\right\} \tag{4.31}$$

The above Equ. 4.31 equivalently is written as

$$\left.\begin{array}{c} Pos\{Prob\{\hat{\tilde{C}}x \leq F_1\} \geq \hat{\theta}^{obj}\} \geq \hat{h}^{obj} \\ Nes\{Prob\{\hat{\tilde{C}}x \leq F_1\} \geq \hat{\theta}^{obj}\} \geq \hat{h}^{obj} \\ \text{subject to} \quad Pos\{Prob\{\hat{\tilde{R}}x \leq \hat{\tilde{r}}_{max}\} \geq \hat{\theta}^{cst}\} \geq \hat{h}^{cst} \\ Nes\{Prob\{\hat{\tilde{R}}x \leq \hat{\tilde{r}}_{max}\} \geq \hat{\theta}^{cst}\} \geq \hat{h}^{cst} \end{array}\right\} \quad (4.32)$$

where $\hat{\tilde{C}} = \sum_{i=1}^{N-1} \hat{\tilde{c}}_1(x_i, x_{i+1}, v_i) + \hat{\tilde{c}}_1(x_N, x_1, v_l)$, $\hat{\tilde{R}} = \sum_{i=1}^{N-1} \hat{\tilde{r}}(x_i, x_{i+1}, v_i) + \hat{\tilde{r}}(x_N, x_1, v_l)$

The above Equ. 4.32 using section 3.11 is equivalently transformed into

$$\left.\begin{array}{c} \sum_{i=1}^{N}\{m_i^c - L*(\hat{h_i}^{obj})\alpha_i^c\}x_i + \Phi^{-1}(\hat{\theta}^{obj})\sqrt{(x^tV^cx)} \leq F_1 \\ \sum_{i=1}^{N}\{m_i^c + L*(1-\hat{h_i}^{obj})\beta_i^c\}x_i + \Phi^{-1}(\hat{\theta}^{obj})\sqrt{(x^tV^cx)} \leq F_1 \\ \text{s.t.} \quad \sum_{i=1}^{N}\{m_i^R - L*(\hat{h_i}^{cst})\alpha_i^R\}x_i + \Phi^{-1}(\hat{\theta}^{cst})\sqrt{(x^tV^Rx + (\sigma_i^r)^2)} \leq \\ m_i^r + L*(\hat{h}_i^{cst})\beta_i^r \\ \sum_{i=1}^{N}\{m_i^R + L*(1-\hat{h_i}^{cst})\beta_i^R\}x_i + \Phi^{-1}(\hat{\theta}^{cst})\sqrt{(x^tV^Rx + (\sigma_i^r)^2)} \leq \\ m_i^r - L*(1-\hat{h}_i^{cst})\alpha_i^r \end{array}\right\} (4.33)$$

Finally the above random-fuzzy model is transformed into the crisp model as given below:

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using a suitable one among the available conveyances in each step from the vehicle types $(v_1, v_2, ..., v_P)$

$$\left.\begin{array}{c} \text{to min } F_1 = \sum_{i=1}^{N}\{m_i^c - L*(\hat{h_i}^{obj})\alpha_i^c\}x_i + \Phi^{-1}(\hat{\theta}^{obj})\sqrt{(x^tV^cx)} \\ \text{s.t.} \quad \sum_{i=1}^{N}\{m_i^R - L*(\hat{h_i}^{cst})\alpha_i^R\}x_i + \Phi^{-1}(\hat{\theta}^{cst})\sqrt{(x^tV^Rx + (\sigma_i^r)^2)} \\ \leq m_i^r + L*(\hat{h}_i^{cst})\beta_i^r \end{array}\right\} \quad (4.34)$$

and

$$\left.\begin{array}{l} \text{to min } \mathrm{F}_1 = \sum_{i=1}^{N}\{m_i^c + L*(1-\hat{h_i}^{obj})\beta_i^c\}x_i + \Phi^{-1}(\hat{\theta}^{obj})\sqrt{(x^tV^cx)} \\ \text{s.t.} \sum_{i=1}^{N}\{m_i^R + L*(1-\hat{h_i}^{cst})\beta_i^R\}x_i + \Phi^{-1}(\hat{\theta}^{cst})\sqrt{(x^tV^Rx + (\sigma_i^r)^2)} \\ \qquad \leq m_i^r - L*(1-\hat{h}_i^{cst})\alpha_i^r \end{array}\right\} \quad (4.35)$$

where $\alpha_i^c, \alpha_i^R, \beta_i^c, \beta_i^R$ and β_i^r are predetermined given values. Again $\hat{h}^{obj}, \hat{h}^{cst}$ are permissible possibility or necessity levels for the objectives and risk/discomfort constraints. Also $\hat{\theta}^{obj}, \hat{\theta}^{cst}$ are permissible probability levels for the objectives and constraints respectively.

Model 4.3A4: CSTSP in Fuzzy Random Environment (FRCSTSP):

In the problem formulated in the section 4.3.2, if costs and risk/discomfort factors are fuzzy random parameters, i.e, $\tilde{\bar{c}}(i,j,k)$ and $\tilde{\bar{r}}(i,j,k)$ respectively and allowable maximum risk/discomfort limit r_{max} is also a fuzzy random variables $\tilde{\bar{r}}_{max}$, then the Equ. 4.9 reduces to:

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using a suitable one among the available conveyances in each step from the vehicle types $(v_1, v_2, ..., v_P)$ so as

$$\left.\begin{array}{l} \text{to minimize } \ Z = \sum_{i=1}^{N-1}\tilde{\bar{c}}(x_i, x_{i+1}, v_i) + \tilde{\bar{c}}(x_N, x_1, v_l) \\ \text{subject to } \ \sum_{i=1}^{N-1}\tilde{\bar{r}}(x_i, x_{i+1}, v_i) + \tilde{\bar{r}}(x_N, x_1, v_l) \leq \tilde{\bar{r}}_{max} \\ where\ x_i \neq x_j, i, j = 1, 2...N, \ \ v_i, v_l \in \{1, 2.., or P\}. \end{array}\right\} \quad (4.36)$$

Above Equ. 4.36 can be reformulated as given below, where the objective function is

$$\sum_{i=1}^{N-1}\tilde{\bar{c}}(x_i, x_{i+1}, v_i) + \tilde{\bar{c}}(x_N, x_1, v_l) \leq F,$$

F is a given crisp value, and equations evaluated using fuzzy random chance constrained programming technique according to the Theorem 3.9.

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using a suitable one among the available conveyance in each step from the vehicle types $(v_1, v_2, ..., v_P)$ to minimize F

$$\left.\begin{array}{c}\text{s.t. } Ch\{\sum_{i=1}^{N-1} \tilde{\hat{c}}(x_i, x_{i+1}, v_i) + \tilde{\hat{c}}(x_N, x_1, v_l) \leq F\}(\gamma) \geq \delta \\ Ch \sum_{i=1}^{N-1} \tilde{\hat{r}}(x_i, x_{i+1}, v_i) + \tilde{\hat{r}}(x_N, x_1, v_l) \leq \tilde{\hat{r}}_{max}\}(\eta) \geq \theta \\ where\ x_i \neq x_j, i, j = 1, 2...N, \ \ v_i, v_l \in \{1, 2.., or P\}.\end{array}\right\} \quad (4.37)$$

Here the parameters $\gamma, \delta, \theta, \eta$ are predetermined confidence levels in [0,1].
Now the above Equ. 4.37 is reformulated as

$$\left.\begin{array}{c}\text{minimize F} \\ \text{s.t } Ch\{\tilde{\hat{C}}x \leq F\}(\gamma) \geq \delta \\ Ch\{\tilde{\hat{R}}_1 x \leq \tilde{\hat{R}}_{max}\}(\eta) \geq \theta \\ x \in X\end{array}\right\} \quad (4.38)$$

where

$$\tilde{\hat{C}} = \sum_{i=1}^{N-1} \tilde{\hat{c}}(x_i, x_{i+1}, v_i) + \tilde{\hat{c}}(x_N, x_1, v_l),$$

$$\tilde{\hat{R}}_1 = \sum_{i=1}^{N-1} \tilde{\hat{r}}(x_i, x_{i+1}, v_i) + \tilde{\hat{r}}_1(x_N, x_1, v_l),$$

$$\tilde{\hat{R}}_{max} = \tilde{\hat{r}}_{max},$$

and X is a fixed set that usually determined by a finite of inequalities involving functions of x as a decision vectors.
It follows from section 3.13.10, the Equ. 4.37 is converted as follows for Probability Possibility measure

$$\left.\begin{array}{c}\text{minimize}\{F\} \\ \text{s.t. } Pr\{\omega | Pos\{\tilde{\hat{C}}x \leq F\} \geq \delta\} \geq \gamma \\ Pr\{\omega | Pos\{\tilde{\hat{R}}_1 x \leq \tilde{\hat{R}}_{max}\} \geq \theta\} \geq \eta \\ x \in X\end{array}\right\} \quad (4.39)$$

and the Probability Necessity measure form as given below

$$\left.\begin{array}{c}\text{minimize}\{F\} \\ \text{s.t. } Pr\{\omega | Nes\{\tilde{\hat{C}}x \leq F\} \geq \delta\} \geq \gamma \\ Pr\{\omega | Nes\{\tilde{\hat{R}}_1 x \leq \tilde{\hat{R}}_{max}\} \geq \theta\} \geq \eta \\ x \in X\end{array}\right\} \quad (4.40)$$

where $\gamma, \delta, \eta, \theta \in [0, 1]$ are the predetermined confidence levels.

To find the crisp values of probability possibility model according the Theorems 3.8 and 3.9 given by the above model Equ. 4.39 is converted as

$$\left.\begin{array}{c} \text{minimize F} = R^{-1}(\delta)\beta^{CT}x + d^{CT}x + \phi^{-1}(1-\gamma)\sqrt{(x^T V^C x)} \\ \text{s.t} \quad R^{-1}(\theta)\beta^{R_{max}} + L^{-1}(\theta)\alpha^{R_1 T}x - (d^{R_1 T}x - d^b) - \\ \phi^{-1}(\eta)\sqrt{(x^T V^{R_1}x + (\sigma^{R_{max}})^2)} \geq 0 \end{array}\right\} \quad (4.41)$$

Similarly for the possibility necessity approaches, according to the Theorems 3.9 and 3.10, the Equ.4.40 converted as

$$\left.\begin{array}{c} \text{minimize F} = d^{CT}x - L^{-1}(1-\delta)\alpha^{CT}x + \phi^{-1}(1-\gamma)\sqrt{(x^T V^C x)} \\ \text{s.t} \quad \phi^{-1}(1-\eta)\sqrt{(x^T V^{R_1}x + (\sigma^{R_{max}})^2)} - L^{-1}(1-\theta)\alpha^{R_{max}} - \\ R^{-1}(\theta)\beta^{R_1 T}x + \\ (d^{R_{max}} - d^{R_1 T}x) \geq 0 \end{array}\right\} \quad (4.42)$$

Model 4.3A5: CSTSP in Bi-random Environment (BRCSTSP):

For the wide range of statistical data values for the long time interval, decision maker are more affiniated towards the bi-random features. In the problem section 4.3.2, if costs and risk/discomfort factors are bi-random parameters, i.e, $\tilde{\tilde{c}}(i, j, k)$ and $\tilde{\tilde{r}}(i, j, k)$ respectively and maximum risk/discomfort limit r_{max} also bi-random variable $\tilde{\tilde{r}}_{max}$, then the Equ. 4.9 reduces to:

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using a suitable one among the available conveyances in each step from the vehicle types $(v_1, v_2, ..., v_P)$ so as

$$\left.\begin{array}{l} \text{to minimize} \quad Z = \sum_{i=1}^{N-1} \tilde{\tilde{c}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{c}}(x_N, x_1, v_l) \\ \text{subject to} \quad \sum_{i=1}^{N-1} \tilde{\tilde{r}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{r}}(x_N, x_1, v_l) \leq \tilde{\tilde{r}}_{max} \\ where \ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\}. \end{array}\right\} \quad (4.43)$$

Above Equ. 4.43 can be reformulated as given, where the objective function

$$\sum_{i=1}^{N-1} \tilde{\tilde{c}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{c}}(x_N, x_1, v_l) \leq F,$$

F is a crisp value, and equations are evaluated using equilibrium chance constrained programming technique.

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using a suitable one among the available conveyances in each step from the vehicle types $(v_1, v_2, ..., v_P)$ to minimize F

$$\left.\begin{array}{c} \text{subject to } Ch^e\{\sum_{i=1}^{N-1}\tilde{\tilde{c}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{c}}(x_N, x_1, v_l) \le F\} \ge \alpha \\ Ch^e \sum_{i=1}^{N-1}\tilde{\tilde{r}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{r}}(x_N, x_1, v_l) \le \tilde{\tilde{r}}_{max}\} \ge \beta \\ where\ x_i \ne x_j, i, j = 1, 2...N,\ \ v_i, v_l \in \{1, 2.., or P\}. \end{array}\right\} \quad (4.44)$$

Here α, β are predetermined confidence levels.
Now the above Equ. 4.44 is reformulated as

$$\left.\begin{array}{c} \text{minimize F} \\ \text{subject to } Ch^e\{\tilde{\tilde{C}}x \le F\} \ge \alpha \\ Ch^e\{\tilde{\tilde{R}}x \le \tilde{\tilde{R}}_{max}\} \ge \beta \\ x \in D \end{array}\right\} \quad (4.45)$$

where $\tilde{\tilde{C}} = \sum_{i=1}^{N-1}\tilde{\tilde{c}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{c}}(x_N, x_1, v_l)$, $\tilde{\tilde{R}} = \sum_{i=1}^{N-1}\tilde{\tilde{r}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{r}}_1(x_N, x_1, v_l)$, $\tilde{\tilde{R}}_{max} = \tilde{\tilde{r}}_{max}$,
and D is a fixed set that usually determined by a finite of inequalities involving functions of x.
Using the Theorem 3.3, the Equ. 4.45 can be written as

$$\left.\begin{array}{c} \text{subject to } Pr\{\omega \in \Omega | Pr\{\tilde{\tilde{C}}(\omega)x \le F\} \ge \alpha\} \ge \alpha \\ Pr\{\omega \in \Omega | Pr\{\tilde{\tilde{R}}(\omega)x \le \tilde{\tilde{R}}_{max}\} \ge \beta\} \ge \beta \\ x \in D \end{array}\right\} \quad (4.46)$$

Finally the above problem using Lemmas 3.3 and 3.4 reduces: to determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ and using a suitable one among the available conveyances in each step from the vehicle types $(v_1, v_2, ..., v_P)$

$$\left.\begin{array}{c} \text{minimize F} = \mu^c x + \Phi^{-1}(\alpha)\sqrt{(x^T V^c x)} + \Phi^{-1}(\alpha)\sqrt{(x^T V^{nc} x)} \\ \text{s.t } \mu^R x + \Phi^{-1}(\beta)\sqrt{(x^T V^R x + (\sigma^{R_{max}})^2)} + \\ \Phi^{-1}(\beta)\sqrt{(x^T V^{nR} x + (\sigma^{R_{nmax}})^2))} \le \mu^{R_{max}}, \\ x \in D. \end{array}\right\} \quad (4.47)$$

Table 4.20: Test TSPLIB Problems by MGA

Instances	Problem Size	Optimum Result	MGA Cost	MGA Iteration	GA Cost	GA Iteration
fri26	26	937	937	**78**	937	269
bays29	29	2020	2020	**61**	2020	451
bayg29	29	1610	1610	**66**	1610	378
dantzig42	42	699	699	**152**	699	612
eil51	51	426	426	**98**	426	341
berlin52	52	7542	7542	1**45**	7542	526
st70	70	675	675	**165**	675	813
eil76	76	538	538	**124**	538	457
pr76	76	108159	108159	**165**	108159	410
rat99	99	1211	1211	**147**	1211	328
kroa100	100	21282	212820	**276**	21282	285

Here α, β are given values. Again $\sigma^{R_{max}}$, $\sigma^{R_{nmax}}$, V^R, V^{nR}, V^c, V^{nc} are standard deviation and variances of maximum of risk/discomfort factors and costs in two fold randomness. Also Φ is a standard normal variate distributions.

Solution Procedures:

The deterministic forms of the uncertain CSTSPs given by Equ. 4.9 for crisp values, Equ.s 4.5, 4.6, 4.19, 4.22, 4.23 and 4.25 for FCSTSP in fuzzy values, Equ. 4.29 for RaCSTSP in random values parameters, Equ.s 4.34 and 4.35 for RFCSTSP in random fuzzy, Equ.s 4.41 and 4.42 for FRCSTSP with fuzzy random values and Equ.4.47 for BRCSTSP with bi-random values are solved by the MGA, developed for this purpose in section 4.4.1.

4.4.3 Numerical Experiments

Testing for MGA:

To judge the effectiveness and feasibility of developed algorithm MGA, we have applied it on the standard TSP problems from TSPLIB[162]. Table- 4.20 gives the results of test functions by both MGA and SGA and on comparison is made in terms of total cost and GA iterations.

Moreover, for a particular test problem bayg29, both standard GA and proposed MGA are used with different P_c's, P_m's and proposed P_s's. The obtained results are presented in Table 4.21.

Table 4.21: Comparison of MGA and SGA with different parameter

Algorithm	Selection	Crossover	Generation	p_c	p_m	p_s	Result
GA	Roulette Wheel	Cyclic	678	0.31	0.43	-	1610
GA	Probabilistic	Cyclic	309	0.31	0.43	-	1610
GA	Probabilistic	Comparison	256	0.4	0.43	-	1610
MGA	Probabilistic	Comparison	176	0.44	0.43	-	1610
MGA	**Probabilistic**	**Comparison**	**66**	**0.34**	**0.43**	**0.3**	1610
MGA	Roulette Wheel	Comparisons	211	0.34	0.43	-	1610
MGA	Roulette Wheel	Cyclic	411	0.5	0.43	-	1610

Model 4.3A: Results of CTSP with risk/discomfort Constraint in Crisp Environment:

Now, we consider a deterministic TSP given by Equ.4.9, whose cost and risk/discomfort matrices are given by Table 4.22. The problem is solved by MGA and the results are presented in Table 4.23.

Here GA parameter are : maximum generation=1000, p_s=0.3, p_c=0.34, p_m=0.4.

Model 4.3A: CSTSP with risk/discomfort Constraint in Crisp Environment:

Now for a CSTSP, we consider three types of conveyances. The cost and risk/discomfort matrices are given for the CSTSP in Table 4.24.

Here we have taken maximum generation=2000, p_s=0.31, p_c=0.34, p_m=0.43. This CSTSP is solved by MGA and the results are presented in Table 4.25.

Model 4.3A1: FCSTSP with risk/discomfort Constraint in Fuzzy Environments:

Here the cost and risk/discomfort values are fuzzy for the FCSTSP. Also we consider three types of conveyances. The fuzzy cost and corresponding fuzzy risk/discomfort matrices for the FCSTSP are given in the Table 4.26. This FCSTSP is solved by MGA and the results are presented in Table 4.27.

Model 4.3A2: RaCSTSP with risk/discomfort Constraint in Random Environment:

Here the cost and risk/discomfort values are random for the RaCSTSP. Also only three types of conveyances are available for transportation. The random cost and risk/discomfort values for the RaCSTSP are random and these values are presented in the form of mean and variances in Table 4.28. This RaCSTSP is

Table 4.22: Input Data: Crisp CTSP (Model 4.3A)

	Crisp Cost Matrix(10×10)									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	35	18	20	17	36	37	42	33	44
2	24	∞	20	28	35	40	30	43	28	14
3	38	27	∞	25	22	35	9	32	40	30
4	28	10	7	∞	20	25	30	35	22	37
5	27	22	35	30	∞	20	25	30	9	28
6	15	30	25	8	28	∞	33	40	32	30
7	38	25	30	22	37	40	∞	32	20	25
8	40	5	32	40	35	25	40	∞	37	38
9	40	40	23	25	20	2	37	32	∞	28
10	28	30	28	20	11	32	37	40	30	∞
	Crisp risk/discomfort Matrix									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	0.5	0.8	0.7	0.82	0.59	0.58	0.59	0.6	0.57
2	0.78	∞	0.81	0.75	0.5	0.6	0.7	0.58	0.75	0.9
3	0.59	0.79	∞	0.85	0.78	0.65	0.81	0.68	0.6	0.7
4	0.72	0.9	0.94	∞	0.8	0.75	0.7	0.65	0.78	0.63
5	0.83	0.79	0.69	0.72	∞	0.82	0.79	0.71	0.9	0.72
6	0.88	0.7	0.75	0.91	0.72	∞	0.67	0.6	0.7	0.77
7	0.68	0.59	0.8	0.7	0.6	0.61	∞	0.68	0.8	0.77
8	0.6	0.94	0.69	0.6	0.59	0.79	0.6	∞	0.59	0.73
9	0.6	0.81	0.77	0.75	0.8	0.99	0.63	0.68	∞	0.72
10	0.85	0.7	0.73	0.53	0.9	0.69	0.64	0.59	0.7	∞

Table 4.23: Results of Crisp CTSP (Model 4.3A)

Algorithm	Path	Value	R_{max}
MGA	8-2-10-5-9-6-1-4-3-7	124.00	Without R_{max}
MGA	8-2-10-5-9-6-1-4-3-7	124.00	8.64
MGA	5-9-6-4-3-7-10-8-2-1	130.00	8.64
MGA	8-2-10-4-3-7-9-1-5	139.00	8.64
MGA	4-8-2-10-5-9-6-1-3-7	140.00	8.64
GA	10-8-2-5-9-6-1-4-3-7	167.00	8.75
MGA	8-5-9-6-1-4-3-7-2-10	176.00	8.00
GA	-2-5-10-4-3-7-9-6-4	192.00	8.00
MGA	7-2-6-9-1-4-8-5-10-3	292.00	6.75

Table 4.24: Input Data: Crisp CSTSP(Model 4.3A)

	Crisp Cost Matrix(10×10) With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	35,36,27	18,39,30	20,33,34	30,21,62	6,23,8	15,36,47	27,38,19	40,31,42	20,31,42
2	35,26,17	∞	40,21,32	18,29,10	35,26,37	40,31,22	40,31,59	33,42,59	18,37,20	24,16,18
3	38,30,29	17,58,34	∞	12,25,14	42,25,46	35,36,34	19,11,8	32,33,25	30,19,41	30,22,33
4	28,20,11	10,22,14	17,8,29	∞	30,19,24	25,16,27	21,31,33	35,36,17	12,23,34	27,48,39
5	17,15,9	42,23,34	35,36,37	20,31,43	∞	30,21,42	45,16,27	30,31,13	19,10,8	28,26,7
6	15,6,7	30,21,29	5,26,28	8,9,12	28,29,40	∞	33,42,24	40,31,22	32,23,35	30,41,32
7	38,39,30	25,54,26	30,38,26	22,43,24	37,58,39	40,21,45	∞	10,41,13	32,33,35	20,15,26
8	40,41,23	25,6,17	32,53,45	40,21,42	35,36,47	25,16,5	40,22,43	∞	22,53,24	37,37,39
9	40,11,33	40,39,36	3,36,37	25,34,29	20,32,21	22,33,25	7,38,39	32,33,14	∞	28,19,26
10	18,27,29	30,21,32	28,19,30	20,31,22	11,33,22	32,12,34	37,28,39	40,41,33	30,51,33	∞
	Crisp risk/discomfort Matrix(10 ×10) With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	.69,.68,.75	.84,.63,.7	.82,.7,.71	.72,.8,.42	.96,.79,.93	.87,.66,.55	.74,.42,.81	.41,.7,.59	.81,.7,.59
2	.67,.76,.84	∞	.61,.8,.7	.83,.73,.92	.67,.76,.65	.41,.71,.79	.41,.71,.43	.69,.6,.42	.83,.64,.81	.77,.85,.3
3	.63,.71,.73	.83,.44,.67	∞	.89,.76,.86	.59,.76,.55	.66,.65,.67	.83,.91,.94	.69,.68,.76	.71,.82,.6	.71,.79,.68
4	.73,.81,.9	.9,.78,.86	.84,.93,.72	∞	.71,.82,.77	.77,.86,.75	.81,.71,.69	.66,.65,.84	.89,.79,.77	.74,.53,.43
5	.84,.86,.92	.59,.78,.67	.66,.65,.64	.82,.71,.59	∞	.71,.81,.59	.57,.85,.74	.71,.7,.88	.82,.91,.93	.74,.75,.93
6	.85,.84,.93	.7,.8,.71	.95,.74,.72	.92,.91,.89	.73,.72,.61	∞	.69,.59,.77	.61,.71,.79	.69,.78,.66	.71,.6,.69
7	.63,.62,.71	.77,.47,.76	.71,.63,.76	.79,.59,.77	.66,.43,.62	.6,.79,.55	∞	.9,.6,.87	.69,.68,.66	.81,.87,.76
8	.61,.6,.78	.76,.95,.84	.69,.47,.56	.61,.81,.6	.67,.66,.55	.6,.85,.95	.61,.8,.59	∞	.79,.48,.77	.64,.64,.62
9	.61,.91,.71	.61,.62,.65	.97,.65,.64	.76,.77,.72	.81,.69,.73	.79,.68,.76	.94,.66,.63	.69,.68,.87	∞	.73,.82,.75
10	.83,.74,.72	.71,.8,.69	.73,.83,.72	.8,.69,.78	.89,.67,.78	.7,.9,.71	.64,.74,.22	.61,.59,.68	.71,.5,.67	∞

Table 4.25: Results for Crisp CSTSP (Model 4.3A)

Algorithm	Path(Vehicle)	Cost	Risk achieved	R_{max}
MGA	1(1)-10(1)-5(2)-4(1)-2(1)-9(1)-3(2)-7(1)-8(3)-6(2)	107.00	8.71	8.75
MGA	9(1)-7(1)-8(1)-6(2)-1(1)-3(1)-4(1)-2(1)-10(1)-5(3)	131.00	8.50	8.75
MGA	2(2)-10(1)-9(1)-6(2)-4(2)-3(3)-7(2)-5(2)-1(3)	141.00	8.50	8.75
MGA	7(1)-8(1)-6(2)-1(1)-10(3)-5(2)-4(3)-2(3)-9(1)-3(2)	144.00	8.19	8.75
GA	2(2)-9(1)-3(3)-7(3)-8(1)-6(2)-1(3)-10(2)-5(2)-4(1)	190.00	8.73	8.75
MGA	5(2)-4(3)-2(1)-9(3)-8(1)-6(3)-1(1)-3(2)-7(1)-10(1)	151.00	8.25	8.25
MGA	2(2)-10(3)-9(1)-3(1)-7(1)-8(1)-6(2)-1(1)-5(2)-4(1)	165.00	7.99	8.00
MGA	7(1)-592)-4(1)-2(3)-9(2)-3(3)-8(1)-6(2)-1(2)-10(1)	240.00	7.25	7.25

Table 4.26: Input Data for FCSTSP (Model 4.3A1)

	Fuzzy Cost Matrix(10 ×10) With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	32,35,36 36,37,39 26,28,29	17,19,20 38,39,42 26,30,31	17,21,22 31,33,34 33,35,36	29,30,31 20,21,23 60,62,63	5,7,10 22,23,25 6,8,9	15,16,18 35,33,37 46,47,48	25,28,29 37,39,43 16,19,20	39,41,42 26,31,33 41,42,43	20,22,23 30,31,34 42,43,45
2	34,35,38 22,26,27 14,17,19	∞	40,41,44 18,21,22 27,32,33	16,18,19 28,29,32 6,10,12	32,35,37 25,26,27 34,37,38	39,40,41 30,31,32 21,23,26	39,40,42 29,30,32 57,59,60	30,33,34 41,42,45 58,59,62	17,19,22 36,37,38 17,20,21	23,24,26 13,16,17 17,18,20
3	36,38,39 29,30,32 28,29,32	16,17,20 54,58,60 31,34,35	∞	10,12,13 24,25,26 12,14,17	40,42,45 23,25,26 45,46,48	33,35,36 34,36,39 33,34,35	17,19,20 11,11,12 5,8,10	30,32,33 30,33,34 24,25,27	28,30,31 18,19,21 40,41,44	29,30,31 19,22,23 32,33,35
4	27,28,30 18,20,21 9,10,12	9,10,11 19,22,23 12,14,15	16,18,20 7,9,10 27,29,30	∞	29,30,33 17,19,20 23,24,25	23,25,26 15,16,18 25,27,28	19,21,22 30,31,32 30,33,34	33,35,36 32,36,38 16,17,18	10,12,13 20,23,24 32,34,35	24,27,29 47,48,49 37,39,40
5	16,18,19 14,15,18 6,8,9	41,42,44 21,23,24 32,34,37	34,35,37 35,36,37 33,38,39	17,20,21 12,13,14 40,43,44	∞	29,30,31 20,21,23 40,41,42	42,45,46 14,16,18 25,27,27	27,30,31 30,31,32 12,13,16	18,19,22 8,10,11 7,8,9	26,28,29 25,26,27 25,27,28
6	13,15,16 5,6,8 5,7,8	26,29,30 20,21,23 27,29,30	4,4,6 25,26,27 27,28,30	6,8,9 7,9,11 10,12,13	26,28,29 26,29,30 38,39,41	∞	31,33,34 40,43,44 23,24,26	39,40,42 30,31,31 20,22,23	30,32,33 22,23,24 35,35,36	28,30,31 40,41,42 30,32,34

7	36,37,39 37,39,40 28,30,32	23,25,26 53,53,55 25,26,27	27,30,32 37,38,39 24,26,27	21,22,24 40,43,44 23,24,25	35,37,38 56,58,60 37,39,40	38,40,41 20,21,22 43,45,46	∞	7,10,11 40,43,44 11,13,14	31,33,34 33,34,35 34,36,37	19,20,22 13,15,16 25,26,28
8	39,41,42 41,42,43 20,23,24	24,26,28 5,6,7 16,17,18	30,32,33 52,53,54 43,45,46	38,40,42 19,21,22 40,42,43	34,35,37 34,36,37 46,47,48	23,25,26 15,16,18 4,5,6	39,40,42 19,21,22 41,43,44	∞	20,22,23 52,53,54 23,24,27	35,37,38 35,36,38 39,40,41
9	38,40,41 10,11,13 31,32,33	39,41,42 38,39,40 34,36,37	4,6,9 34,36,37 36,37,39	23,25,26 33,34,36 28,29,30	20,21,23 31,32,33 20,21,22	22,23,25 31,33,34 23,25,26	5,7,8 36,38,39 38,39,41	30,32,33 32,33,34, 11,13,15	∞	27,28,30 18,19,20 24,26,27
10	15,17,18 25,26,28 25,29,30	28,30,31 20,21,22 31,32,34	26,28,29 18,19,20 28,30,32	18,20,21 29,31,32 21,22,24	9,11,12 32,33,34 20,22,24	30,32,34 10,12,13 33,34,35	35,38,39 26,28,29 38,39,41	40,41,43 41,42,43 30,33,34	29,31,32 51,52,54 30,32,33	∞
	Fuzzy risk/discomfort Matrix(10×10) With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	.7,.65,.63 .69,.67,.66 .75,.72,.7	.85,.81,.78 .59,.57,.56 .65,.63,.6	.8,.78,.77 .7,.67,.65 .69,.71,.7	.75,.72,.71 .8,.81,.83 .37,.32,.29	.9,.87,.85 .78,.74,.72 .89,.84,.81	.85,.83,.8 .64,.61,.59 .51,.5,.47	.7,.67,.61 .6,.58,.54 .79,.75,.73	.54,.5,.47 .65,.6,.58 .54,.5,.48	.79,.75,.7 .61,.58,.54 .48,.42,.41
2	.58,.55,.5 .7,.66,.61 .8,.75,.71	∞	.56,.41,.47 .76,.71,.69 .68,.61,.59	.78,.77,.71 .67,.62,.6 .9,.85,.82	.65,.61,.59 .75,.68,.65 .6,.58,.5	.53,.5,.47 .68,.64,.61 .7,.65,.62	.59,.52,.48 .69,.63,.6 .31,.26,.2	.7,.63,.59 .51,.45,.4 .32,.34,.19	.75,.7,.68 .6,.57,.53 .7,.69,.62	.69,.64,.61 .8,.76,.71 .81,.76,.7
3	.55,.51,.48 .6,.56,.53 .61,.58,.56	.72,.69,.62 .38,.31,.26 .6,.58,.51	∞	.81,.76,.7 .71,.68,.66 .8,.76,.71	.51,.46,.4 .7,.64,.61 .48,.44,.4	.59,.55,.52 .61,.58,.56 .62,.6,.57	.8,.75,.71 .9,.86,.81 .89,.86,.81	.65,.6,.59 .64,.6,.58 .68,.65,.61	.58,.55,.51 .8,.76,.71 .55,.5,.48	.67,.61,.58 .76,.71,.68 .64,.6,.57
4	.69,.64,.62 .78,.75,.71 .85,.83,.8	.86,.81,.79 .76,.71,.69 .81,.78,.74	.79,.75,.72 .9,.85,.82 .7,.64,.6	∞	.65,.63,.6 .76,.72,.7 .78,.71,.69	.69,.65,.62 .78,.75,.71 .68,.67,.65	.78,.74,.71 .68,.65,.61 .6,.54,.5	.6,.56,.52 .59,.58,.56 .79,.76,.72	.85,.82,.8 .78,.74,.71 .71,.69,.64	.68,.63,.59 .5,.45,.41 .6,.54,.5
5	.8,.76,.71 .81,.79,.75 .88,.81,.79	.55,.52,.49 .75,.74,.72 .61,.58,.54	.6,.58,.4 .58,.55,.5 .59,.58,.54	.78,.75,.71 .65,.62,.61 .55,.51,.48	∞	.62,.58,.55 .81,.75,.72 .55,.51,.45	.51,.45,.41 .81,.78,.75 .71,.68,.66	.67,.62,.59 .66,.61,.58 .82,.79,.75	.8,.76,.7 .88,.81,.78 .9,.87,.81	.69,.66,.62 .7,.68,.65 .9,.87,.83
6	.8,.75,.71 .81,.79,.76 .88,.85,.81	.65,.63,.6 .75,.72,.7 .66,.61,.59	.85,.82,.78 .7,.68,.62 .65,.62,.6	.88,.84,.79 .87,.84,.8 .85,.81,.78	.7,.67,.63 .6,.58,.55 .58,.54,.49	∞	.64,.6,.58 .55,.51,.46 .7,.68,.65	.55,.52,.48 .65,.63,.6 .76,.71,.68	.68,.61,.58 .73,.7,.68 .62,.58,.55	.65,.61,.58 .55,.52,.48 .65,.62,.6
7	.58,.54,.49 .56,.52,.48 .65,.62,.58	.65,.63,.6 .44,.38,.33 .71,.65,.6	.64,.6,.58 .6,.58,.55 .67,.64,.6	.7,.68,.65 .55,.51,.45 .71,.68,.64	.56,.54,.51 .38,.32,.28 .55,.53,.51	.55,.51,.46 .75,.71,.68 .52,.47,.4	∞	.85,.81,.78 .55,.54,.51 .75,.76,.72	.65,.61,.59 .58,.54,.5 .65,.61,.58	.78,.74,.69 .71,.68,.64 .65,.62,.58
8	.56,.52,.49 .54,.52,.51 .5,.43,.4	.7,.68,.65 .9,.88,.84 .8,.81,.78	.64,.6,.58 .41,.38,.37 .51,.45,.4	.56,.52,.5 .76,.74,.7 .56,.52,.49	.62,.58,.53 .62,.57,.55 .52,.48,.45	.55,.52,.48 .8,.77,.7 .88,.83,.8	.55,.54,.51 .78,.72,.7 .54,.53,.5	∞	.78,.76,.73 .43,.4,.36 .73,.7,.68	.58,.56,.51 .6,.54,.5 .58,.54,.49
9	.56,.51,.48 .88,.85,.81 .68,.65,.51	.58,.52,.5 .59,.57,.56 .58,.55,.53	.9,.85,.82 .62,.61,.58 .6,.54,.5	.7,.68,.64 .74,.7,.67 .68,.52,.58	.78,.75,.71 .65,.61,.58 .74,.7,.68	.74,.7,.68 .64,.61,.59 .67,.64,.6	.85,.81,.8 .62,.6,.57 .58,.54,.49	.62,.6,.58 .65,.61,.6 .79,.75,.72	∞	.69,.65,.63 .78,.73,.7 .72,.7,.68
10	.78,.71,.69 .7,.67,.64 .69,.64,.6	.66,.61,.58 .77,.74,.7 .78,.76,.71	.69,.65,.62 .8,.76,.74 .68,.65,.63	.74,.7,.68 .65,.6,.57 .76,.71,.68	.83,.78,.75 .62,.58,.56 .75,.71,.66	.65,.61,.58 .87,.83,.78 .68,.64,.59	.59,.54,.5 .68,.64,.61 .59,.55,.51	.55,.52,.47 .52,.48,.54 .64,.6,.58	.64,.59,.58 .45,.41,.37 .61,.59,.58	∞

Table 4.27: Optimum Results of FCSTSP (Model 4.3A1)

Method	α	β	Algo.	DM	Path(Vehicle)	Obj Value	Fuzzy Cost	Risk Value	R_{max}
POS. NES.	0.95	0.8	MGA	ODM	10(1)-5(2)-9(2)-1(1)-6(1)-4(2)-3(3)-7(1)-8(2)-2(3)	99	80,100,115	8.72,8.31,7.93	9.25.,9,8.5
				PDM	10(1)-5(2)-9(2)-1(1)-6(1)-4(2)-3(3)-7(1)-8(2)-2(3)	114.35	80,100,115	8.72,8.31,7.93	9.25.,9,8.5
				ODM	5(3)-9(1)-3(3)-7(3)-8(2)-2(1)-10(2)-6(1)-4(3)-1(1)	126.15	110,127,142	8.33,8.08,7.73	9.25.,9,8.5
				PDM	7(3)-8(2)-4(3)-6(2)-1(2)-5(1)-9(3)-10(2)-3(2)-2(2)	139.3	126,140,154	8.21,7.97,7.65	9.25.,9,8.5
				ODM	8(3)-6(2)-1(1)-4(1)-2(2)-10(1)-5(3)-9(1)-3(3)-7(3)	103.25	84,104,118	8.42,8.09,7.74	8.75,8.5,8.25
				PDM	1(3)-5(3)-9(1)-3(3)-7(3)-8(2)-2(2)-10(2)-6(2)-4(3)	119.5	101,120,135	8.43,8.2,7.84	8.75,8.5,8.25
			GA	ODM	4(3)-1(1)-5(3)-9(1)-3(3)-7(3)-8(3)-2(2)-10(2)-6(3)	126.15	110,127,141	8.31,7.98,7.61	8.75,8.5,8.25
				PDM	5(3)-8(3)-2(1)-4(2)-9(1)-3(3)-7(2)-10(2)-6(2)-1(2)	138.1	121,139,156	8.16,7.93,7.59	8.75,8.5,8.25
	.8	.9	MGA	ODM	6(3)-4(3)-1(2)-5(3)-9(1)-3(2)-7(3)-8(3)-2(2)-10(2)	125.25	111,126,141	8.42,8.19,7.83	8.5.8,7.75
				PDM	10(2)-2(1)-4(3)-1(2)-5(3)-9(3)-8(2)-612)-3(3)-7(1)	138.35	126,139,156	8.21,7.96,7.54	8.5.8,7.75
	α	ω	Algo	DM	Path(Vehicle)	Obj Value	Fuzzy Cost	Risk Value	R_{max}
GMIV	-	0.5	MGA	-	10(1)-5(2)-9(2)-1(1)-6(1)-4(2)-3(3)-7(1)-8(2)-2(3)	99.16	80,100,115	8.72,8.31,7.93	9.25.,9,8.5
		0.25	MGA	-	10(1)-5(2)-9(2)-1(1)-6(1)-4(2)-3(3)-7(1)-8(2)-2(3)	96.25	80,100,115	8.72,8.31,7.93	9.25.,9,8.5
		0.5	MGA	-	5(3)-9(1)-3(3)-7(3)-8(2)-2(1)-10(2)-6(1)-4(3)-1(1)	126.66	110,127,142	8.33,8.08,7.73	9.25.,9,8.5
		.75	MGA	-	7(3)-8(2)-4(3)-6(2)-1(2)-5(1)-9(3)-10(2)-3(2)-2(2)	129.16	126,140,154	8.21,7.97,7.65	9.25.,9,8.5
Crede bility	0.6	-	MGA	-	5(2)-6(2)-1(1)-4(1)-2(2)-10(1)-8(1)-9(1)-3(3)-7(3)	126.5	97,116,12	8.42,8.09,7.74	8.75,8.5,8.25
	0.5	-	MGA	-	1(3)-5(3)-9(1)-3(3)-7(3)-8(2)-2(2)-10(2)-6(2)-4(3)	119.5	101,120,135	8.43,8.2,7.84	8.75,8.5,8.25
	0.6	-	GA	-	4(3)-1(1)-5(3)-9(1)-3(3)-7(3)-8(3)-2(2)-10(2)-6(3)	126.15	110,127,141	8.31,7.98,7.61	8.75,8.5,8.25
	0.5	-	GA	-	5(3)-8(3)-2(1)-4(2)-9(1)-3(3)-7(2)-10(2)-6(2)-1(2)	138.1	121,139,156	8.16,7.93,7.59	8.75,8.5,8.25
EVM	-	.5	MGA	-	6(3)-4(3)-1(2)-5(3)-9(1)-3(2)-7(3)-8(3)-2(2)-10(2)	126	111,126,141	8.42,8.19,7.83	8.5.8,7.75
					10(2)-2(1)-4(3)-1(2)-5(3)-9(3)-8(2)-612)-3(3)-7(1)	140	126,139,156	8.21,7.96,7.54	8.5.8,7.75

solved by MGA and the results are given in Table 4.29.

Model 4.3A3: RFCSTSP with risk/discomfort Constraint in Random-Fuzzy Environment:

Here the cost and risk/discomfort matrices are random-fuzzy values for the RFCSTSP. Also only three types of conveyances are available for transportation. Assume that mean, $\tilde{m}^c$ is a triangular fuzzy number. The random-fuzzy cost and risk/discomfort matrices for the RFCSTSP are given in Table 4.30, where first part is a TFN and second part is a variance.

Here we have take permissible probability levels $\hat{\theta}^{obj} = \hat{\theta}^{cst}$=0.94. We derive L(x)=1-x, the left and right spreads respectively are $\alpha^c = m^c - \hat{h}^{obj}$, $\beta^c = m^c - 2 * \hat{h}^{obj}$ and $\alpha^R = m^R - \hat{h}^{cst}$, $\beta^R = m^R - 2 * \hat{h}^{cst}$, $\alpha^r = m^r - \hat{h}^{cst}$, $\beta^r = m^r - 2 * \hat{h}^{cst}$ for the cost and risks. With these input data, RFCSTSP is solved by MGA and the results are given in Table 4.31.

Model 4.3A4: FRCSTSP with risk/discomfort Constraint in Fuzzy Random Environment:

Here the costs and risk/discomfort parameters are Fu-Ra values for the FRCSTSP. Also we consider three types of conveyances. The extended operations on the basis of min-max cannot be directly applied to fuzzy numbers with discrete supports, but fuzzy numbers in LR-representation are helpful for computational. Assume that the costs are LR-type Fu-Ra as ($\hat{c}$, α, β) where $\hat{c}$ is a normal variate and α, β are left and right spreads of the LR- fuzzy variables. Similarly for risk/discomforts are taken as LR-type Fu-Ra variables ($\hat{r}$, α, β) where $\hat{r}$ is a normal random variate and α, β are left and right spreads of the LR- fuzzy variables. The Fu-Ra cost matrix for the CSTSP and the corresponding fuzzy random risk/discomfort matrix are presented in Table 4.32.

In the Table 4.32, risk/discomfort data are LR-type fuzzy numbers presented in the tuple where mean also a normal variate N(m, σ). Probability levels γ = η = 0.9 and set L(x)=1-x, left and right spreads are also taken in the Table 4.32. For input data in Table 4.31, the FRCSTSP is solved by MGA and the results are presented in Table 4.33.

Model 4.3A5: BRCSTSP with risk/discomfort Constraint in Bi-random Environment:

Here the cost and risk/discomfort factors are in bi-random values for the BRC-

Table 4.28: Input Data for RaCSTSP (Model 4.3A2)

	Random Cost Matrix(10 ×10) for RCSTSP With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	(32,1.1) (37,1.21) (28,1.02)	(19,.9) (39,1.07) (30,1.11)	(21,1.02) (33,1.15) (35,1.17)	(30,1.01) (21,.98) (62,1.2)	(7,1.23) (23,1.02) (8,1.19)	(16,1.11) (36,1.03) (47,.97)	(28,1.04) (39,1.12) (19,1.18)	(41,1.12) (31,1.13) (42,1.03)	(21,1.02) (31,1.1) (43,1.01)
2	(35,1.12) (26,1.18) (17,1.13)	∞	(41,1.03) (21,1.17) (32,1.32)	(18,1.11) (29,1.12) (10,1.03)	(35,1.07) (26,1.2) (37,1.2)	(40,1.02) (31,1.2) (23,1.31)	(40,1.13) (30,1.15) (59,1.14)	(33,1.03) (42,1.21) (59,1.16)	(19,1.2) (37,1.13) (20,1.3)	(24,1.19) (16,1.12) (18,1.03)
3	(38,1.29) (30,1.13) (29,1.15)	(17,1.21) (58,1.43) (34,1.32)	∞	(12,1.25) (25,1.21) (14,1.11)	(42,1.23) (25,1.23) (46,1.24)	(35,1.21) (36,1.4) (34,1.12)	(19,1.13) (11,1.1) (8,1.3)	(32,1.1) (33,1.21) (25,1.16)	(30,1.11) (19,1.22) (41,1.41)	(30,1.21) (22,1.16) (33,1.33)
4	(28,1.14) (20,1.1) (10,1.31)	(10,1.2) (22,1.32) (14,1.2)	(18,1.21) (9,1.4) (29,1.31)	∞	(30,1.13) (19,1.15) (24,1.21)	(25,1.23) (16,1.12) (27,1.13)	(21,1.4) (31,1.4) (33,1.19)	(35,1.3) (36,1.2) (17,1.23)	(12,1.21) (23,1.31) (34,1.2)	(27,1.6) (48,1.2) (39,1.28)
5	(18,1.31) (15,1.2) (8,1.2)	(42,1.2) (23,1.31) (34,1.21)	(35,1.12) (36,1.41) (38,1.34)	(20,1.31) (13,1.31) (43,1.15)	∞	(30,1.21) (21,1.36) (41,1.5)	(45,1.16) (16,1.02) (27,1.31)	(30,1.24) (31,1.27) (13,1.02)	(19,1.34) (10,1.01) (8,1.04)	(28,1.42) (26,1.47) (27,1.21)
6	(15,1.31) (6,1.65) (7,1.27)	(29,1.15) (21,1.75) (29,1.15)	(4,1.32) (26,1.62) (28,1.72)	(8,1.41) (9,1.7) (12,1.04)	(28,1.61) (29,1.21) (39,1.37)	∞	(33,1.26) (42,1.31) (24,1.32)	(40,1.53) (31,1.32) (22,1.65)	(32,1.21) (23,1.34) (35,1.21)	(30,1.54) (41,1.52) (32,1.52)
7	(37,1.6) (39,1.43) (30,1.32)	(25,1.21) (53,1.6) (26,1.54)	(30,1.5) (38,1.71) (26,1.56)	(22,1.61) (43,1.31) (24,1.76)	(37,1.98) (58,1.21) (40,1.21)	(40,1.76) (21,1.65) (45,1.61)	∞	(10,1.31) (43,1.65) (13,1.21)	(33,1.54) (34,1.71) (36,1.37)	(20,1.04) (15,1.2) (26,1.6)
8	(41,1.27) (42,1.43) (23,1.15)	(26,1.43) (6,1.32) (17,1.23)	(32,1.34) (53,1.43) (45,1.17)	(40,1.21) (21,1.21) (42,1.31)	(35,1.53) (36,1.21) (47,1.32)	(25,1.53) (16,1.06) (5,1.03)	(40,1.27) (21,1.03) (43,1.04)	∞	(22,1.31) (53,1.62) (24,1.02)	(37,1.76) (36,1.78) (40,1.02)
9	(40,1.72) (11,1.21) (32,1.02)	(41,1.56) (39,1.56) (36,1.42)	(6,1.24) (36,1.42) (37,1.76)	(25,1.71) (34,1.57) (29,1.08)	(21,1.04) (32,1.3) (21,1.02)	(23,1.32) (33,1.06) (25,1.03)	(7,1.01) (38,1.02) (39,1.21)	(32,1.32) (33,1.76) (13,1.52)	∞	(28,1.41) (19,1.32) (26,1.72)
10	(17,1.51) (26,1.01) (29,1.21)	(30,1.31) (21,1.04) (32,1.92)	(28,1.15) (19,1.21) (30,1.72)	(20,1.72) (31,1.02) (22,1.51)	(11,1.82) (33,1.27) (22,1.19)	(32,1.52) (12,1.18) (34,1.17)	(38,1.02) (28,1.13) (39,1.16)	(41,1.62) (42,1.81) (33,1.21)	(31,1.52) (52,1.37) (32,1.15)	∞
	Random risk/discomfort Matrix(10 ×10) for RCSTSP With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	(.62,1.1) (.54,1.21) (.28,1.02)	(.75,.9) (.53,1.07) (.64,1.11)	(.7,1.02) (.61,1.15) (.59,1.17)	(.66,1.01) (.78,.98) (.34,1.2)	(.87,1.23) (.71,1.02) (.88,1.19)	(.8,1.11) (.58,1.03) (.49,.97)	(.68,1.04) (.52,1.12) (.76,1.18)	(.5,1.12) (.64,1.13) (.55,1.03)	(.74,1.02) (.63,1.1) (.52,1.01)
2	(.6,1.12) (.65,1.18) (.79,1.13)	∞	(.54,1.03) (.74,1.17) (.63,1.32)	(.77,1.11) (.62,1.12) (.85,1.03)	(.6,1.07) (.68,1.2) (.58,1.2)	(.55,1.02) (.64,1.2) (.7,1.31)	(.54,1.13) (.66,1.15) (.35,1.14)	(.62,1.03) (.53,1.21) (.32,1.16)	(.76,1.2) (.58,1.13) (.73,1.3)	(.71,1.19) (.78,1.12) (.74,1.03)
3	(.58,1.29) (.64,1.13) (.66,1.15)	(.77,1.21) (.35,1.43) (.62,1.32)	∞	(.79,1.25) (.7,1.21) (.81,1.11)	(.54,1.23) (.745,1.23) (.49,1.24)	(.59,1.21) (.59,1.4) (.62,1.12)	(.76,1.13) (.85,1.1) (.86,1.3)	(.62,1.1) (.61,1.21) (.7,1.16)	(.66.11) (.76,1.22) (.52,1.41)	(.61,1.21) (.72,1.16) (.62,1.33)
4	(.65,1.14) (.76,1.1) (.84,1.31)	(.86,1.2) (.73,1.32) (.79,1.2)	(.78,1.21) (.9,1.4) (.65,1.31)	∞	(.66,1.13) (.79,1.15) (.71,1.21)	(.7,1.23) (.77,1.12) (.7,1.13)	(.77,1.4) (.63,1.4) (.63,1.19)	(.69,1.3) (.6,1.2) (.77,1.23)	(.82,1.21) (.71,1.31) (.59,1.2)	(.69,1.6) (.47,1.2) (.54,1.28)
5	(.8,1.31) (.8,1.2) (.88,1.2)	(.54,1.2) (.69,1.31) (.6,1.21)	(.6,1.12) (.6,1.41) (.56,1.34)	(.75,1.31) (.82,1.31) (.51,1.15)	∞	(.65,1.21) (.76,1.36) (.54,1.5)	(.5,1.16) (.8,1.02) (.68,1.31)	(.63,1.24) (.64,1.27) (.8,1.02)	(.76,1.34) (.84,1.01) (.86,1.04)	(.68,1.42) (.48,1.47) (.64,1.21)
6	(.8,1.31) (.89,1.65) (.85,1.27)	(.69,1.15) (.79,1.75) (.7,1.15)	(.89,1.32) (.76,1.62) (.65,1.72)	(.85,1.41) (.88,1.7) (.8,1.04)	(.7,1.61) (.68,1.21) (.53,1.37)	∞	(.63,1.26) (.55,1.31) (.73,1.32)	(.55,1.53) (.67,1.32) (.74,1.65)	(.63,1.21) (.72,1.34) (.7,1.21)	(.65,1.54) (.52,1.52) (.61,1.52)
7	(.55,1.6) (.57,1.43) (.66,1.32)	(.7,1.21) (.42,1.6) (.7,1.54)	(.67,1.5) (.59,1.71) (.71,1.56)	(.72,1.61) (.52,1.31) (.69,1.76)	(.62,1.98) (.37,1.21) (.54,1.21)	(.54,1.76) (.76,1.65) (.5,1.61)	∞	(.84,1.31) (.58,1.65) (.82,1.21)	(.62,1.54) (.62,1.71) (.6,1.37)	(.84,1.04) (.79,1.2) (.68,1.6)
8	(.55,1.23) (.55,1.43) (.72,1.15)	(.7,1.43) (.78,1.32) (.77,1.02)	(.65,1.34) (.42,1.43) (.5,1.32)	(.58,1.21) (.74,1.21) (.54,1.03)	(.59,1.53) (.6,1.21) (.48,1.05)	(.68,1.53) (.76,1.06) (.88,1.31)	(.57,1.27) (.72,1.03) (.52,1.38)	∞	(.72,1.31) (.44,1.62) (.61,1.73)	(.58,1.76) (.6,1.78) (.57,1.28)
9	(.54,1.72) (.84,1.21) (.57,1.02)	(.51,1.56) (.56,1.56) (.59,1.42)	(.88,1.24) (.6,1.42) (.6,1.76)	(.7,1.71) (.61,1.57) (.67,1.08)	(.72,1.04) (.67,1.3) (.75,1.02)	(.71,1.32) (.61,1.06) (.74,1.03)	(.87,1.01) (.62,1.02) (.58,1.21)	(.7,1.32) (.63,1.76) (.82,1.52)	∞	(.68,1.41) (.74,1.32) (.7,1.72)
10	(.8,1.51) (.7,1.01) (.64,1.21)	(.68,1.31) (.74,1.04) (.65,1.92)	(.69,1.15) (.55,1.21) (.66,1.72)	(.76,1.72) (.64,1.02) (.73,1.51)	(.8,1.82) (.61,1.27) (.74,1.19)	(.61,1.52) (.8,1.18) (.54,1.17)	(.58,1.02) (.68,1.13) (.58,1.16)	(.56,1.62) (.55,1.81) (.57,1.21)	(.63,1.52) (.42,1.37) (.6,1.15)	∞

Table 4.29: Results of RaCSTSP (Model 4.3A2)

K1	K2	Algorithm	Path(Vehicle)	Costs	R_{max}
0.5	0.5	MGA	10(1)-5(2)-9(2)-1(1)-6(1)-4(2)-3(3)-7(1)-8(2)-2(3)	50.80	8.5
		MGA	5(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-6(2)-1(3)-7(3)	54.32	8.5
		MGA	7(3)-4(2)-1(3)-5(1)-6(1)-2(2)-3(3)-10(1)-8(2)-9(3)	56.60	8.5
0.5	0.5	GA	5(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-6(2)-1(3)-7(3)	64.32	8.5
0.4	0.6	MGA	10(1)-5(2)-9(2)-1(1)-6(1)-4(2)-3(3)-7(1)-8(2)-2(3)	41.36	8.5.
0.6	0.4	MGA	10(1)-5(2)-9(2)-1(1)-6(1)-4(2)-3(3)-7(1)-8(2)-2(3)	60.24	8.5

Table 4.30: Input Data for RFCSTSP (Model 4.3A3)

	Random-Fuzzy Cost Matrix(10×10) for RFCSTSP With Three Conveyances				
i/j	1	2	3	4	5
1	∞	[(32,35,36),1.21] [(36,37,39),1.21] [(26,28,29),1.08]	[(17,19,20),.98] [(38,39,42),1.32] [(26,30,31),1.03]	[(17,21,22),1.76] [(31,33,34),1.16] [(33,35,36),1.23]	[(29,30,31),1.13] [(20,21,23),1.13] [(60,62,63),1.05]
2	[(34,35,38),1.34] [(22,26,27),1.12] [(14,17,19),1.54]	∞	[(40,41,44),1.42] [(18,21,22),1.14] [(27,32,33),1.36]	[(16,18,19),1.13] [(28,29,32),1.17] [(6,10,12),1.12]	[(32,35,37),1.45] [25,26,27),1.18] [(34,37,38),1.4]
3	[(36,38,39),1.18] [(29,30,32),1.41] [(28,29,32),1.72]	[(16,17,20),1.43] [(54,58,60),1.31] [(31,34,35),1.32]	∞	[(10,12,13),1.17] [(24,25,26),1.17] [(12,14,17),1.03]	[(40,42,45)1.54] [(23,25,26),1.02] [(45,46,48),1.13]
4	[(27,28,30),1.42] [(18,20,21),1.46] [(9,10,12),1.14]	[(9,10,11),1.17] [(19,22,23),1.32] [(12,14,15),1.17]	[(16,18,20),1.18] [(7,9,10),1.62] [(27,29,30),1.14]	∞	[(29,30,33),.9] [(17,19,20),1.54] [(23,24,25),1.76]
5	[(16,18,19,1.17] [(14,15,18),1.3] [(6,8,9),1.3]	[(41,42,44),1.17] [(21,23,24),1.3] [(32,34,37),1.3]	[(34,35,37),1.14] [(35,36,37),1.3] [(33,38,39),1.3]	[(17,20,21)1.2] [(12,13,14),1.38] [(40,43,44),1.16]	∞
6	[(13,15,16),1.3] [(5,6,8),1.3] [(5,7,8),1.3]	[(26,29,30),1.54] [(20,21,23),1.17] [(27,29,30),1.3]	[(4,4,6),1.17] [(25,26,27),1.41] [(27,28,30),1.3]	[(6,8,9),1.3]),1.13] [(7,9,11),1.2] [(10,12,13),1.24]	[(26,28,29),1.34] [(26,29,30),1.73] [(38,39,41),1.3]
7	[(36,37,39),1.71] [(37,39,40),1.43] [(28,30,32),1.43]	[(23,25,26),1.16] [(53,53,55),1.13] [(25,26,27),1.31]	[(27,30,32),1.3] [(37,38,39),1.3] [(24,26,27),.98]	[(21,22,24),1.3] [(40,43,44),1.17] [(23,24,25),1.3]	[(35,37,38),1.43] [(56,58,60),1.3] [(37,39,40),1.23]
8	[(39,41,42),1.37] [(41,42,43),1.14] [(20,23,24),1.46]	[(24,26,28),1.43] [(5,6,7),1.33] [(16,17,18),1.23]	[(30,32,33),1.54] [(52,53,54),1.22] [(43,45,46),1.79]	[(38,40,42),1.27] [(19,21,22),1.3] [(40,42,43),1.3]	[(34,35,37),1.3] [(34,36,37),1.25] [(46,47,48),1.3]
9	[(38,40,41),1.41] [(10,11,13),1.02] [(31,32,33),1.37]	[(39,41,42),1.21] [(38,39,40),1.28] [(34,36,37),1.11]	[(4,6,9),1.16] [(34,36,37),1.45] [(36,37,39),1.19]	[(23,25,26),1.3] [(33,34,36),1.3] [(28,29,30),1.3]	[(20,21,23),1.3] [(31,32,33),1.41] [(20,21,22),1.3]
10	[(15,17,18),1.12] [(25,26,28),1.13] [(25,29,30),1.2]	[(28,30,31),1.34] [(20,21,22),1.33] [(31,32,34),1.63]	[(26,28,29),1.32] [(18,19,20),1.23] [(28,30,32),1.13]	[(18,20,21),1.3] [(29,31,32),1.43] [(21,22,24),1.53]	[(9,11,12),1.47] [(32,33,34),1.63] [(20,22,24),1.37]
	Random-Fuzzy Cost Matrix(10 ×10) for RFCSTSP With Three Conveyances				
i/j	6	7	8	9	10
1	[(5,7,10),1.32] [(22,23,25),1.16] [(6,8,9),1.06]	[(15,16,18),.99) [(35,33,37),1.14] [(46,47,48),1.23]	[(25,28,29),1.1] [(37,39,43),1.11] [(16,19,20),1.9]	[(39,41,42),1.13] [(26,31,33),1.15] [(41,42,43)1.22]	[(20,22,23),1.12] [(30,31,34),1.09] [(42,43,45),1.41]
2	[(39,40,41),1.2] [(30,31,32),1.34] [(21,23,26),1.76]	[(39,40,42),1.67] [(29,30,32),1.32] [(57,59,60),1.33]	[(30,33,34),1.13] [(41,42,45),1.41] [(58,59,62),1.72]	[(17,19,22),1.16] [(36,37,38),1.3] [(17,20,21),1.8]	[(23,24,26),1.14] [(13,16,17),1.17] [(17,18,20),1.17]

3	[(33,35,36),1.13] [(34,36,39),1.13] [(33,34,35),1.5]	[(17,19,20),1.15] [(11,11,12),1.17] [(5,8,10),1.14]	[(30,32,33),1.98] [(30,33,34),1.07] [(24,25,27),1.53]	[(28,30,31),1.09] [(18,19,21),1.73] [(40,41,44),1.72]	[(29,30,31),1.31] [(19,22,23),1.32] [(32,33,35),1.36]
4	[(23,25,26),1.3] [(15,16,18),1.43] [(25,27,28),1.9]	[(19,21,22),.78] [(30,31,32),1.52] [(30,33,34),1.31]	[(33,35,36),1.7] [(32,36,38),1.15] [(16,17,18),1.7]	[(10,12,13),1.6] [(20,23,24),1.76] [(32,34,35),1.45]	[(24,27,29),1.65] [(47,48,49),1.17] [(37,39,40),1.76]
5	[(29,30,31),1.26] [(20,21,23),1.3] [(40,41,42),1.15]	[(42,45,46),1.23] [(14,16,18),1.3] [(25,27,27),1.54]	[(27,30,31),1.18] [(30,31,32),1.3] [(12,13,16),1.71]	[(18,19,22),1.3] [(8,10,11),1.3] [(7,8,9),1.3]	[(26,28,29),1.51] [(25,26,27),1.3] [(25,27,28),1.3]
6	∞	[(31,33,34),1.21] [(40,43,44),1.3] [(23,24,26),1.3]	[(39,40,42),1.3] [(30,31,31),1.3] [(20,22,23),1.3]	[(30,32,33),1.3] [(22,23,24),1.3] [(35,35,36),1.28]	[(28,30,31),1.3] [(40,41,42),1.47] [(30,32,34),1.3]
7	[(38,40,41),1.14] [(20,21,22),1.16] [(43,45,46),1.24]	∞	[(7,10,11),1.3] [(40,43,44),1.3] [(11,13,14),1.3]	[(31,33,34),1.3] [(33,34,35),1.45] [(34,36,37),1.3]	[(19,20,22),1.46] [(13,15,16),1.3] [(25,26,28),1.3]
8	[(23,25,26),1.3] [(15,16,18),1.3] [(4,5,6),1.3]	[(39,40,42),1.3] [(19,21,22),1.04] [(41,43,4),1.12]	∞	[(20,22,23),1.67] [(52,53,54),1.61] [(23,24,27),1.3]	[(35,37,38),1.3] [(35,36,38),1.3] [(39,40,41),1.15]
9	[(22,23,25),1.3] [(31,33,34),1.68] [(23,25,26),1.3]	[(5,7,8),1.17] [(36,38,39),1.3] [(38,39,41),1.3]	[(30,32,33),1.7] [(32,33,34),1.27] [(11,13,15),1.3]	∞	[(27,28,30),1.04] [(18,19,20),1.3] [(24,26,27),1.3]
10	[(30,32,34),1.49] [(10,12,13),1.41] [(33,34,35),1.57]	[(35,38,39),1.3] [(26,28,29),1.8] [(38,39,41),1.17]	[(40,41,43),1.23] [(41,42,43),1.3] [(30,33,34),1.15]	[(29,31,32),1.25] [(51,52,54),1.3] [(30,32,33),1.2]	∞
	Random-Fuzzy risk/discomfort Matrix(10×10) for RFCSTSP With Three Conveyances				
i/j	1	2	3	4	5
1	∞	[(.69,.65,.61),1.12] [(.36,.37,.39),1.21] [(.26,.28,.29),1.08]	[(.72,.7,.68),.1.13 [(.38,.39,.42),1.32] [(.26,.30,.31),1.03]	[(.73,.71,.62),1.76] [(.31,.33,.34),1.16] [(.33,.35,.36),1.23]	[(.61,.30,.31),1.13] [(.20,.21,.23),1.13] [(.60,.62,.63),1.05]
2	[(.34,.35,.38),1.34] [(.22,.26,.27),1.12] [(.14,.17,.19),1.54]	∞	[(.40,.41,.44),1.42] [(.18,.21,.22),1.14] [(.27,.32,.33),1.36]	[(.16,.18,.19),1.13] [(.28,.29,.32),1.17] [(.6,.10,.12),1.12]	[(.32,.35,.37),1.45] [.25,.26,.27),1.18] [(.34,.37,.38),1.4]
3	[(.36,.38,.39),1.18] [(.29,.30,.32),1.41] [(.28,.29,.32),1.72]	[(.16,.17,.20),1.43] [(.54,.58,.61),1.31] [(.31,.34,.35),1.32]	∞	[(.10,.12,.13),1.17] [(.24,.25,.26),1.17] [(.12,.14,.17),1.03]	[(.40,.42,.45)1.54] [(.23,.25,.26),1.02] [(.45,.46,.48),1.13]
4	[(.27,.28,.30),1.42] [(.18,.20,.21),1.46] [(.9,.10,.12),1.14]	[(.9,.10,.11),1.17] [(.19,.22,.23),1.32] [(.12,.14,.15),1.17]	[(.16,.18,.20),1.18] [(.7,.9,.10),1.62] [(.27,.29,.30),1.14]	∞	[(.29,.30,.33),.9] [(.17,.19,.20),1.54] [(.23,.24,.25),1.76]
5	[(.16,.18,.19,1.17] [(14.,.15,.18),1.3] [(.6,.8,.9),1.3]	[(.41,.42,.44),1.17] [(.21,.23,.24),1.3] [(.32,.34,.37),1.3]	[(.34,.35,.37),1.14] [(.35,.36,.37),1.3] [(.33,.38,.39),1.3]	[(.17,.20,..21)1.2] [(.12,.13,.14),1.38] [(.40,.43,.44),1.16]	∞
6	[(.13,.15,.16),1.3] [(.5,.6,.8),1.3] [(.5,.7,.8),1.3]	[(.26,.29,.30),1.54] [(.20,.21,.23),1.17] [(.27,.29,.30),1.3]	[(.4,.4,.6),1.17] [(.25,.26,.27),1.41] [(.27,.28,.30),1.3]	[(.6,.8,.9),1.3]),1.13] [(.7,.9,.11),1.2] [(.10,.12,.13),1.24]	[(.26,.28,.29),1.34] [(.26,.29,.30),1.73] [(.38,.39,.41),1.3]
7	[(.36,.37,.39),1.71] [(.37,.39,.40),1.43] [(.28,.3,.32),1.43]	[(.23,.25,.26),1.16] [(.53,.53,.55),1.13] [(.25,.26,.27),1.31]	[(.27,.30,.32),1.3] [(.37,.38,.39),1.3] [(.24,.26,.27),.98]	[(.21,.22,.24),1.3] [(.4,.43,.44),1.17] [(.23,.24,.25),1.3]	[(.35,.37,.38),1.43] [(.56,.58,.60),1.3] [(.37,.39,.40),1.23]
8	[(.39,.41,.42),1.37] [(.41,.42,.43),1.14] [(.2,.23,.24),1.46]	[(.24,.26,.28),1.43] [(.5,.6,.7),1.33] [(.16,.17,.18),1.23]	[(.30,.32,.33),1.54] [(.52,.53,.54),1.22] [(.43,.45,.46),1.79]	[(.38,.40,.42),1.27] [(.19,.21,.22),1.3] [(.4,.42,.43),1.3]	[(.34,.35,.37),1.3] [(.34,.36,.37),1.25] [(.46,.47,.48),1.3]
9	[(.38,.40,.41),1.41] [(.1,.11,.13),1.02] [(.31,.32,.33),1.37]	[(.39,.41,.42),1.21] [(.38,.39,.4),1.28] [(.34,.36,.37),1.11]	[(.4,.6,.9),1.16] [(.34,.36,.37),1.45] [(.36,.37,.39),1.19]	[(.23,.25,.26),1.3] [(.33,.34,.36),1.3] [(.28,.29,.30),1.3]	[(.20,.21,.23),1.3] [(.31,.32,.33),1.41] [(.2,.21,.22),1.3]
10	[(.15,.17,.18),1.12] [(.25,.26,.28),1.13] [(.25,.29,.30),1.2]	[(.28,.30,.31),1.34] [(.2,.21,.22),1.33] [(.31,.32,.34),1.63]	[(.26,.28,.29),1.32] [(.18,.19,.20),1.23] [(.28,.30,.32),1.13]	[(.18,.20,.21),1.3] [(.29,.31,.32),1.43] [(.21,.22,.24),1.53]	[(.9,.11,.12),1.47] [(.32,.33,.34),1.63] [(.20,.22,.24),1.37]
	Random-Fuzzy risk/discomfort Matrix(10 ×10) for RCSTSP With Three Conveyances				
i/j	6	7	8	9	10
1	[(.5,.7,.10),1.32] [(.22,.23,.25),1.16] [(.6,.8,.9),1.06]	[(.15,.16,.18),.99) [(.35,.33,.37),1.14] [(.46,.47,.48),1.23]	[(.25,.28,.29),1.1] [(.37,.39,.43),1.11] [(.16,.19,.20),1.9]	[(.39,.41,.42),1.13] [(.26,.31,.33),1.15] [(.41,.42,.43)1.22]	[(.20,.22,.23),1.12] [(.30,.31,.34),1.09] [(.42,.43,.45),1.41]
2	[(.39,.40,.41),1.2] [(.30,.31,.32),1.34] [(.21,.23,.26),1.76]	[(.39,.40,.42),1.67] [(.29,.30,.32),1.32] [(.57,.59,.60),1.33]	[(.30,.33,.34),1.13] [(.41,.42,.45),1.41] [(.58,.59,.62),1.72]	[(.17,.19,.22),1.16] [(.36,.37,.38),1.3] [(.17,.20,.21),1.8]	[(.23,.24,.26),1.14] [(.13,.16,.17),1.17] [(.17,.18,.20),1.17]
3	[(.33,.35,.36),1.13] [(.34,.36,.39),1.13] [(.33,.34,.35),1.5]	[(.17,.19,.20),1.15] [(.11,.11,.12),1.17] [(.05,.08,.1),1.14]	[(.30,.32,.33),1.98] [(.3,.33,.34),1.07] [(.24,.25,.27),1.53]	[(.28,.30,.31),1.09] [(.18,.19,.21),1.73] [(.4,.41,.44),1.72]	[(.29,.30,.31),1.31] [(.19,.22,.23),1.32] [(.32,.33,.35),1.36]
4	[(.23,.25,.26),1.3] [(.15,.16,.18),1.43] [(.25,.27,.28),1.9]	[(.19,.21,.22),.78] [(.30,.31,.32),1.52] [(.3,.33,.34),1.31]	[(.33,.35,.36),1.7] [(.32,.36,.38),1.15] [(.16,.17,.18),1.7]	[(.1,.12,.13),1.6] [(.2,.23,.24),1.76] [(.32,.34,.35),1.45]	[(.24,.27,.29),1.65] [(.47,.48,.49),1.17] [(.37,.39,.4),1.76]
5	[(.29,.3,.31),1.26] [(.2,.21,.23),1.3] [(.4,.41,.42),1.15]	[(.42,.45,.46),1.23] [(.14,.16,.18),1.3] [(.25,.27,.27),1.54]	[(.27,.3,.31),1.18] [(.3,.31,.32),1.3] [(.12,.13,.16),1.71]	[(.18,.19,.22),1.3] [(.08,.1,.11),1.3] [(.07,.08,.09),1.3]	[(.26,.28,.29),1.51] [(.25,.26,.27),1.3] [(.25,.27,.28),1.2]

6	∞	[(.31,.33,.34),1.21] [(.4,.43,.44),1.3] [(.23,.24,.26),1.3]	[(.39,.40,.42),1.3] [(.3,.31,.31),1.3] [(.2,.22,.23),1.3]	[(.3,.32,.33),1.3] [(.22,.23,.24),1.3] [(.35,.35,.36),1.28]	[(.28,.3,.31),1.3] [(.4,.41,.42),1.47] [(.3,.32,.34),1.3]
7	[(.38,.4,.41),1.14] [(.2,.21,.22),1.16] [(.43,.45,.46),1.24]	∞	[(.07,.1,.11),1.3] [(.4,.43,.44),1.3] [(.11,.13,.14),1.3]	[(.31,.33,.34),1.3] [(.33,.34,.35),1.45] [(.34,.36,.37),1.3]	[(.19,.2,.22),1.46] [(.13,.15,.16),1.3] [(.25,.26,.28),1.3]
8	[(.23,.25,.26),1.3] [(.15,.16,.18),1.3] [(.04,.05,.06),1.3]	[(.39,.4,.42),1.3] [(.19,.21,.22),1.04] [(.41,.43,.4),1.12]	∞	[(.2,.22,.23),1.67] [(.52,.53,.54),1.61] [(.23,.24,.27),1.3]	[(.35,.37,.38),1.3] [(.35,.36,.38),1.43] [(.39,.4,.41),1.15]
9	[(.22,.23,.25),1.3] [(.31,.33,.34),1.68] [(.23,.25,.26),1.3]	[(.05,.07,.08),1.17] [(.36,.38,.39),1.3] [(.38,.39,.41),1.3]	[(.3,.32,.33),1.7] [(.32,.33,.34),1.27] [(.11,.13,.15),1.3]	∞	[(.27,.28,.3),1.04] [(.18,.19,.2),1.3] [(.24,.26,.27),1.3]
10	[(.3,.35,.38),1.49] [(.1,.12,.13),1.41] [(.33,.34,.35),1.57]	[(.35,.38,.39),1.3] [(.26,.28,.29),1.8] [(.38,.39,.41),1.17]	[(.4,.41,.48),1.23] [(.41,.42,.43),1.3] [(.3,.33,.34),1.15]	[(.29,.31,.32),1.25] [(.51,.52,.54),1.47] [(.3,.32,.33),1.2]	∞

Table 4.31: Results of RFCSTSP (Model 4.3A3)

$\hat{h}^{obj}$	$\hat{h}^{cst}$	Algorithm	DM	Path(Vehicle)	Costs	R_{max}
0.95	0.95	MGA	PDM	3(1)-10(3)-2(3)-9(3)-7(3)-8(1)-6(2)-1(1)-5(2)-4(3)	152.68	8.5
			ODM	3(1)-10(3)-2(3)-9(3)-7(3)-8(1)-6(2)-1(1)-5(2)-4(3)	144.23	8.5
		MGA	PDM	5(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-6(2)-1(3)-7(3)	156.52	8.5
			ODM	5(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-6(2)-1(3)-7(3)	146.52	8.5
		MGA	PDM	10(2)-1(1)-9(2)-5(1)-7(2)-3(1)-4(2)-2(3)-8(1)-6(1)	156.61	6.75
			ODM	10(2)-1(1)-9(2)-5(1)-7(2)-3(1)-4(2)-2(3)-8(1)-6(1)	148.34	6.75
		GA	PDM	6(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-5(2)-1(3)-7(3)	172.21	6.0
			ODM	6(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-5(2)-1(3)-7(3)	162.45	6.0
0.95	0.7	MGA	PDM	6(2)-8(1)-7(1)-1(1)-10(2)-5(3)-2(3)-3(1)-4(2)-9(3)	145.29	6.75
			ODM	6(2)-8(1)-7(1)-1(1)-10(2)-5(3)-2(3)-3(1)-4(2)-9(3)	141.78	6.75
0.7	0.95	MGA	PDM	4(3)-7(1)-8(1)-6(2)-1(2)-5(1)-9(2)-2(1)-10(1)-3(1)	164.96	6.5
			ODM	4(3)-7(1)-8(1)-6(2)-1(2)-5(1)-9(2)-2(1)-10(1)-3(1)	154.13	6.5
0.8	0.75	MGA	PDM	3(1)-4(3)-2(3)-10(2)-5(3)-9(3)-8(1)-6(1)-1(3)-7(2)	151.21	6.0
			ODM	3(1)-4(3)-2(3)-10(2)-5(3)-9(3)-8(1)-6(1)-1(3)-7(2)	147.36	6.0

Table 4.32: Input Data: FRCSTSP (Model 4.3A4)

	Fuzzy Random Cost Matrix(10×10) for FRCSTSP With Three Conveyances				
i/j	1	2	3	4	5
1	∞	[(c,5,6),c∼ N(35,1)] [(c,3,3),c∼N(36,2)] [(c,2,2),c∼N(26,2)]	[(c,1,2),c∼N(17,2)] [(c,3,2),c∼N(38,2)] [(c,3,3),c∼N(26,2)]	[(c,1,2),c∼N(16,3)] [(c,3,4),c∼N(31,2)] [(c,5,6),c∼N(33,2)]	[(c,3,3),c∼N(29,2)] [(c,1,2),c∼N(20,2)] [(c,2,3),c∼N(60,4)]
2	[(c,3,3),c∼N(31,1)] [(c,2,2),c∼N(22,3)] [(c,1,1),c∼N(14,4)]	∞	[(c,4,4),c∼N(40,2)] [(c,2,2),c∼N(18,2)] [(c,2,3),c∼N(27,1)]	[(c,1,1),c∼N(16,2)] [(c,2,3),c∼N(28,2)] [(c,1,1),c∼N(6,3)]	[(c,3,3),c∼N(32,2)] [(c,2,2),c∼N(25,2)] [(c,3,4),c∼N(38,1)]
3	[(c,3,3),c∼N(36,2)] [(c,3,3),c∼N(26,3)] [(c,2,3),c∼N(28,2)]	[(c,1,2),c∼N(16,3)] [(c,5,6),c∼N(54,1)] [(c,3,3),c∼N(31,2)]	∞	[(c,1,1),c∼N(10,2)] [(c,2,2),c∼N(24,1)] [(c,1,1),c∼N(12,1)]	[(c,4,4)c∼N(40,1)] [(c,2,2),c∼N(26,3)] [(c,4,4),c∼N(45,5]
4	[(c,2,3),c∼N(26,1)] [(c,2,2),c∼N(18,2)] [(c,1,1),c∼N(9,2)]	[(c,1,1),c∼N(9,2)] [(c,2,2),c∼N(19,1)] [(c,1,1),c∼N(12,4)]	[(c,1,2),c∼N(16,4)] [(c,5,4),c∼N(7,2)] [(c,2,3),c∼N(27,2)]	∞	[(c,3,3),c∼N(29,3)] [(c,1,2),c∼N(17,3)] [(c,2,2),c∼N(23,4)]

5	[(c,1,1,c∼N(16,1)] [(c,1,1),c∼N(14,1)] [(c,1,1),c∼N(6,2)]	[(c,4,4),c∼N(41,3)] [(c,2,2),c∼N(22,2)] [(c,3,3),c∼N(32,3)]	[(c,3,3),c∼N(34,2)] [(c,6,7),c∼N(36,2)] [(c,3,3),c∼N(33,2)]	[(c,2,2),c∼N(17,2)] [(c,1,1),c∼N(12,2)] [(c,4,4),c∼N(38,3)]	∞
6	[(c,1,1),c∼N(15,1)] [(c,1,4),c∼N(6,2)] [(c,2,3),c∼N(6,2)]	[(c,2,3),c∼N(26,2)] [(c,2,2),c∼N(20,1)] [(c,2,3),c∼N(26,2)]	[(c,1,1),c∼N(5,5)] [(c,6,7),c∼N(36,2)] [(c,2,3),c∼N(26,3)]	[(c,2,3),c∼N(6,2)])] [c,1,1),c∼N(13,3)] [(c,1,1),c∼N(10,3)]	[(c,2,2),c∼N(26,1)] [(c,2,3),c∼N(26,2)] [(c,3,4),c∼N(38,4)]
7	[(c,3,3),c∼N(36,1)] [(c,3,4),c∼N(37,2)] [(c,3,3),c∼N(26,1)]	[(c,2,2),c∼N(36,3)] [(c,3,5),c∼N(53,4)] [(c,2,2),c∼N(25,2)]	[(c,3,3),c∼N(27,1)] [(c,3,3),c∼N(37,2)] [(c,2,2),c∼N(26,3)]	[(c,2,4),c∼N(20,2)] [(c,4,4),c∼N(43,1)] [(c,2,2),c∼N(20,1)]	[(c,3,3),c∼N(35,2)] [(c,5,4),c∼N(56,2)] [(c,3,4),c∼N(39,3)]
8	[(c,4,4),c∼N(39,2)] [(c,4,3),c∼N(41,2)] [(c,2,2),c∼N(20,2)]	[(c,2,2),c∼N(24,1)] [(c,1,1),c∼N(6,4)] [(c,1,1),c∼N(16,3)]	[(c,3,3),c∼N(30,3)] [(c,3,4),c∼N(53,1)] [(c,4,4),c∼N(40,2)]	[(c,4,4),c∼N(38,2)] [(c,2,2),c∼N(20,3)] [(c,4,3),c∼N(40,2)]	[(c,3,3),c∼N(34,3)] [(c,6,3),c∼N(32,2)] [(c,1,2),c∼N(43,1)]
9	[c,4,1),c∼N(38,2)] [(c,1,1),c∼N(10,4)] [(c,3,3),c∼N(31,2)]	[(c,4,4),c∼N(39,3)] [(c,3,4),c∼N(38,3)] [(c,3,3),c∼N(34,1)]	[(c,1,2),c∼N(4,2)] [(,3,3),c∼N(34,5)] [(c,3,3),c∼N(36,1)]	[(c,2,2),c∼N(23,2)] [(c,3,3),c∼N(33,3)] [(c,2,3),c∼N(28,1)]	[(c,1,3),c∼N(20,6)] [(c,3,3),c∼N(31,4)] [(c,2,2),c∼N(20,2)]
10	[(c,1,1),c∼N(15,2)] [(c,2,2),c∼N(25,1)] [(c,2,3),c∼N(25,2)]	[(c,3,3),c∼N(28,3)] [(c,2,2),c∼N(20,2)] [(c,3,3),c∼N(31,3)]	[(c,2,2),c∼N(28,3)] [(c,1,2),c∼N(18,3)] [(c,3,3),c∼N(28,2)]	[(c,2,2),c∼N(18,2)] [(c,3,2),c∼N(29,2)] [(c,2,2),c∼N(21,5)]	[(c,1,1),c∼N(9,2)] [(c,3,3),c∼N(32,2)] [(c,2,4),c∼N(20,4)]
	Fuzzy Random Cost Matrix(10 ×10) for RCSTSP With Three Conveyances				
i/j	6	7	8	9	10
1	[(c,1,1),c∼N(5,2)] [(c,2,2),c∼N(22,3)] [(c,1,2),c∼N(6,1)]	[(c,1,1),c∼N(15,1)) [(c,3,3),c∼N(35,3)] [(c,4,4),c∼N(46,6)]	[(c,2,3),c∼N(25,3)] [(c,3,4),c∼N(37,2)] [(c,1,2),c∼N(16,2)]	[(c,1,2),c∼N(39,3)] [(c,3,1),c∼N(26,4)] [(c,2,3),c∼N(41,2)]	[(c,2,2),c∼N(20,3)] [(c,1,3),c∼N(30,2)] [(c,4,5),c∼N(42,4)]
2	[(c,4,1),c∼N(39,2)] [(c,3,3),c∼N(30,1)] [(c,3,2),c∼N(30,1)]	[(c,1,2),c∼N(39,3)] [(c,3,3),c∼N(29,1)] [(c,5,6),c∼N(57,2)]	[(c,3,3),c∼N(30,2)] [(c,2,4),c∼N(41,2)] [(c,5,6),c∼N(58,1)]	[(c,1,2),c∼N(17,1)] [(c,3,8),c∼N(36,2)] [(c,2,2),c∼N(17,2)]	[(c,2,2),c∼N(23,2)] [(c,1,1),c∼N(13,2)] [(c,1,2),c∼N(17,3)]
3	[(c,3,3),c∼N(33,1)] [(c,3,3),c∼N(34,3)] [(c,4,5),c∼N(33,4)]	[(c,1,2),c∼N(17,3)] [(c,1,2),c∼N(11,4)] [(c,1,1),c∼N(5,1)]	[(c,3,3),c∼N(30,1)] [(c,3,3),c∼N(30,2)] [(c,2,2),c∼N(24,2)]	[(c,3,3),c∼N(28,2)] [(c,1,2),c∼N(18,1)] [(c,4,4),c∼N(40,4)]	[(c,3,3),c∼N(29,1)] [(c,2,2),c∼N(19,4)] [(c,3,3),c∼N(32,3)]
4	[(c,2,2),c∼N(23,2)] [(c,6,8),c∼N(15,1)] [(c,2,2),c∼N(25,3)]	[(c,1,2),c∼N(19,5)] [(c,1,3),c∼N(30,2)] [(c,3,3),c∼N(30,4)]	[(c,3,3),c∼N(33,1)] [(c,3,8),c∼N(32,5)] [(c,1,1),c∼N(16,3)]	[(c,2,3),c∼N(10,3)] [(c,2,4),c∼N(20,1)] [(c,3,5),c∼N(32,3)]	[(c,2,2),c∼N(24,4)] [(c,4,9),c∼N(47,2)] [(c,3,4),c∼N(37,2]
5	[(c,3,3),c∼N(29,1)] [(c,2,2),c∼N(20,2)] [(c,4,4),c∼N(40,2)]	[(c,5,6),c∼N(42,2)] [(c,1,1),c∼N(14,1)] [(c,2,2),c∼N(25,3)]	[(c,3,3),c∼N(27,1)] [(c,3,3),c∼N(30,4)] [(c,1,6),c∼N(12,1)]	[(c,1,2),c∼N(18,3)] [(c,1,1),c∼N(8,1)] [(c,8,9),c∼N(7,1)]	[(c,2,2),c∼N(26,1)] [(c,2,2),c∼N(25,1)] [(c,2,2),c∼N(25,1)]
6	∞	[(c,3,3),c∼N(31,1)] [(c,4,4),c∼N(40,1)] [(c,2,6),c∼N(23,1)]	[(c,4,2),c∼N(39,3)] [(c,3,3),c∼N(30,1)] [(c,2,2),c∼N(20,1)]	[(c,3,3),c∼N(30,1)] [(c,2,2),c∼N(22,1)] [(c,3,3),c∼N(35,1)]	[(c,3,3),c∼N(28,4)] [(c,4,4),c∼N(40,1)] [(c,3,3),c∼N(30,1)]
7	[(c,4,4),c∼N(38,1)] [(c,2,2),c∼N(20,1)] [(c,4,6),c∼N(43,1)]	∞	[(c,1,1),c∼N(7,1)] [(c,4,4),c∼N(40,1)] [(c,1,4),c∼N(11,1)]	[(c,3,3),c∼N(31,1)] [(c,3,3),c∼N(33,1)] [(c,3,7),c∼N(34,3)]	[(c,2,2),c∼N(19,1)] [(c,1,1),c∼N(13,1)] [(c,6,8),c∼N(25,2)]
8	[(c,2,2),c∼N(23,1)] [(c,1,1),c∼N(15,1)] [(c,1,2),c∼N(4,2)]	[(c,4,4),c∼N(39,1)] [(c,2,2),c∼N(19,1)] [(c,3,4),c∼N(41,4)]	∞	[(c,2,2),c∼N(20,1)] [(c,3,4),c∼N(52,1)] [(c,4,2),c∼N(23,1)]	[(c,3,3),c∼N(35,1)] [(c,6,8),c∼N(35,3)] [(c,4,4),c∼N(39,1)]
9	[(c,2,2),c∼N(22,1)] [(c,3,3),c∼N(31,1)] [(c,2,2),c∼N(23,1)]	[(c,1,3),c∼N(5,1)] [(c,3,3),c∼N(36,1)] [(c,3,4),c∼N(38,1)]	[(c,3,3),c∼N(30,1)] [(c,3,3),c∼N(32,1)] [(c,1,1),c∼N(11,2)]	∞	[(c,2,3),c∼N(27,1)] [(c,1,2),c∼N(18,1)] [(c,2,7),c∼N(24,3)]
10	[(c,3,3),c∼N(30,1)] [(c,1,1),c∼N(10,1)] [(c,3,3),c∼N(33,1)]	[(c,3,39),c∼N(3,1)] [(c,8,9),c∼N(26,1)] [(c,3,4),c∼N(38,1)]	[(c,4,3),c∼N(40,2)] [(c,4,4),c∼N(41,3)] [(c,3,3),c∼N(30,1)]	[(c,3,3),c∼N(29,1)] [(c,5,5),c∼N(51,5)] [(c,3,3),c∼N(30,1)]	∞

	Fuzzy Random risk/discomfort Matrix(10×10) for FRCSTSP With Three Conveyances				
i/j	1	2	3	4	5
1	∞	(.69, .05, .01) (.36, .03, .03) (.26, .02, .02)	(.72, .07, .08) (.38, .03, .04) (.26, .03, .03)	(.73, .01, .02), (.31, .03, .03) (.33, .03, .03)	(.61, .03, .01) (.2, .02, .02) (.6, .06, .06)
2	(.34, .03, .03) (.22, .02, .02) (.14, .01, .01)	∞	(.4, .04, .04) (.18, .02, .02) (.27, .03, .02)	(.16, .01, .01) (.28, .02, .03) (.06, .01, .02)	(.32, .03, .03) (.25, .02, .02) (.34, .03, .03)
3	(.36, .03, .03) (.29, .03, .03) (.28, .02, .03)	(.16, .01, .02) (.54, .08, .01) (.31, .03, .03)	∞	(.1, .02, .03) (.24, .02, .02) (.12, .01, .01)	(.4, .04, .03) (.23, .02, .02) (.45, .04, .04)
4	(.27, .02, .03) (.18, .02, .02) (.9, .01, .01)	(.09, .01, .01) (.19, .02, .02) (.12, .01, .01)	(.16, .01, .02) (.7, .09, .01) (.27, .02, .03)	∞	(.29, .03, .03) (.17, .01, .02) (.23, .02, .02)
5	(.16, .01, .01) (.14, .01, .01) (.6, .01, .02)	(.41, .04, .04) (.21, .02, .02) (.32, .03, .02)	(.34, .03, .03) (.35, .03, .03) (.33, .03, .03)	(.17, .02, .02) (.12, .01, .01) (.4, .04, .03)	∞
6	(.13, .05, .01) (.5, .06, .08) (.5, .07, .03)	(.26, .02, .03) (.2, .02, .02) (.27, .02, .03)	(.4, .04, .03) (.25, .02, .02) (.27, .02, .03)	(.6, .08, .09) (.7, .09, .01) (.1, .01, .01)	(.26, .02, .02) (.26, .02, .03) (.38, .03, .01)
7	(.36, .03, .03) (.37, .03, .04) (.28, .03,.03)	(.23, .02, .02) (.53, .05, .05) (.25, .02, .02)	(.27, .03, .03) (.37, .03, .03) (.24, .02, .27)	(.21, .02, .02) (.4, .04, .04) (.23, .02, .021)	(.35, .03, .03) (.56, .05, .01) (.37, .03,.04)
8	(.39, .04, .04) (.41, .04, .04) (.2, .02, .02)	(.24, .02, .02) (.5, .01, .02) (.16, .01, .01)	(.3, .03, .03) (.52, .05, .05) (.43, .04, .04)	(.38, .04, .04) (.19, .02, .02) (.4, .04, .04)	(.34, .03, .03) (.34, .03, .031) (.46, .04, .04)
9	(.38, .04, .04) (.1, .01, .01) (.31, .03, .03)	(.39, .04, .04) (.38, .03, .04) (.34, .03, .03)	(.4, .01, .02) (.34, .03, .03) (.36, .03, .03)	(.23, .025, .02) (.33, .03, .03) (.28, .02, .03)	(.2, .021, .023) (.31, .03, .03) (.2, .02, .02)
10	(.15, .01, .01) (.25, .02, .02) (.25, .02, .03)	(.28, .03, .03) (.2, .02, .02) (.31, .03, .03)	(.26, .02, .02) (.18, .01, .02) (.28, .03, .03)	(.18, .02, .02) (.29, .03, .03) (.21, .02, .02)	(.9, .01, .01) (.32, .03, .03) (.2, .01, .01)
	Fuzzy Random risk/discomfort Matrix(10 ×10) for FRCSTSP With Three Conveyances				
i/j	6	7	8	9	10
1	(.5, .07, .01) (.22, .02, .02) (.6, .08, .09)	(.15, .01, .01) (.35, .03, .03) (.46, .07, .08)	(.25, .02, .02) (.37, .03, .04) (.16, .01, .02)	(.39, .04, .05) (.26, .03, .03) (.41, .04, .04)	(.2, .02, .03) (.3, .03, .03) (.42, .01, .04)
2	(.39, .04, .04) (.3, .03, .03) (.21, .02, .02)	(.39, .04, .04) (.29, .03, .03) (.57, .05, .06)	(.3, .03, .03) (.41, .04, .04) (.58, .05, .06)	(.17, .01, .02) (.36, .03, .03) (.17, .02, .02)	(.23, .02, .02) (.13, .01, .01) (.17, .01, .02)
3	(.33, .03, .03) (.34, .03, .03) (.33, .03, .03)	(.17, .01, .02) (.11, .01, .01) (.05, .01, .01)	(.3, .03, .04) (.3, .03, .03) (.24, .02, .02)	(.28, .03, .03) (.18, .01, .02) (.4, .04, .04)	(.29, .03, .03) (.19, .02, .02) (.32, .03, .03)
4	(.23, .02, .02) (.15, .01, .01) (.25, .02, .02)	(.19, .02, .02) [(.3, .03, .03) (.3, .03, .03)	(.33, .03, .03) (.32, .03, .03) (.16, .01, .01)	(.1, .01, .013) (.2, .02, .02) (.32, .03, .03)	(.24, .02, .029) (.47, .04, .04) (.37, .03, .04)
5	(.29, .03, .03) (.2, .02, .02) (.4, .04, .04)	(.42, .04, .04) (.14, .01, .02) (.25, .02, .02)	(.27, .03, .03) (.3, .03, .02) (.12, .02, .06)	(.18, .01, .02) (.08, .01, .01) (.07, .01, .01)	(.26, .02, .02) (.25, .02, .02) (.25, .02, .02)
6	∞	(.31, .03, .04) (.4, .04, .04) (.23, .02, .02)	(.39, .04, .04) (.3, .03, .031) (.2, .02, .02)	(.3, .03, .03) (.22, .02, .02) (.35, .03, .03)	(.28, .03, .03) (.4, .04, .04) (.3, .03, .03)
7	(.38, .04, .04) (.2, .02, .02) (.43, .04, .04)	∞	(.07, .001, .001) (.4, .04, .04) (.11, .01, .01)	(.31, .03, .03) (.33, .03, .03) (.34, .03, .03)	(.19, .02, .02) (.13, .01, .01) (.25, .02, .02)
8	(.23, .02, .02) (.15, .01, .01) (.04, .01, .01)	(.39, .04, .04) (.19, .02, .02) (.41, .03, .03)	∞	(.2, .02, .02) (.52, .05, .05) (.23, .02, .02)	(.35, .03, .03) (.35, .03, .03) (.39, .04, .04)
9	(.22, .02, .02) (.31, .03, .03) [(.23, .05, .06)	(.05, .07, .08) (.36, .03, .03) (.38, .03, .04)	(.3, .03, .03) (.32, .03, .03) (.11, .01, .01)	∞	(.27, .02, .03) (.18, .01, .02) (.24, .02, .02)
10	(.3, .03, .03) (.1, .012, .01) (.33, .02, .01)	(.35, .03, .03) (.26, .02,.03) (.38, .03, .04)	(.4, .04, .04) (.41, .04, .04) (.3, .01, .01)	(.29, .03, .03) (.51, .05, .05) (.3, .03, .03)	∞

Table 4.33: Results of FRCSTSP (Model 4.3A4)

δ	θ	Algorithm	DM	Path(Vehicle)	Costs	R_{max}
0.9	0.9	MGA	PDM	4(3)-10(3)-2(3)-9(3)-7(3)-8(1)-6(2)-1(1)-5(2)-3(1)	148.56	8.5
			ODM	4(3)-10(3)-2(3)-9(3)-7(3)-8(1)-6(2)-1(1)-5(2)-3(1)	140.13	8.5
		MGA	PDM	6(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-5(2)-1(1)-7(2)	151.21	8.5
			ODM	6(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-5(2)-113)-7(2)	147.18	8.5
		MGA	PDM	1(2)-10(1)-9(2)-5(1)-7(2)-3(1)-4(2)-2(3)-8(1)-6(1)	166.25	6.75
			ODM	1(2)-10(1)-9(2)-5(1)-7(2)-3(1)-4(2)-2(3)-8(1)-6(1)	151.31	6.75
		GA	PDM	6(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-5(2)-1(3)-7(3)	169.21	6.0
			ODM	6(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-5(2)-1(3)-7(3)	162.45	6.0
0.96	0.7	MGA	PDM	3(2)-8(1)-7(1)-1(1)-10(2)-5(3)-2(3)-4(1)-6(2)-9(3)	155.76	6.75
			ODM	4(2)-8(1)-7(1)-1(1)-10(2)-5(3)-2(3)-3(1)-6(2)-9(3)	142.18	6.75
0.79	0.9	MGA	PDM	5(3)-7(1)-8(1)-6(2)-1(2)-4(1)-9(2)-2(1)-10(1)-3(1)	161.34	6.5
			ODM	4(3)-7(1)-8(1)-6(2)-1(2)-5(1)-9(2)-2(1)-10(1)-3(1)	164.13	6.5
0.85	0.75	MGA	PDM	1(2)-4(3)-2(3)-10(2)-5(3)-9(3)-8(1)-6(1)-3(3)-7(2)	168.45	6.0
			ODM	1(1)-4(3)-2(3)-10(2)-5(3)-9(3)-8(1)-6(1)-1(3)-7(2)	146.93	6.0

STSP. Also we consider three types of conveyances. We set two fold randomness of the given data in the form of mean and variances. The bi-random cost matrix for the CSTSP and corresponding bi-random risk/discomfort matrix are given in Table 4.34. Again, with these input data, we solve the BRCSTSP by MGA and the near optimum results are presented in Table 4.35.

4.4.4 Discussion

To validate the feasibility and effectiveness of the proposed algorithm, we have applied the MGA on some standard TSP problem taken from TSPLIB [162]. The proposed algorithm was implemented in C++ with the parameters as 100 chromosomes, 2000 iterations in maximum. Table 4.20 shows the comparisons between MGA and GA for the some standard TSP problems. It is seen that the number of iterations is less in MGA than SGA.

Again in Table 4.21, we survey the importance's of parameter of selection (p_s) in proposed MGA. It indicates that for the optimal solution of the standard TSP **bayg29**, p_s navigates the sample space better for p_s=0.3. In this case, optimum results are obtained quickly for 66 iterations only. Here also, MGA performs better than the SGA.

In Table 4.22, we consider 10×10 crisp costs and risk/discomfort matrices for a CTSP. The optimum results are presented in Table 4.23. It is observed that

Table 4.34: Input Data: BRCSTSP (Model 4.3A5)

	Bi-random Cost Matrix(10 *10) for BRCSTSP With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	(32,1.1) (37,1.21) (28,1.02)	(19,.9) (39,1.07) (30,1.11)	(21,1.02) (33,1.15) (35,1.17)	(30,1.01) (21,.98) (62,1.2)	(7,1.23) (23,1.02) (8,1.19)	(16,1.11) (36,1.03) (47,.97)	(28,1.04) (39,1.12) (19,1.18)	(41,1.12) (31,1.13) (42,1.03)	(21,1.02) (31,1.1) (43,1.01)
2	(35,1.12) (26,1.18) (17,1.13)	∞	(41,1.03) (21,1.17) (32,1.32)	(18,1.11) (29,1.12) (10,1.03)	(35,1.07) (26,1.2) (37,1.2)	(40,1.02) (31,1.2) (23,1.31)	(40,1.13) (30,1.15) (59,1.14)	(33,1.03) (42,1.21) (59,1.16)	(19,1.2) (37,1.13) (20,1.3)	(24,1.19) (16,1.12) (18,1.03)
3	(38,1.29) (30,1.13) (29,1.15)	(17,1.21) (58,1.43) (34,1.32)	∞	(12,1.25) (25,1.21) (14,1.11)	(42,1.23) (25,1.23) (46,1.24)	(35,1.21) (36,1.4) (34,1.12)	(19,1.13) (11,1.1) (8,1.3)	(32,1.1) (33,1.21) (25,1.16)	(30,1.11) (19,1.22) (41,1.41)	(30,1.21) (22,1.16) (33,1.33)
4	(28,1.14) (20,1.1) (10,1.31)	(10,1.2) (22,1.32) (14,1.2)	(18,1.21) (9,1.4) (29,1.31)	∞	(30,1.13) (19,1.15) (24,1.21)	(25,1.23) (16,1.12) (27,1.13)	(21,1.4) (31,1.4) (33,1.19)	(35,1.3) (36,1.2) (17,1.23)	(12,1.21) (23,1.31) (34,1.2)	(27,1.6) (48,1.2) (39,1.28)
5	(18,1.31) (15,1.2) (8,1.2)	(42,1.2) (23,1.31) (34,1.21)	(35,1.12) (36,1.41) (38,1.34)	(20,1.31) (13,1.31) (43,1.15)	∞	(30,1.21) (21,1.36) (41,1.5)	(45,1.16) (16,1.02) (27,1.31)	(30,1.24) (31,1.27) (13,1.02)	(19,1.34) (10,1.01) (8,1.04)	(28,1.42) (26,1.47) (27,1.21)
6	(15,1.31) (6,1.65) (7,1.27)	(29,1.15) (21,1.75) (29,1.15)	(4,1.32) (26,1.62) (28,1.72)	(8,1.41) (9,1.7) (12,1.04)	(28,1.61) (29,1.21) (39,1.37)	∞	(33,1.26) (42,1.31) (24,1.32)	(40,1.53) (31,1.32) (22,1.65)	(32,1.21) (23,1.34) (35,1.21)	(30,1.54) (41,1.52) (32,1.52)
7	(37,1.6) (39,1.43) (30,1.32)	(25,1.21) (53,1.6) (26,1.54)	(30,1.5) (38,1.71) (26,1.56)	(22,1.61) (43,1.31) (24,1.76)	(37,1.98) (58,1.21) (40,1.21)	(40,1.76) (21,1.65) (45,1.61)	∞	(10,1.31) (43,1.65) (13,1.21)	(33,1.54) (34,1.71) (36,1.37)	(20,1.04) (15,1.2) (26,1.6)
8	(41,1.27) (42,1.43) (23,1.15)	(26,1.43) (6,1.32) (17,1.23)	(32,1.34) (53,1.43) (45,1.17)	(40,1.21) (21,1.21) (42,1.31)	(35,1.53) (36,1.21) (47,1.32)	(25,1.53) (16,1.06) (5,1.03)	(40,1.27) (21,1.03) (43,1.04)	∞	(22,1.31) (53,1.62) (24,1.02)	(37,1.76) (36,1.78) (40,1.02)
9	(40,1.72) (11,1.21) (32,1.02)	(41,1.56) (39,1.56) (36,1.42)	(6,1.24) (36,1.42) (37,1.76)	(25,1.71) (34,1.57) (29,1.08)	(21,1.04) (32,1.3) (21,1.02)	(23,1.32) (33,1.06) (25,1.03)	(7,1.01) (38,1.02) (39,1.21)	(32,1.32) (33,1.76) (13,1.52)	∞	(28,1.41) (19,1.32) (26,1.72)
10	(17,1.51) (26,1.01) (29,1.21)	(30,1.31) (21,1.04) (32,1.92)	(28,1.15) (19,1.21) (30,1.72)	(20,1.72) (31,1.02) (22,1.51)	(11,1.82) (33,1.27) (22,1.19)	(32,1.52) (12,1.18) (34,1.17)	(38,1.02) (28,1.13) (39,1.16)	(41,1.62) (42,1.81) (33,1.21)	(31,1.52) (52,1.37) (32,1.15)	∞
	Bi-random risk/discomfort Matrix(10 ×10) for BRCSTSP With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	(.32,1.1) (.37,1.21) (.28,1.02)	(.19,.9) (.39,1.07) (.30,1.11)	(.21,1.02) (.33,1.15) (.35,1.17)	(.30,1.01) (.21,.98) (.62,1.2)	(.07,1.23) (.23,1.02) (.08,1.19)	(.16,1.11) (.36,1.03) (.47,.97)	(.28,1.04) (39,1.12) (.19,1.18)	(.41,1.12) (.31,1.13) (.42,1.03)	(.21,1.02) (.31,1.1) (43,1.01)
2	(.35,1.12) (.26,1.18) (.17,1.13)	∞	(.41,1.03) (.21,1.17) (.32,1.32)	(.18,1.11) (.29,1.12) (.10,1.03)	(.35,1.07) (.26,1.2) (.37,1.2)	(.40,1.02) (.31,1.2) (.23,1.31)	(.40,1.13) (.30,1.15) (.59,1.14)	(.33,1.03) (.42,1.21) (.59,1.16)	(.19,1.2) (.37,1.13) (.20,1.3)	(.24,1.19) (.16,1.12) (.18,1.03)
3	(.38,1.29) (.30,1.13) (.29,1.15)	(.17,1.21) (.58,1.43) (.34,1.32)	∞	(.12,1.25) (.25,1.21) (.14,1.11)	(.42,1.23) (.25,1.23) (.46,1.24)	(.35,1.21) (.36,1.4) (.34,1.12)	(.19,1.13) (.11,1.1) (.08,1.3)	(32,1.1) (.33,1.21) (.25,1.16)	(.30,1.11) (.19,1.22) (.41,1.41)	(.30,1.21) (.22,1.16) (.33,1.33)
4	(.28,1.14) (.20,1.1) (.10,1.31)	(.10,1.2) (.22,1.32) (.14,1.2)	(.18,1.21) (.09,1.4) (.29,1.31)	∞	(.30,1.13) (.19,1.15) (.24,1.21)	(.25,1.23) (.16,1.12) (.27,1.13)	(.21,1.4) (.31,1.4) (.33,1.19)	(.35,1.3) (.36,1.2) (.17,1.23)	(.12,1.21) (.23,1.31) (.34,1.2)	(.27,1.6) (.48,1.2) (.39,1.28)
5	(.18,1.31) (.15,1.2) (.08,1.2)	(.42,1.2) (.23,1.31) (.34,1.21)	(.35,1.12) (.36,1.41) (.38,1.34)	(.20,1.31) (.13,1.31) (.43,1.15)	∞	(.30,1.21) (.21,1.36) (.41,1.5)	(.45,1.16) (.16,1.02) (.27,1.31)	(.30,1.24) (.31,1.27) (.13,1.02)	(.19,1.34) (.10,1.01) (.08,1.04)	(.28,1.42) (.26,1.47) (.27,1.21)
6	(.15,1.31) (.06,1.65) (.07,1.27)	(.29,1.15) (.21,1.75) (.29,1.15)	(.04,1.32) (.26,1.62) (.28,1.72)	(.08,1.41) (.09,1.7) (.12,1.04)	(.28,1.61) (.29,1.21) (.39,1.37)	∞	(.33,1.26) (.42,1.31) (.24,1.32)	(40,1.53) (.31,1.32) (.22,1.65)	(.32,1.21) (.23,1.34) (.35,1.21)	(.30,1.54) (.41,1.52) (.32,1.52)
7	(.37,1.6) (.39,1.43) (.30,1.32)	(.25,1.21) (.53,1.6) (.26,1.54)	(.30,1.5) (.38,1.71) (.26,1.56)	(.22,1.61) (.43,1.31) (.24,1.76)	(.37,1.98) (.58,1.21) (.40,1.21)	(.40,1.76) (.21,1.65) (.45,1.61)	∞	(.10,1.31) (.43,1.65) (.13,1.21)	(.33,1.54) (.34,1.71) (.36,1.37)	(.20,1.04) (.15,1.2) (.26,1.6)
8	(.41,1.23) (.42,1.43) (.23,1.15)	(.26,1.43) (.06,1.32) (.17,1.02)	(.32,1.34) (.53,1.43) (.45,1.32)	(.40,1.21) (.21,1.21) (.42,1.03)	(.35,1.53) (.36,1.21) (.47,1.05)	(.25,1.53) (.16,1.06) (.05,1.31)	(.40,1.27) (.21,1.03) (.43,1.38)	∞	(.22,1.31) (.53,1.62) (.24,1.73)	(.37,1.76) (.36,1.78) (.40,1.28)
9	(.40,1.72) (.11,1.21) (.32,1.02)	(.41,1.56) (.39,1.56) (.36,1.42)	(.06,1.24) (.36,1.42) (.37,1.76)	(.25,1.71) (.34,1.57) (.29,1.08)	(.21,1.04) (.32,1.3) (.21,1.02)	(.23,1.32) (.33,1.06) (.25,1.03)	(.07,1.01) (.38,1.02) (.39,1.21)	(.32,1.32) (.33,1.76) (.13,1.52)	∞	(.28,1.41) (.19,1.32) (.26,1.72)
10	(.17,1.51) (.26,1.01) (.29,1.21)	(.30,1.31) (.21,1.04) (.32,1.92)	(.28,1.15) (.19,1.21) (.30,1.72)	(.20,1.72) (.31,1.02) (.22,1.51)	(.11,1.82) (.33,1.27) (.22,1.19)	(.32,1.52) (.12,1.18) (.34,1.17)	(.38,1.02) (.28,1.13) (.39,1.16)	(.41,1.62) (.42,1.81) (.33,1.21)	(.31,1.52) (.52,1.37) (.32,1.15)	∞

Table 4.35: Results of BRCSTSP (Model 4.3A5)

α	β	Algorithm	Path(Vehicle)	Value	R_{max}
0.95	0.95	MGA	2(2)-10(3)-3(3)-9(3)-7(3)-8(1)-6(2)-1(1)-5(2)-4(3)	56.31	9.5
		MGA	5(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-6(2)-1(3)-7(3)	59.61	9.5
0.8	0.9	MGA	8(1)-6(2)-1(2)-9(1)-3(1)-4(2)-2(2)-10(1)-5(3)-7(3)	58.45	8.75
		MGA	3(1)-4(3)-2(3)-10(2)-5(3)-9(3)-8(1)-6(1)-1(3)-7(2)	71.59	8.75
0.7	0.9	MGA	7(2)-8(1)-6(2)-1(1)-10(2)-5(3)-2(3)-3(1)-4(2)-9(3)	59.48	8.5
		MGA	10(2)-1(1)-9(2)-5(1)-7(2)-3(1)-4(2)-2(3)-8(1)-6(1)	64.54	8.5
0.75	0.75	MGA	3(2)-7(1)-8(1)-6(2)-1(2)-5(1)-9(2)-2(1)-10(1)-4(1)	63.42	8.0
		MGA	1(3)-10(2)-8(1)-6(1)-9(1)-2(1)-7(1)-5(3)-3(1)-4(1)	65.21	8.0
0.95	0.75	MGA	3(1)-4(3)-2(3)-10(2)-5(3)-9(3)-8(1)-6(1)-1(3)-7(2)	57.79	7.5
		GA	5(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-6(2)-1(3)-7(3)	72.49	7.5

CTSP without any total risk factor as a goal gives the lowest minimum cost and as the total risk/discomfort decreases, total cost increases. It is realistically true in our day-to-day life. For a particular value of risk/discomfort, the some near-optimum results along with the optimum one are given. Due some reasons if the TS fails to implement the optimum results, he/she may to achieve the most feasible near-optimum solution.

Again, we form a CSTSP with the three conveyances i.e. ($10\times10\times3$) costs and risk/discomfort matrices are presented in Table 4.24. Along each route, the corresponding conveyances are in parentheses. The optimum results of CSTSP are given in Table 4.25. Here also as total risk/discomfort goes down, the corresponding travelling cost increases.

A ($10\times10\times3$) FCSTSP is presented in Table 4.26 where both costs and risk/discomfort factors along with the targeted total risk/discomfort are triangular fuzzy numbers. The optimum results of FCSTSP in both optimistic and pessimistic senses with different possibility and necessity levels, also other three different models results with GMIV, Credibility and EVM strategies result are presented in Table 4.27. As expected, optimistic model fetches less travelling cost than the pessimistic model.

In Table 4.28, the costs and risk/discomfort factors for the same size CSTSP are random having normal distribution. Mean and variance for cost and risk/discomfort parameter are presented together in first bracket. The model is a combination of E- and V- models and the probabilistic constraint is made deterministic using chance constraint. The optimum results are available in Table 4.29. Here, it is observed that when E-model is given more importance i.e. if more weight are given to E-model, cost increases and on the other hand, importance to V-model reduces the cost.

For random-fuzzy CSTSP, random-fuzzy input data and optimum results are presented in Tables 4.30 and 4.31 respectively. Here, costs and risk/discomfort factors are L-L fuzzy numbers. For a fixed θ=0.94, results in possibility and necessity approaches are given where as before, optimistic representation gives better result (less cost) than the pessimistic ones.

Again in the case of fuzzy-random CSTSP, fuzzy-random input data and optimum results are presented in Tables 4.32 and 4.33 respectively. Here, costs and risk/discomfort factors are LR-type fuzzy numbers and the mean values is a normal N(m,σ) variate. For a fixed δ=η (=0.9), results in possibility and necessity approaches are given where as before, optimistic representation gives better result (less cost) than the pessimistic ones.

Similarly for bi-random costs and risk/discomfort factors presented in Table 4.34, optimum results are obtained with different probability levels-α and β for objective (cost) and constraint (risk/discomfort factor) respectively and presented in Table 4.35. In all cases, the near-optimum solutions are available. Also MGA gives better results than the SGA.

4.5 Model-4.4: A Rough Set based Genetic Algorithm for Constrained Solid TSP with Interval Valued Costs and Times

This model presents a Rough Set based Genetic Algorithms (RSGAs) to solve constrained Solid Travelling Salesman Problems (CSTSPs) with restricted conveyances (CSTSPwR) having uncertain travel costs and times as interval values. In the proposed RSGAs, a rough set based age dependent selection technique and an age oriented min-point crossover are used along with three types of generation and p_m- dependent random mutations. A number of benchmark problems from standard data set, TSPLIB [162] are tested against the proposed algorithm and existing simple GA (SGA) and hence the efficiency of the new algorithms are established. We have modeled CSTSPwRs where some conveyances are not allowed to run in some particular routes. CSTSPwRs are formulated as constrained linear programming problems and solved by both proposed RSGAs and SGA. These are illustrated numerically by some empirical data and the results from the above methods are compared. Statistical significance of the proposed algorithms are demonstrated through statistical analysis using standard deviation (SD). Moreover, as a non-parametric test, Friedman test is performed with the proposed algorithms. In addition, a Post hoc paired comparison is done and the out performances of the RSGAs are established.

4.5.1 Proposed RSGAs

Here RSGA is developed with the rough set based age dependent selection, min-point crossover and p_m- dependent random mutation and used among a set of potential solutions to get a new set of solutions. As usual, it is continued until terminating conditions are encountered.

i. Representation:

Here a complete tour on N cities represents a solution. So an N dimensional integer vector $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$ is used to represent a solution (path), where $x_{i1}, x_{i2}, ..., x_{iN}$ represent N consecutive cities in a tour. Population size number M, and i-th solution $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$, where $x_{i1}, x_{i2}, ..., x_{iN}$, are randomly generated by random number generator between 1 to N with maintaining the TSP conditions such as no repetition of cities (nodes) and also satisfying the necessary constraints. Fitnesses are evaluated through summing the costs between the consecutive cities (nodes) of each solution (chromosome). The solution f(X_i) represent for the i_{th} solution fitness in the solution space. Since the maximum population size M, so M numbers of solutions (chromosomes) are generated randomly.

ii. Rough Set Based Age Dependent Selection:

In Last et al. [88], for the first time, an attempt was made to improve the performance of genetic algorithms by providing a new fuzzy-based extension of the Life Time feature. They used a Fuzzy Logic Controller (FLC) to adopt the crossover probability as a function of the chromosomes age. Also Fdez et al. [50], Roy et al. [147] used it in some refinement of the mechanism on inventory control system. They used the age of the chromosome in fuzzy environment. Here we model the age in rough environment as rough set is more uncertain than the fuzzy set in the uncertain paradigms. So rough set based age is more effective. The general principle is that for both young and old individuals, the crossover probability is naturally low, while there is a certain age interval, where this probability is high. The concepts of young, old, and middle-aged are modeled as linguistic variables. Here, we use these linguistic variables in rough environment.

In our proposed RSGAs, the age of a chromosome is determined based on their fitness values and then a ' **rough set based age dependent selection' (REA)** technique is applied. Here the age of each chromosome lie in a region of the common age represented by a rough set. These regions are termed as young, middle and old for RSGA-I. So for the age of each chromosome, a linguistic value - young, middle or old is created. Now according to the age distributions of the members (in pair) of the mating pool, similar linguistic variables such as low, medium and high are generated for the said chromosomes to fix p_c's. Using the trust measures of rough set, the probability of crossover, p_c for each chromosome is assigned by the corresponding linguistic variables.

Now, we have M solutions in a generation with fitnesses represented by f(X_i) for the i_{th} chromosomes. At the time of initialization, each chromosome age is defined as null. Now in each generation, the age is counted as using the following mechanism :

$$age(x_i) = \begin{cases} avg(age) + \frac{k*(avgfit - f(X_i))}{(avgfit - minfit)}, If avgfit > f(X_i), \\ \frac{avg(age)}{2} + \frac{k*(f(X_i) - avgfit)}{(maxfit - avgfit)}, otherwise \end{cases} \tag{4.48}$$

where avgfit is the average fitness values, maxfit and minfit are maximum and minimum fitness values of the last generation, k=$\frac{maxfit+minfit}{2}$. Also avg(age) means the average age of the set of chromosomes. Here the maximum and minimum ages depend on the requirement of the problems.

Now since age is calculated as crisp values, we construct the common rough values form it,

Rough Age=([r_1*avg age, r_2*avg age], [r_3*avg age, r_4*avg age]),

r_1=$\frac{Max\ Age - Avg\ Age}{Avg\ Age}$, r_2=$\frac{Max\ Age + Min\ Age}{2}$, r_3=$\frac{Max\ Age - Min\ Age}{2}$, r_4=$\frac{Avg\ Age - Min\ Age}{Avg\ Age}$

According to the age of the chromosome, it belongs to any one of the common rough age defined as Young, Middle and Old. For common rough age ([a,b],[c,d]), it is described as below

$$Age = \begin{cases} Young & for\ c \le age < a \\ Middle & \text{for } a \le age \le b \\ Old & \text{for } b < age \le d \end{cases} \tag{4.49}$$

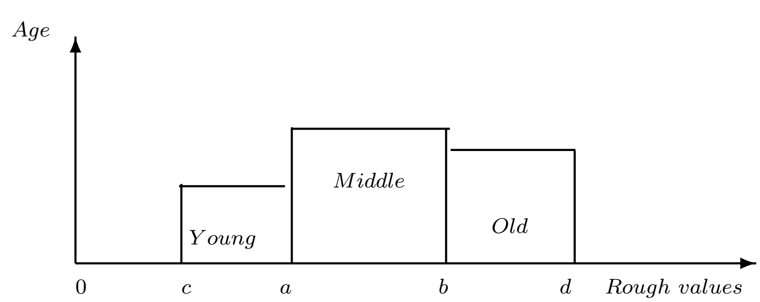


Fig.4.5.1 : Rough age distribution of Interval.

Table 4.36: Rough trust based linguistics

Chromosomes	Young	Middle	Old
Young	Low	Medium	Low
Middle	Medium	High	Medium
Old	Low	Medium	Low

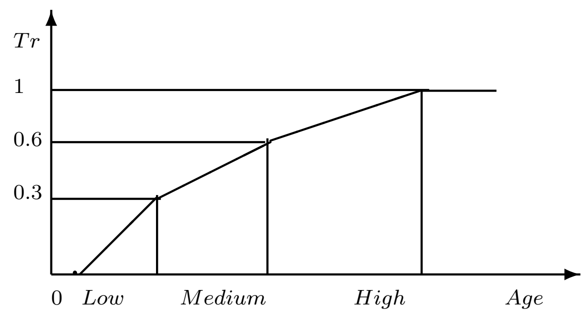


Fig.4.5.2 : Rough age distribution of p_c.

Out-comes of different types of ages are given in Eq. 4.49. The above Eq. 4.49, shows that if common rough age region is ([a,b],[c,d]), then the space (c to a) refers to as young age, (a to b) as middle age and (b to d) as old age. Also the pictorial representation of this is given in Fig. 4.5.1. Then in the Table 4.36, according to the linguistic values, rough trust based p_c are assigned which is shown in Fig.4.5.2.

iii. Rough Extended Age Based Selection:

To have more accurate classification, we make five classifications instead of above three and then, the region of common age is divided into very young, young, middle, old and very old for RSGA-II. As before, combining the eligible parents, the very low, low, medium, high and very high linguistic variables are assigned for p_c's of chromosomes.

Now, we consider the age in a different extended linguistic code i.e. Young, Middle and Old are replaced by Very Young, Young, Middle, Old and Very Old scale. So it is more realistic in the sense of classification and acceptable to design for the real world problems. According to the requirement of the five linguistic values, the trust measure levels are expanded in five sections which are shown in Eqs. 3.5.1 and 3.5.1. Determined p_c values of the extended linguistics are also given below in Fig. 4.5.3.

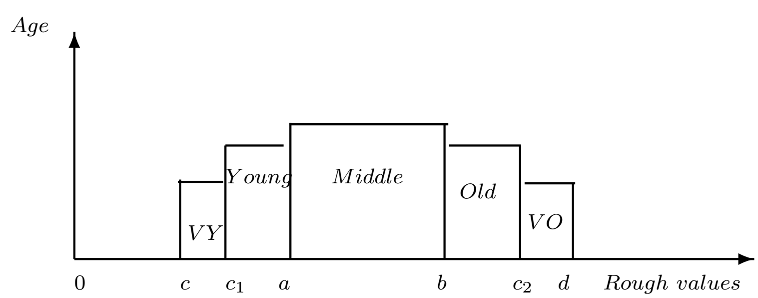
According to the extended age of the chromosome, it belongs to the any one

Table 4.37: Rough extended trust based linguistic

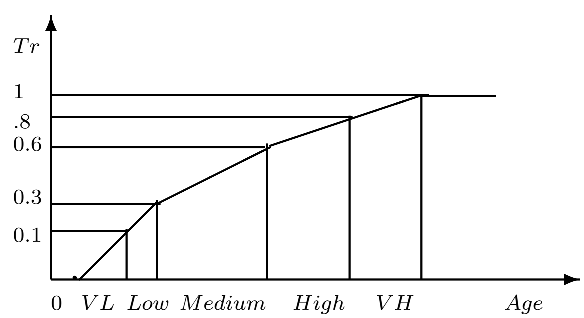
Chromosomes	Very Young	Young	Middle	Old	Very Old
Very Young	Very Low	Low	Medium	Low	Very Low
Young	Low	Low	High	Low	Very Low
Middle	Medium	High	Very High	High	Medium
Old	Low	Low	High	Low	Very Low
Very Old	Very Low	Very Low	Medium	Very Low	Very Low

of the common rough age intervals like Very Young, Young, Middle, Old and Very Old. The common rough age ([a,b],[c,d]) is extended to $0 \leq c \leq c_1 \leq a \leq b \leq c_2 \leq d$ and is described as below,

$$Age = \begin{cases} Very\,Young & for\; c \leq age < c_1 \\ Young & \text{for } c_1 \leq age < a \\ Middle & \text{for } a \leq age \leq b \\ Old & \text{for } b < age \leq c_2 \\ Very\,Old & \text{for } c_2 < age \leq d \end{cases} \tag{4.50}$$



$Fig.4.5.3 : Rough\ extended\ age\ distribution\ of\ Interval.$



$Fig.4.5.4 : Rough\ extended\ age\ distribution\ of\ p_c.$

iv. Min-Point Crossover

In addition to this, an unique crossover technique - **'min-point crossover'** is introduced here for extracting minimum travelling cost as a TSP demands it.

Thus the developed crossover depends upon the basic requirement of total minimum travel cost. First, randomly selected two paths(say, parents) are modified. Then new paths (i.e. children) are created from the modified parents comparing the costs between the nodes(i.e. cities) in form of convex combinations of uncertain interval values. The node with minimum cost is selected for this purpose.

A. Determination of Crossover Probability(p_c): Probability of crossover (p_c) for a pair of chromosomes (X_i,X_j) is determined as below:

(a). p_c's for rough set based age selection

(i). At first age levels, (young, middle, old) of X_i and X_j are determined by making trust measure of rough values with respect to their ages in common rough age region given in Fig. 4.5.2.

(ii). After determination of age intervals of the chromosomes, their crossover probabilities are determined as linguistic variables (low, medium, high) as in Fig. 4.5.1 using rough trust measure which is presented in Table 4.36 and trust levels are given as Eqs. 3.5 and 3.5.

(b). p_c's for extended rough set based age selection

(i). Proceeding as above, p_c of each chromosome for the extended rough set based age selection are determined according to Table 4.37 and given in Fig. 4.5.3 and 4.5.4.

B. Crossover Mechanism:

To select two individuals (parents) from the matting pool, generate the random number, between [0,1]. If $r < p_c$ then select that population for first parent (say P_{r1}). Similarly choose the another parents (say P_{r2}). Let these are P_{r1}: a_1, a_2,..., a_N and P_{r2}: s_1, s_2,..., s_N. Here (a_1, a_2,..., a_N) and (s_1, s_2,..., s_N) are nodes within (1, 2, 3,..., N), and these are numbers of cities. Then choose a city randomly from 1 to N, say a_i = s_p(i=1, 2, ..., N), p=(1, 2,..., N). Parents are modified by placing a_i or s_p in the first place of P_{r1} and P_{r2} respectively. Now modified parents are given by P_{r1}: a_i, a_1, a_2,.., a_{i-1}, a_{i+1},....a_N, P_{r2}: s_p, s_1, s_2,., s_{p-1}, s_{p+1},...., s_N. To get the first child (Ch_1), placing a_i in the first place of Ch_1, compare the next route a_i to a_1 and a_i to s_1, minimum cost route be selected in Ch_1. As consider the uncertain interval values, so we determine the values as convex combination of the upper and lower interval values, such as for any route cost between two nodes a_i to a_2 becomes as **c** (a_i, a_1) =$\lambda * \mathbf{CL_{i1}} + (\mathbf{1} - \lambda) * \mathbf{CR_{i1}}$, $\lambda \in rand[0, 1]$, CL and CR are, lower and upper values of the corresponding intervals between two nodes. This crossover mechanism is given in the section 4.3.1(c)(iii).

Algorithm for Min-point crossover:

input: Matting Pool, p_c, Total number of nodes (N).

output: Offspring (child).

1. **Start**
2. **for** i=1 to Pop Size
3. Random number generator r∈ [0,1]
3. Choose pair of chromosomes according to p_c
4. Randomly generate node between 1 to N (say a_r)
5. Replace a_r at first place of each parents chromosomes // modified parents
6. Determine min-point value of each corresponding node
7. **for** j=1 to N
8. Compare min-point value
9. Check the existence of corresponding node in child
10. Concatenated node to the child (offspring)
11. **end for**
12. Replace a_r at end place of each parents chromosomes// modified parents
13. Compare min-point value from end of the each corresponding nodes
14. **for** j=1 to N
15. Compare min-point value
16. Check the existence of corresponding node in child
17. Concatenated node to the child (offspring)
18. **end for**
19. Replace the child's in offspring's set
20. **end for**
21. End Algorithm

v. Three Different Forms of p_m Dependent Random Mutations:

(a). Selection for mutation: For each solution of P(t), generate a random number r from the range [0,1]. If $r < p_m$ then the solution is taken for mutation where p_m be the probability of mutation.

(b). Mutation Process: At first determine the total number of mutated node (T). To mutate a solution $X = (x_1, x_2, ..., x_N)$, number of mutated node T= p_m* N, N=total number of nodes.

i. Random location method (Type-I): Generate two different integer randomly between [1, N]. Interchange the nodes x_i, x_j in order to generate two random integers up-to T times to get new solutions which replace the parent solution.

ii. Fixed location method (Type-II): If T becomes even, then selected T consecutive numbers of nodes in a solution $X = (x_1, x_2, ...x_N)$ and select any of the two node x_i and x_j and interchange their places. So here change is done up-to T/2 times not generating any random number. On the other hand if T becomes odd, then interchanges similarly the places of the solutions up to (T/2)+1 times.

iii. Generation Oriented Mutation (Variable Method): Here we model a new form of mutation mechanism where probability of mutation (p_m) are determined as follows

$p_m = \frac{k}{Current\ generation\ number}$, k$\in$[0,1].

So, here p_m decreases smoothly as generation increases. After calculating the p_m, mutation operation is applied in both of the two methods, Type-I and Type-II methods.

Type-I: Algorithm for generation dependent random mutation

c. Procedure Mutation:

input: Total number of nodes (N), Offspring's.
output: Mutated offspring (child).

1. **Start**
2. Set g=current generation number
3. $p_m = \frac{k}{g}$, k$\in$[0,1]
4. Determine T= p_m*N // total number of mutated nodes
5.   **for** i=0 to pop_size
5.   r=rand(0,1)
6.   **if**(r< p_m){
6.   Select chromosome depending p_m
7.   **for** j=1 to T
8.   Randomly select two different nodes between [1,N]
9.   Replace the places of the selected two nodes
10.   **end for**
11.   eEnd if
11.   **end for**
12. **End Algorithm**

Type-II: Algorithm for generation dependent fixed location mutation

1. **Start**
2. Set g=current generation number
3. p$_m$=$\frac{k}{g}$, k∈[0,1]
4. Determine T= p$_m$*N // total number of mutated node
5. **for** i=0 to pop_size
5. r=rand(0,1)
6. **if**(r$<$ p$_m$){
6. Select chromosome depending p$_m$
7. Select T consecutive nodes location in chromosome
8. **for** j=1 to $\frac{T}{2}$ or ($\frac{T}{2}+1$)// According T even or odd
9. Replace the places of the any two nodes
10. **end for**
11. **end for**
12. **End Algorithm**

With the above selection, crossover and mutation, the proposed RSGA is as follows:

Algorithm for RSGA

Input: max_ gen, pop_ size, Max_age, Min_ age, Input Data (cost, risk matrix).

Output: The optimum and near optimum solutions.

1. **Start**
2. g ← 0 // g: iteration/generation number
3. **Initialize P(g)** // randomly generate initial population $P(g)$
4. **Evaluate f(P(g))**; //Evaluate fitness of each chromosome
5. **while** (g $\leq max\_gen$)
6. Evaluate the average fitness
7. **if** average fitness $>$ current fitness
8. age(x$_i$)=avg(age)+$\frac{k*(avgfit-f(X_i))}{(avgfit-minfit)}$
9. **else**
10. age(x$_i$)=$\frac{avg(age)}{2}+\frac{k*(f(X_i)-avgfit)}{(maxfit-avgfit)}$
11. **if** (age(x$_i$)$>$ maximum age)
12. age(x$_i$)= maximum age
13. **else if** (age(x$_i$)$<$ minimum age)

14. age(x_i)= minimum age
15. Determine average age
16. Determine common rough age
17. Switch (Choice)
18. Case I:// **RSGA-I**
    (a). Developed linguistic variables young, middle, old
    (b). **for** each pair of parents **do**
    (c). Trust based p_c created
    (d). **end for**
19. Case-II:// **RSGA-II**
    (a). Developed very young, young, middle, old, very old
    (b). **for** each pair of parents **do**
    (c). Extended trust based p_c created
    (d). **end for**// end switch
20. **for** i=1 to Pop Size//**min-point crossover**
21. Choose pair of chromosomes according to p_c
22. Randomly generate node between 1 to N (say a_r)
23. Replace a_r at first place of each parents chromosomes
24. Determine min-point value of each corresponding node
25. **for** j=1 to N
26. Compare min-point value
27. Check the existence of corresponding node in child
28. Concatenated node to the child (offspring)
29. **end for**
30. Replace a_r at end place of each parents chromosomes
31. Compare min-point value from end of the each corresponding nodes
32. **for** j=1 to N
33. Compare min-point value
34. Check the existence of corresponding node in child
35. Concatenated node to the child (offspring)
36. **end for**
37. Replace the child's in offspring's set
38. **end for**
39. Switch (Choice) // **Mutation**
40. Case-I(**simple**):

(a). **for** i=0 to pop_size
(b). Select chromosome depending p_m
(c). Randomly select two different nodes between [1,N]
(d). Replace the places of the selected two nodes
(e). **end for**

41. Case-II(**variable**):
(a). $p_m=\frac{k}{g}$, k∈[0,1]
(b). Determine T= p_m*N // total number of mutated node
(c). **for** i=0 to pop_size
(d). Select chromosome depending p_m
(e). **for** j=1 to T //**Type -I**
(f). Randomly select two different nodes between [1,N]
(g). Replace the places of the selected two nodes
(h). **end for**
(i). **end for**

42. Case-III(**variable**):
(a). $p_m=\frac{k}{g}$, k∈[0,1]
(b). Determine T= p_m*N
(c). **for** i=0 to pop_size
(d). Select chromosome depending p_m
(e). **for** j=1 to $\frac{T}{2}$ or $(\frac{T}{2}+1)$//According T even or odd(**Type-II**)
(f). Replace the places of the any two nodes
(g). **end for**
(h). **end for**

43. Store the new off springs into offspring set
44. **Reproduce a new P(g)**
45. **Evaluate f(P(g))**;//evaluate the fitness of reproduce chromosome
46. Store the local optimum and near optimum solutions
47. g ← g+1
48. endwhile
49. Store the global optimum and near optimum results
50. End Algorithm.

Flowchart of this algorithm is depicted in Fig. 4.1.

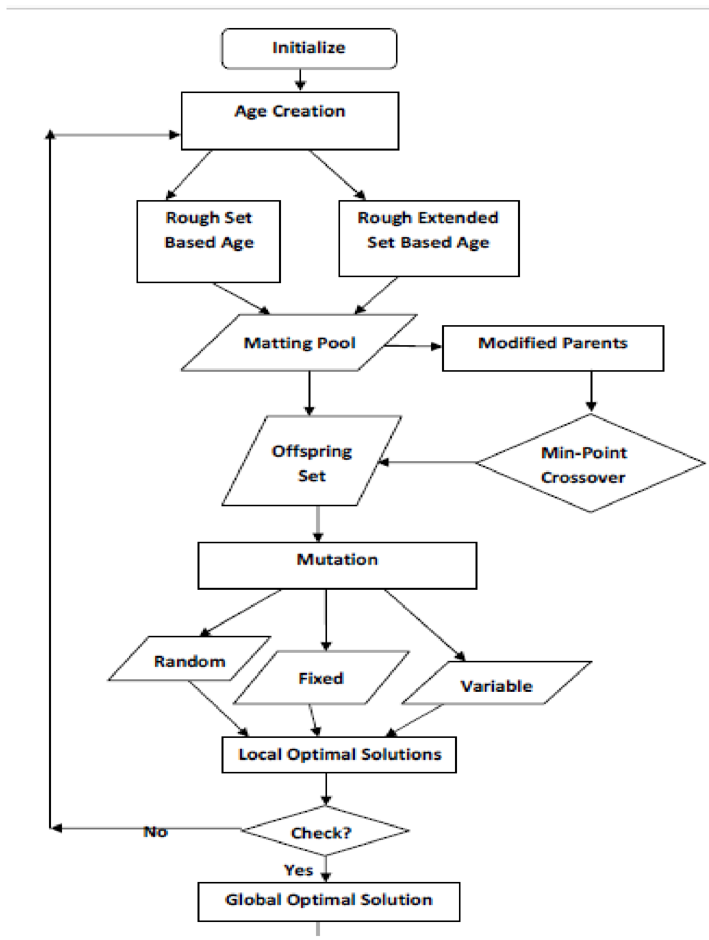


Figure 4.1: Flow chart of RSGA

Termination Criteria

RSGA-I (Rough set based) and RSGA-II (Rough extended set based) algorithms are terminated when any one of the following conditions is satisfied (which over is earlier):

(a) the best solution does not improve within some consecutive generations

(b) number of generations reaches user defined iterations.

The same termination criteria are used for SGA, SGA-I, II, III, IV, V and FGA which are different combinations of the GA operators presented in Table 4.39.

4.5.2 Mathematical Formulation and Its crisp equivalence

Model 4.4A: STSP with time Constraints (CSTSP):

In a STSP, a salesman has to travel N cities by choosing any one of the available P different types of conveyances using minimum cost restricting total travel time within maximum allowable time. Times taken to travel from one city another using different conveyances are different. Let $c(i, j, k)$ be the cost for travelling from i-th city to j-th city using k-th type conveyance and $t(i, j, k)$ be the time taken for this travel. These values including maximum allowable are crisp interval numbers. Then the salesman has to determine a complete tour (x_1, x_2, ...,x_N, x_1) and corresponding conveyance types (v_1, v_2, ..., v_P) to be used for the tour, where $x_i \in \{1, 2, ..N\}$ for $i = 1, 2, ..., N$, $v_i \in \{1, 2, ..P\}$ for $i = 1, 2, ..., N$ and all x_is are distinct. Then the problem can be mathematically formulated as:

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one of the available conveyance in each step from the vehicle types $(v_1, v_2, ..., v_P)$ so as

$$\left.\begin{array}{l} \text{to min } \; Z = \sum_{i=1}^{N-1} [c_{L_{i,i+1}}, c_{R_{i,i+1}}](x_{i,i+1}, v_i) + [c_{L_{N,1}}, c_{R_{N,1}}](x_{N,1}, v_l), \\ \text{s.t } \; \sum_{i=1}^{N-1} [t_{L_{i,i+1}}, t_{R_{i,i+1}}](x_{i,i+1}, v_i) + [t_{L_{N,1}}, t_{R_{N,1}}](x_{N,1}, v_l) \\ \qquad \subseteq [t_{maxL}, t_{maxR}], \\ where\ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\} \end{array}\right\} \quad (4.51)$$

Model 4.4A1: STSP using restricted conveyances with time Constraints (CSTSPwR):

In real life, it is seen that in all stations, all types of conveyances may not available due to the geographical position of the station,weather conditions, etc. So it is more realistic, that restricted conveyances are available in different stations. Considering the availability of the conveyances, we design the STSP with restricted condition with time constraints as below:

Let $c(i, j, k)$ be the cost for travelling from i-th city to j-th city using k-th type conveyance and $t(i, j, k)$ be the time taken in travelling from i-th city to j-th city using k-th type conveyance. Then the salesman has to determine a complete tour (x_1, x_2, ...,x_N, x_1) and corresponding conveyance types (v_1, v_2, ..., v_S) to be used for the tour, where $x_i \in \{1, 2, ..N\}$ for $i = 1, 2, ..., N$, $v_i \in \{1, 2, ..S\}$ for $i = 1, 2, ..., N$ and all x_is are distinct. Also $v_i \in \{1, 2, ..S\}$ provides maximum available S($\leq$ P) types of conveyances. Then the problem can be mathematically formulated as:

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one available corresponding conveyance in each step from the vehicle types $(v_1, v_2, ..., v_S)$ so as

$$\left.\begin{array}{rl} \text{Minimize} & Z = \sum_{i=1}^{N-1}[c_{L_{i,i+1}}, c_{R_{i,i+1}}](x_{i,i+1}, v_i) + [c_{L_{N,1}}, c_{R_{N,1}}](x_{N,1}, v_l), \\ \text{s.t} & \sum_{i=1}^{N-1}[t_{L_{i,i+1}}, t_{R_{i,i+1}}](x_{i,i+1}, v_i) + [t_{L_{N,1}}, t_{R_{N,1}}](x_{N,1}, v_l) \\ & \subseteq [t_{maxL}, t_{maxR}], \\ & where\ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{v_1, v_2.., v_S\} \end{array}\right\} \tag{4.52}$$

Thus the above model written as

$$\left.\begin{array}{c} \text{Minimize} \quad Z = [C_L, C_R](x, v), \\ \text{s.t} \quad [T_L, T_R](x, v) \subseteq [t_{maxL}, t_{maxR}] \\ where\ x_i \neq x_j, i, j = 1, 2...N, \quad v_i \in \{v_1, v_2.., v_S\} \end{array}\right\} \tag{4.53}$$

$$and \qquad [C_L, C_R](x, v) = \sum_{i=1}^{N-1}[c_{L_{i,i+1}}, c_{R_{i,i+1}}](x_{i,i+1}, v_i) + [c_{L_{N,1}}, c_{R_{N,1}}](x_{N,1}, v_l)$$
$$[T_L, T_R](x, v) = \sum_{i=1}^{N-1}[t_{L_{i,i+1}}, t_{R_{i,i+1}}](x_{i,i+1}, v_i) + [t_{L_{N,1}}, t_{R_{N,1}}](x_{N,1}, v_l).$$

The interval valued objective and constraints are transformed as given in section 3.2 following Karmakar et al. [78]. Thus the crisp version of the above model is:

$$minimize\ Z = ({C_L}^{m_1} * {C_R}^{m_2})^{\frac{1}{m_1+m_2}}, for\ m_1, m_2 \in (0, 1).$$

with one of the following constraints.

According Moore's approaches [79]

$$\left.\begin{array}{r} \text{subject to} \quad t_{maxL} \leq T_L, \\ T_R \leq t_{maxR}. \end{array}\right\} \tag{4.54}$$

Ishibuchi and Tanaka's approaches [79]

$$\left.\begin{array}{r} \text{subject to} \quad T_L \leq t_{maxL}, \\ \frac{(T_L+T_R)}{2} \leq \frac{(t_{maxR}+t_{maxL})}{2}. \end{array}\right\} \tag{4.55}$$

Chanas and Kuchta's approaches [79] for $0 \leq s_0 < s_1 \leq 1$,

$$\left.\begin{array}{c} \text{s.t} \quad (T_L + s_0 * (T_R - T_L)) \leq (t_{maxL} + s_0 * (t_{maxR} - t_{maxL})), \\ \frac{(T_L+s_0*(T_R-T_L)+T_L+s_1*(T_R-T_L))}{2} \leq \\ \frac{(t_{maxL}+s_0*(t_{maxR}-t_{maxL})+t_{maxL}+s_1*(t_{maxR}-t_{maxL}))}{2} \end{array}\right\} \tag{4.56}$$

Hu and Wang's approaches [79]

$$\left.\begin{array}{r}\text{subject to } \frac{(T_L+T_R)}{2} \leq \frac{(t_{maxR}+t_{maxL})}{2}. \\ (t_{maxR}-t_{maxL}) \leq (T_R - T_L)\end{array}\right\} \quad (4.57)$$

Mahato and Bhunia's approaches [79]

$$\left.\begin{array}{r}\text{subject to } T_L \leq t_{maxL}. \quad (optimistic\ case) \\ \text{Pessimistic case } \frac{(T_L+T_R)}{2} \leq \frac{(t_{maxR}+t_{maxL})}{2},\ Type-I\ \&\ Type-II, \\ (t_{maxR}-t_{maxL}) \leq (TT_L),\ for\, Type-III\ intervals.\end{array}\right\} (4.58)$$

4.5.3 Numerical Experiments

(a) Testing with problems from TSPLIB[162]

To validate the feasibility and effectiveness of the proposed algorithms, we apply the proposed RSGAs on some standard TSP problems taken from TSPLIB[162]. The proposed algorithm was implemented in C++ with following parameters as 100 chromosomes and 2000 iterations (maximum). The best optimal results are presented.

Comparison of results of test problems by RSGA-II and SGA :

Table 4.38 gives the results of the test problems using RSGA-II and SGA, the results are compared in terms of optimal cost, iterations and computational time (CPU time in minutes). It is seen that the number of iterations and computational times are less in RSGA-II than SGA. In each instance, average result (Avg), best found results (Cost) and the standard deviations (SD) are presented.

(b) Comparison RSGAs w. r. to different operators:

Moreover, for a particular test problem bayg29, both SGA and proposed RSGAs are used with different operators and parameters (p$_c$'s, p$_m$'s, p$_s$'s). The obtained results are presented in Tables 4.39 and 4.40.

In Table 4.39, we survey the importance's of different types of selection, crossover and mutation parameters in the proposed algorithms. It indicates that for the optimal solution of the standard TSP **bayg29**, optimal result is found in the rough extended age based selection mechanism with min point crossover and fixed mutation. These results are obtained quickly by 64 iterations only. Here also, RSGAs perform better than SGA. In this testing, "probabilistic" selection takes less generations than that required for "Roulette Wheel" selection. Again

Table 4.38: Test TSPLIB Problems by RSGA

Instances	BKS	RSGA-II				SGA			
		Cost Avg SD	Iter.	Time	Run	Cost Avg SD	Iter.	Time	Run
fri26	937	937 935.38 1.64	**67**	0.23	25	937 928.46 2.71	269	3.58	25
bays29	2020	2020 2017.27 1.87	**64**	1.56	25	937 2014.53 3.56	451	4.42	25
bayg29	1610	1610 1609.76 0.78	**43**	1.32	25	1610 1603.31 2.37	378	4.56	25
dantzig42	699	699 697.21 1.46	**123**	1.41	25	699 695.47 3.95	612	5.36	25
eil51	426	426 425.86 0.63	**98**	1.78	40	426 422.43 2.49	341	4.21	40
berlin52	7542	7542 7540.37 1.29	**145**	2.1	40	7542 7537.56 3.01	526	4.37	40
st70	675	675 674.3 0.75	**165**	2.9	40	675 671.73 1.72	813	7.01	40
eil76	538	538 537.75 0.98	**124**	2.76	40	538 535.47 1.58	457	4.27	40
pr76	108159	108159 108156.32 2.42	**175**	3.01	40	108159 108143.9 6.78	410	3.49	40
rat99	1211	1211 1210.7 1.2	**149**	4.2	40	1211 1207.43 2.17	328	6.12	40
kroa100	21282	21282 21276.81 2.79	**249**	6.53	50	21282 21267.34 5.98	285	12.34	50

Table 4.39: Comparison of RSGAs, SGAs with different parameters

Algorithm	Selection	Crossover	Generation	p_c	p_m	p_s	Avg	SD	Result	Run
SGA-I	Roulette Wheel	Cyclic	678	0.3	0.4	-	1603.31	2.37	1610	25
SGA-II	Probabilistic	Cyclic	309	0.31	0.4	0.3	1605.8	1.83		
SGA-III	Probabilistic	Comparison	256	0.4	0.4	-	1604.72	3.17		
SGA-IV	Probabilistic	Comparison	176	0.4	0.4	0.3	1607.81	1.54		
RSGA-I	**Rough age based**	**Min point**	**66**	**-**	**0.4**	-	1609.21	0.94		
RSGA-II	**Rough extended age based**	**Min point**	**64**	**-**	**0.4**	-	1609.76	0.78		
SGA-V	Roulette Wheel	Min point	211	0.4	0.4	-	1608.32	2.05		
SGA-I	Roulette Wheel	Cyclic	411	0.5	0.4	-	1605.54	2.39		

Table 4.40: Comparison of RSGAs w. r. to different Mutations

Algorithm	Selection	Crossover	Mutation	Generation	P_m	Avg	SD	Result	Run
RSGA-I	**Rough Age Based**	**Min Point**	Simple	753	0.4	1607.34	0.78	1610	25
				598	0.3	1607.92	1.09		
				634	0.2	1606.54	0.85		
			Random	256	0.4	1608.46	0.75		
				145	0.3	1608.43	0.91		
				98	0.2	1606.63	1.82		
			Fixed	66	0.4	1608.21	0.77		
				71	0.3	1608.52	0.97		
				87	0.2	1607.78	1.03		
			Variable	**47**	-	1609.16	0.88		
				54	-	1609.02	0.93		
				56	-	1609.09	0.77		
RSGA-II	**Rough Extended Age Based**	**Min Point**	Simple	664	0.4	1607.53	1.28	1610	25
				564	0.3	1608.05	0.59		
				605	0.2	1607.03	0.96		
			Random	234	0.4	1608.57	0.67		
				121	0.3	1608.85	0.94		
				87	0.21	1608.43	1.09		
			Fixed	64	0.4	1608.34	1.11		
				68	0.3	1608.78	0.52		
				68	0.2	1608.45	0.99		
			Variable	**43**	-	1609.76	0.78		
				56	-	1609.44	0.97		
				45	-	1609.55	0.81		

keeping every thing same, with RW selection, higher value of p_c requires more number of generations and hence it is undesirable.

In Table 4.40, optimum results for the standard TSP, ”bayg29” are obtained in different environments using different selection and mutation techniques. It is observed that though all approaches furnish the same optimum result, the RSGAs with Min-point crossover and variable mutation takes minimum number of generations. In all cases, RSGAs with ”simple” mutation requires maximum numbers of generations for optimum results, where as ”random” and ”fixed” mutations take the value in between these numbers.

Model 4.4A: Experiments for CTSP with and with out time constraint:

Here we consider a deterministic TSP of 10×10 size given by Eq. 4.52, whose cost and time matrices are given in Table 4.41.

For the input data in Table 4.41, the problem given by Eq. 4.54 is solved by RSGAs and SGAs and the optimum results are presented below in Table 4.42. Here we took maximum generation=100, independent run of each algorithms and deterministic constraint are obtained using only Moore approaches.

From Table 4.42, it is observed that for the cases of CTSP having ”without

Table 4.41: Input Data: Interval CTSP (Model 4.4A1)

	Crisp Cost Matrix(10 $\times$10)									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	[35, 39]	[18, 23]	[20, 27]	[17, 19]	[36, 45]	[37, 42]	[42, 49]	[33, 34]	[44, 48]
2	[24, 30]	∞	[20, 26]	[28, 32]	[35, 39]	[40,44]	[30, 36]	[43, 47]	[28, 34]	[14,16]
3	[38, 42]	[27, 34]	∞	[25,28]	[22,26]	[35, 36]	[9,13]	[32,35]	[40,42]	[30,33]
4	[28, 32]	[10,14]	[7,12]	∞	[20,22]	[25,28]	[30,33]	[35,39]	[22,25]	[37,42]
5	[27,32]	[22,26]	[35,38]	[30,33]	∞	[20,24]	[25,30]	[30,33]	[9,13]	[28,33]
6	[15,17]	[30,33]	[25,30]	[8,12]	[28,30]	∞	[33,36]	[40,44]	[32,34]	[30,36]
7	[38, 44]	[25,32]	[30,33]	[22,24]	[37,39]	[40,44]	∞	[32,35]	[20,22]	[25,27]
8	[40,45]	[5,9]	[32,35]	[40,44]	[35,38]	[25,26]	[40,44]	∞	[37,39]	[38,42]
9	[40,42]	[40,46]	[23,26]	[25,29]	[20,25]	[2,5]	[37,45]	[32,35]	∞	[28,34]
10	[28,33]	[30,34]	[28,32]	[20,25]	[11,15]	[32,36]	[37,39]	[40,44]	[30,34]	∞
	Interval Time Matrix									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	[.5,.53]	[.8,.9]	[.7,.79]	[.82, .9]	[.59,.64]	[.58,.6]	[.59, .62]	[.6,64]	[.57, .6]
2	[.78,.84]	∞	[.81,.88]	[.75,.8]	[.5,.56]	[.6,.64]	[.7,.76]	[.58,.62]	[.75,.79]	[.9,.92]
3	[.59,.63]	[.79,.82]	∞	[.85,.87]	[.78,.84]	[.65,.7]	[.81,.86]	[.68,.72]	[.6,.64]	[.7,.76]
4	[.72,.76]	[.9,.92]	[.94,.95]	∞	[.8,.84]	[.75,.8]	[.7,.76]	[.65,.66]	[.78,.8]	[.63,.7]
5	[.83,.88]	[.79,.86]	[.69,.74]	[.72,.74]	∞	[.82,.88]	[.79,.82]	[.71,.73]	[.9,.92]	[.72,.74]
6	[.88,.9]	[.7,.74]	[.75,.77]	[.91,.92]	[.72,.75]	∞	[.67,.69]	[.6,.7]	[.7,.73]	[.77,.8]
7	[.68,.7]	[.59,.6]	[.8,.84]	[.7,.73]	[.6,.65]	[.61,.65]	∞	[.68,.7]	[.8,.83]	[.77,.8]
8	[.6,.64]	[.94,.95]	[.69,.73]	[.6,.63]	[.59,.62]	[.79,.81]	[.6,.62]	∞	[.59,.63]	[.73,.76]
9	[.6,.63]	[.81,.83]	[.77,.8]	[.75,.78]	[.8,.82]	[.9,.99]	[.63,.65]	[.68,.7]	∞	[.72,.74]
10	[.85,.9]	[.7,.76]	[.73,.75]	[.53,.55]	[.9,.96]	[.69,.73]	[.64,.66]	[.59,.63]	[.7,.74]	∞

Table 4.42: Optimum Results of CTSP in Crisp (Model 4.4A)

Algorithm	Selection	Mutation	Path	Gen	Value	Avg	SD	T_{max}	Run
Proposed RSGA	Rough Age (RSGA-I)	Random	6-4-3-7-10-5-9-8-2-1	87	[130, 166]	146.21	1.21	Without T_{max}	20
		Fixed	6-4-3-7-10-5-9-8-2-1	71	[130, 166]	147.53	0.92		
		Variable	6-4-3-7-10-5-9-8-2-1	42	[130, 166]	147.97	0.53		
	Rough Extended (RSGA-II)	Random	5-7-2-10-3-4-6-9-1-5-8	66	[102,117]	107.74	1.32		
		Fixed	5-7-2-10-3-4-6-9-1-5-8	59	[102,117]	108.75	1.45		
		Variable	8-2-10-5-9-6-1-4-3-7	28	[92,127]	104.32	0.82		
Proposed RSGA	Rough Age (RSGA-I)	Random	8-2-10-5-9-6-1-4-3-7	92	[155,182]	163.45	0.57	[7.0, 8.25]	
		Fixed	8-2-10-5-9-6-1-4-3-7	64	[155,182]	164.37	1.78		
		Variable	9-5-6-4-3-7-10-8-2-1	32	[143,165]	153.37	0.66		
	Rough Extended (RSGA-II)	Random	6-8-2-10-4-3-7-9-1-5	76	[149,169]	157.56	0.89		
		Fixed	6-8-2-10-4-3-7-9-1-5-7	53	[149,169]	157.8	0.56		
		Variable	4-8-2-10-5-9-6-1-3-7	28	[134,152]	142.76	1.16		
SGA-I	RW	Simple	10-8-2-5-9-6-1-4-3-7	188	[192, 259]	223.43	0.77	[7, 7.75]	
RSGA-II	REA	Variable	4-1-2-5-9-6-10-8-3-7	48	[165,189]	174.45	1.22	[6.5, 7.5]	
SGA-I	RW	Simple	1-2-5-10-4-3-7-9-6-4	231	[232,354]	290.7	0.99	[6.5, 7.5]	
RSGA-II	REA	Variable	7-2-6-9-1-4-8-5-10-3	45	[272, 312]	287.17	0.97	[5.5, 6.75]	

T_{max}” and with same ”T_{max}”, better results are obtained with respect to the costs as well as number of generations by RSGA-II and variable mutation. Also when T_{max} decreases, corresponding cost increases according to the realistic conditions of the present investigation.

Model 4.4A1: Experiment for CSTSPwR with time Constraint:

Now for a CSTSPwR, we consider maximum available three types of conveyances. The cost and time matrices for the CSTSPwR of 10×10 size are presented in Table 4.43.

Here we took maximum generation as 200 with 20 independent runs, and for proposed RSGA-II, REA selection with variable mutation is used. Only feasible constraints which are satisfied in the corresponding interval are considered. The optimum results are presented in Table 4.44.

Comparing the corresponding results from the given table, we see that Table 4.44 supports the usual expectation i.e., as the total travel times decrease, the corresponding total costs increase for all approaches. Considering all the approaches, Moore's approach with RSGA-II furnishes the lowest travel cost. This cost is much less than the corresponding costs with SGA-I for both values of T_{max}.

Model 4.4A1: CSTSPwR with virtual data set (for large TSP)

For the large scale data set, we randomly generate the costs within a range. Here, costs c_{Lij} and c_{Rij} ($i \neq j$) are taken for first, second and third conveyances respectively as follows:

c_{Lij}=10(1+ random integer with [0,8]), c_{Rij}=c_{Lij} +0.5(1+ random integer with [0,8])

c_{Lij}=9(1+ random integer with [0,8]), c_{Rij}=c_{Lij} +0.5(1+ random integer with [0,8])

c_{Lij}=11(1+ random integer with [0,8]), c_{Rij}=c_{Lij} +0.5(1+ random integer with [0,8])

Similarly randomly generated time matrix for three conveyances as follows

t_{Lij}=0.25(1+ random number with [0,1]), t_{Rij}=t_{Lij} +0.15(1+ random number with [0,1])

t_{Lij}=0.2(1+ random number with [0,1]), t_{Rij}=t_{Lij} +0.12(1+ random number with [0,1])

t_{Lij}=0.3(1+ random number with [0,1]), t_{Rij}=t_{Lij} +0.18(1+ random number on [0,1])

Table 4.43: Input Data: CSTSPwR (Model 4.4A1)

	Cost Matrix(10 *10) With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	[32,35] [36,42] [26,28]	[17,19] [38,42] [26,31]	[17,21] [31,33] [33,39]	[29,30] [20,21] [60,65]	[5,7] [22,27] [6,9]	[15,18] [35,37] [46,48]	[25,29] [37,43] [16,20]	[39,42] [26,33] [41,43]	[20,23] [30,34] [42,45]
2	[34,38] [22,27] [14,19]	∞	[40,44] [18,22] [27,33]	[16,19] [28,32] [6,12]	[32,37] [25,27] [34,38]	[39,41] [30,32] [21,26]	[39,42] [29,32] [57,60]	[30,34] [41,45] [58,62]	[17,22] [36,38] [17,21]	[23,26] [13,17] [17,20]
3	[36,39] [29,32] [28,32]	[16,20] [54,60] [31,35]	∞	[10,14] [24,26] [12,17]	[40,45] [23,26] [45,48]	[33,36] [34,39] [33,35]	[17,20] [11,13] [5,10]	[30,33] [30,34] [24,27]	[28,32] [18,21] [40,44]	[29,31] [19,23] [32,35]
4	[27,30] [18,21] [9,12]	[9,11] [19,23] [12,15]	[16,20] [7,10] [27,30]	∞	[29,33] [17,20] [23,24]	[23,26] [15,18] [25,28]	[19,22] [30,32] [30,33]	[33,36] [32,38] [16,18]	[10,13] [20,24] [32,35]	[24,29] [47,49] [37,40]
5	[16,19] [14,18] [6,9]	[41,44] [21,24] [32,34]	[34,37] [35,37] [33,39]	[17,21] [12,14] [40,44]	∞	[29,34] [20,23] [40,42]	[42,46] [14,18] [25,27]	[27,30] [30,32] [12,16]	[18,22] [8,11] [7,9]	[26,29] [25,27] [25,28]
6	[13,16] [5,8] [5,8]	[26,30] [20,23] [27,30]	[4,6] [25,27] [27,30]	[6,9] [7,11] [10,13]	[26,29] [26,30] [38,41]	∞	[31,34] [40,44] [23,26]	[39,42] [30,31] [20,23]	[30,33] [22,23] [35,36]	[28,31] [40,42] [30,34]
7	[36,39] [37,40] [28,32]	[23,26] [53,55] [25,27]	[27,32] [37,39] [24,27]	[21,24] [40,44] [23,25]	[35,38] [56,60] [37,40]	[38,41] [20,22] [43,46]	∞	[7,11] [40,44] [11,14]	[31,34] [33,35] [34,37]	[19,22] [13,16] [25,28]
8	[39,42] [41,43] [20,24]	[24,28] [5,7] [16,18]	[30,33] [52,54] [43,46]	[38,42] [19,22] [40,43]	[34,37] [34,37] [46,48]	[23,26] [15,18] [4,6]	[39,42] [19,22] [41,44]	∞	[20,23] [52,54] [23,27]	[35,38] [35,38] [39,41]
9	[38,41] [10,13] [31,33]	[39,42] [38,40] [34,37]	[4,9] [34,37] [36,39]	[23,26] [33,36] [28,30]	[20,23] [31,33] [20,22]	[22,25] [31,34] [23,26]	[5,8] [36,39] [38,41]	[30,33] [32,34] [11,15]	∞	[27,30] [18,20] [24,27]
10	[15,18] [25,28] [25,30]	[28,31] [20,22] [31,34]	[26,29] [18,20] [28,32]	[18,21] [29,32] [21,24]	[9,12] [32,34] [20,24]	[30,34] [10,13] [33,35]	[35,39] [26,29] [38,41]	[40,43] [41,43] [30,34]	[29,32] [51,54] [30,33]	∞
	Time Matrix(10 *10) With maximum available three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	[.7,.74] [.65,.66] [.7,.75]	[.81,.85] [.54,.57] [.6,.65]	[.68,.71] [.67,.7] [.63,.7]	[.65,.69] [.78,.83] [.3,.39]	[.79,.85] [.72,.74] [.8,.84]	[.68,.78] [.6,.69] [.45,.47]	[.7,.74] [.56,.64] [.67,.73]	[.54,.64] [.6,.65] [.45,.5]	[.71,.75] [.57,.64] [.4,.43]
2	[.5,.55] [.57,.66] [.68,.75]	∞	[.45,.47] [.66,.71] [.58,.61]	[.71,.77] [.56,.62] [.79,.85]	[.6,.65] [.67,.68] [.56,.58]	[.5,.54] [.58,.64] [.57,.65]	[.5,.65] [.56,.63] [.21,.26]	[.57,.59] [.45,.49] [.32,.34]	[.65,.68] [.56,.57] [.57,.69]	[.6,.64] [.68,.76] [.71,.76]
3	[.5,.56] [.56,.59] [.51,.58]	[.67,.69] [.3,.36] [.62,.68]	∞	[.71,.76] [.67,.71] [.68,.76]	[.45,.48] [.63,.66] [.4,.44]	[.5,.55] [.56,.59] [.56,.57]	[.68,.75] [.79,.86] [.68,.7]	[.56,.6] [.56,.6] [.6,.68]	[.5,.54] [.68,.76] [.45,.5]	[.6,.68] [.7,.76] [.54,.6]
4	[.6,.66] [.7,.75] [.78,.83]	[.78,.81] [.66,.76] [.71,.78]	[.7,.75] [.8,.85] [.57,.64]	∞	[.6,.63] [.7,.77] [.67,.71]	[.6,.66] [.71,.75] [.56,.67]	[.7,.74] [.6,.65] [.56,.59]	[.52,.56] [.51,.58] [.69,.76]	[.8,.88] [.71,.74] [.61,.69]	[.6,.69] [.41,.47] [.51,.54]
5	[.68,.76] [.71,.79] [.68,.79]	[.45,.52] [.7,.74] [.56,.58]	[.46,.58] [.5,.55] [.59,.65]	[.7,.75] [.6,.66] [.5,.54]	∞	[.56,.58] [.68,.75] [.45,.51]	[.41,.45] [.73,.78] [.57,.68]	[.56,.62] [.56,.61] [.72,.79]	[.68,.74] [.73,.81] [.69,.78]	[.6,.66] [.67,.78] [.79,.87]
6	[.68,.75] [.74,.79] [.78,.85]	[.56,.63] [.67,.72] [.56,.61]	[.78,.82] [.57,.68] [.56,.62]	[.76,.84] [.77,.84] [.78,.81]	[.57,.67] [.56,.59] [.48,.54]	∞	[.54,.6] [.45,.51] [.67,.78]	[.45,.52] [.54,.63] [.67,.74]	[.58,.61] [.73,.77] [.52,.58]	[.56,.61] [.45,.52] [.56,.62]
7	[.45,.54] [.46,.52] [.56,.62]	[.56,.63] [.34,.38] [.61,.65]	[.54,.6] [.56,.58] [.6,.64]	[.67,.73] [.45,.51] [.57,.68]	[.5,.54] [.28,.32] [.45,.53]	[.45,.51] [.67,.71] [.42,.47]	∞	[.78,.81] [.45,.54] [.65,.76]	[.56,.61] [.5,.54] [.54,.61]	[.67,.74] [.67,.74] [.51,.62]
8	[.45,.52] [.5,.55] [.45,.53]	[.67,.69] [.79,.88] [.58,.68]	[.54,.6] [.31,.38] [.41,.45]	[.5,.55] [.7,.77] [.5,.59]	[.52,.58] [.56,.64] [.42,.48]	[.45,.52] [.68,.77] [.78,.83]	[.44,.54] [.64,.72] [.45,.53]	∞	[.68,.76] [.35,.4] [.63,.7]	[.48,.56] [.5,.55] [.5,.59]
9	[.56,.58] [.8,.85] [.6,.65]	[.5,.55] [.5,.57] [.5,.55]	[.79,.85] [.6,.65] [.46,.54]	[.6,.68] [.7,.77] [.56,.58]	[.67,.75] [.56,.65] [.64,.7]	[.67,.7] [.6,.65] [.56,.64]	[.75,.81] [.56,.6] [.45,.54]	[.52,.6] [.5,.61] [.69,.75]	∞	[.6,.65] [.7,.77] [.62,.7]
10	[.57,.71] [.57,.67] [.59,.64]	[.66,.71] [.7,.74] [.68,.76]	[.59,.65] [.68,.76] [.58,.65]	[.64,.7] [.55,.6] [.6,.71]	[.73,.78] [.56,.66] [.55,.71]	[.45,.61] [.7,.83] [.58,.64]	[.49,.54] [.56,.64] [.49,.55]	[.45,.52] [.42,.54] [.54,.6]	[.54,.59] [.35,.41] [.51,.59]	∞

Table 4.44: Results of CSTSPwR(Model 4.4A1)

Algorithm	Optimum Path(Vehicle)	Cost	Avg	SD	Approaches	T_{max}	Run
RSGA-II	1(1)-10(1)-5(2)-4(1)-2(1)-9(1)-3(2)-7(1)-8(3)-6(2)	[98,112]	104.37	1.26	Moore	[7.75,8.75]	20
	9(1)-7(1)-8(1)-6(2)-1(1)-3(1)-4(1)-2(1)-10(1)-5(3)	[120,136]	127.68	1.17	Ishibuchi & Tanka		
	8(3)-2(2)-10(1)-9(1)-6(2)-4(2)-3(3)-7(2)-5(2)-1(3)	[116,128]	121.32	0.95	Chanas and Kuchta's		
	7(1)-8(1)-6(2)-1(1)-10(3)-5(2)-4(3)-2(3)-9(1)-3(2)	[128,156]	141.23	1.67	Hu and Wang's		
	10(3)-2(1)-9(1)-6(2)-4(2)-3(3)-7(2)-5(2)-1(3)-8(1)	[136,152]	143.752	0.89	Mahato and Bhunia's		
SGA-I	2(2)-9(1)-3(3)-7(3)-8(1)-6(2)-1(3)-10(2)-5(2)-4(1)	[183,215]	197.6	2.41	Moore		
RSGA-II	6(1)-4(3)-2(1)-9(3)-8(1)-5(2)-1(1)-3(2)-7(1)-10(1)	[124,136]	127.54	1.73	Moore	[6.5,7.5]	
	8(3)-10(3)-9(1)-3(1)-7(1)-2(1)-6(2)-1(1)-5(2)-4(1)	[133,156]	143.2	1.18	Ishabuchi & Tanka		
	6(1)-9(1)-10(2)-3(1)-7(1)-8(1)-2(2)-1(1)-5(2)-4(1)	[125,142]	132.7	1.52	Chanas and Kuchata's		
	4(1)-7(2)-9(1)-3(3)-10(1)-8(1)-6(2)-1(1)-5(2)-2(2)	[164,177]	169.32	1.19	Hu and Wang's		
	4(2)-10(3)-9(1)-3(1)-7(1)-8(1)-5(2)-1(1)-6(2)-2(1)	[145,158]	150.7	1.02	Mahato and Bhunia's		
SGA-I	2(3)-6(3)-9(2)-3(2)-7(1)-1(2)-8(2)-10(1)-5(2)-4(1)	[217,246]	230.5	1.09	Moore		

Data sets are randomly generated using rand() function of C programming language. For the CSTSPwR, cost and time matrices for different size problems (N=20, 40,60, 80,100, 150 and 200) are generated randomly. To derive the results, we considered the CSTSPwR formulations in Eqs. 4.54 - 4.58 and solved using RSGA-II. For comparison, SGA-I with Moore's approach (Eq. 4.54) only is taken. The optimum results of this randomly generated CSTSPwRs are presented in Table 4.45. From Table 4.45, it is evident that for all sizes of CSTSPwR, Moore's approach gives the best results i.e. minimum costs are less than the other four approaches. For comparison, here SGA-I with Moore's approach is used and it is seen that RSGA-II fetches much better results than SGA-I for all sizes of cost matrix. SGA-I gives the worst result than all other approaches for a particular size of cost matrix.

4.5.4 Statistical test for RSGAs

i. Against different test problems only:

Performance of the proposed method is statistically tested against 25 separate runs and calculating the average value, standard deviation and percentage relative error according to optimal solution against some standard test problems. The results obtained by proposed method are given in Table 4.46. Examining the Table 4.46, it is concluded that the proposed method, RSGA-II has generated the closer results to the optimal solutions with minimal standard deviations for the problems fri26, bays29, bayg29, dantzig42, eil51, berlin52, st70, pr76, rat99, Lin105 and Eil101. It can be seen that except one problem kroa200, for all other fourteen problems, best results by RSGA-II are the same as the corresponding

Table 4.45: For Virtual Data (Model 4.4A1)

Instances (Cities)	Algorithm	Approaches	Costs	T_{max}	Run
20×20	RSGA-II	Moore	[232, 278]	[14.4, 21.8]	20
		Ishibuchi	[254, 289]		
		Chanas and Kuchta's	[267, 308]		
		Hu and Wang's	[282, 320]		
		Mahato and Bhunia's	[280, 316.5]		
	SGA-I	Moore	[328, 360]	[21.25, 34.7]	
40×40	RSGA-II	Moore	[466, 513]	[19.7, 28.4]	
		Ishibuchi	[490, 567]		
		Chanas and Kuchta's	[546, 587]		
		Hu and Wang's	[544, 593.5]		
		Mahato and Bhunia's	[576, 632.5]		
	SGA-I	Moore	[598, 648]	[23.7, 27.3]	
60×60	RSGA-II	Moore	[690, 810]	[23.5, 33.6]	
		Ishibuchi	[719, 842]		
		Chanas and Kuchta's	[765, 896]		
		Hu and Wang's	[784, 867]		
		Mahato and Bhunia's	[793, 923]		
	SGA-I	Moore	[839, 978]	[24.25, 34.5]	
80×80	RSGA-II	Moore	[1042, 1224]	[26.5, 64.7]	
		Ishibuchi	[1196, 1376]		
		Chanas and Kuchta's	[1264, 1433]		
		Hu and Wang's	[1335, 1479]		
		Mahato and Bhunia's	[1273, 1452]		
	SGA-I	Moore	[1598, 1864]	[41.25, 62.7]	
100×100	RSGA-II	Moore	[1425, 1663]	[46.5, 77.8]	
		Ishibuchi	[1562, 1678]		
		Chanas and Kuchta's	[1647, 1783]		
		Hu and Wang's	[1724, 1890]		
		Mahato and Bhunia's	[1738, 1975]		
	SGA-I	Moore	[1983, 2251]	[64.2, 76.4]	
150×150	RSGA-II	Moore	[2042, 2490]	[70.6, 121.7]	
		Ishibuchi	[2368,2789]		
		Chanas and Kuchta's	[2945, 3272]		
		Hu and Wang's	[3243, 3364]		
		Mahato and Bhunia's	[2173, 2505]		
	SGA-I	Moore	[3581, 3743]	[91.25, 135.2]	
200×200	RSGA-II	Moore	[2724, 3602]	[104.3, 176.2]	
		Ishibuchi	[3192, 3945]		
		Chanas and Kuchta's	[3347, 4178]		
		Hu and Wang's	[3441, 4732]		
		Mahato and Bhunia's	[3703, 4926]		
	SGA-I	Moore	[4398, 5231]	[141.5, 188.9]	

Table 4.46: Dispersion Results of RSGA-II

Instances	BKS	Best	Worst	Average	SD[b]	Error(%)
fri26	937	937	939	937.32	1.31	0.19
bays29	2020	2020	2034	2020.25	2.37	1.21
bayg29	1610	1610	1616	1610.42	0.46	0.24
dantzig42	699	699	704	700.71	1.52	1.49
eil51	426	426	429	427.15	0.98	0.17
berlin52	7542	7542	7567	7544.45	0.76	1.37
st70	675	675	686	679.4	1.43	0.23
eil76	538	538	557	543.3	23.57	0.53
pr76	108159	108159	108343	108211.73	2.12	2.70
rat99	1211	1211	1220	1217.5	0.74	0.29
Kroa100	21282	21282	21604	21432.30	56.17	1.07
Lin105	14379	14379	14431	14387.25	1.35	0.94
Eil101	629	629	646	629.7	1.23	0.07
Ch105	6528	6528	6636	6543.7	31.62	3.46
Kroa200	29368	29468	29874	297036.15	103.28	2.87

best results available in the literature.

ii. Against different test problems and different algorithms:

In Table 4.47, average values of SDs and the corresponding errors have been calculated for eleven problems using seven methods. In all cases, the average results given by RSGA-I and RSGA-II are less than the corresponding average results by SGA-I, II, III, IV and V. Moreover, as the SD's in RSGA-I and -II are quite small except three cases, it indicates that these methods are stable, results in different runs do not differ much from the mean. We also obtain the least percentage relative error in different cases. These errors are also very small indicating that derived average solutions are nearer to the best known solution in the literature. Thus the proposed methods have produced closer results to optimum.

iii. The Friedman Test:

To compare the performance of the algorithms SGA-I, II, III, IV, V, RSGA-I and RSGA-II, we perform the Friedman test (cf Derrac et al. [40]). Since it is a non parametric statistical procedure whose main aim is to detect significant difference between the behavior of two or more algorithms.

The assumptions of Friedman test are:

- The results over instances (problems from TSPLIB) are mutually independent (i.e. the results within one instances does not influence the results within other instances)

Table 4.47: Results of RSGA and Other Methods

Algorithm	Problem	fri26	bays29	bayg29	dantzig42	eil51	berlin52	st70	eil76	pr76	rat99	kroa100
	BKS⇒	937	2020	1610	699	426	7542	675	538	108159	1211	21282
SGA-I	Avg	989.23	2076.9	1639.5	731.4	452.3	7667.4	722.84	584.5	108354.5	1246.7	21757.67
	SD	5.93	1.73	20.48	5.78	7.65	7.32	5.90	5.5	31.58	24.7	34.9
	Error(%)	1.75	0.78	1.82	1.80	0.68	0.84	2.89.	0.97	4.3	2.57	3.85
SGA-II	Avg	984.3	2075.2	1637.8	732.6	452.7	7666.8	721.7	580.63	108360.5	1244.65	21731.3
	SD	2.37	1.80	1.48	2.70	26.58	10.32	4.64	6.56	52.8	24.7	87.98
	Error(%)	1.52	0.92	0.74	1.79	0.81	0.59	3.45	2.7	3.65	0.86	2.97
SGA-III	Avg	979.73	2074.4	1637.71	728.47	450.8	7655.41	718.58	578.3	108725.42	1243.6	21702.3
	SD	2.23	1.71	2.48	1.23	11.56	1.32	2.19	01.35	2.15	1.7	32.5
	Error(%)	0.74	2.72	1.95	0.56	1.79	1.02	0.86	1.75	3.22	1.63	2.7
SGA-IV	Avg	966.37	2056.21	1633.8	721.4	445.92	7617.46	712.72	562.2	108674.61	1233.7	21678.7
	SD	4.13	1.98	2.54	3.01	2.51	1.33	2.32	1.56	2.58	2.37	84.89
	Error(%)	2.7	3.01	1.72	2.8	1.81	0.93	1.06	2.45	0.87	1.36	2.11
SGA-V	Avg	958.52	2050.3	1627.43	717.42	442.7	7612.9	701.25	558.2	108521.75	1231.53	21502.26
	SD	2.63	2.81	1.48	2.17	1.65	0.82	1.9	0.76	2.08	4.7	78.91
	Error(%)	1.12	1.78	0.95	2.36	1.02	1.9	0.93	1.78	4.45	2.31	3.27
RSGA-I	Avg	953.2	2036.17	1621.43	710.12	432.8	7589.6	686.2	544.3	108344.8	1223.49	21457.2
	SD	2.76	2.75	0.54	1.78	1.15	1.02	2.31	0.61	2.58	1.03	67.8
	Error(%)	0.78	1.39	0.3	1.51	0.67	1.59	0.76	0.25	2.72	0.35	1.23
RSGA-II	Avg	937.32	2020.25	1610.42	700.7	427.15	7544.45	679.4	543.3	108211.5	1217.5	21432.3
	SD	1.31	2.37	0.46	1.52	0.98	0.76	1.43	23.57	2.12	0.71	56.17
	Error(%)	0.19	1.21	0.24	1.49	0.17	1.37	0.23	0.53	2.7	0.29	1.07

- Within each instance, the observations (average objective values) can be ranked.

Hypothesis:

H_0: Each ranking of the algorithms within each problem is equally likely, (i.e., there is no difference between them)

H_1: At least one of the algorithms tends to yield larger average objective values than at least one of the other algorithms

Here number of algorithms (k)=7, number of instances (b)=11. The Friedman ranking table is given in Table 4.48 which is prepared according to the average results of Table 4.47.

Now $A_2=\sum_{i=1}^{b}\sum_{j=1}^{k}[R(X_{ij})]^2$, $R_j=\sum_{1}^{b} R(X_{ij})$ for j=1, 2,..,k, and $B_2=\frac{1}{b}\sum_{j=1}^{k} R_j^2$.

The test statistic is given by: $T_2=\frac{(b-1)[B_2-bk(k+1)^2/4]}{A_2-B_2}$

Hence from the Table 4.48, we calculate

$$A_2=473+402+299+196+115+44+11=1540,$$

$$B_2=\frac{1}{11}[71^2+66^2+57^2+46^2+35^2+22^2+11^2]=1508.36$$

With the values of A_2 and B_2, calculate the test statistic,

$$T_2=\frac{(11-1)[1508.36-11\times 7(7+1)^2/4]}{1540-1508.36}=87.34$$

Using a table for the F distribution with a significance level α= 0.01, we find that

$$F_{(1-\alpha),(k-1),(b-1)(k-1)}=F_{0.99,6,60}=3.12$$

Table 4.48: Ranking of the Friedman Test

Algorithms(k)	SGA-I	SGA-II	SGA-III	SGA-IV	SGA-V	RSGA-I	RSGA-II
Instances(b)	R(X_{b1})	R(X_{b2})	R(X_{b3})	R(X_{b4})	R(X_{b5})	R(X_{b6})	R(X_{b7})
fri26	7	6	5	4	3	2	1
bays29	7	6	5	4	3	2	1
bayg29	7	6	5	4	3	2	1
dantzig42	6	7	5	4	3	2	1
eil51	6	7	5	4	3	2	1
berlin52	7	6	5	4	3	2	1
st70	7	6	5	4	3	2	1
eil76	7	6	5	4	3	2	1
pr76	3	4	7	6	5	2	1
rat99	7	6	5	4	3	2	1
kroa100	7	6	5	4	3	2	1
Average Rank	6.45	6	5.18	4.18	3.18	2	1
Summation	71	66	57	46	35	22	11

Since $T_2 > F_{0.99,6,60}$, we reject the null hypothesis. Hence there exist some algorithms whose performances are significantly different from the others.

iv. (Post Hoc) Paired Comparisons:

Here if the algorithms a and b are considered different after the rejection of the null hypothesis from the Friedman test, following the Post Hoc paired comparison technique (cf. Derrac et al. [39]), calculate the absolute differences of the summation of the ranks of algorithms a and b and declare a and b different if :

$$|R_a - R_b| > t_{1-\frac{\alpha}{2}} \left[\frac{2b(A_2-B_2)}{(b-1)(k-1)}\right]^{\frac{1}{2}}$$

where $t_{1-\frac{\alpha}{2}}$ is the 1-$\frac{\alpha}{2}$ quantile of the t-distribution with (b-1)(k-1) degrees of freedom. Here $t_{1-\frac{\alpha}{2}}$ for α=0.01 and 60 degrees of freedom is 2.660 and the critical value for the difference is: $2.66[\frac{2*11(1540-1508.36)}{(11-1)(7-1)}]$= 9.06.

The Table 4.49 summarizes the paired comparisons, underline values indicated that the algorithms are different. From the Table 4.49, we conclude that, RSGA-I and RSGA-II have outperformed than all other algorithms and RSGA-II is the best out performer amongst the other algorithms.

4.5.5 Discussion

Here, a GA (RSGAs) has been proposed with rough set based selection, min-point crossover and generation dependent mutation processes. Here rough

Table 4.49: Paired Comparison of the Friedman Test

$\|R_i - R_j\|$	SGA-II	SGA-III	SGA-IV	SGA-V	RSGA-I	RSGA-II
SGA-I	5	14	25	36	$\underline{49}$	$\underline{60}$
SGA-II	-	9	20	31	$\underline{44}$	$\underline{55}$
SGA-III	-	-	11	22	$\underline{35}$	$\underline{46}$
SGA-IV	-	-	-	11	$\underline{24}$	$\underline{35}$
SGA-V	-	-	-	-	$\underline{13}$	$\underline{24}$
RSGA-I	-	-	-	-	-	$\underline{11}$

set based age dependent selection with 3 and 5 (extended) classifications, min-point crossover and three different p_m dependent mutations are developed. For STSPs, p_m oriented random mutation accelerates to get wide variety of node combinations. If p_m is high, then the mutation rate is also much high. So it is in fine tuning to the optimization problem, particularly this type of node oriented problems such as TSP, vehicle routing problem, network optimization etc. Again in accounting the complexity in mutation mechanism, type-I is much high against the type-II because in type-I, randomly exchange occurs with searching the node in each step of the mutation where as type-II does no search in location exchange. But type-I is more affective to find the global optimum. For type-III, its complexity is better against other two and efficiency is much high. With these new features, RSGAs are used for test problems from TSPLIB and its efficiency is proved. The supremacy of RSGA is established through the Friedman test and Post hoc paired comparison. Later, two TSP problems-constrained TSP and constrained Solid TSP are solved and the optimum results along with near optimum results are presented. The developed RSGAs are quite general, these can be used for the decision making problems in other areas such inventory control system, supply-chain, portfolio management, etc. Moreover, RSGAs will be very useful for the large problems with large scale data. The proposed RSGAs can be extended/modified to be applied for the optimization of multi-objective problems.

4.6 Model-4.5: A Rough extended Genetic Algorithm to solve Constrained Solid Travelling Salesman Problem with Bi-Fuzzy Costs[4]

In this model, a Rough extended Genetic Algorithm (ReGA) is proposed to solve constrained solid travelling salesman problems (CSTSPs) with crisp and bi-fuzzy costs. In the proposed ReGA, a 'rough set based selection' (7-point scale) technique and comparison crossover with generation dependent mutation are developed. In CSTSP, the costs and risk/discomfort factors are in the form of crisp and bi-fuzzy in nature. In this investigation, CSTSPs are illustrated numerically by some standard test data from TSPLIB [162] using ReGA. In each environment, some statistical significance studies through ANOVA due to different risk/discomfort factors and other system parameters are presented.

4.6.1 Proposed ReGA

The proposed algorithm ReGA consists of the rough set based selection (7-point), comparison crossover and generation dependent random mutation. The proposed ReGA and its procedures are presented below:

i. Representation:

Here a complete tour of N cities represents a solution. So an N dimensional integer vector $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$ and $Y_i = (v_{i1}, v_{i2}, ..., v_{iP})$ are used to represent a solution, where $x_{i1}, x_{i2}, ..., x_{iN}$ represent N consecutive cities in a tour and $Y_i = (v_{i1}, v_{i2}, ..., v_{iP})$ represents the corresponding conveyances. Populations of such solutions $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$, and $Y_i = (v_{i1}, v_{i2}, ..., v_{iP})$ i = 1, 2, ..., N, are randomly generated by random number generator. Let the population size be M.

ii. Rough set based selection:

The above M such solutions have fitnesses represented by $f(x_i)$ of the i_{th} chromosomes. At the time of initialization, each chromosome age is defined as null. Now in every generation the age is counted as using the Equ. 4.48. Here the maximum and minimum ages depend on the requirement of the problems.

[4]This portion is published in **Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA) 2015, Springer**, Advances in Intelligent Systems and Computing 404, DOI 10.1007/978-81-322-2695-6_36, with title *Constrained Solid Travelling Salesman Problem Solving by Rough GA Under Bi-Fuzzy Coefficients.*

Table 4.50: Rough extended trust based linguistic

Gene	VVY	VY	Y	M	O	VO	VVO
VVY	VVL	VL	VL	L	VL	VL	VVL
VY	VL	VL	L	M	L	VL	VL
Y	VL	L	L	H	L	L	VL
M	L	M	H	VH	H	M	L
O	VL	L	L	H	L	L	VL
VO	VL	VL	L	M	L	VL	VL
VVO	VVL	VL	VL	VL	VL	VL	VVL

Now since the ages are calculated as crisp values, we construct the common rough values form it,
Rough Age=([r_1*avg age, r_2*avg age], [r_3*avg age, r_4*avg age]), where
r_1=$\frac{Max\ Age-Avg\ Age}{Avg\ Age}$, r_2=$\frac{Max\ Age+Min\ Age}{2}$, r_3=$\frac{Max\ Age-Min\ Age}{2}$, r_4=$\frac{Avg\ Age-Min\ Age}{Avg\ Age}$

According to the extended age of the chromosome in Equ. 3.24, (mathematical expression are given in section 3.5.1), it belongs to any one of the common rough age and corresponding p$_c$s are created of each chromosome as VVL, VL, L, M, H, VH, VVH. The common rough age ([a,b],[c,d]) is extended to $0 \leq c \leq e \leq f \leq a \leq b \leq g \leq h \leq d$ and is described as below,

$$Age = \begin{cases} VeryVeryYoung(VVY)\ for\, c \leq age < e \\ VeryYoung(VY) \quad \text{for } e \leq age < f \\ Young(Y) \quad \text{for } f \leq age < a \\ Middle(M) \quad \text{for } a \leq age \leq b \\ Old(O) \quad \text{for } b < age \leq g \\ VeryOld(VO) \quad \text{for } g < age \leq h \\ VeryVeryOld(VVO) \quad \text{for } h < age \leq d \end{cases} \tag{4.59}$$

iii. Comparison Crossover:

(a). Determination Probability of Crossover (p$_c$): Probability of crossover (p$_c$), for a pair of chromosomes (X$_i$,X$_j$) is determined as below:

A. p$_c$s for rough set based age selection

(i) At first age levels, (VVY, VY, M, O, VO, VVO) of X$_i$ and X$_j$ are determined by making trust measure of rough values w.r.to their ages in common rough age region given in Equ. 4.59.

(ii) After determination of age intervals of the chromosomes, their crossover probabilities are determined as linguistic variables (VVL, Vl, L, M, H, VH, VVH) as in Fig 3.5.1 using rough trust measure which is presented in Table 4.50 and trust levels are given as Equ. 4.59.

(b). Crossover Mechanism: Here we used comparison crossover method. We choose two individuals(parents) to produce new individuals(child's). To get optimal result of a TSP, we take a tour from one node(city)to next node(city) with minimum cost/value. we construct the crossover mechanism according to the section 4.3.1(c) (iii).

iv. Generation Dependent Random Mutation

(a) Selection for mutation: For each solution of P(t), generate a random number r from the range [0,1]. If $r < p_m$ then the solution is taken for mutation where p_m be the probability of mutation.

(b) Generation Oriented Mutation(Variable Method): Here we model a new form of mutation mechanism where probability of mutation (p_m) are determined as follows

$$p_m=\frac{k}{\sqrt{Current\ generation\ number}},\ k\in[0,1].$$

Here p_m decreases smoothly as generation increases. After calculating the p_m, then mutation operation is performed as the conventional random mutation.
Here we randomly choose two nodes from each chromosome and exchange their place and replace the chromosome in the new offspring set.

v. Algorithm of ReGA

Input: max_ gen, pop_ size, Max_age, Min_ age, Problem Data (cost matrix, risk matrix).

Output: The optimum and near optimum solutions.

1. **Start**
2. g ← 0 // g: generation number
3. **Initialize** $\mathbf{P(g)}$
4. **Evaluate** $\mathbf{f(P(g))}$;
5. **while** (g $\leq max\_gen$)
6. Evaluate the average fitness
7. **if** average fitness $>$ current fitness
8. age(x_i)=avg(age)+$\frac{k*(avgfit-f(X_i))}{(avgfit-minfit)}$
9. **else**
10. age(x_i)=$\frac{avg(age)}{2}+\frac{k*(f(X_i)-avgfit)}{(maxfit-avgfit)}$
11. **if** (age(x_i)$>$ maximum age)
12. age(x_i)= maximum age

13. **else if** (age(x_i)< minimum age)
14. age(x_i)= minimum age
15. Determine average age
16. Determine common rough age
17. Developed VVY, VY, M, O, VO, VVO
18. **for** each pair of parents **do**
19. Trust based p_c created
20. **end for**
21. **for** i=1 to Pop Size//**Comparison crossover**
22. Choose pair of chromosomes according to p_c
23. Randomly generate node between 1 to N (say a_r)
24. Replace a_r at first place of each parents
25. Determine value at each corresponding node
26. **for** j=1 to N
27. Compare minimum value
28. Check the node existence in child
29. Concatenated node to the child (offspring)
30. **end for**
31. Replace a_r at end place of each parents
32. Compare minimum value from end of the parents
33. Repeat step 26 to step 30
34. Replace the children in offspring's set
35. **end for**
36. $p_m=\frac{k}{g}$, k∈[0,1]
37. **for** i=0 to pop_size
38. Select chromosome depending on p_m
39. Randomly select two different nodes in [1,N]
40. Swap the places of the selected two nodes
41. **end for**
42. Store the new off springs into offspring set
43. **Reproduce a new P(g)**
44. **Evaluate f(P(g))**
45. Store the local optimum and near optimum solutions
46. g ← g+1
47. endwhile

48. Store the global optimum and near optimum results
49. End Algorithm.

vi. Complexity of the ReGA:

(a). Time Complexity:

The time complexities of selection operator, crossover operator and mutation operator in genetic algorithm are O(MN), O(Mp_cN^2), O(Mp_mN^2) respectively, where M is the size of the population. Normally $p_c > p_m$, so O(Mp_cN^2) > O(Mp_mN^2)>O(MN). If s_0 is the number of generations (outer iterations), so the time complexity of the outer loop is O(s_0MN^2). The time complexity of initial population generation and fitness function calculation are O(MN) <O(s_0MN^2). As O(MN) < O(s_0MN^2), the time complexity of the GA is O(s_0MN^2).

(b). Space Complexity:

Genetic algorithm needs to save the populations, so it needs MN of the space. Normally M>N, so GA space complexity is O(MN).

4.6.2 Mathematical Formulation and Its crisp equivalence

Model 4.5A: STSP with risk/discomfort Constraints (CSTSP):

Let $c(i,j,k)$ be the cost for travelling from i-th city to j-th city using k-th type conveyance and $r(i,j,k)$ be the risk/discomfort factor in travelling from i-th city to j-th using k-th type conveyance. Then the salesman has to determine a complete tour (x_1, x_2, ...,x_N, x_1) and corresponding conveyance types (v_1, v_2, ..., v_P) to be used for the tour, where $x_i \in \{1,2,..N\}$ for $i = 1,2,...,N$, $v_i \in \{1,2,..P\}$ for $i = 1,2,...,N$ and all x_i are distinct. Then the problem can be mathematically formulated as:

$$\left.\begin{array}{l} \text{to minimize} \quad Z = \sum_{i=1}^{N-1} c(x_i, x_{i+1}, v_i) + c(x_N, x_1, v_l), \\ \text{subject to} \quad \sum_{i=1}^{N-1} r(x_i, x_{i+1}, v_i) + r(x_N, x_1, v_l) \le r_{max}, \\ where\ x_i \neq x_j, i,j = 1,2...N, \quad v_i, v_l \in \{1,2..,or P\} \end{array}\right\} \tag{4.60}$$

Model 4.5A1: CSTSP in bi-fuzzy Environment (BFCSTSP):
In the above problem Equ. 4.60, if costs and risk/discomfort factors are bi-fuzzy variables, i.e, $\tilde{\tilde{c}}(i,j,k)$ and $\tilde{\tilde{r}}(i,j,k)$ respectively, risk/discomfort limit r_{max} is also bi-fuzzy number $\tilde{\tilde{r}}_{max}$, then the above problem reduces to (according Theorem 3.2).

$$\left.\begin{array}{l}\text{to minimize } \; Z=\sum_{i=1}^{N-1}\tilde{\tilde{c}}(x_i,x_{i+1},v_i)+\tilde{\tilde{c}}(x_N,x_1,v_l),\\ \text{subject to } \; \sum_{i=1}^{N-1}\tilde{\tilde{r}}(x_i,x_{i+1},v_i)+\tilde{\tilde{r}}(x_N,x_1,v_l)\leq \tilde{\tilde{r}}_{max},\\ where\ x_i\neq x_j, i,j=1,2...N, \;\; v_i,v_l\in\{1,2..,orP\}\end{array}\right\} \tag{4.61}$$

The problem in Equ. 4.61 under Pos-Pos measures are equivalently written as below:

$$\left.\begin{array}{c}\text{minimize } f\\ Pos\{\theta|Pos\{|\tilde{\tilde{C}}(\theta)^T x\leq f\}\geq\delta\}\geq\gamma\\ Pos\{\theta|Pos\{|\tilde{\tilde{R}}(\theta)^T x\leq\tilde{\tilde{r}}_{max}(\theta)^T\}\geq\theta\}\geq\eta\end{array}\right\} \tag{4.62}$$

The problem in Equ. 4.61 under Nes-Nes are equivalently written as below:

$$\left.\begin{array}{c}\text{minimize } f\\ Nes\{\theta|Nes\{|\tilde{\tilde{C}}(\theta)^T x\leq f\}\geq\delta\}\geq\gamma\\ Nes\{\theta|Nes\{|\tilde{\tilde{R}}(\theta)^T x\leq\tilde{\tilde{r}}_{max}(\theta)^T\}\geq\theta\}\geq\eta\end{array}\right\} \tag{4.63}$$

The Equ. 4.62 is transformed as

$$\left.\begin{array}{c}\text{minimize } c^T-L^{-1}(\delta)\alpha_1^{cT}-L^{-1}(\gamma)\alpha_2^{cT}\\ \text{s.t. } r_{max}-R^T+R^{-1}(\theta)\beta_1^{r_{max}}+L^{-1}(\theta)\alpha_1^{r_{max}T}\\ +L^{-1}(\eta)(\alpha_2^{RT}+\beta_2^{r_{max}}\geq 0\end{array}\right\} \tag{4.64}$$

The Equ. 4.63 are equivalently written as below:

$$\left.\begin{array}{c}\text{minimize } c^T+R^{-1}(1-\delta)\beta_1^{cT}+R^{-1}(1-\gamma)\beta_2^{cT}\\ \text{s.t. } r_{max}-R^T-L^{-1}(1-\eta)(\alpha_2^{r_{max}}+\beta_2^{RT})\\ -L^{-1}(1-\theta)\alpha_1^{r_{max}}-R^{-1}(\theta)\beta_1^{RT}\geq 0\end{array}\right\} \tag{4.65}$$

where $\alpha_1^c,\alpha_2^c,\alpha_1^R,\alpha_2^R,\alpha_1^{r_{max}},\alpha_2^{r_{max}},\beta_1^c,\beta_2^c,\beta_1^R,\beta_2^R,\beta_1^{r_{max}},\beta_2^{r_{max}}$ are corresponding left and right spreads of the reference function of LR fuzzy numbers and $\theta,\eta,\delta,\gamma$ are predetermined confidence levels.

Table 4.51: Test TSPLIB Problems by ReGA

Instances	Result	ReGA Cost Avg	RGA Iteration	RGA SD Error(%)	SGA Cost Avg	SGA Iteration	SGA SD Error(%)
fri26	937	937 938.74	**43**	0.73 0.87	937 940.71	269	2.65 3.46
bays29	2020	2020 2023.4	**53**	1.48 1.65	2020 2027.79	451	2.81 3.21
bayg29	1610	1610 1611.52	**62**	0.45 0.98	1610 1615.71	378	3.57 2.63
dantzig42	699	699 700.35	**140**	1.72 0.07	699 704.75	612	3.27 2.87
eil51	426	426 427.38	**79**	0.68 1.17	426 429.38	341	2.01 2.78
berlin52	7542	7542 7548.75	**120**	1.62 0.63	7542 7562.29	526	4.31 2.57
st70	675	675 676.25	**154**	1.38 1.01	675 679.45	813	2.4 4.25
eil76	538	538 540.73	**113**	0.97 0.69	538 543.27	457	2.47 1.64
pr76	108159	108159 108180.34	**151**	1.05 0.74	108159 108243.39	410	2.13 4.06
rat99	1211	1211 1213.76	**135**	1.34 0.57	1211 1217.43	328	3.63 3.36
kroa100	21282	21282 21284.75	**262**	1.78 1.05	21282 21289.9	285	4.73 3.65

4.6.3 Numerical Experiments

Testing for ReGA:

We select some standard TSP problems from TSPLIB [162]. Table 4.51 gives the results of the test problems using both ReGA and SGA and a comparison is made in terms of total cost and iterations. Here SGA is the combinations of RW selection, cyclic crossover with well known random mutation. We have taken the results under 25 independent runs. The best optimal solution with standard deviation (SD) and error are presented.

Model 4.5A: CSTSP with Risks/Discomforts Constraint in Crisp values

Now for a CSTSP, we consider the TSP formulation with three types of conveyances as Equ. 4.9. The cost matrix for the CSTSP and corresponding risk/discomfort matrix are presented in Table 4.52. With these input data, we solve the CSTSP using ReGA and SGA. The optimum results are given in Table 4.53. Here we have taken maximum generation=1000, and we see that as risk factor decreases the corresponding cost increases as per real life expectation.

Table 4.52: Input Data: Crisp CSTSP (Model 4.5A)

	Crisp Cost Matrix(5 *5) With Three Conveyances				
i/j	1	2	3	4	5
1	∞	(35,36,27)	(18,39,30)	(20,33,34)	(30,21,62)
2	(35,26,17)	∞	(40,21,32)	(18,29,10)	(35,26,37)
3	(38,30,29)	(17,58,34)	∞	(12,25,14)	(42,25,46)
4	(28,20,11)	(10,22,14)	(17,8,29)	∞	(30,19,24)
5	(17,15,9)	(42,23,34)	(35,36,37)	(20,31,43)	∞
	Crisp Risks/Discomforts Matrix(5*5) With Three Conveyances				
i/j	1	2	3	4	5
1	∞	(.69,.68,.75)	(.84,.63,.7)	(.82,.7,.71)	(.72,.8,.42)
2	(.67,.76,.84)	∞	(.61,.8,.7)	(.83,.73,.92)	(.67,.76,.65)
3	(.63,.71,.73)	(.83,.44,.67)	∞	(.89,.76,.86)	(.59,.76,.75)
4	(.73,.81,.9)	(.9,.78,.86)	(.84,.93,.72)	∞	(.71,.82,.77)
5	(.84,.86,.92)	(.59,.78,.67)	(.66,.65,.64)	(.82,.71,.59)	∞

Table 4.53: Results of crisp CSTSP (Model 4.5A)

Algorithm	Path(Vehicle)	Cost	Risk	R_{max}
ReGA	3(1)-4(1)-2(2)-5(3)-1(1)	70	4.50	4.75
	2(3)-1(1)-3(1)-4(2)-5(2)	89	4.52	
	1(1)-3(1)-2(2)-5(1)-4(3)	92	4.3	
	1(2)-5(1)-4(1)-2(1)-3(3)	93	4.54	
	4(1)-5(2)-1(2)-2(2)-3(2)	127	4.16	
SGA	2(3)-3(3)-1(3)-5(1)-4(2)	142	4.7	4.75
ReGA	4(2)-5(3)-2(1)-1(1)-3(2)	138	4.25	4.25
	1(2)-3(2)-2(2)-5(1)-4(3)	154	3.68	4.00
	4(3)-5(1)-2(3)-3(3)-1(2)	160	3.71	

Model 4.5A1: CSTSP in bi-fuzzy Environments (BFCSTSP):

Here we take the costs and risk/discomfort factors as bi-fuzzy for the CSTSP as Equ. 4.64 and Equ. 4.65. Also we consider three types of conveyances. Here we use triangular LR Fu-Fu variables where (ξ, α, β) is LR fuzzy variable with known left and right spreads. Here ξ is also a triangular fuzzy variable whose values are the corresponding values given in in Table 4.52.

Here predetermined confidence levels δ= γ=0.9, θ=η=0.95 and reference function L(x)=R(x)=1-x are taken. Left and right spreads of the LR fuzzy numbers are given in the Table 4.54. With these input data, the BFCSTSP under possibility (optimistic) and necessity (pessimistic) measures are solved by ReGA and the optimum results are presented in Table 4.55 as a DM may be a optimistic (ODM) or pessimistic (PDM).

Table 4.54: Input Data: (BFCSTSP) (Model 4.5A1)

i/j	1	2	3	4	5
1	∞	(ξ, 2, 2) (ξ, 4, 4) (ξ, 1, 1)	(ξ, 3, 3) (ξ, 5, 5) (ξ, 2, 2)	(ξ,4, 4) (ξ, 6, 6) (ξ, 7, 7)	(ξ, 5, 5) (ξ, 3, 3) (ξ, 4, 4)
2	(ξ, 3, 3) (ξ, 4, 4) (ξ, , 2)	∞	(ξ, 5, 5) (ξ, 7, 7) (ξ, 6, 6)	(ξ, 1, 1) (ξ, 3, 3) (ξ, 8, 8)	(ξ, 2, 2) (ξ, 4, 4) (ξ, 3, 3)
3	(ξ, 6, 6) (ξ, 8, 8) (ξ, 7, 7)	(ξ, 1, 1) (ξ, 4, 4) (ξ, 3, 3)	∞	(ξ, 5, 5) (ξ, 4, 4) (ξ, 6, 6)	(ξ, 1, 1) (ξ, 2, 2) (ξ, 9, 9)
4	(ξ, 6, 6) (ξ, 4, 4) (ξ,5, 5)	(ξ, 3, 3) (ξ, 7, 7) (ξ, 3, 3)	(ξ,5, 5) (ξ, 3, 3) (ξ, 6, 6)	∞	(ξ, 6, 6) (ξ, 4, 4) (ξ, 7, 7)
5	(ξ, 4, 4) (ξ, 3, 3) (ξ, 6, 6)	(ξ, 2, 2) (ξ, 7, 7) (ξ, 6, 6)	(ξ, 8, 8) (ξ, 7, 7) (ξ, 6, 6)	(ξ, 9, 9) (ξ, 5, 5) (ξ, 6, 6)	∞
	Bi-fuzzy Risks/Discomforts Matrix(5 *5) With Three Conveyances				
i/j	1	2	3	4	5
1	∞	(ξ, .12, .12) (ξ, .02, .02) (ξ, .07, .07)	(ξ, .13, .13) (ξ, .03, .03) (ξ, .04, .04)	(ξ, .14, .14) (ξ, .04, .04) (ξ, .06, .06)	(ξ, .15, .15) (ξ, .05, .05) (ξ, .08, .08)
2	(ξ, .16, .16) (ξ, .24, .24) (ξ, .14, .14)	∞	(ξ, .17, .17) (ξ, .16, .16) (ξ, .06, .06)	(ξ, .01, .01) (ξ, .17, .17) (ξ, .1, .1)	(ξ, .11, .11) (ξ, .21, .21) (ξ, .2, .2)
3	(ξ, .06, .06) (ξ, .13, .13) (ξ, .16, .16)	(ξ, .18, .18) (ξ, .11, .11) (ξ, .22, .22)	∞	(ξ, .03, .03) (ξ, .16, ,16) (ξ, .25, .25)	(ξ, .04, .04) (ξ, .05, .05) (ξ, .01, .01)
4	(ξ, .07, .07) (ξ, .04, .04) (ξ, .05, .05)	(ξ, .13, .13) (ξ, .07, .07) (ξ, .06, .06)	(ξ, .15, .15) (ξ, .13, .13) (ξ, .14, .14)	∞	(ξ, .26, .26) (ξ, .14, .14) (ξ, .2, .)
5	(ξ, .11, .11) (ξ, .03, .03) (ξ, .05, .05)	(ξ, .2, .2) (ξ, .1, .1) (ξ, .06, .06)	(ξ, .19, .19) (ξ, .13, .13) (ξ, .17, .17)	(ξ, .18, .18) (ξ, .12, .12) (ξ, .16, .16)	∞

Table 4.55: Optimum Results of BFCSTSP (Model 4.5A1)

DM	Path(Vehicle)	Obj Value	Risk	R_{max}
ODM	3(2)-1(1)-4(2)-5(3)-2(3)	63.5	4.37	4.5
PDM	5(1)-1(2)-4(2)-3(3)-2(3)	68.9	4.48	4.5
ODM	3(1)-4(3)-2(1)-5(3)-1(1)	72.5	4.2	4.5
PDM	4(3)-1(2)-5(1)-3(2)-2(1)	79.4	4.32	4.5
ODM	1(1)-4(1)-2(2)-5(3)-3(3)	81.5	4.03	4.5
PDM	1(3)-5(3)-3(3)-2(2)-4(3)	96.5	4.23	4.5
ODM	2(3)-1(1)-5(3)-3(3)-4(2)	93.5	3.78	4.25
PDM	5(3)-2(1)-4(2)-3(3)-1(2)	118.2	3.81	4.25
ODM	4(3)-1(2)-5(3)-3(2)-2(2)	102.2	3.6	4
PDM	2(1)-4(3)-1(2)-5(3)-3(3)	129.5	3.91	4

Table 4.56: Results for virtual data (Model 4.5A1)

Instances (Cities)	Costs	R_{max}
15×15	142	5.5
20×20	196	6.5
25×25	244	7.5
30×30	273	9.5
35×35	398	11.25
40×40	446	13.0
45×45	518	15.5
50×50	692	18.0
80×50	1145	23.7
100×100	1468	32.5
150×150	2354	41.4
200×200	3623	73.9

Table 4.57: Number of win for different algorithms

Problem	fri26	bays29	bayg29	dantzig42	eil51	berlin52	st70	eil76	pr76	rat99	kroa100
RGA	84	91	78	90	71	79	87	75	97	77	81
FGA	67	76	63	82	57	68	63	59	71	69	64
SGA	59	43	56	41	51	57	49	37	56	52	55

Model 4.5A: CSTSP for virtual data:

Here CSTSP are solved by ReGA with large scale data of different sizes, which are randomly generated for different cities and the results are presented in Table 4.56.

4.6.4 Statistical test for ReGA

ANOVA Test:

To test the statistical significance of the proposed algorithm, ReGA, we perform the ANOVA and parametric F-tests. To compare the efficiency of the developed algorithm, another two established heuristic techniques Fuzzy age based GA (FGA developed by Last et al. [88] and used by Roy et al. [147]) and SGA are used. Here 100 independent runs for individual algorithm are considered. Different steps of this ANOVA are as follows:

For calculation of different steps of ANOVA, we subtract 50 (with out lose of

Table 4.58: ANOVA :Subtracted table from Table 4.57

Problem	fri26	bays29	bayg29	dantzig42	eil51	berlin52	st70	eil76	pr76	rat99	kroa100	Mean
X_1	34	41	28	40	21	29	37	25	47	27	31	$\overline{X}_1$=32.74
X_2	17	26	13	32	7	18	13	9	21	19	14	$\overline{X}_2$=17.18
X_3	9	-7	6	-9	1	7	-1	-13	6	2	5	$\overline{X}_3$=0.55

Table 4.59: ANOVA summary table

Source of variation	Sum of square	df	Mean of square	F
Between groups	SS_B=5701.19	J-1=2	$MS_B=\frac{SS_B}{J-1}$=2850.6	$\frac{MS_B}{MS_W}$=31.28
Within groups	SS_W=1640	J(I-1)=20	$MS_W=\frac{SS_W}{J(I-1)}$=91.11	
Total	SS_T=7341.19	IJ-1=32		

generality) from each numbers and the Table 4.57 is reduced to Table 4.58 where X_1, X_2 and X_3 represents ReGA, FGA and SGA respectively.

Here, total sample size of each algorithm is equal and say, I=11 (TSPLIB problems) and number of algorithm is, J=3. Mean of the sample means, $\overline{\overline{X}}$=16.82. Different values of ANOVA are calculated and presented in Table 4.58.

Here, critical F values, $F_{0.05(2,20)} \approx 3.57$. As the computed F (cf. Table 10) is higher than the standard critical F values (=3.57) for 0.05 level of significance, it may be inferred that there is a significant differences between the groups. When F ratio is found to be significant in an ANOVA with more than two groups, it should be followed by a multiple comparison test to find which group means differ significantly from each other. Scheffe's multiple comparison F- test is done for this purpose to find out whether ReGA & SGA and/or ReGA & FGA are significant. For the first pair i.e., for ReGA & SGA, we calculate F value given by $F=\frac{(\overline{X}_1-\overline{X}_3)^2}{MS_W(\frac{1}{I}+\frac{1}{J})}$=26.80. Similarly, for the second pair i.e., for ReGA and FGA, calculated F= 6.26. As both calculated F values are greater than the standard value (3.57), there is a significant difference between ReGA & SGA and also ReGA & FGA. From Table 4.59, it is observed that the mean ($\overline{X}_1$) of X_1 is higher than the other two means ($\overline{X}_2$ and $\overline{X}_3$). Thus significant differences between the algorithms are observed and therefore, it can be concluded that ReGA is better compared to the other two algorithms

4.6.5 Discussion

In this investigation, a proposed algorithm for GA, called ReGA is proposed and illustrated in CSTSP formulated in different environments. In ReGA, a new rough 7 -point age based selection and comparison crossover are used along with generation dependent random mutation. Such CSTSPs are here formulated with crisp and bi-fuzzy costs and risk/discomfort levels and solved by the proposed ReGA. Here, development of ReGA is in general form and it can be applied in other discrete problems of optimization such as network optimization, graph theory, solid transportation problems, vehicle routing, VLSI chip design, etc. In spite of the better results by ReGA, there is a lot of scope for development in ReGA, specially for the CSTSPs.

4.7 Conclusion

In this chapter, GAs have been developed with five selection operations namely probabilistic selection, probabilistic selection with p_s parameter, rough selection with 3-point, 5-point and 7-point scale, three crossover accordingly adaptive crossover, comparison crossover and min-point crossover and three types of mutation operators such as nodes oriented, generation dependent and location based mutation. Each of the algorithms such as IGA, AGA, MGA, RSGA-I, RSGA-II and ReGA is established solving the standard NP- hard problems from TSPLIB[162].

This chapter contains some constrained STSPs under different environments such as crisp, fuzzy, random, fuzzy-random, random-fuzzy, bi-fuzzy, bi-random, rough and fuzzy-rough. The models are solved by the proposed algorithms in crisp, fuzzy(possibility, necessity, GMIV, credibility and EVM approach), rough(Expectation, trust), random(chance constraint), bi-random, bi-fuzzy, random-fuzzy, fuzzy-random environments. Some virtual data are generated for TSPs and the large size CSTSPs are solved by these algorithms in crisp environment.

Some major statistical tests are done to establish the efficiency of the algorithms, these are Friedman test, Post Hoc analysis, standard deviation, mean, percentage error and ANOVA. Except some cases, proposed algorithms performed much better.

Chapter 5

Single Objective Optimization Using Hybrid Heuristics

5.1 Introduction

Evolutionary algorithms can be combined with more traditional optimization techniques. This is as simple as the use of a conjugate-gradient minimization after primary search with an evolutionary algorithm. It may also involve simultaneous application of algorithms like the use of evolutionary search for the structure of a model coupled with gradient search for parameter values. Further, evolutionary computation can be used to optimize the performance of neural networks, fuzzy systems, production systems, wireless systems and other program structures. It can also be used to perform heuristic initialization of the population, so that search begins with some reasonably good points, rather than a random set. Goldberg [61] described techniques for adding knowledge-directed crossover and mutation. He also discussed the hybridization of GAs with other search techniques. Pure genetic algorithms use only the encoding and objective function. This may help to use in problem specific information. In hybrid schemes GAs are used to get close to optimum value, then conventional optimization schemes like greedy search, gradient search or stochastic hill climbing, etc.

The hybrid algorithm in this section is designed to use heuristics for initialization of population and improvement of offspring produced by crossover and mutations for a Traveling Salesman Problem (TSP). The initialization heuristic algorithm is used to initialize a part of the hybridization, remaining part of the algorithm will be done by another heuristics one case GA and other case PSO and GA. The offspring is obtained by crossover between two parents selected randomly. The tour improvement heuristics: swap operator and swap sequence are used to bring the offspring to a global minimum.

This chapter contains two models, first model is the hybridization of heuristic ACO and GA and second model combinations of ACO, swap sequence based PSO with GA. For the first time, pheromones are classified by rough set and according their pheromone p_cs are created. Present investigation develops 4DTSP and r-4DTSP that are given in section 1.7.3. The efficiency of the proposed intelligent hybrid algorithms are established through some statistical tests such as standard deviation, mean, error, ANOVA etc., and the standard test problems from TSPLIB [162] was solved. Again by proposed algorithms, 4DTSP and r-4DTSP under bi-fuzzy and bi-rough environments are solved.

5.2 Model-5.1: An Intelligent Hybrid Algorithm for 4- Dimensional TSP[1]

In this chapter, the first model presented is the development and application of a hybridized algorithmic approach to solve a 4- dimensional Travelling Salesman Problem (4DTSP) where different paths with various number of conveyances are available to travel between two cities. The algorithm is a hybridization of rough set based ant colony optimization (rACO) with developed genetic algorithm (GA). The initial solutions are produced by ACO which acts as a selection operation of GA and then GA is developed with an extended rough set based selection (7-point scale), comparison crossover and generation dependent mutation. The said hybrid algorithm rough set based Ant Colony Optimization (rACO) with Genetic Algorithm (rACO-GA) is tested against some test functions and supremacy of the proposed algorithm is established. The 4DTSPs are formulated with crisp and bi-fuzzy costs. In each environment, some statistical significant studies due to different time constraint values and other system parameters are presented. The models are illustrated with some numerical data.

[1]communicated to **Journal of Industrial Information Integration. (Elsevier)**

5.2.1 Proposed Intelligent Hybrid rACO-GA

Here an intelligent hybrid algorithm rACO-GA using the rough set based pheromone selection (7-point), comparison crossover and generation dependent random mutation for GA are proposed. The proposed rACO-GA and its procedures are presented below:

(i) Representation:

Here a complete tour of N cities represents a solution of ants. So an N dimensional integer vector $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$, $Y_i = (r_{i1}, r_{i2}, ..., r_{is})$ and $Z_i = (v_{i1}, v_{i2}, ..., v_{iP})$ are used as cities, routes and vehicles to represent a solution, where x_{i1}, x_{i2}, ..., x_{iN} represent N consecutive cities in a tour. In the algorithm, initially an ant colony system is used to produce a set of paths (tours) for a salesman, which is a set of potential solutions for the GA.

(ii) Rough set based ACO (rACO):

In the present algorithm, τ_{ij} represents amount of pheromone which lies on the path between nodes i and j, $iter1$ and $iter2$ represent iteration counters, $maxiter$ and $maxgen$ represent maximum iteration and generation numbers of the hybrid algorithm. n represents number of ants, N is the population size and number of nodes/cities, r and k stands for different routes and vehicles in the problem, where r$\in \{1, 2, .., s\}$ and v$\in \{1, 2, .., p\}$.

(a) Pheromone Initialization:

As the aim of a TSP is to minimize the cost and time for a tour, it is assumed that initial value of pheromone τ_{ijrk}=$\frac{1}{\sqrt{c_{ijrk}}}$, where c_{ijrk} is the cost for travelling from i to j-th city along r-th route using k-th vehicle.

(b) Path Construction:

To construct a path X_m for mth ant, following steps are followed:

a. Let S=$\{1, 2, ..., N\}$ and $l = 1$

b. x_{ml}= a random element from the set S.

c. Let S=$NS - \{x_{ml}\}$

d. Let node i be the present position of an ant i.e., $x_{ml} = i$. Then next node $j \in S$ is selected through the r_{th} route using k_{th} vehicle by the ant with a probability p_{ijrk} given by the formula

$$p_{ijrk} = \frac{\tau_{ijrk}^{\delta_1}}{\sum_{j \in S} \tau_{ijrk}^{\delta_1}}$$

where δ_1 is a user defined parameter which controls the relative importance of pheromone concentration.

e. $l = l + 1, x_{ml} = j.$

f. if $l < N$, goto step (c).

n-such paths are constructed for different n ants.

(c) Pheromone Evaporation:

For evaporation of pheromone, the following formula is used

$$\tau_{ijrk} = (1 - \rho)\tau_{ijrk}$$

where ρ is in $[0, 1]$. The constant ρ, specifies the rate at which pheromone evaporates, causing ants to forget previous decisions.

(d) Pheromone Updating:

After the completion of a tour by all ants, pheromone is increased on the paths through which the ants have travelled. Depending upon the nature of the present problem, pheromone is updated using the following rules.

$\tau_{ijrk} = (1 - \rho)\tau_{ijrk} + \frac{\rho}{n} \sum_{i=1}^{n} \tau_{ijrk}^{best}$, where ρ refers to the rate of evaporation and n be the ants, τ_{ijrk} is highest value of pheromone.

(iii) Rough set based pheromone classification:

After updating of the pheromone quantity, we classify the pheromones depending on the minimum , average and maximum pheromone information. Since pheromone are represented by crisp values, we construct the common rough values from it,

Rough Pheromone =([r_1*avg ph , r_2*avg ph], [r_3*avg ph , r_4*avg ph]), where r_1=$\frac{Max\ -Avg}{Avg}$, r_2=$\frac{Max\ +Min}{2}$, r_3=$\frac{Max\ -Min}{2}$, r_4=$\frac{Avg\ -Min}{Avg}$, avg ph means average pheromone.

This pheromone of the chromosome, belongs to any one of the common rough pheromone values and corresponding p_c's are created for each chromosome as

Table 5.1: Rough Extended Trust Based Linguistic

Gene	VVS	VS	S	M	H	VH	VVH
VVS	VVL	VL	VL	L	VL	VL	VVL
VS	VL	VL	L	M	L	VL	VL
S	VL	L	L	H	L	L	VL
M	L	M	H	VH	H	M	L
H	VL	L	L	H	L	L	VL
VH	VL	VL	L	M	L	VL	VL
VVH	VVL	VL	VL	L	VL	VL	VVL

VVL, VL, L, M, H, VH, VVH. For this purpose, a mathematical equation Equ. 3.24 is developed in section 3.5.1. The common rough variables ([a,b],[c,d]) is extended to $0 \leq c \leq e \leq f \leq a \leq b \leq g \leq h \leq d$ and is described as below,

$$Pheromone = \begin{cases} VeryVerySmall(VVS) \; for\, c \leq pheromone < e \\ VerySmall(VjS) \quad \text{for } e \leq pheromone < f \\ Small(S) \quad \text{for } f \leq pheromone < a \\ Medium(M) \quad \text{for } a \leq pheromone \leq b \\ High(H) \quad \text{for } b < pheromone \leq g \\ VeryHigh(VH) \quad \text{for } g < pheromone \leq h \\ VeryVeryHigh(VVH) \;\; \text{for } h < pheromone \leq d \end{cases} \tag{5.1}$$

(iv) Comparison Crossover:

(a) Determination of Probability of Crossover (p_c): For a pair of chromosomes (X_i, X_j), we construct the following rough set. At first the states of X_i and X_j i.e, (VVS, VS, S, M, H, VH, VVH) are determined by making trust measures of rough values w.r.to their pheromones in common rough pheromone region given in the proposed method. After the determination of states of pheromone intervals of the chromosomes, their crossover probabilities are determined as linguistic variables (VVL, VL, L, M, H, VH, VVH) using rough trust measures which are presented in Table 5.1 following Equ. 5.1.

(b) Crossover Mechanism: For crossover, we choose two individuals (parents) to produce new individuals (children). To get optimal result of a TSP, we take a tour from one node (city)to next node (city) with minimum cost/value using following algorithm (cf. section 4.3.1)(c)(iii).

(v) Generation Dependent Random Mutation:

(a) Generation Dependent Mutation(Variable Method): Here we model a new form of mutation mechanism where probability of mutation (p_m) are determined by

$p_m = \frac{k}{\sqrt{Current\ generation\ number}}$, $k \in [0,1]$.

(b) Selection for mutation: For each solution of P(t), generate a random number r from the range [0,1]. If $r < p_m$, then the solution is taken for mutation. Here p_m decreases gradually as generation increases. After calculating the p_m, mutation operation follows the conventional random mutation. Here we randomly choose two nodes from each chromosome and exchange their positions and replace the chromosome in the new offspring set.

(vi) Termination Criteria:

Hybrid algorithm is terminated if any one of the following conditions is satisfied (which ever is earlier):

(a) the best solution does not improve within 20 consecutive generations

(b) number of generations reaches user defined iterations (generations).

Hybrid Algorithm :

Input: Set $iter_{ACO}(= 0)$, $iter_{GA}(= 0)$, $maxiter$ and Max_{gen} (S_0), Population Size (pop_size), Number of ants (n), Probability of Mutation (p_m), Problem Data (cost and risk matrices).

Output: The optimum and near optimum paths/tour.

1. Start

2. Set initial generation $iter_{ACO} = 0$, $iter_{GA} = 0$ and $Max_{gen}(S_0)$.

3. Initialize pheromone τ_{ijrk} for $i = 1, 2, ..., N$ and $j = 1, 2, ..., N$ using r_{th} route and k_{th} vehicle.

4. For $(iter_{ACO} \leq maxiter)$

5. Construct path of n ants, i.e., n tours $X_i = (x_{i1rk}, x_{i2rk}, .., x_{iNrk}, x_{i1rk})$, $i = 1, 2, .., n$ using τ_{ijrk}.

6. Made pheromone evaporation.

7. Update pheromone for all the paths.

8. $iter_{ACO} = iter_{ACO}+1$

9. End for

10. Set initial solution obtained from ACO.

11. For $(iter_{GA} \leq S_0)$

12. Sum the pheromone of all individual chromosomes.

13. Cluster the pheromone.
14. Develop the linguistic as VVLP, VLP, LP, etc.
15. Trust based p_c created.
16. Crossover operation performed.
17. Mutation operation performed.
18. Update the chromosome.
19. Update the pheromone.
20. Find best optimum and near optimum solutions.
21. $iter_{GA} = iter_{GA}+1$
22. End for
23. Store global and near optimum solutions.
24. End

5.2.2 Mathematical Formulation and Its crisp equivalence

Model 5.1A: STSP (3DTSP) with Time Constraints:

Let $c(i, j, k)$ and $t(i, j, k)$ be the cost and time respectively for travelling from i-th city to j-th city using k-th type conveyance. Then the salesman has to determine a complete tour (x_1, x_2, ...,x_N, x_1) and corresponding conveyance types (v_1, v_2, ..., v_P) to be used for the tour, where $x_i \in \{1, 2, ..N\}$ for $i = 1, 2, ..., N$, $v_i \in \{1, 2, ..P\}$ for $i = 1, 2, ..., N$ and all x_i's are distinct. Then the problem can be mathematically formulated as:

$$\left.\begin{array}{r}\text{minimize}\quad Z = \sum_{i=1}^{N-1} c(x_i, x_{i+1}, v_i) + c(x_N, x_1, v_l),\\ \text{subject to}\quad \sum_{i=1}^{N-1} t(x_i, x_{i+1}, v_i) + t(x_N, x_1, v_l) \le t_{max},\\ where\ x_i \neq x_j, i, j = 1, 2...N,\quad v_i, v_l \in \{1, 2.., or\, p\}\end{array}\right\} \tag{5.2}$$

along with sub tour elimination criteria

$$\left.\begin{array}{r}\sum_{i \in S}^{N} \sum_{j \in S}^{N} x_{ij} \le |S| - 1, \forall S \subset Q\\ where\ x_{ij} \in \{0, 1\}, i, j = 1, 2.., N..\end{array}\right\} \tag{5.3}$$

.

Model 5.1B: 4DTSP with time Constraints (4DTSP):

Let $c(i,j,r,k)$ and $t(i,j,r,k)$ be the cost and time respectively for travelling from i-th city to j-th city by the r-th route using k-th type conveyance. Then the salesman has to determine a complete tour (x_1, x_2, ...,x_N, x_1) and corresponding available route types (r_1, r_2, ..., r_s) with conveyance types (v_1, v_2, ..., v_p) to be used for the tour, where $x_i \in \{1, 2, ..N\}$ for $i = 1, 2, ..., N$, $r_i \in \{1, 2, ..s\}$ and $v_i \in \{1, 2, ..p\}$ for $i = 1, 2, ..., N$ and all x_i's are distinct. Then the problem can be mathematically formulated as:

$$\left.\begin{array}{c} \text{minimize } \; Z = \sum_{i=1}^{N-1} c(x_i, x_{i+1}, r_i, v_i) + c(x_N, x_1, r_l, v_l), \\ \text{subject to } \; \sum_{i=1}^{N-1} t(x_i, x_{i+1}, r_i, v_i) + t(x_N, x_1, r_l, v_l) \le t_{max}, \\ where\ x_i \neq x_j, i, j = 1, 2...N,\ r_i, r_l \in \{1, 2.., or\, s\},\ v_i, v_l \in \{1, 2.., or\, p\} \end{array}\right\} \quad (5.4)$$

Model 5.1C: 4DTSP in bi-fuzzy Environment (BF4DTSP):

In the above Equ. 5.4, if costs and times are bi-fuzzy variables, i.e, $\tilde{\tilde{c}}(i,j,r,k)$ and $\tilde{\tilde{t}}(i,j,r,k)$ respectively, time limit t_{max} is also bi-fuzzy number $\tilde{\tilde{t}}_{max}$, then following the Theorem 3.1 [171], the above problem reduces to

$$\left.\begin{array}{c} \text{minimize } \; Z = \sum_{i=1}^{N-1} \tilde{\tilde{c}}(x_i, x_{i+1}, r_i, v_i) + \tilde{\tilde{c}}(x_N, x_1, v_l v_l), \\ \text{subject to } \; \sum_{i=1}^{N-1} \tilde{\tilde{t}}(x_i, x_{i+1}, r_i, v_i) + \tilde{\tilde{t}}(x_N, x_1, r_l, v_l) \le \tilde{\tilde{t}}_{max}, \\ where\ x_i \neq x_j, i, j = 1, 2...N,\ r_i, r_l \in \{1, 2.., ors\},\ v_i, v_l \in \{1, 2.., or\, p\} \end{array}\right\} \quad (5.5)$$

Equ. 5.5 can be reformulated as

$\sum_{i=1}^{N-1} \tilde{\tilde{c}}(x_i, x_{i+1}, r_i, v_i) + \tilde{\tilde{c}}(x_N, x_1, r_i, v_l) \le f$, where f be a given crisp value. Using Bifuzzy-Chance Constraint Multi objective Programing (CCMOP) 3.13.3 [103], and Theorems 3.1 and 3.2, we have

$$\left.\begin{array}{c} \text{minimize } f \\ Pos\{\theta | Pos\{|\tilde{\tilde{C}}(\theta)^T x \le f\} \ge \delta\} \ge \gamma \\ Pos\{\theta | Pos\{|\tilde{\tilde{T}}(\theta)^T x \le \tilde{\tilde{T}}_{max}(\theta)^T\} \ge \theta\} \ge \eta \end{array}\right\} \quad (5.6)$$

The objective function for Nes-Nes [171] is equivalently written as:

$$\left.\begin{array}{c} \text{minimize } f \\ Nes\{\theta | Nes\{|\tilde{\tilde{C}}(\theta)^T x \le f\} \ge \delta\} \ge \gamma \\ Nes\{\theta | Nes\{|\tilde{\tilde{T}}(\theta)^T x \le \tilde{\tilde{T}}_{max}(\theta)^T\} \ge \theta\} \ge \eta \end{array}\right\} \quad (5.7)$$

where $\tilde{\tilde{C}} = \sum_{i=1}^{N-1} \tilde{\tilde{c}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{c}}(x_N, x_1, v_l)$, $\tilde{\tilde{T}} = \sum_{i=1}^{N-1} \tilde{\tilde{t}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{t}}_1(x_N, x_1, v_l)$, $\tilde{\tilde{T}}_{max} = \tilde{\tilde{t}}_{max}$.
The Equs. 5.6 and 5.7 are transformed following possibility necessity measures as .

$$\left.\begin{array}{c} \text{minimize } c^T - L^{-1}(\delta)\alpha_1^{cT} - L^{-1}(\gamma)\alpha_2^{cT} \\ \text{s.t. } T_{max} - R^T + R^{-1}(\theta)\beta_1^{T_{max}} + L^{-1}(\theta)\alpha_1^{T_{max}T} \\ +L^{-1}(\eta)(\alpha_2^{RT} + \beta_2^{T_{max}} \ge 0 \end{array}\right\} \quad (5.8)$$

and

$$\left.\begin{array}{c} \text{minimize } c^T + R^{-1}(1-\delta)\beta_1^{cT} + R^{-1}(1-\gamma)\beta_2^{cT} \\ \text{s.t. } T_{max} - R^T - L^{-1}(1-\eta)(\alpha_2^{T_{max}} + \beta_2^{RT}) \\ -L^{-1}(1-\theta)\alpha_1^{T_{max}} - R^{-1}(\theta)\beta_1^{RT} \ge 0 \end{array}\right\} \quad (5.9)$$

where $\alpha_1^c, \alpha_2^c, \alpha_1^R, \alpha_2^R, \alpha_1^{T_{max}}, \alpha_2^{T_{max}}, \beta_1^c, \beta_2^c, \beta_1^R, \beta_2^R, \beta_1^{r_{max}}, \beta_2^{r_{max}}$ are corresponding left and right spreads of the reference function of LR fuzzy numbers and $\theta, \eta, \delta, \gamma$ are predetermined confidence levels.

5.2.3 Numerical Experiments

Testing for rACO-GA:

The performance of the proposed hybrid algorithm (HA) rACO-GA was found for 15 standard benchmark problems using TSPLIB [162]. Table 5.2 gives the results of rACO-GA along with the standard GA and ACO. The results are compared in terms of total cost. Under 20 independent runs, the average result, best found solution with standard deviation (SD) and relative error are presented in Table 5.2.

Table 5.2: Test TSPLIB Problems by rACO-GA

Instances	Average	Result		Best	Found	Result	Error & SD		
	HA	ACO	GA	HA	ACO	GA	HA	ACO	GA
fri26	938.51	939.63	941.64	937	937	937	0, 0.45	0.02, 0.76	0.16, 0.56
bays29	2021.23	2022.78	2022.56	2020	2020	2020	0, 0.43	0.16,.78	0.61, 1.04
bayg29	1610.34	1611.02	1610.97	1610	1610	1610	0, 0.12	0.11, 0.37	0.04, 0.76
dantzig42	699.27	703.51	700.07	699	703	699	0, 0.67	1.23, 0.98	0.45, 0.68
eil51	427.8	432.98	429.31	426	430	426	0, 0.98	3.65, 1.53	1.78, 0.93
berlin52	7548.9	7936.35	7654.87	7542	7883	7623	0.06, 1.76	18.54, 2.49	2.43, 1.07
st70	677.34	699.51	682.17	675	687	675	0.03, 1.02	5.87, 3.78	2.67, 1.45
eil76	539.65	567.27	545.86	538	547	547	0.78, 0.67	2.76, 1.93	3.87, 3.65
pr76	108265.76	108634.71	108572.32	108159	108346	108258	0.45, 0.99	12.67, 8.75	7.65, 3.95
rat99	1212.52	1236.46	1218.71	1211	1223	1211	0.34, 0.67	1.72, 1.98	1.23, 0.87
kroa100	21321.78	21567.82	21431.75	21282	21427	21378	0.56, 1.85	4.72, 2.95	2.12, 3.17
kroc100	20834.87	20956.23	20971.75	20750	20802	20831	0.58, 2.73	5.71, 0.98	2.45, 1.79
kroa150	26600.76	26952.34	26743.89	26524	26871	26701	0.87, 2.56	3.61, 4.12	2.91, 0.93
krob200	29450.7	30887.34	29965.27	29413	29944	29789	2.31, 3.02	15.47, 6.82	10.72, 6.14
pr299	49765.6	52945.78	50831.43	48743	49765	49391	4.97, 5.92	23.81, 10.21	10.89, 8.37

Table 5.3: Parameters for HA, ACO and SGA

Size (N)	Maxgen	Iter_{ACO}	Iter_{GA}	Maxiter	Ant number(n)	popsize	p_c	p_m	δ_1
$N \leq 50$	200	80	120	100	30	50	0.35	0.1	0.2
$50 < N \leq 100$	300	120	180	200	50	100	0.3	0.15	0.2
$100 < N \leq 150$	400	200	300	300	80	100	0.35	0.2	0.3
$150 < N \leq 200$	500	200	400	400	100	130	0.4	0.2	0.3
$200 < N \leq 250$	600	250	450	400	100	150	0.45	0.2	0.3
$250 < N \leq 300$	900	400	500	500	100	150	0.45	0.25	0.3

The parameters of the HA are set as in Table 5.3 for different nodes of the TSP. As the size of the TSP increases, pop-size, Maxgen, ant numbers for convergence of the optimal solution also increases.

Model 5.1B: 4DTSP with time Constraint in Crisp Environment:
Now for a 4DTSP, where we consider three types of conveyances and maximum three types of route as in Equ. 5.4. The cost and time matrices for the 4DTSP are presented in Table 5.4.

Here we consider a deterministic 2DTSP from Equ. 5.2 using a single vehicle. The problem is solved by rACO-GA and the results are presented in Table 5.5.

To determine these results, we have taken maximum generation=1000, and we see that as time decreases, the corresponding tour cost increases as in real life situation. Again, we consider a deterministic 3DTSP given by Equ. 5.2. The

Table 5.4: Input Data: Crisp 4DTSP (Model 5.1B)

	Crisp Cost Matrix(10*10) With Three Route and Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	(35,36,27) (24,34,25) (17,23,26)	(18,39,30) (19,24,26) (30,24,31)	(20,33,34) (23,27,22) (23,22,28)	(30,21,62) (32,14,18) (31,43,32)	(23,24,27) (28,36,29) (57,28,39)	(41,37,21) (31,45,62) (24,11,28)	(17,15,9) (67,38,29) (11,34,13)	(35,36,37) (45,38,29) (19,28,17)	(23,45,18) (47,39,20) (17,29,10)
2	(35,26,17) (33,34,28) (22,27,29)	∞	(40,21,32) (57,28,39) (13,27,19)	(18,29,10) (18,39,20) (15,21,32)	(35,26,37) (27,36,30) (31,54,23)	(17,27,15) (45,25,16) (43,25,28)	(18,23,16) (23,26,22) (19,28,38)	(21,24,15) (41,39,20) (23,25,27)	(18,28,19) (17,28,19) (32,37,33)	(35,36,37) (27,26,29) (23,27,28)
3	(38,30,29) (23,45,18) (17,28,35)	(17,58,34) (23,24,27) (37,27,19)	∞	(12,25,14) (44,38,37) (39,23,43)	(42,25,46) (29,30,46) (43,33,54)	(19,27,35) (34,27,18) (21,26,16)	(29,19,24) (27,28,17) (15,17,19)	(17,17,19) (18,27,16) (21,27,28)	(17,16,19) (24,22,29) (21,26,28)	(15,18,19) (17,18,19) (17,22,28)
4	(28,20,11) (18,19,16) (56,23,19)	(10,22,14) (18,28,32) (333,46,28)	(17,8,29) (37,11,44) (48,29,10)	∞	(30,19,24) (30,17,11) (41,37,21)	(31,32,18) (17,27,15) (32,37,33)	(17,43,23) 11,34,13) (30,21,62)	(23,27,29) (35,26,17) (36,28,22)	(35,36,37) (28,36,29) (17,10,19)	(21,28,29) (33,21,38) (67,26,38)
5	(17,15,9) (34,29,11) (17,29,10)	(42,23,34) (45,19,20) (15,29,30)	(35,36,37) (29,10,28) (37,25,18)	(20,31,43) (36,29,13) (52,19,38)	∞	(32,37,33) (28,36,29) (35,26,17)	(28,36,29) (32,15,33) (17,34,23)	(17,19,10) (17,18,14) (29,27,27)	(21,22,29) (22,29,30) (35,36,37)	(28,28,19) (34,33,37) (43,36,23)
6	(22,25,17) (17,27,15) (23,24,27)	(17,15,9) 11,34,13) (43,25,28)	(32,37,33) (45,48,10) (23,24,27)	(43,25,28) (54,38,20) (28,29,17)	(23,24,27) (55,38,43) (45,56,57)	∞	(22,26,17) (28,36,29) (47,46,35)	(17,16,19) (17,54,29) (35,28,47)	(22,17,16) (28,39,10) (24,34,25)	(31,28,29) (39,40,29) (48,29,10)
7	(21,24,2) (30,21,62) (30,21,62)	(35,26,17) (43,25,28) (43,25,28)	(32,37,33) (24,34,25) (48,29,10)	(17,27,15) (53,67,18) (18,15,13)	(23,24,27) (18,15,13) (18,28,29)	(48,29,10) (33,27,26) (28,25,29)	∞	(30,38,40) (23,24,27) (35,28,19)	(56,53,61) (28,39,28) (53,67,18)	(17,28,19) (18,15,13) (18,28,29)
8	(43,25,28) (11,34,13) (43,25,28)	(53,67,18) (18,15,13) (30,21,62)	(18,15,13) (18,28,29) (45,56,27)	(34,56,15) (45,56,27) (35,26,17)	(23,24,27) (28,25,26) (17,27,15)	(17,27,15) (17,27,15) (45,56,27)	(17,15,9) (17,10,11) (17,12,11)	∞	(17,27,15) (23,24,27) (23,17,19)	(45,56,27) (32,18,19) (24,27,20)
9	(18,15,13) (18,15,13) (19,18,17)	(17,15,9) 11,34,13) (17,27,15)	(45,56,27) (35,26,17) (23,24,27)	(54,37,29) (24,34,25) (18,15,13)	(23,24,27) (18,28,29) (45,56,27)	(48,29,10) (17,27,15) (19,18,17)	(19,18,17) (20,26,19)	12,34,13) (17,19,10) (28,36,29)	(11,34,13) ∞ (53,67,18)	(37,45,28) (54,37,29) (22,32,16)
10	(21,34,13) (30,21,62) (43,25,28)	(43,25,28) (11,34,13) (23,24,27)	12,33,13) (16,34,13) (23,24,27)	(11,34,23) (23,24,27) (18,15,13)	(17,27,15) (24,34,25) (17,27,15)	(48,29,10) (53,67,18) (35,36,37)	(17,27,15) (18,28,29) (18,28,29)	(54,37,29) (45,56,27) (28,36,29)	(54,37,29) (19,18,17) (17,27,15)	∞
	Crisp time Matrix(10×10) With Three route and Conveyances respectively									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞ (.26,.22,.25) (.11,.16,.17)	(.69,.68,.75) (.32,.45,.71) (.16,.18,.19)	(.84,.63,.7) (.24,.62,.44) (.18,.19,.31)	(.82,.7,.71) (.36,.64,.72) (.25,.28,.29)	(.72,.8,.42) (.32,.42,.26) (.27,.28,.29)	(.45,.34,.28) (.45,.56,.73) (.23,.25,.32)	(.33,.42,.45) (.23,.45,.36) (.31,.33,.34)	(.22,.32,.42) (.21,.52,.33) (.41,.43,.45)	(.42,.62,.45) (.24,.26,.27) (.32,.34,.36)	(.43,.53,.52) (.32,.28,.35) (.43,.46,.47)
2	.7,.66,.61 .8,.75,.71	∞	.76,.71,.69 .68,.61,.59	.67,.62,.6 .9,.85,.82	.75,.68,.65 .6,.58,.5	.68,.64,.61 .7,.65,.62	.69,.63,.6 .31,.26,.2	.51,.45,.4 .32,.34,.19	.6,.57,.53 .7,.69,.62	.8,.76,.71 .81,.76,.7
3	.55,.51,.48 .6,.56,.53 .61,.58,.56	.72,.69,.62 .38,.31,.26 .6,.58,.51	∞	.81,.76,.7 .71,.68,.66 .8,.76,.71	.51,.46,.4 .7,.64,.61 .48,.44,.4	.59,.55,.52 .61,.58,.56 .62,.6,.57	.8,.75,.71 .9,.86,.81 .89,.86,.81	.65,.6,.59 .64,.6,.58 .68,.65,.61	.58,.55,.51 .8,.76,.71 .55,.5,.48	.67,.61,.58 .76,.71,.68 .64,.6,.57
4	.69,.64,.62 .78,.75,.71 .85,.83,.8	.86,.81,.79 .76,.71,.69 .81,.78,.74	.79,.75,.72 .9,.85,.82 .7,.64,.6	∞	.65,.63,.6 .76,.72,.7 .78,.71,.69	.69,.65,.62 .78,.75,.71 .68,.67,.65	.78,.74,.71 .68,.65,.61 .6,.54,.5	.6,.56,.52 .59,.58,.56 .79,.76,.72	.85,.82,.8 .78,.74,.71 .71,.69,.64	.68,.63,.59 .5,.45,.41 .6,.54,.5
5	.8,.76,.71 .81,.79,.75 .88,.81,.79	.55,.52,.49 .75,.74,.72 .61,.58,.54	.6,.58,.4 .58,.55,.5 .59,.58,.54	.78,.75,.71 .65,.62,.61 .55,.51,.48	∞	.62,.58,.55 .81,.75,.72 .55,.51,.45	.51,.45,.41 .81,.78,.75 .71,.68,.66	.67,.62,.59 .66,.61,.58 .82,.79,.75	.8,.76,.7 .88,.81,.78 .9,.87,.81	.69,.66,.62 .7,.68,.65 .9,.87,.83
6	.8,.75,.71 .81,.79,.76 .88,.85,.81	.65,.63,.6 .75,.72,.7 .66,.61,.59	.85,.82,.78 .7,.68,.62 .65,.62,.6	.88,.84,.79 .87,.84,.8 .85,.81,.78	.7,.67,.63 .6,.58,.55 .58,.54,.49	∞	.64,.6,.58 .55,.51,.46 .7,.68,.65	.55,.52,.48 .65,.63,.6 .76,.71,.68	.68,.61,.58 .73,.7,.68 .62,.58,.55	.65,.61,.58 .55,.52,.48 .65,.62,.6
7	.58,.54,.49 .56,.52,.48 .65,.62,.58	.65,.63,.6 .44,.38,.33 .71,.65,.6	.64,.6,.58 .6,.58,.55 .67,.64,.6	.7,.68,.65 .55,.51,.45 .71,.68,.64	.56,.54,.51 .38,.32,.28 .55,.53,.51	.55,.51,.46 .75,.71,.68 .52,.47,.4	∞	.85,.81,.78 .55,.54,.51 .75,.76,.72	.65,.61,.59 .58,.54,.5 .65,.61,.58	.78,.74,.69 .71,.68,.64 .65,.62,.58
8	.56,.52,.49 .54,.52,.51 .5,.43,.4	.7,.68,.65 .9,.88,.84 .8,.81,.78	.64,.6,.58 .41,.38,.37 .51,.45,.4	.56,.52,.5 .76,.74,.7 .56,.52,.49	.62,.58,.53 .62,.57,.55 .52,.48,.45	.55,.52,.48 .8,.77,.7 .88,.83,.8	.55,.54,.51 .78,.72,.7 .54,.53,.5	∞	.78,.76,.73 .43,.4,.36 .73,.7,.68	.58,.56,.51 .6,.54,.5 .58,.54,.49
9	.56,.51,.48 .88,.85,.81 .68,.65,.51	.58,.52,.5 .59,.57,.56 .58,.55,.53	.9,.85,.82 .62,.61,.58 .6,.54,.5	.7,.68,.64 .74,.7,.67 .68,.52,.58	.78,.75,.71 .65,.61,.58 .74,.7,.68	.74,.7,.68 .64,.61,.59 .67,.64,.6	.85,.81,.8 .62,.6,.57 .58,.54,.49	.62,.6,.58 .65,.61,.6 .79,.75,.72	∞	.69,.65,.63 .78,.73,.7 .72,.7,.68
10	.78,.71,.69 .7,.67,.64 .69,.64,.6	.66,.61,.58 .77,.74,.7 .78,.76,.71	.69,.65,.62 .8,.76,.74 .68,.65,.63	.74,.7,.68 .65,.6,.57 .76,.71,.68	.83,.78,.75 .62,.58,.56 .75,.71,.66	.65,.61,.58 .87,.83,.78 .68,.64,.59	.59,.54,.5 .68,.64,.61 .59,.55,.51	.55,.52,.47 .52,.48,.54 .64,.6,.58	.64,.59,.58 .45,.41,.37 .61,.59,.58	∞

Table 5.5: Results of 2DTSP in Crisp (Model 5.1A)

Algorithm	Path	Value	T_{max}
rACO-GA	2-6-1-9-5-10-8-4-3-7	147	Without T_{max}
	2-6-1-9-5-10-8-4-3-7	147	8.51
	7-2-6-4-3-5-10-8-9-1	154	8.57
	7-1-4-3-10-9-6-8-5-2	173	8.25
	5-2-8-10-9-6-1-3-4-7	189	8.1
ACO	6-3-9-7-5-2-1-10-8-4	193	8.7
GA	2-8-5-7-6-10-4-3-9-1	197	8.7
rACO-GA	4-8-9-1-3-7-2-10-5-6	204	8.00
ACO	3-8-5-7-6-10-4-2-9-1	227	8.0
GA	8-2-1-3-4-10-7-9-6-5	221	8.00
rACO-GA	8-2-7-9-1-3-5-6-10-4	356	7.5
ACO	5-6-2-7-8-10-3-9-4-1	392	7.5
GA	10-6-2-7-8-5-3-9-4-1	398	7.5

problem is solved by rACO-GA and the results are presented in Table 5.6.

Next we take a deterministic 4DTSP given by Equ. 5.4 where three types of routes and vehicles are considered. The problem is solved by rACO-GA and the results are presented in Table 5.7.

Model 5.1C: 4DTSP with time constraint in bi-fuzzy Environments (BF-4DTSP)

Here we take the cost and time constraint as bi-fuzzy values for the 4DTSP as in Equs. 5.8 and 5.9. Also we consider three types of routes and conveyances. We use triangular LR Fu-Fu variables where (ξ, α, β) is LR fuzzy variable with known left and right spreads. Also ξ is a triangular fuzzy variable connecting with the corresponding components in Table 5.4.

Here predetermined confidence levels δ= γ=0.9, θ=η=0.95 and reference function L(x)=R(x)=1-x are taken. Left and right spreads of the LR fuzzy numbers are given in the Table 5.8.

Table 5.6: Results of 3DTSP in Crisp (Model 5.1A)

Algorithm	Path(Vehicle)	Cost	Time	T_{max}
rACO-GA	9(1)-7(2)-8(3)-4(1)-3(1)-2(2)-5(1)-1(1)-10(2)-6(2)	170	8.75	8.75
	2(2)-1(3)-10(1)-3(1)-6(2)-7(1)-4(2)-5(2)-10(1)-9(2)	193	8.62	
	6(1)-9(2)-10(1)-7(2)-3(1)-8(2)-5(1)-4(1)-2(1)-1(3)	205	8.59	
	6(1)-10(2)-5(1)-7(1)-4(2)-3(3)-1(2)-10(3)-9(1)-2(1)	213	8.54	
	6(1)-7(2)-9(2)-8(1)-4(1)-5(2)-1(2)-2(2)-3(2)-10(1)	228	8.46	
ACO	3(2)-10(1)-8(1)-2(3)-3(3)-1(3)-5(1)-4(2)-6(2)-8(1)	242	8.7	8.75
GA	4(1)-5(1)-8(1)-3(3)-2(1)-10(3)-5(1)-4(2)-6(2)-7(2)	247	8.7	8.75
rACO-GA	3(2)-7(1)-4(1)-3(1)-1(1)-5(2)-10(2)-8(1)-6(1)-2(3)	282	7.95	8.00
	7(2)-9(1)-8(1)-10(2)-1(2)-3(2)-6(2)-5(1)-4(3)-2(1)	315	7.71	7.75
	10(1)-7(2)-6(1)-5(3)-4(2)-2(3)-3(1)-1(2)-8(2)-9(1)	376	7.58	

Table 5.7: Results of 4DTSP in Crisp (Model 5.1B)

Algorithm	Path(Route, Vehicle)	Cost	Time	T_{max}
rACO-GA	10(2,1)-7(3,2)-8(1,3)-4(2,1)-3(1,1)-2(1,2)-5(2,1)-1(3,1)-9(1,2)-6(2,2)	183	8.75	8.75
	2(1,2)-10(2,3)-1(1,1)-4(1,2)-6(1,2)-7(3,1)-3(2,2)-5(1,2)-10(2,1)-9(2,2)	187	8.67	
	6(1,3)-9(2,1)-10(1,1)-7(1,2)-3(1,3)-8(2,2)-5(3,1)-4(2,1)-2(1,1)-1(2,3)	216	8.53	
	6(2,1)-10(2,2)-5(1,1)-7(2,1)-4(2,3)-3(3,1)-1(2,1)-10(3,1)-9(2,1)-2(3,1)	219	8.42	
	6(1,3)-7(2,1)-9(2,1)-8(1,1)-4(2,1)-5(2,2)-1(1,2)-2(3,2)-3(1,2)-10(3,3)	245	8.34	
ACO	3(1,2)-10(2,1)-8(3,1)-2(2,3)-3(3,1)-1(1,1)-5(2,1)-4(1,2)-6(2,2)-8(1,2)	253	8.73	8.75
GA	4(3,3)-5(1,2)-8(3,1)-3(2,3)-2(1,1)-10(2,3)-5(2,1)-4(1,2)-6(1,2)-7(2,2)	262	8.7	8.75
rACO-GA	3(2,3)-7(1,2)-4(3,1)-3(2,1)-1(1,1)-5(2,1)-10(2,2)-8(1,3)-6(1,1)-2(3,2)	303	7.91	8.00
	8(3,2)-7(2,1)-9(3,1)-10(2,3)-1(2,2)-3(2,1)-6(2,1)-5(1,2)-4(3,3)-2(1,2)	338	7.66	7.75
	10(1,2)-7(,12)-6(3,1)-5(3,2)-4(2,2)-2(1,3)-3(2,1)-1(3,2)-8(2,2)-9(2,1)	381	7.48	

Table 5.8: Input Data: BF-4DTSP (Model 5.1C)

	Fuzzy Cost Matrix (10 *10) with three route and conveyances respectively									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	(ξ, 2, 2) (ξ, 4, 4) (ξ, 1, 1)	(ξ, 3, 3) (ξ, 5, 5) (ξ, 2, 2)	(ξ,4, 4) (ξ, 6, 6) (ξ, 7, 7)	(ξ, 5, 5) (ξ, 3, 3) (ξ, 4, 4)	(ξ, 9, 9) (ξ, 10, 10) (ξ, 6, 6)	(ξ, 5, 5) (ξ, 8, 8) (ξ, 7, 7)	(ξ, 6, 6) (ξ, 8, 8) (ξ, 5, 5)	(ξ, 2, 2) (ξ, 4, 4) (ξ, 3, 3)	(ξ, 5, 5) (ξ, 5, 5) (ξ, 1, 1)
2	(ξ, 3, 3) (ξ, 4, 4) (ξ, 2, 2)	∞	(ξ, 5, 5) (ξ, 7, 7) (ξ, 6, 6)	(ξ, 1, 1) (ξ, 3, 3) (ξ, 8, 8)	(ξ, 2, 2) (ξ, 4, 4) (ξ, 3, 3)	(ξ, 5, 5) (ξ, 5, 5) (ξ, 2, 2)	(ξ, 5, 5) (ξ, 5, 5) (ξ, 5, 5)	(ξ, 5, 5) (ξ, 7, 7) (ξ, 5, 5)	(ξ, 5, 5) (ξ, 5, 5) (ξ, 5, 5)	(ξ, 5, 5) (ξ, 6, 6) (ξ, 5, 5)
3	(ξ, 6, 6) (ξ, 8, 8) (ξ, 7, 7)	(ξ, 1, 1) (ξ, 4, 4) (ξ, 3, 3)	∞	(ξ, 5, 5) (ξ, 4, 4) (ξ, 6, 6)	(ξ, 1, 1) (ξ, 2, 2) (ξ, 9, 9)	(ξ, 5, 5) (ξ, 5, 5) (ξ, 3, 3)	(ξ, 5, 5) (ξ, 1, 1) (ξ, 6, 6)	(ξ, 5, 5) (ξ, 2, 2) (ξ, 3, 3)	(ξ, 5, 5) (ξ, 3, 3) (ξ, 5, 5)	(ξ, 5, 5) (ξ, 4, 4) (ξ, 1, 1)
4	(ξ, 6, 6) (ξ, 4, 4) (ξ,5, 5)	(ξ, 3, 3) (ξ, 7, 7) (ξ, 3, 3)	(ξ,5, 5) (ξ, 3, 3) (ξ, 6, 6)	∞	(ξ, 6, 6) (ξ, 4, 4) (ξ, 7, 7)	(ξ, 6, 6) (ξ, 1, 1) (ξ, 5, 5)	(ξ, 4, 4) (ξ, 5, 5) (ξ, 5, 5)	(ξ, 7, 7) (ξ, 2, 2) (ξ, 5, 5)	(ξ, 8, 8) (ξ, 3, 3) (ξ, 5, 5)	(ξ, 9, 9) (ξ, 5, 5) (ξ, 5, 5)

5	$(\xi, 4, 4)$ $(\xi, 3, 3)$ $(\xi, 6, 6)$	$(\xi, 2, 2)$ $(\xi, 7, 7)$ $(\xi, 6, 6)$	$(\xi, 8, 8)$ $(\xi, 7, 7)$ $(\xi, 6, 6)$	$(\xi, 9, 9)$ $(\xi, 5, 5)$ $(\xi, 6, 6)$	∞	$(\xi, 8, 8)$ $(\xi, 2,2)$ $(\xi, 7, 7)$	$(\xi, 5, 5)$ $(\xi, 5, 5)$ $(\xi, 3, 3)$	$(\xi, 1, 1)$ $(\xi, 7, 7)$ $(\xi, 2, 2)$	$(\xi, 5, 5)$ $(\xi, 5, 5)$ $(\xi, 11, 11$	$(\xi, 8, 8)$ $(\xi, 6, 6)$ $(\xi, 3, 3)$
6	$(\xi, 9, 9)$ $(\xi, 11, 11)$ $(\xi, 10, 10)$	$(\xi, 11, 11)$ $(\xi, 10, 10)$ $(\xi, 5, 5)$	$(\xi, 7, 7)$ $(\xi, 5, 5)$ $(\xi, 3, 3)$	$(\xi, 12, 12)$ $(\xi, 9, 9)$ $(\xi, 7, 7)$	$(\xi, 5, 5)$ $(\xi, 1, 1)$ $(\xi, 9, 9)$	∞	$(\xi, 10, 10)$ $(\xi, 11, 11)$ $(\xi, 2, 2)$	$(\xi, 17, 17)$ $(\xi, 12, 12)$ $(\xi, 12, 12)$	$(\xi, 13, 13)$ $(\xi, 13, 13)$ $(\xi, 17, 17)$	$(\xi, 14, 14)$ $(\xi, 14, 14)$ $(\xi, 15, 15)$
7	$(\xi, 5, 5)$ $(\xi, 9, 9)$ $(\xi, 11, 11)$	$(\xi, 3, 3)$ $(\xi, 5, 5)$ $(\xi, 3, 3)$	$(\xi, 4, 4)$ $(\xi, 1, 1)$ $(\xi, 4, 4)$	$(\xi, 1, 1)$ $(\xi, 8, 8)$ $(\xi, 8, 8)$	$(\xi, 2, 2)$ $(\xi, 8, 8)$ $(\xi, 7, 7)$	$(\xi, 8, 8)$ $(\xi, 4, 4)$ $(\xi, 2, 2)$	∞	$(\xi, 2, 2)$ $(\xi, 17, 17)$ $(\xi, 10, 10)$	$(\xi, 12, 12)$ $(\xi, 6, 6)$ $(\xi, 6, 6)$	$(\xi, 16, 16)$ $(\xi, 7, 7)$ $(\xi, 1, 1)$
8	$(\xi, 10, 10)$ $(\xi, 5, 5)$ $(\xi, 11, 11)$	$(\xi, 5, 5)$ $(\xi, 3, 3)$ $(\xi, 9, 9)$	$(\xi, 2, 2)$ $(\xi, 1, 1)$ $(\xi, 2, 2)$	$(\xi, 11, 11$ $(\xi, 8, 8)$ $(\xi, 8, 8)$	$(\xi, 3, 3)$ $(\xi, 9, 9)$ $(\xi, 3, 3)$	$(\xi, 2, 2)$ $(\xi, 4, 4)$ $(\xi, 17, 17)$	$(\xi, 12, 12)$ $(\xi, 7, 7)$ $(\xi, 11, 11)$	∞	$(\xi, 6, 6)$ $(\xi, 2, 2)$ $(\xi, 12, 12)$	$(\xi, 6, 6)$ $(\xi, 1, 1)$ $(\xi, 6, 6)$
9	$(\xi, 10, 10)$ $(\xi, 11, 11)$ $(\xi, 10, 10)$	$(\xi, 4, 4)$ $(\xi, 5, 5)$ $(\xi, 2, 2)$	$(\xi, 2, 2)$ $(\xi, 18, 18)$ $(\xi, 8, 8)$	$(\xi, 13, 13)$ $(\xi, 10, 10)$ $(\xi, 9, 9)$	$(\xi, 8, 8)$ $(\xi, 8, 8)$ $(\xi, 6, 6)$	$(\xi, 10, 10)$ $(\xi, 3, 3)$ $(\xi, 6, 6)$	$(\xi,15, 15)$ $(\xi, 5, 5)$ $(\xi, 5, 5)$	$(\xi, 1, 1)$ $(\xi, 1, 1)$ $(\xi, 11, 11)$	∞	$(\xi, 10, 10)$ $(\xi, 12, 12)$ $(\xi, 4, 4)$
10	$(\xi, 5, 5)$ $(\xi, 8, 8)$ $(\xi, 11, 11)$	$(\xi, 10, 10)$ $(\xi, 7, 7)$ $(\xi, 4, 4)$	$(\xi, 4, 4)$ $(\xi, 5, 5)$ $(\xi, 10, 10)$	$(\xi, 12, 12)$ $(\xi, 1, 1)$ $(\xi, 2, 2)$	$(\xi, 5, 5)$ $(\xi, 8, 8)$ $(\xi, 9, 9)$	$(\xi, 12, 12)$ $(\xi, 2, 2)$ $(\xi, 4, 4)$	$(\xi, 5, 5)$ $(\xi, 6, 6)$ $(\xi, 17, 17)$	$(\xi, 1, 1)$ $(\xi, 16, 16)$ $(\xi, 6, 6)$	$(\xi, 1, 1)$ $(\xi, 6, 6)$ $(\xi, 11, 11)$	∞
	Bi-fuzzy Time Matrix(10×10) With three routes and conveyances respectively									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	$(\xi, .12, .12)$ $(\xi, .02, .02)$ $(\xi, .07, .07)$	$(\xi, .13, .13)$ $(\xi, .03, .03)$ $(\xi, .04, .04)$	$(\xi, .14, .14)$ $(\xi, .04, .04)$ $(\xi, .06, .06)$	$(\xi, .15, .15)$ $(\xi, .05, .05)$ $(\xi, .08, .08)$	$(\xi, .07, .07)$ $(\xi, .13, .13)$ $(\xi, .03, .03)$	$(\xi, .11, .11)$ $(\xi, .06, .06)$ $(\xi, .12, .12)$	$(\xi, .03, .03)$ $(\xi, .01, .01)$ $(\xi, .12, .12)$	$(\xi, .1, .1)$ $(\xi, .11, .11)$ $(\xi, .13, .13)$	$(\xi, .14, .14)$ $(\xi, .13, .13)$ $(\xi, .14, .14)$
2	$(\xi, .1, .1)$ $(\xi, .24, .24)$ $(\xi, .14, .14)$	∞	$(\xi, .17, .17)$ $(\xi, .16, .16)$ $(\xi, .06, .06)$	$(\xi, .01, .01)$ $(\xi, .17, .17)$ $(\xi, .1, .1)$	$(\xi, .11, .11)$ $(\xi, .2, .2)$ $(\xi, .2, .2)$	$(\xi, .07, .07)$ $(\xi, .06, .06)$ $(\xi, .1, .1)$	$(\xi, .16, .16)$ $(\xi, .03, .03)$ $(\xi, .11, .11)$	$(\xi, .01, .01)$ $(\xi, .07, .07)$ $(\xi, .03, .03)$	$(\xi, .06, .06)$ $(\xi, .1, .1)$ $(\xi, .04, .04)$	$(\xi, .02, .02)$ $(\xi, .15, .15)$ $(\xi, .05, .05)$
3	$(\xi, .06, .06)$ $(\xi, .13, .13)$ $(\xi, .16, .16)$	$(\xi, .18, .18)$ $(\xi, .11, .11)$ $(\xi, .22, .22)$	∞	$(\xi, .03, .03)$ $(\xi, .16 ,16)$ $(\xi, .25, .25)$	$(\xi, .04, .04)$ $(\xi, .05, .05)$ $(\xi, .01, .01)$	$(\xi, .1, .1)$ $(\xi, .11, .11)$ $(\xi, .11, .11)$	$(\xi, .04, .04)$ $(\xi, .15, .15)$ $(\xi, .03, .03)$	$(\xi, .1, .1)$ $(\xi, .05, .05)$ $(\xi, .15, .15)$	$(\xi, .2, .2)$ $(\xi, .1, .1)$ $(\xi, .07, .07)$	$(\xi, .15, .15)$ $(\xi, .1, .1)$ $(\xi, .12, .12)$
4	$(\xi, .07, .07)$ $(\xi, .04, .04)$ $(\xi, .05, .05)$	$(\xi, .13, .13)$ $(\xi, .07, .07)$ $(\xi, .06, .06)$	$(\xi, .15, .15)$ $(\xi, .13, .13)$ $(\xi, .14, .14)$	∞	$(\xi, .26, .26)$ $(\xi, .14, .14)$ $(\xi, .2, .2)$	$(\xi, .04, .04)$ $(\xi, .2, .2)$ $(\xi, .07, .07)$	$(\xi, .05, .05)$ $(\xi, .15, .15)$ $(\xi, .05, .05)$	$(\xi, .05, .05)$ $(\xi, .03, .03)$ $(\xi, .15, .15)$	$(\xi, .25, .25)$ $(\xi, .05, .05)$ $(\xi, .13, .13)$	$(\xi, .03, .03)$ $(\xi, .04, .04)$ $(\xi, .05, .05)$
5	$(\xi, .11, .11)$ $(\xi, .03, .03)$ $(\xi, .05, .05)$	$(\xi, .2, .2)$ $(\xi, .1, .1)$ $(\xi, .06, .06)$	$(\xi, .19, .19)$ $(\xi, .13, .13)$ $(\xi, .17, .17)$	$(\xi, .18, .18)$ $(\xi, .12, .12)$ $(\xi, .16, .16)$	∞	$(\xi, .03, .03)$ $(\xi, .1, .1)$ $(\xi, .11, .11)$	$(\xi, .04, .04)$ $(\xi, .07, .07)$ $(\xi, .04, .04)$	$(\xi, .15, .15)$ $(\xi, .16, .16)$ $(\xi, .17, .17)$	$(\xi, .03, .03)$ $(\xi, .03, .03)$ $(\xi, .08, .08)$	$(\xi, .07, .07)$ $(\xi, .07, .07)$ $(\xi, .04, .04)$
6	$(\xi, .04, .04)$ $(\xi, .04, .04)$ $(\xi, .16, .16)$	$(\xi, .02, .02)$ $(\xi, .2, .2)$ $(\xi, .12, .12)$	$(\xi, .1, .1)$ $(\xi, .02, 0.2)$ $(\xi, .22, .22)$	$(\xi, .2, .2)$ $(\xi, .12, .12)$ $(\xi, .1, .1)$	$(\xi, .16, .16)$ $(\xi, .21, .21)$ $(\xi, .08, .08)$	∞	$(\xi, .15, .15)$ $(\xi, .2, .2)$ $(\xi, .2, .2)$	$(\xi, .2, .2)$ $(\xi, .04, .04)$ $(\xi, .15, .15)$	$(\xi, .04, .04)$ $(\xi, .1, .1)$ $(\xi, .12, .12)$	$(\xi, .16, .16)$ $(\xi, .12, .12)$ $(\xi, .13, .13)$
7	$(\xi, .2, .2)$ $(\xi, .26, .26)$ $(\xi, .1, .1)$	$(\xi, .06, .06)$ $(\xi, .2, .2)$ $(\xi, .26, .26)$	$(\xi, .16, .16)$ $(\xi, .02, .02)$ $(\xi, .21, .21)$	$(\xi, .09, .09)$ $(\xi, .21, .21)$ $(\xi, .22, .22)$	$(\xi, .07, .07)$ $(\xi, .06, .06)$ $(\xi, .1, .1)$	$(\xi, .16, .16)$ $(\xi, .16, .16)$ $(\xi, .23, .23)$	∞	$(\xi, .1, .1)$ $(\xi, .1, .1)$ $(\xi, .06, .06)$	$(\xi, .15, .15)$ $(\xi, .13, .13)$ $(\xi, .25, .25)$	$(\xi, .02, .02)$ $(\xi, .06, .06)$ $(\xi, .27, .27)$
8	$(\xi, .23, .23)$ $(\xi, .03 .03)$ $(\xi, .01, .01)$	$(\xi, .16, .16)$ $(\xi, .01, .01)$ $(\xi, .21, .21)$	$(\xi, .1, .1)$ $(\xi, .1, .1)$ $(\xi, .11, .11)$	$(\xi, .07, .07)$ $(\xi, .08, .08)$ $(\xi, .16, .16)$	$(\xi, .02, .02)$ $(\xi, .11, .11)$ $(\xi, .02, .02)$	$(\xi, .12, .12)$ $(\xi, .1, .1)$ $(\xi, .04, .04)$	$(\xi, .04, .04)$ $(\xi, .02, .02)$ $(\xi, .01, .01)$	∞	$(\xi, .05, .05)$ $(\xi, .07, .07)$ $(\xi, .05, .05)$	$(\xi, .06, .06)$ $(\xi, .09, .09)$ $(\xi, .08, .08)$
9	$(\xi, .07, .07)$ $(\xi, .2, .2)$ $(\xi, .03, .03)$	$(\xi, .21, .21)$ $(\xi, .08, .08)$ $(\xi, .07, .07)$	$(\xi, .08, .08)$ $(\xi, .19, .19)$ $(\xi, .1, .1)$	$(\xi, .1, .1)$ $(\xi, .16, .16)$ $(\xi, .21, .21)$	$(\xi, .11, .11)$ $(\xi, .24, .24)$ $(\xi, .08, .08)$	$(\xi, .24, .24)$ $(\xi, .15, .15)$ $(\xi, .24, .24)$	$(\xi, .15, .15)$ $(\xi, .17, .17)$ $(\xi, .17, .17)$	$(\xi, .11, .11)$ $(\xi, .16, .16)$ $(\xi, .03, .03)$	∞	$(\xi, .17, .17)$ $(\xi, .03, .03)$ $(\xi, .05, .05)$
10	$(\xi, .2, .2)$ $(\xi, .07, .07)$ $(\xi, .16, .16)$	$(\xi, .21, .21)$ $(\xi, .21, .21)$ $(\xi, .03, .03)$	$(\xi, .18, .18)$ $(\xi, .08, .08)$ $(\xi, .06, .06)$	$(\xi, .24, .24)$ $(\xi, .17, .17)$ $(\xi, .1, .1)$	$(\xi, .03, .03)$ $(\xi, .03, .03)$ $(\xi, .16, .16)$	$(\xi, .1, .1)$ $(\xi, .16, .16)$ $(\xi, .17, .17)$	$(\xi, .03, .03)$ $(\xi, .15, .15)$ $(\xi, .11, .11)$	$(\xi, .15, .15)$ $(\xi, .11, .11)$ $(\xi, .21, .21)$	$(\xi, .16, .16)$ $(\xi, .06, .06)$ $(\xi, .07, .07)$	∞

Table 5.9: Optimum Results of BF-4DTSP (Model 5.1C)

DM	Path(Route,Vehicle)	Obj Value	Time	T_{max}
ODM	9(1,2)-8(1,3)-3(2,2)-6(2,3)-1(1,2)-4(2,1)-5(3,2)-2(3,3)-7(2,2)-10(1,3)	152.5	8.52	8.75
PDM	5(1,2)-1(2,3)-9(1,2)-4(3,2)-10(3,3)-3(1,3)-6(2,3)-7(1,3)-2(2,3)-8(1,3)	176.5	8.74	8.75
ODM	10(1,2)-8(2,1)-6(1,2)-3(2,1)-7(1,2)-9(1,3)-4(2,3)-2(2,1)-5(2,3)-1(1,1)	182.5	8.34	8.75
PDM	9(1,2)-7(1,3)-6(2,3)-4(3,2)-1(1,2)-8(1,1)-10(3,1)-5(2,1)-3(3,2)-2(1,1)	204.5	8.67	8.75
ODM	10(1,2)-9(2,3)-1(1,2)-7(2,1)-8(1,2)-4(3,1)-2(1,2)-5(3,3)-3(2,3)-6(1,2)	294.5	8.1	8.5
PDM	7(2,1)-1(3,2)-8(1,3)-5(2,3)-8(2,2)-3(2,3)-2(3,2)-9(1,3)-4(1,3)-10(1,1)	324.5	8.43	8.5
ODM	2(1,3)-1(2,1)-7(1,3)-8(2,3)-5(1,3)-9(1,3)-10(2,3)-6(2,3)-3(2,3)-4(1,2)	256.5	7.58	8.25
PDM	5(3,3)-10(1,2)-8(2,3)-6(1,3)-2(2,1)-9(1,3)-7(1,3)-4(2,2)-3(1,3)-1(2,1)	294.5	8.17	8.25
ODM	9(1,2)-10(2,3)-4(3,2)-8(2,3)-1(2,1)-7(1,2)-6(2,1)-5(3,2)-3(2,2)-2(3,2)	267.5	7.76	8.00
PDM	10(2,3)-7(3,2)-2(1,2)-6(1,2)-8(1,3)-9(1,3)-4(3,1)-1(2,1)-5(3,5)-3(3,3)	345.5	7.97	8.00

Table 5.10: Dispersion Results of rACO-GA

Instances	BKS	Best	Worst	Average	SD^b	Error(%)
fri26	937	937	939	937.21	0.97	0.01
bays29	2020	2020	2030	2020.17	1.72	0.86
bayg29	1610	1610	1616	1610.28	0.56	1.27
dantzig42	699	699	704	700.03	0.58	1.09
eil51	426	426	429	426.75	1.03	1.61
berlin52	7542	7542	7559	7544.02	1.02	2.53
st70	675	675	682	678.14	.93	1.51
eil76	538	538	552	541.23	3.43	1.34
pr76	108159	108159	108276	108203.9	1.92	1.49
rat99	1211	1211	1218	1216.32	1.25	2.91
kroa100	21282	21282	21578	21419.2	6.23	2.69
lin105	14379	14379	14413	14384.13	2.57	1.49
eil101	629	629	637	629.56	0.71	1.87
ch105	6528	6528	6621	6539.12	11.71	2.63
pr136	96772	97832	99496	98324.7	5.79	4.21

5.2.4 Statistical Test

Dispersion Tests for rACO-GA:

Performance of the proposed method is statistically tested running it 25 times and calculating the average value, standard deviation and percentage relative error according to optimal solution against some standard test problems. The results obtained by proposed method are given in Table 5.10.

Examining the Table 5.10, it is concluded that the proposed method, rACO-GA has generated the closer results to the optimal solutions with minimal standard deviations for the problems fri26, bays29, dantzig42, st70 and eil101 . It can be seen that except one problem pr136, for all other fourteen problems, best results by rACO-GA are the same as the corresponding best results in literature.

5.2.5 Discussion

In this investigation, an intelligent hybrid algorithm rACO-GA is proposed and illustrated in 4DTSP formulated in different environments. In rACO-GA, a rough (7 -point) set based selection and comparison crossover are used along with generation dependent random mutation. 4DTSP introduced for in the area of TSPs and regarded as highly NP-hard combinatorial optimization problems. Such 4DTSPs are here formulated crisp and bi-fuzzy costs and time boundary and solved by the proposed intelligent hybrid algorithm. Here, development of rACO-GA is in general form and it can be applied in other discrete problems such as network optimization, graph theory, solid transportation problems, vehicle routing, covering salesman problem, VLSI chip design, etc. In spite of the better results by rACO-GA, there is a lot of scope for development in rACO-GA, specially for the 4DTSPs. In the 4DTSP with routes and conveyances, we have assigned a route and conveyance arbitrarily during each crossover and mutation for the optimum selection of the routes. This is a limitation of the present 4DTSPs.

5.3 Model-5.2: A new Evolutionary Hybrid Algorithm for restricted 4- Dimensional TSP (r-4DTSP) in Uncertain Environment [2]

In this model, a hybridized soft computing technique is proposed to solve a restricted 4- dimensional TSP (r-4DTSP) where different paths with various number of conveyances are available to travel between two cities. Here some restrictions on paths and conveyances are imposed. The algorithm is a hybridization of ant colony optimization (ACO) and swap operator based particle swarm optimization (PSO) with genetic algorithm (GA). The initial solutions are produced by ACO which are used as swarm in PSO and then a modified GA with selection, comparison crossover and generation dependent mutation is used. The said hybrid algorithm (ACO-PSO-GA) is tested against some test functions and efficiency of the proposed algorithm is established. The r-4DTSPs are considered with crisp and bi-rough costs. In each environment, some statistical significant studies due to different time constraint values and other system parameters are presented. The models are illustrated with some numerical data.

5.3.1 Proposed hybrid ACO-PSO-GA

The proposed evolutionary hybrid algorithm, ACO-PSO-GA using common ACO for initial solution, the swap sequence based PSO and GA with rough set based pheromone update selection (7-point), comparison crossover and generation dependent random mutation. The proposed ACO-PSO-GA and its procedures are presented below:

(i) Representation:

Here a complete tour of N cities represents a solution of ants. So an N dimensional integer vector X_i = (x_{i1}, x_{i2}, ..., x_{iN}), Y_i = (r_{i1}, r_{i2}, ..., r_{is}) and Z_i = (v_{i1}, v_{i2}, ..., v_{iP}) are used as cities with route, and vehicles to represent a solution, where x_{i1}, x_{i2}, ..., x_{iN} represent N consecutive cities in a tour. In the algorithm, initially ACO is used to produce a set of paths (tours) for the salesman, which is a set of potential solutions for the PSO and after updating of the path, then the GA part of the algorithm is used.

[2]This portion is communicated in **Swarm And Evolutionary Computing**, with title *A new Evolutionary Hybrid Algorithm for restricted 4- Dimensional TSP (r-4DTSP) in Uncertain Environment.*

(ii) Ant Colony Optimization (ACO):

Here in the proposed algorithm, τ_{ij} represents amount of pheromone which lies on the path between nodes i and j, $iter1$, $iter2$ and $iter3$ represent iteration counter, $maxiter_1$, $maxiter_2$ and $maxgen$ represent maximum iteration number of the ACO, PSO algorithm and maximum generation number in GA part, n and N represent number of ants or population size and number of nodes/cities respectively, r and k stand respectively for different routes and vehicles in the problem. where r$\in \{1, 2, .., s\}$ and v$\in \{1, 2, .., p\}$. The remaining part of ACO algorithm are same as given in last model in section 5.2.1.(ii).

(iii) Particle Swarm Optimization:

After finding the paths by above ACO, we use the swap sequence for updating the paths. A PSO normally starts with a set of potential solution (called swarm) of the decision making problem. Individual solutions (swarm) are called particles and food is analogous to optimal solution. Here each particle i has a position vector x_i(t), a velocity V_i(t), the position at which the best fitness X_{pbest}(t) encountered by the particle, best position of the all particles X_{gbest}(t) in current generation t. In the next generation (t+1), the position and velocity of the particle are changed to X_i(t+1) and V_i(t+1) following the equations:

$$\left\{ \begin{array}{l} V_i(t+1) = wV_i(t) + c_1 r_1 (X_{pbest}(t) - X_i(t)) + c_2 r_2 (X_{gbest}(t) - X_i(t)), \\ X_i(t+1) = X_i(t) + V_i(t+1) \end{array} \right\} \tag{5.10}$$

where c_1, c_2 are acceleration constants, w is the inertia weight and r_1, r_2 are two random distinct values in [0,1]. For the TSP where swap sequence and swap operations are used to find velocity of a particle and its updating Equ. 5.10. For swap sequence based PSO, different nodes /cities are used to update a solution. A sequence of swap operators known as swap sequence are used which to transform a solution to updated solution.

(a) Swap Operator:

Let us consider a solution sequence of TSP with N nodes, X = (x_1, x_2, x_3,..,x_N, x_1), where $x_1 \in \{1, 2, 3, .., N\}$ and each x_i is distinct. Swap operator, SO(i,j) is defined as exchange of nodes x_i and x_j in solution sequence X. Now $\acute{X}$= X+ SO(i,j) as a new sequence of operator SO(i,j) on X. Here "+" is an operator but not as algebraic sum. For an example, consider TSP with seven nodes and X=(x_1, x_2, x_3, x_4,x_5,x_6,x_7)= (2,3,1,4,6,5,7). If the swap operator is SO(3,5), then $\acute{X}$= X+ SO(i,j)= (2,3,1,4,6,5,7) + SO(3,5) = (2,3,6,4,1,5,7). Here 3rd and 5th positions are exchanged.

(b) Swap Sequence:

The swap sequence SS is made up with one or more swap operators. Consider SS= (SO_1, SO_2,..., SO_n), where SO_1, SO_2,..., SO_n are swap operators, here the order of the swap operator in SS is important. All the swap operators of the swap sequence act on the solution in order. This can be formulated as below:

$$\acute{X} = X + SS = X + (SO_1, SO_2, ..., SO_n) = (((X+SO_1)+SO_2)...+SO_n)$$

Different swap sequences are used on the same solution may produce a same new solution. Then for a Basic Swap Sequence (BSS) is form which has the least swap operator. Several swap sequence are merged into a new swap sequence. Here we use the operator $\oplus$ for merging two swap sequences.

(c) Basic Swap Sequence:

Let us consider two solutions, A and B, to construct BSS namely SS which act on B to get A, SS= A$\ominus$B, We can swap the nodes in B according to A from left to right to get SS. Consider A:(1, 2, 3, 4, 5), B:(2, 3, 1, 5, 4), now A(1)=B(3)=1, so first swap operator is SO(1,3), $\acute{B}$=B $\oplus$ SO(1,3), similarly found SO(2,3) and SO(4,5). Thus Basic swap sequence SS= A $\ominus$ B = (SO(1,3), SO(2,3), SO(4,5)).

(d) Discrete PSO Updating:

Now the original PSO updated for TSP is as follows:

$$\left\{ \begin{array}{l} V_i(t+1) = wV_i(t) \oplus c_1 r_1 (X_{pbest}(t) \ominus X_i(t)) \oplus c_2 r_2 (X_{gbest}(t) \ominus X_i(t)), \\ X_i(t+1) = X_i(t) \oplus V_i(t+1) \end{array} \right\} \tag{5.11}$$

The given parameters r_1, r_2, c_1, c_2 and w are now defined as follows, $c_1 r_1 (X_{pbest}(t) \ominus X_i(t))$ gives all swap operators in BSS. Similarly for the $c_2 r_2 (X_{gbest}(t) \ominus X_i(t))$ also.

(e) Pseudo Code of PSO:

for i = 1 to n **do**
  $X_i(0) = X_i(t-1)$
  $X_{pbesti}(0) = X_i(0)$
  $V_k(0)$ = SO(i,j), i,j$\in \{1, 2, .., N\}$, i$\neq$j.
end for
t = 1
X_{gbest}= Minimum cost solution from solution set $\{X_1(0), X_2(0), ..., X_{n_i}(0)\}$
end do
for i = 1 to n_i **do**

Determine $V_i(t)$ and $X_i(t)$ using Equ. 5.11.
If f(X_{pbesti}(t-1)) $>$ f($X_i(t)$)
$X_{pbesti}(t) = X_i(t)$
else
$X_{pbesti}(t) = X_{pbesti}(t-1)$
end if
If f(X_{gbest}) $>$ f(X_i(t))
X_{gbest} (t)= X_i(t)
end if
end for

(iv) Genetic Algorithm:

(a) Rough set based pheromone classification:

After finding the solution from discrete PSO, we again collect pheromone quantity, then classify the pheromones depending on the minimum, average and maximum pheromone information. Since pheromones are represented by crisp values, we construct the common rough values from it,
Rough Pheromone =([r_1*avg ph , r_2*avg ph], [r_3*avg ph , r_4*avg ph]),
where r_1=$\frac{Max\ -Avg}{Avg}$, r_2=$\frac{Max\ +Min}{2}$, r_3=$\frac{Max\ -Min}{2}$, r_4=$\frac{Avg\ -Min}{Avg}$

According to the pheromone of the chromosome, it belongs to any one of the common rough pheromone values and corresponding p_c's are created of each chromosome as VVL, VL, L, M, H, VH, VVH. The common rough variables ([a,b],[c,d]) is extended to $0 \leq c \leq e \leq f \leq a \leq b \leq g \leq h \leq d$ and is described as below,

$$Pheromone = \begin{cases} VeryVerySmall(VVS) & for\, c \leq pheromone < e \\ VerySmall(VjS) & \text{for } e \leq pheromone < f \\ Small(S) & \text{for } f \leq pheromone < a \\ Medium(M) & \text{for } a \leq pheromone \leq b \\ High(H) & \text{for } b < pheromone \leq g \\ VeryHigh(VH) & \text{for } g < pheromone \leq h \\ VeryVeryHigh(VVH) & \text{for } h < pheromone \leq d \end{cases} \tag{5.12}$$

(b) Comparison Crossover:

(i) Determination of Probability of Crossover (p_c):

For a pair of chromosomes (X_i, X_j), we construct the following rough set. At first, the states of X_i and X_j i.e, (VVS, VS, S, M, H, VH, VVH) are determined by making trust measures of rough values w.r.t their pheromones in common rough pheromone region given in Equ. 5.12 . After the determination of states of pheromone intervals of the chromosomes, their crossover probabilities are determined as linguistic variables (VVL, Vl, L, M, H, VH, VVH) using rough trust measures which are presented in Table 5.1 following Equ. 5.12.

(ii) Crossover Mechanism:

The procedure are given in section 4.3.1(iii).

(c) Generation Dependent Random Mutation:

(i) Generation Dependent Mutation(Variable Method): Here we model formulate a modified form of mutation mechanism where probability of mutation (p_m) are determined by

$$p_m=\frac{k}{\sqrt{1+Current\ generation\ number}},\ k\in[0,1].$$

(ii) Selection for mutation: For each solution of P(t), generate a random number r from the range [0,1]. If $r < p_m$, then the solution is taken for mutation. Here p_m decreases gradually as generation increases. After calculating the p_m, mutation operation follows the conventional random mutation. Here we randomly choose two nodes from each chromosome and exchange their positions and replace the chromosome in the new offspring set.

(v) Hybrid Algorithm (ACO-PSO-GA):

Input: Set $iter_{ACO} = 0$, $iter_{GA} = 0$, $maxiter$ and Max_{gen} (S_0), Population Size (pop_size), Number of ants (n), Probability of Mutation (p_m), Problem Data (cost matrix, time matrix, route and vehicle set).

Output: The optimum and near optimum solutions.

1. Start

2. Set $iter_{ACO} = 0$, $iter_{PSO} = 0$, $iter_{GA} = 0$ and $Max_{gen}(S_0)$.

3. Initialize pheromone τ_{ijrk} for $i = 1, 2, ..., N$ and $j = 1, 2, ..., N$ using r_{th} route and k_{th} vehicle.

4. For $(iter_{ACO} \leq maxiter)$

5. Construct path of n ants, i.e., n tours $X_i = (x_{i1rk}, x_{i2rk}, .., x_{iNrk}, x_{i1rk})$, $i = 1, 2, .., n$ using τ_{ijrk}.

6. Make pheromone evaporation.

7. Update pheromone for all the paths by equation in section 3.2.4.

8. $iter_{ACO}$ = $iter_{ACO}$+1

9. End for

10. Set initial solution obtained from ACO.

11. For $(iter_{PSO} \leq maxiter_1)$
12. Initialize the $X_i(t)$, $Y_i(t)$, $Z_i(t)$
13. Determine X_{pbest}, $Xgbest$
14. Update by Equ. 5.11
15. end for
16. Store the best solutions
17. For $(iter_{GA} \leq S_0)$
18. Sum the pheromone of all individual chromosomes.
19. Clustere the pheromone.
20. Develop the linguistic VVP, VLP, LP, MP, HP, VHP, VVHP
21. Trust based p_c created.
22. Crossover operation.
23. Mutation operation.
24. Update the chromosome.
25. Update the pheromone.
26. Find best optimum and near optimum solutions.
27. $iter_{GA}$ = $iter_{GA}$+1
28. End for
29. Store global and near optimum solutions.
30. End

5.3.2 Mathematical Formulation and Its crisp equivalence

Model 5.2A: 4DTSP in restricted routes with time Constraints (r-4DTSP):
In real life, it is seen that in all stations, all types routes may not be available due to the geographical position of the station,weather conditions, etc. So it is more realistic, that restricted routes be considered to travel different stations. Let $c(i, j, r, k)$ and $t(i, j, r, k)$ be the cost and time respectively for travelling from i-th city to j-th city by the r-th route using k-th type conveyance. Then the salesman has to determine a complete tour (x_1, x_2, ...,x_N, x_1) and corresponding available route types (r_{m1}, r_{m2}, ..., r_{ms}) with conveyance types (v_{q1}, v_{q2}, ..., v_{qp}) providing maximum available $s_1(\leq$ S) and $p_1(\leq$ P) types of routes and conveyances to be used for the tour, where $x_i \in \{1, 2, ..N\}$ for $i = 1, 2, ..., N$, $r_{mi} \in \{1, 2, ..s_1\}$ and $v_{qi} \in \{1, 2, ..p_1\}$ for $i = 1, 2, ..., N$ and all x_i's are distinct. Then the problem

can be mathematically formulated as:

$$\left.\begin{array}{c}\text{minimize } Z=\sum_{i=1}^{N-1} c(x_i,x_{i+1},r_{mi},v_{qi})+c(x_N,x_1,r_{ml},v_{ql}),\\ \text{subject to } \sum_{i=1}^{N-1} t(x_i,x_{i+1},r_{mi},v_{qi})+t(x_N,x_1,r_{ml},v_{ql})\le t_{max},\\ where\ x_i\neq x_j, i,j=1,2...N, m=1,2,...s_1\,, q=1,2,..,p_1,\\ r_{mi},r_{ml}\in\{1,2..,or\ s_1\}, v_{qi},v_{ql}\in\{1,2..,or\ p_1\},\end{array}\right\} \quad (5.13)$$

Model 5.2B: r-4DTSP in bi-rough Environment (BR-r-4DTSP):

In the above problem Equ. 5.13, if costs and times are bi-rough variables, i.e, $\hat{\hat{c}}(i,j,r,k)$ and $\hat{\hat{t}}(i,j,r,k)$ respectively, time limit t_{max} is also bi-rough number $\hat{\hat{t}}_{max}$, then following the section 3.13.7, the above problem reduces to

$$\left.\begin{array}{c}\text{to minimize } Z=\sum_{i=1}^{N-1} \hat{\hat{c}}(x_i,x_{i+1},r_{mi},v_{qi})+\hat{\hat{c}}(x_N,x_1,r_{m1}v_{ql}),\\ \text{subject to } \sum_{i=1}^{N-1} \hat{\hat{t}}(x_i,x_{i+1},r_{mi},v_{qi})+\hat{\hat{t}}(x_N,x_1,r_{ml},v_{ql})\le \hat{\hat{t}}_{max},\\ where\ x_i\neq x_j, i,j=1,2...N, m=1,2,...s_1\,, q=1,2,..,p_1,\\ r_{mi},r_{ml}\in\{1,2..,or\ s_1\}, v_{qi},v_{ql}\in\{1,2..,or\ p_1\}.\end{array}\right\} \quad (5.14)$$

Equ. 5.14 can be reformulated as

$\sum_{i=1}^{N-1} \hat{\hat{c}}(x_i,x_{i+1},r_{mi},v_{qi})+\hat{\hat{c}}(x_N,x_1,r_{mi},v_{ql})\le f$, where f be a given crisp value. Using Bi-rough CCMOP in section 3.13.7, we have

$$\left.\begin{array}{c}\text{minimize } f\\ Ch\{\theta|\{|\hat{\hat{C}}(\theta)^T x\le f\}\ge\delta\}\ge\gamma\\ Ch\{\theta|\{|\hat{\hat{T}}(\theta)^T x\le \hat{\hat{T}}_{max}(\theta)^T\}\ge\theta\}\ge\eta\end{array}\right\} \quad (5.15)$$

The objective function for Ex-Tr are equivalently written as in section 3.13.8 below:

$$\left.\begin{array}{c}\text{minimize } f\\ Ex\{\lambda|Tr\{|\hat{\hat{C}}(\lambda)^T x\le f\}\ge\beta\}\\ Ex\{\lambda|Tr\{|\hat{\hat{T}}(\lambda)^T x-\hat{\hat{t}}_{max}(\lambda)^T\le w\}\ge\eta\}\end{array}\right\} \quad (5.16)$$

where $\hat{\hat{C}} = \sum_{i=1}^{N-1} \hat{\hat{c}}(x_i, x_{i+1}, v_i) + \hat{\hat{c}}(x_N, x_1, v_l)$,
$\hat{\hat{T}} = \sum_{i=1}^{N-1} \hat{\hat{t}}(x_i, x_{i+1}, v_i) + \hat{\hat{t}}_1(x_N, x_1, v_l)$,
$\hat{\hat{T}}_{max} = \hat{\hat{t}}_{max}$.

The objective function for Ex-Tr are equivalently written as below:

$$\text{minimize } f = \begin{cases} u - r + 2\alpha(s+r), & \text{if } u - r \le f \le u - p \\ \frac{u(p+q+r+s) - r(q+p) - p(s+r) + 2\alpha(s+r)(q+p)}{p+q+r+s} & \text{if } u - p \le f \le u + q \\ u - r + (2\alpha - 1)(s+r) & \text{if } u + q \le f \le u + s \end{cases} \tag{5.17}$$

$$\text{s.t. } w \ge \begin{cases} u_1 - r_1 + 2\eta(s_1+r_1), & \text{if } u_1 - r_1 \le w \le u_1 - p_1 \\ \frac{u_1(p_1+q_1+r_1+s_1) - r_1(q_1+p_1) - p_1(s_1+r_1) + 2\eta(s_1+r_1)(q_1+p_1)}{p_1+q_1+r_1+s_1}, & u_1 - p_1 \le w \le u_1 + q_1 \\ u_1 - r_1 + (2\eta - 1)(s_1+r_1) & \text{if } u_1 + q_1 \le w \le u_1 + s_1 \end{cases} \tag{5.18}$$

where f, w are crisp values and u and u_1 are expectation of rough variables, α, η which are predetermined confidence levels.

5.3.3 Numerical Experiments

Testing for hybrid ACO-PSO-GA:

The proposed ACO-PSO-GA algorithm was proposed on 15 standard benchmarked problems from TSPLIB [162]. Table 5.11 gives the results of hybrid ACO-PSO-GA along with the results by SGA, ACO and their hybridization ACO-GA. We compare the results in terms of total cost. The the average results and best found solution are obtained under 20 independent runs.

The parameters for the hybrid ACO-PSO-GA are set as those in Table 5.12 for different nodes of the TSP. As the size of the TSP increases , the pop-size, Maxgen, ant numbers for convergence for the optimal solution also increases.

Table 5.11: Test TSPLIB Problems by ACO-PSO-GA

Instances	Average	Result			Best	Found	Result	
	ACO-PSO-GA	ACO-GA	ACO	GA	ACO-PSO-GA	ACO-GA	ACO	GA
fri26	937.73	938.51	939.63	939.64	937	937	937	937
bays29	2020.45	2021.23	2022.78	2022.56	2020	2020	2020	2020
bayg29	1610.01	1610.34	1611.02	1610.97	1610	1610	1610	1610
dantzig42	699.12	699.27	703.51	700.07	699	699	703	699
eil51	427.26	427.8	432.98	429.31	426	426	430	426
berlin52	7544.81	7548.9	7936.35	7654.87	7542	7542	7883	7623
st70	678.11	677.34	699.51	682.17	675	675	687	675
eil76	538.31	539.65	567.27	545.86	538	538	547	547
pr76	108194.65	108265.76	108634.71	108572.32	108159	108159	108346	108258
rat99	1211.21	1212.52	1236.46	1218.71	1211	1211	1223	1211
kroa100	21298.06	21321.78	21567.82	21431.75	21282	21282	21427	21378
kroc100	20802.35	20834.87	20956.23	20971.75	20750	20750	20802	20831
kroa150	26616.38	26600.76	26952.34	26743.89	26524	26524	26871	26701
krob200	29367.65	29450.7	30887.34	29965.27	29413	29413	29944	29789
pr299	48906.14	49765.6	52945.78	50831.43	48743	48743	49765	49391

Table 5.12: Parameters for Hybrid Algorithm

Size (N)	Maxgen	Iter$_{PSO}$	Iter$_{ACO}$	Iter$_{GA}$	Maxiter	Ant number(n)	popsize	p$_c$	p$_m$	δ_1
N$\leq$ 50	200	30	80	120	100	30	50	0.35	0.1	0.2
50$< N \leq$ 100	300	40	120	180	200	50	100	0.3	0.15	0.2
100$< N \leq$ 150	400	40	200	300	300	80	100	0.35	0.2	0.3
150$< N \leq$ 200	500	50	200	400	400	100	130	0.4	0.2	0.3
200$< N \leq$ 250	600	60	250	450	400	100	150	0.45	0.2	0.3
250$< N \leq$ 300	900	80	400	500	500	100	150	0.45	0.25	0.3

Model 5.2A: r-4DTSP with time Constraint in Crisp Environment

For r-4DTSP, here we consider three types of conveyances and maximum three types of route as in Equ. 5.13. The cost and time matrices for the r-4DTSP are presented in Table 5.13.

From the Equ. 5.13, the equations for 3DTSP and 2DTSP are obtained taking only one route and one route along with one conveyance respectively. Taking the data from Table 5.13 for 1st route only, results of 3DTSP are obtained by the proposed algorithm. Similarly for 2DTSP, data for the first route and first conveyance are used.

Here we consider a deterministic 4DTSP given by Equ. 5.13 removing the route restrictions. The problem is solved by ACO-PSO-GA and the results are presented in Table 5.16.

Again we consider a deterministic restricted 4DTSP given by Equ. 5.13. The problem is solved by ACO-PSO-GA and the results are presented in Table 5.17.

Table 5.13: Input Data: Crisp r-4DTSP (Model 5.2A)

	Crisp Cost Matrix(10×10) With Three Route and Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	(35,36,27) (24,34,25) (17,23,26)	(18,39,30) (19,24,26) (30,24,31)	(20,33,34) (23,27,22) (23,22,28)	(30,21,62) (32,14,18) (31,43,32)	(23,24,27) (28,36,29) (57,28,39)	(41,37,21) (31,45,62) (24,11,28)	(17,15,9) (67,38,29) (11,34,13)	(35,36,37) (45,38,29) (19,28,17)	(23,45,18) (47,39,20) (17,29,10)
2	(35,26,17) (33,34,28) (22,27,29)	∞	(40,21,32) (57,28,39) (13,27,19)	(18,29,10) (18,39,20) (15,21,32)	(35,26,37) (27,36,30) (31,54,23)	(17,27,15) (45,25,16) (43,25,28)	(18,23,16) (23,26,22) (19,28,38)	(21,24,15) (41,39,20) (23,25,27)	(18,28,19) (17,28,19) (32,37,33)	(35,36,37) (27,26,29) (23,27,28)
3	(38,30,29) (23,45,18) (17,28,35)	(17,58,34) (23,24,27) (37,27,19)	∞	(12,25,14) (44,38,37) (39,23,43)	(42,25,46) (29,30,46) (43,33,54)	(19,27,35) (34,27,18) (21,26,16)	(29,19,24) (27,28,17) (15,17,19)	(17,17,19) (18,27,16) (21,27,28)	(17,16,19) (24,22,29) (21,26,28)	(15,18,19) (17,18,19) (17,22,28)
4	(28,20,11) (18,19,16) (56,23,19)	(10,22,14) (18,28,32) (333,46,28)	(17,8,29) (37,11,44) (48,29,10)	∞	(30,19,24) (30,17,11) (41,37,21)	(31,32,18) (17,27,15) (32,37,33)	(17,43,23) 11,34,13) (30,21,62)	(23,27,29) (35,26,17) (36,28,22)	(35,36,37) (28,36,29) (17,10,19)	(21,28,29) (33,21,38) (67,26,38)
5	(17,15,9) (34,29,11) (17,29,10)	(42,23,34) (45,19,20) (15,29,30)	(35,36,37) (29,10,28) (37,25,18)	(20,31,43) (36,29,13) (52,19,38)	∞	(32,37,33) (28,36,29) (35,26,17)	(28,36,29) (32,15,33) (17,34,23)	(17,19,10) (17,18,14) (29,27,27)	(21,22,29) (22,29,30) (35,36,37)	(28,28,19) (34,33,37) (43,36,23)
6	(22,25,17) (17,27,15) (23,24,27)	(17,15,9) 11,34,13) (43,25,28)	(32,37,33) (45,48,10) (23,24,27)	(43,25,28) (54,38,20) (28,29,17)	(23,24,27) (55,38,43) (45,56,57)	∞	(22,26,17) (28,36,29) (47,46,35)	(17,16,19) (17,54,29) (35,28,47)	(22,17,16) (28,39,10) (24,34,25)	(31,28,29) (39,40,29) (48,29,10)
7	(30,21,62) (30,21,62)	(35,26,17) (43,25,28) (43,25,28)	(32,37,33) (24,34,25) (48,29,10)	(17,27,15) (53,67,18) (18,15,13)	(23,24,27) (18,15,13) (18,28,29)	(48,29,10) (33,27,26) (28,25,29)	∞	(30,38,40) (23,24,27) (35,28,19)	(56,53,61) (28,39,28) (53,67,18)	(17,28,19) (18,15,13) (18,28,29)
8	(43,25,28) (11,34,13) (43,25,28)	(53,67,18) (18,15,13) (30,21,62)	(18,15,13) (18,28,29) (45,56,27)	(34,56,15) (45,56,27) (35,26,17)	(23,24,27) (28,25,26) (17,27,15)	(17,27,15) (17,27,15) (45,56,27)	(17,15,9) (17,10,11) (17,12,11)	∞	(17,27,15) (23,24,27) (23,17,19)	(45,56,27) (32,18,19) (24,27,20)
9	(18,15,13) (18,15,13) (19,18,17)	(17,15,9) 11,34,13) (17,27,15)	(45,56,27) (35,26,17) (23,24,27)	(54,37,29) (24,34,25) (18,15,13)	(23,24,27) (18,28,29) (45,56,27)	(48,29,10) (17,27,15) (19,18,17)	(19,18,17) (20,26,19)	12,34,13) (17,19,10) (28,36,29)	∞	(37,45,28) (54,37,29) (22,32,16)
10	(21,34,13) (30,21,62) (43,25,28)	(43,25,28) (11,34,13) (23,24,27)	12,33,13) (16,34,13) (23,24,27)	(11,34,23) (23,24,27) (18,15,13)	(17,27,15) (24,34,25) (17,27,15)	(48,29,10) (53,67,18) (35,36,37)	(17,27,15) (18,28,29) (18,28,29)	(54,37,29) (45,56,27) (28,36,29)	(54,37,29) (19,18,17) (17,27,15)	∞
	Crisp time Matrix(10*times*10) With Three route and Conveyances respectively									
-i/j	1	-2	3	4	5	6	7	8	9	10
1	∞	(.69,.68,.75) (.32,.45,.71) (.16,.18,.19)	(.84,.63,.7) (.24,.62,.44) (.18,.19,.31)	(.82,.7,.71) (.36,.64,.72) (.25,.28,.29)	(.72,.8,.42) (.32,.42,.26) (.27,.28,.29)	(.45,.34,.28) (.45,.56,.73) (.23,.25,.32)	(.33,.42,.45) (.23,.45,.36) (.31,.33,.34)	(.22,.32,.42) (.21,.52,.33) (.41,.43,.45)	(.42,.62,.45) (.24,.26,.27) (.32,.34,.36)	(.43,.53,.52) (.32,.28,.35) (.43,.46,.47)
2	.7,.66,.61 .8,.75,.71 .68,.7,.61	∞	.76,.71,.69 .68,.61,.59 .6,.61,.4	.67,.62,.6 .9,.85,.82 .29,.65,.32	.75,.68,.65 .6,.58,.5 .56,.48,.35	.68,.64,.61 .7,.65,.62 .17,.35,.52	.69,.63,.6 .31,.26,.2 .41,.56,.22	.51,.45,.4 .32,.34,.19 .42,.44,.12	.6,.57,.53 .7,.69,.62 .37,.29,.52	.8,.76,.71 .81,.76,.7 .61,.46,.73
3	.55,.51,.48 .6,.56,.53 .61,.58,.56	.72,.69,.62 .38,.31,.26 .6,.58,.51	∞	.81,.76,.7 .71,.68,.66 .8,.76,.71	.51,.46,.4 .7,.64,.61 .48,.44,.4	.59,.55,.52 .61,.58,.56 .62,.6,.57	.8,.75,.71 .9,.86,.81 .89,.86,.81	.65,.6,.59 .64,.6,.58 .68,.65,.61	.58,.55,.51 .8,.76,.71 .55,.5,.48	.67,.61,.58 .76,.71,.68 .64,.6,.57
4	.69,.64,.62 .78,.75,.71 .85,.83,.8	.86,.81,.79 .76,.71,.69 .81,.78,.74	.79,.75,.72 .9,.85,.82 .7,.64,.6	∞	.65,.63,.6 .76,.72,.7 .78,.71,.69	.69,.65,.62 .78,.75,.71 .68,.67,.65	.78,.74,.71 .68,.65,.61 .6,.54,.5	.6,.56,.52 .59,.58,.56 .79,.76,.72	.85,.82,.8 .78,.74,.71 .71,.69,.64	.68,.63,.59 .5,.45,.41 .6,.54,.5
5	.8,.76,.71 .81,.79,.75 .88,.81,.79	.55,.52,.49 .75,.74,.72 .61,.58,.54	.6,.58,.4 .58,.55,.5 .59,.58,.54	.78,.75,.71 .65,.62,.61 .55,.51,.48	∞	.62,.58,.55 .81,.75,.72 .55,.51,.45	.51,.45,.41 .81,.78,.75 .71,.68,.66	.67,.62,.59 .66,.61,.58 .82,.79,.75	.8,.76,.7 .88,.81,.78 .9,.87,.81	.69,.66,.62 .7,.68,.65 .9,.87,.83
6	.8,.75,.71 .81,.79,.76 .88,.85,.81	.65,.63,.6 .75,.72,.7 .66,.61,.59	.85,.82,.78 .7,.68,.62 .65,.62,.6	.88,.84,.79 .87,.84,.8 .85,.81,.78	.7,.67,.63 .6,.58,.55 .58,.54,.49	∞	.64,.6,.58 .55,.51,.46 .7,.68,.65	.55,.52,.48 .65,.63,.6 .76,.71,.68	.68,.61,.58 .73,.7,.68 .62,.58,.55	.65,.61,.58 .55,.52,.48 .65,.62,.6
7	.58,.54,.49 .56,.52,.48 .65,.62,.58	.65,.63,.6 .44,.38,.33 .71,.65,.6	.64,.6,.58 .6,.58,.55 .67,.64,.6	.7,.68,.65 .55,.51,.45 .71,.68,.64	.56,.54,.51 .38,.32,.28 .55,.53,.51	.55,.51,.46 .75,.71,.68 .52,.47,.4	∞	.85,.81,.78 .55,.54,.51 .75,.76,.72	.65,.61,.59 .58,.54,.5 .65,.61,.58	.78,.74,.69 .71,.68,.64 .65,.62,.58
8	.56,.52,.49 .54,.52,.51 .5,.43,.4	.7,.68,.65 .9,.88,.84 .8,.81,.78	.64,.6,.58 .41,.38,.37 .51,.45,.4	.56,.52,.5 .76,.74,.7 .56,.52,.49	.62,.58,.53 .62,.57,.55 .52,.48,.45	.55,.52,.48 .8,.77,.7 .88,.83,.8	.55,.54,.51 .78,.72,.7 .54,.53,.5	∞	.78,.76,.73 .43,.4,.36 .73,.7,.68	.58,.56,.51 .6,.54,.5 .58,.54,.49
9	.56,.51,.48 .88,.85,.81 .68,.65,.51	.58,.52,.5 .59,.57,.56 .58,.55,.53	.9,.85,.82 .62,.61,.58 .6,.54,.5	.7,.68,.64 .74,.7,.67 .68,.52,.58	.78,.75,.71 .65,.61,.58 .74,.7,.68	.74,.7,.68 .64,.61,.59 .67,.64,.6	.85,.81,.8 .62,.6,.57 .58,.54,.49	.62,.6,.58 .65,.61,.6 .79,.75,.72	∞	.69,.65,.63 .78,.73,.7 .72,.7,.68
10	.78,.71,.69 .7,.67,.64 .69,.64,.6	.66,.61,.58 .77,.74,.7 .78,.76,.71	.69,.65,.62 .8,.76,.74 .68,.65,.63	.74,.7,.68 .65,.6,.57 .76,.71,.68	.83,.78,.75 .62,.58,.56 .75,.71,.66	.65,.61,.58 .87,.83,.78 .68,.64,.59	.59,.54,.5 .68,.64,.61 .59,.55,.51	.55,.52,.47 .52,.48,.54 .64,.6,.58	.64,.59,.58 .45,.41,.37 .61,.59,.58	∞

Table 5.14: Results of 2DTSP in Crisp (Model 5.2A)

Algorithm	Path	Value	T_{max}
ACO-PSO-GA	3-7-2-1-5-9-10-4-6-8	137	Without T_{max}
	3-7-2-1-5-9-10-4-6-8	139	8.54
	4-7-8-1-5-9-10-3-6-2	145	8.51
	2-6-3-1-9-5-4-7-8-10	153	8.42
	4-6-2-8-5-9-10-7-3-1	156	8.25
	5-8-2-1-5-9-10-3-6-7	167	8.02
ACO-GA	2-6-1-9-5-10-8-4-3-7	147	Without T_{max}
	2-6-1-9-5-10-8-4-3-7	147	8.51
	7-2-6-4-3-5-10-8-9-1	154	8.57
	7-1-4-3-10-9-6-8-5-2	173	8.25
	5-2-8-10-9-6-1-3-4-7	189	8.1
ACO	6-3-9-7-5-2-1-10-8-4	193	8.7
GA	2-8-5-7-6-10-4-3-9-1	197	8.7
ACO-GA	4-8-9-1-3-7-2-10-5-6	204	8.00
ACO-PSO-GA	5-7-3-2-4-6-8-10-9-1	193	
	8-7-3-2-4-6-5-10-9-1	206	
ACO-GA	4-8-9-1-3-7-2-10-5-6	204	
ACO	3-8-5-7-6-10-4-2-9-1	227	
GA	8-2-1-3-4-10-7-9-6-5	221	
ACO-PSO-GA	8-2-7-9-4-3-5-6-10-1	216	7.5
ACO-PSO-GA	4-8-9-1-3-7-2-10-5-6	204	
ACO-GA	4-8-9-1-3-7-2-10-5-6	204	
ACO	5-6-2-7-8-10-3-9-4-1	392	
GA	10-6-2-7-8-5-3-9-4-1	398	

Table 5.15: Results of 3DTSP in Crisp (Model 5.2A)

Algorithm	Path(Vehicle)	Cost	Time	T_{max}
ACO-PSO-GA	9(1)-7(2)-8(3)-4(1)-3(1)-2(2)-5(1)-1(1)-10(2)-6(2)	170	8.75	8.75
	2(2)-1(3)-10(1)-3(1)-6(2)-7(1)-4(2)-5(2)-10(1)-9(2)	193	8.62	
	6(1)-9(2)-10(1)-7(2)-3(1)-8(2)-5(1)-4(1)-2(1)-1(3)	205	8.59	
	6(1)-10(2)-5(1)-7(1)-4(2)-3(3)-1(2)-10(3)-9(1)-2(1)	213	8.54	
	6(1)-7(2)-9(2)-8(1)-4(1)-5(2)-1(2)-2(2)-3(2)-10(1)	228	8.46	
ACO-GA	4(1)-5(1)-8(1)-3(3)-2(1)-10(3)-5(1)-4(2)-6(2)-7(2)	247	8.7	
ACO	3(2)-10(1)-8(1)-2(3)-3(3)-1(3)-5(1)-4(2)-6(2)-8(1)	242	8.7	
GA	3(2)-5(1)-8(2)-4(1)-2(1)-10(3)-5(1)-4(2)-6(2)-7(1)	247	8.7	
ACO-PSO-GA	3(2)-7(1)-4(1)-3(1)-1(1)-5(2)-10(2)-8(1)-6(1)-2(3)	282	7.95	8.00
	7(2)-9(1)-8(1)-10(2)-1(2)-3(2)-6(2)-5(1)-4(3)-2(1)	315	7.71	7.75
	10(1)-7(2)-6(1)-5(3)-4(2)-2(3)-3(1)-1(2)-8(2)-9(1)	376	7.58	

Table 5.16: Results of 4DTSP in Crisp (Model 5.2A)

Algorithm	Path(Route, Vehicle)	Cost	Time	T_{max}
ACO-PSO-GA	10(2,1)-7(3,2)-8(1,3)-4(2,1)-3(1,1)-2(1,2)-5(2,1)-1(3,1)-9(1,2)-6(2,2)	183	8.75	8.75
	2(1,2)-10(2,3)-1(1,1)-4(1,2)-6(1,2)-7(3,1)-3(2,2)-5(1,2)-10(2,1)-9(2,2)	187	8.67	
	6(1,3)-9(2,1)-10(1,1)-7(1,2)-3(1,3)-8(2,2)-5(3,1)-4(2,1)-2(1,1)-1(2,3)	216	8.53	
	6(2,1)-10(2,2)-5(1,1)-7(2,1)-4(2,3)-3(3,1)-1(2,1)-10(3,1)-9(2,1)-2(3,1)	219	8.42	
	6(1,3)-7(2,1)-9(2,1)-8(1,1)-4(2,1)-5(2,2)-1(1,2)-2(3,2)-3(1,2)-10(3,3)	245	8.34	
ACO-GA	4(3,3)-5(1,2)-8(3,1)-3(2,3)-2(1,1)-10(2,3)-5(2,1)-4(1,2)-6(1,2)-7(2,2)	262	8.7	
ACO	3(1,2)-10(2,1)-8(3,1)-2(2,3)-3(3,1)-1(1,1)-5(2,1)-4(1,2)-6(2,2)-8(1,2)	253	8.73	
GA	4(3,3)-5(1,2)-8(3,1)-3(2,3)-2(1,1)-10(2,3)-5(2,1)-4(1,2)-6(1,2)-7(2,2)	262	8.7	
ACO-PSO-GA	3(2,3)-7(1,2)-4(3,1)-3(2,1)-1(1,1)-5(2,1)-10(2,2)-8(1,3)-6(1,1)-2(3,2)	303	7.91	8.00
	8(3,2)-7(2,1)-9(3,1)-10(2,3)-1(2,2)-3(2,1)-6(2,1)-5(1,2)-4(3,3)-2(1,2)	338	7.66	7.75
	10(1,2)-7(,12)-6(3,1)-5(3,2)-4(2,2)-2(1,3)-3(2,1)-1(3,2)-8(2,2)-9(2,1)	381	7.48	

Table 5.17: Results of r-4DTSP in Crisp (Model 5.2A)

Algorithm	Path(Route, Vehicle)	Cost	Time	T_{max}
ACO-PSO-GA	10(1,1)-7(3,1)-8(1,3)-4(2,1)-3(1,1)-2(1,2)-5(2,1)-1(3,1)-9(1,2)-6(2,2)	192	8.75	8.75
	2(1,2)-10(2,2)-1(1,1)-4(1,2)-6(2,2)-7(3,1)-3(2,1)-5(1,2)-10(2,1)-9(2,2)	201	8.67	
	6(1,3)-9(2,1)-10(1,1)-7(1,2)-3(1,3)-8(2,2)-5(3,1)-4(2,1)-2(1,1)-1(2,3)	229	8.53	
	6(2,1)-10(2,2)-5(1,1)-7(2,1)-4(2,3)-3(3,1)-1(2,1)-10(3,1)-9(2,1)-2(3,1)	236	8.42	
	6(1,3)-7(2,1)-9(2,1)-8(1,1)-4(2,1)-5(2,2)-1(1,2)-2(3,2)-3(1,2)-10(3,3)	278	8.34	
ACO-GA	4(3,3)-5(1,2)-8(3,1)-3(2,3)-2(1,1)-10(2,3)-5(2,1)-4(1,2)-6(1,2)-7(2,2)	281	8.7	
ACO	3(1,2)-10(2,1)-8(3,1)-2(2,3)-3(3,1)-1(1,1)-5(2,1)-4(1,2)-6(2,2)-8(1,2)	253	8.73	
GA	4(3,3)-5(1,2)-8(3,1)-3(2,3)-2(1,1)-10(2,3)-5(2,1)-4(1,2)-6(1,2)-7(2,2)	262	8.7	
ACO-PSO-GA	3(2,3)-7(1,2)-4(3,1)-3(2,1)-1(1,1)-5(2,1)-10(2,2)-8(1,3)-6(1,1)-2(3,2)	303	7.91	8.00
	8(3,2)-7(2,1)-9(3,1)-10(2,3)-1(2,2)-3(2,1)-6(2,1)-5(1,2)-4(3,3)-2(1,2)	338	7.66	7.75
	10(1,2)-7(,12)-6(3,1)-5(3,2)-4(2,2)-2(1,3)-3(2,1)-1(3,2)-8(2,2)-9(2,1)	381	7.48	

Model 5.2B: r-4DTSP with time Constraint in bi-rough Environments (BR-4DTSP)

Here we take the cost and time as bi-rough values for the r-4DTSP as Equ. 5.17 and Equ. 5.18. Also we consider maximum three types routes and conveyances. We use bi-rough variables ($[\xi - p_1,\ \xi + q_1]$,$[\xi - r_1,\ \xi + s_1]$). For bi-rough values, we consider $p_1 = 2$, $q_1 = 2$, $r_1 = 3$, $s_1 = 3$ according to the Table 5.18. Since ξ is a rough variable connecting with the corresponding components in Table 5.18. For time matrix ($[\xi - p_1,\ \xi + q_1]$,$[\xi - r_1,\ \xi + s_1]$), we consider $p_1 = .01$, $q_1 = .01$, $r_1 = .2$, $s_1 = .2$.

Model 5.2B: r-4DTSP for virtual data

Here CSTSP are solved by ACO-PSO-GA with large scale data which are randomly generated for different cities and the results are presented in Table 5.20.

Table 5.18: Input Data: r-4DTSP(rough) (Model 5.2B)

	Rough **Cost** Matrix(10 ×10) for RCMOSTSP With Three Conveyances							
i/j	1	2	3	4	5	6	7	8
1	∞	([29,30],[27,32]) ([35,37],[34,39]) ([24,25],[23,28])	([13,15],[12,17]) ([36,37][34,39]) ([29,30],[27,31])	([20,21],[18,22]) ([31,33],[30,34]) ([29,30],[28,35])	([28,29],[26,31]) ([19,20],[18,21]) ([58,59],[57,62])	([23,26],[21, 27]) ([21,23],[20,25]) ([7,8],[6,10])	([15,16],[13,17]) ([34,36],[32,37]) ([44,46],[43,47])	([26,28],[23,29]) ([37,38],[35,39]) ([17,18],[16,20])
2	([33,34],[33,35] ([23,24],[22,26]) ([15,16],[14,17])	∞	([38,39],[37,41]) ([20,21],[19,22]) ([29,30],[28,32])	([15,16],[14,18]) ([28,29],[27,30]) ([9,10],[8,11])	([33,34],[32,35]) ([25,26],[24,27]) ([33,35],[32.37])	([39,40],[37,41]) ([28,29],27,31]) ([2,22],[20,23])	([39,40],[38,41]) ([29,30],[28,31]) ([57,59],[56,61])	([32,33],[31,34]) ([40,41],[39,42]) ([54,55],[53,59])
3	([34,35],[33,38]) ([28,29],[27,30]) ([28,29],[27,30])	([15,17],[13,18]) ([54,56],[53,58]) ([30,31],[29,34])	∞	([11,12],10,13]) ([22,24],[21,25]) ([13,14],[11,15])	([39,40],[37,42]) ([23,24],[22,25]) ([44,45],[43,46])	([33,35],[32,36]) ([33,34],[31,36]) ([32,33],31,34])	([18,19],[17,20]) ([10,11],[9,13]) ([7,8],[6,10])	([29,32],[28,33]) ([32,33],[31,30]) ([23,25],[22,26])
4	([26,28],[25,29]) ([17,18],[16,20]) ([9,10],[8,11])	([9,10],[8,11]) ([19,20],[18,22]) ([14,15],[13,17])	([15,16],[14,18]) ([8,9],[7,10]) ([27,29],[26,30])	∞	([28,30],[27,31]) ([18,19],[17,20]) ([22,23],[21,24])	([23,25],[22,26]) ([14,16],[13,17]) ([25,27],[26,28])	([19,21],[18,22]) ([30,31],[29,33]) ([31,33],[30,34])	([33,35],[32,36]) ([33,34],[32,36]) ([15,16],[14,17])
5	([15,17],[14,18]) ([13,15],[12,16]) ([6,7],[5,8])	([39,40],[38,42]) ([21,23],[20,24]) ([31,34],[30,35])	([33,35],[32,36]) ([33,34],[32,36]) ([35,37],[34,38])	([18,19],[17,20]) ([11,13],[10,14]) ([42,43],[41,44])	∞	([29,30],[28,32]) ([20,21],,[19,22]) ([40,41],[39,43])	([43,44],[42,45]) ([15,16],[13,17]) ([25,27],[24,28])	([28,29],[27,30]) ([29,30],[27,31]) ([12,13],[11.14])
6	([15,16],14,18]) ([6,7].[5,8]) ([7,8],[6,10])	([27,28],[26,29]) ([21,22],[20,23]) ([28,29],[27,30])	([4,6],[3,8]) ([25,26],[24,27]) ([26,28],[25,29])	([6,7],[5,8]) ([7,9],[6,10]) ([11,12],[10,15])	([26,27],[28,30]) ([27,29],[26,30]) ([38,39],[37,40])	∞	([32,33],[30,34]) ([41,42],[40,44]) ([23,24],[22,25])	([39,40],[38,42]) ([29,31],[28,30]) ([21,22],[20,25])
7	([33,34],[35,37]) ([36,39],[35,40]) ([28,30],[27,31])	([25,26],[23,28]) ([48,49],[47,53]) ([25,26],[23,27])	([28,29],[27,30]) ([37,38],[36,39]) ([24,25],[23,26])	([21,22],[20,23]) ([40,43],[39,44]) ([23,24],[22,25])	([36,37],[35,38]) ([55,56],[54,58]) ([39,40],[38,41])	([39,40],[38,42]) ([20,21],[19,25]) ([43,44],[42,45])	∞	([8,9],[7,10]) ([39,40],[38,43]) ([11,13],[10,14])
8	([39,40],[37,41]) ([41,42],[40,44]) ([22,23],[21,24])	([23,25],[22,26]) ([5,6],[4,7]) ([15,17],[14,18])	([29,32],[28,33]) ([49,53],[48,54]) ([44,45],[43,47])	([38,40],[37,41]) ([19,21],[18,22]) ([39,40],[38,42])	([35,36],[33,38]) ([33,36],[31,37]) ([45,47],[44,48])	([23,25],[22,27]) ([13,16],[12,18]) ([5,6],[4,7])	([40,41],[39,42]) ([20,21],[19,22]) ([41,43],[39,44])	∞
	Rough **Time** Matrix(10×10) for RCSTSP With Three Conveyances							
i/j	1	2	3	4	5	6	7	8
1	∞	([.56,.58],[.55,.62]) ([.52,.54],[.51,.55]) ([.25,.27],[.23,.28])	([.71,.73],[.7,.75]) ([.51,.53],[.5,.56]) ([.63,.64],[.61,.67])	([.68,.69],[.67,.7]) ([.57,.6],[.54,.61]) ([.54,.56],[.53,.59])	([.62,.64],[.61,.66]) ([.71,.73],[.7,.78]) ([.31,.33],[.3,.34])	([.81,.83],[.8,.87]) ([.69,.71],[.68,.73]) ([.81,.83],[.8,.88])	([.76,.77],[.75,.8]) ([.55,.58],[.53,.59]) ([.47,.49],[.45,.5])	([.67,.68],[.66,.69]) ([.48,.52],[.47,.54]) ([.65,.66],[.64,.68])
2	([.54,.55],[.53,.6]) ([.64,.65],[.61,.67]) ([.72,.73],[.7,.79])	∞	([.51,.52],[.5,.54]) ([.71,.72],[.69,.74]) ([.61,.63],[.6,.64])	([.7,71],[.67,.77]) ([.6,.62],[.57,.63]) ([.76,.77],[.74,.85])	([.63,.64],[.61,.6]) ([.61,.62],[.58,.68]) ([.55,.56],[.52,.58])	([.53,.55],[.51,.56]) ([.6,.63],[.57,.64]) ([.66,.67],[.65,.7])	([.5,.51],[.52,.54]) ([.61,.62],[.6,.66]) ([.33,.35],[.31,.36])	([.6,.62],[.57,.63]) ([.51,.52],[.5,.53]) ([.27,.29],[.26,.32])
3	([.55,.56],[.53,.58]) ([.6,.62],[.59,.64]) ([.61,.63],[.6,.66])	([.71,.72],[.7,.77]) ([.33,.34],[.31,.35]) ([.6,62],[.57,.64])	∞	([.76,.77],[.75,.79]) ([.67,.7],[.66,.72]) ([.76,.77],[.73,.8])	([.53,.54],[.51,.54]) ([.7,.71],[.69,.74]) ([.43,.44],[.42,.49])	([.53,.54],[.5,.59]) ([.55,.56],[.53,.59]) ([.6,.62],[.57,.58])	([.71,.72],[.7,.74]) ([.81,.83],[.79,.85]) ([.83,.84],[.81,.86])	([.57,.59],[.56,.62]) ([.6,.61],[.59,.63]) ([.68,.67],[.66,.67])
4	([.61,.62],[.6,.65]) ([.73,.74],[.7,.76]) ([.76,.78],[.74,.84])	([.81,.82],[.79,.85]) ([.71,.73],[.7,.74]) ([.76,.77],[.75,.79])	([.77,.78],[.76,.79]) ([.85,.87],[.83,.9]) ([.63,.65],[.62,.66])	∞	([.65,.66],[.63,.67]) ([.73,.74],[.7,.79]) ([.67,.68],[.65,.71])	([.68,.69],[.67,.7]) ([.71,.72],[.73,.77]) ([.66,.69,],[.64,.7])	([.71,.73],[.7,.77]) ([.6,.63],[.56,.65]) ([.61,.63],[.6,.66])	([.67,.68],[.64,.69]) ([.56,.58],[.55,.6]) ([.71,.73],[.7,.77])
5	([.76,.77],[.74,.8]) ([.76,.77],[.73,.8]) ([.83,.84],[.8,.88])	([.52,.54],[.51,.55]) ([.67,.68,],[.65,.69]) ([.56,.58],[.55,.6])	([.56,.57],[.55,.6]) ([.56,.58],[.55,.6]) ([.51,.53],[.5,56])	([.73,.75],[.72,.76]) ([.78,.79],[.76,.82]) ([.49,.51],[.47,.52])	∞	([.63,.65],[.6,.66]) ([.73,.74],[.7,.76]) ([.53,.54],[.5,.55])	([.47,.48],[.44,.5]) ([.76,.78],[.75,.8]) ([.67,.68][.66,.69])	([.61,.63],[.58,.64]) ([.63,.64],[.62,.66]) ([.74,.76],[.73,.8])
6	([.78,.8],[.7,.81]) ([.87,.89],[.85,.9]) ([.8,.81],[.78,.85])	([.67,.68],[.59,.69]) ([.77,.79],[.76,.8]) ([.67,.7],[.66,.71])	([.86,.88],[.83,.89]) ([.73,.74],[.72,.76]) ([.63,.65],[.6,.66])	([.8,.85],[.78,.88]) ([.83,.88],[.81,.89]) ([.76,.8],[.73,.81])	([.69,.7],[.66,.71]) ([.67,.68],[.66,.7]) ([.51,.53],[.5,.54])	∞	([.6,.63],[.56,.64]) ([.53,.55],[.51,.56]) ([.73,.74],[.7,.75])	([.51,.55],[.5,.56]) ([.61,.63],[.6,.67]) ([.7,.78])
7	([.55,.56],[.5,.57]) ([.55,.57],[.53,.6]) ([.63,64],[.6,.67])	([.66,.67],[.65,.68]) ([.41,.42],[.4,43]) ([.68,.7],[.66,.71])	([.63,.64],[.62,.67]) ([.56,.59],[.55,.6]) ([.67,.68],[.66,.71])	([.71,.72],[.69,.75]) ([.49,.52],[.47,.56]) ([.65,.69],[.64,.7])	([.61,.62],[.6,.63]) ([.33,.37],[.31,.39]) ([.49,.54],[.48,.55])	([.5,.54],[.49,.56]) ([.73,.76],[.71,.77]) ([.5,.56],[.49,.57])	∞	([.78,.79],[.77,.84]) ([.55,.56],[.54,.58]) ([.79,.8],[.77,.82])
8	([.51,.55],[.5,.56]) ([.51,.55],[.5,.56]) ([.7,.72],[.69,.73])	([.67,.7],[.65,.71]) ([.73,.74],[.7,.78]) ([.73,.74],[.7,.77])	([.63,.65],[.6,.66]) ([.41,.42],[.39,.43]) ([.49,.5],[.47,.52])	([.57,.58],[.55,.6]) ([.7,.71],[.67,.72]) ([.52,.54],[.5,.55])	([.56,.57],[.55,.59]) ([.56,.6],[.55,.57]) ([.45,.48],[.43,.49])	([.66,.68],[.65,.69]) ([.76,.77],[.75,.78]) ([.87,.88],[.86,.89])	([.56,.57],[.55,.59]) ([.71,.72],[.7,.74]) ([.5,.52],[.49,.55])	∞

Table 5.19: Optimum Results of BR-r-4DTSP (Model 5.2B)

Path(Route,Vehicle)	Obj Value	Time	T_{max}
7(1,1)-3(2,2)-6(1,3)-1(1,2)-4(3,1)-5(3,2)-2(1,3)-8(2,1)	136	7.47	
3(1,2)-1(2,3)-4(3,2)-5(1,3)-6(2,3)-7(1,3)-2(2,3)-8(1,3)	138.5	7.6	
8(2,1)-7(1,2)-3(2,1)-6(1,2)-4(2,3)-2(2,1)-5(2,3)-1(2,1)	141.5	7.34	7.75
7(1,3)-6(2,3)-4(3,2)-1(1,2)-8(1,1)-5(2,1)-3(3,2)-2(1,1)	154.5	7.67	
1(1,2)-7(2,1)-8(1,2)-4(3,1)-2(1,2)-5(3,3)-3(2,3)-6(1,2)	177.5	7.1	
7(2,1)-1(3,2)-8(1,3)-5(2,3)-8(2,2)-3(2,3)-2(3,2)-4(1,3)	184.5	7.25	7.5
3(1,1)-1(3,1)-7(2,3)-8(2,3)-5(2,3)-6(2,1)-2(1,3)-4(1,2)	226.5	7.15	7.25
4(2,1)-8(2,3)-6(1,3)-2(2,1)-7(1,3)-5(2,2)-3(1,3)-1(2,1)	234.5	7.17	
4(3,2)-8(2,3)-1(2,1)-7(1,2)-6(2,1)-5(3,2)-3(2,2)-2(3,2)	260	6.6	
7(1,2)-2(3,2)-6(1,2)-8(1,3)-4(3,1)-1(2,1)-5(3,5)-3(3,3)	285.5	6.97	7.00

Table 5.20: Results with virtual data (Model 5.2B)

Instances (Cities)	Costs	T_{max}
15×15	254	9.5
20×20	365	13.7
25×25	457	18.5
30×30	565	25.5
35×35	951	31.4
40×40	1462	44.3
45×45	1824	61.5
50×50	2568	73.1
80×80	7145	78.3
100×100	1512	131.5
150×150	27410	185.5
250×250	38652	276.1

5.3.4 Statistical Test

Performance of the proposed method is statistically tested running it 25 times and calculating the average value, standard deviation and percentage relative error according to optimal solution against some standard test problems. The results obtained by proposed method are given in Table 5.21. Examining the Table 5.21, it is concluded that the proposed method, hybrid algorithm has generated the closer results to the optimal solutions with minimal standard deviations for the problems bayg29, eil51, berlin52 and rat99. It can be seen that three problems eil76, kroa100 and kroa200 have large size of SD and all other problems close to the standard results. Only kroa200 not found the best results by ACO-PSO-GA but all other are the same as the corresponding best results in literature.

5.3.5 Discussion

In this investigation, a new evolutionary hybrid algorithm ACO-PSO-GA is proposed and illustrated in r-4DTSP formulated in different environments. In the proposed algorithm, where initial solutions are generated by ACO, then swap operator based discrete PSO used and at end GA is applied with a rough 7 - point pheromone based selection, comparison crossover along with generation

Table 5.21: Dispersion Tests of ACO-PSO-GA

Instances	BKS	Best	Worst	Average	SD[b]	Error(%)
fri26	937	937	939	937.32	1.31	0.19
bays29	2020	2020	2034	2020.25	2.37	1.21
bayg29	1610	1610	1616	1610.42	0.46	0.24
dantzig42	699	699	704	700.71	1.52	1.49
eil51	426	426	429	427.15	0.98	0.17
berlin52	7542	7542	7567	7544.45	0.76	1.37
st70	675	675	686	679.4	1.43	0.23
eil76	538	538	557	543.3	23.57	0.53
pr76	108159	108159	108343	108211.73	2.12	2.70
rat99	1211	1211	1220	1217.5	0.74	0.29
kroa100	21282	21282	21604	21432.30	56.17	1.07
lin105	14379	14379	14431	14387.25	1.35	0.94
eil101	629	629	646	629.7	1.23	0.07
ch105	6528	6528	6636	6543.7	31.62	3.46
kroa200	29368	29468	29874	29736.15	103.28	2.87

dependent random mutation. For the first time restricted 4DTSP are introduced in the area of TSPs and regarded as highly NP-hard combinatorial optimization problems. Such r-4DTSPs are formulated with crisp and bi-rough costs and time boundary and solved by the proposed ACO-PSO-GA. Here, development of ACO-PSO-GA is in general form and it can be applied in other discrete problems such as network optimization, graph theory, solid transportation problems, vehicle routing, VLSI chip design, etc. In spite of the better results by ACO-PSO-GA, there is a lot of scope for development in ACO-PSO-GA, specially for the r-4DTSPs. In the four dimensional TSPs with conveyances and routes, we have assigned a conveyance and route arbitrarily during each crossover and mutation for the optimum selection of the routes and conveyances. This is a limitation of the present r-4DTSPs.

5.4 Conclusion

In this chapter, we formulated two hybridized evolutionary algorithms and solved 4DTSP and restricted 4DTSP under crisp, bi-fuzzy and bi-rough environments. The above 4DTSP and r-4DTSP are also new in TSP family. To the best of our knowledge, there is no direct application of PSO to TSP till now. Here we have also presented a sequence based PSO algorithm along with other two bio-inspired heuristics ACO and GA to solve proposed TSPs. Here, real-life complex problems such as courier services, online retailer business, etc, which are exponentially increasing in the third world countries, can be modelled like r-4DTSP and solved through proposed hybrid heuristics. This method/its modified form can be used to solve the decision-making problems easily in other areas such as network optimization, routing, VLSI chip design, social networking, supply chain, logistics etc,. The proposed algorithms can be extended to solve multi objective optimization problems.

Part III

Multi-Objective Optimization Using a Heuristic Method

Chapter 6

Multi-Objective Optimization Using Heuristics Algorithms

6.1 Introduction

This chapter aims at presenting the general problem of decision making in unknown, complex or changing environment by an extension of static multi-objective optimization problem. General optimization problem is defined, which encompasses not just dynamics, but also change in the optimization problem itself, with focus on changing number of objectives used to evaluate potential solutions. In order to solve a defined problem, a variant of multi-objective genetic algorithm was used. Since the chapter focuses on the performance of the algorithm as well as used for solving the problem, but tends to demonstrate the approach, experimental results produced by tests with MOGA are presented. These experimental results clearly demonstrate that MOGA successfully furnished the population of potential solutions to the problem for different test cases, such as homogeneous, non-homogeneous, and the problem with changing number of objectives.

Using various approaches, such as estimation of behavior of the system with statistically known disturbances, introduction of adaptation of controller parameters etc. [7], a wider problem domain can be encompassed, nevertheless it is still very clearly defined in advance. An approach to apply a multi-objective evolutionary algorithm to solving a defined dynamic multi-objective problem of search for solution was demonstrated in this chapter.

Dynamic multi-objective problems defined based on a class of test functions with known features have been chosen in order to evaluate the application of the proposed approach, in conditions when the features of the problem are known (features of test functions, Pareto front etc.). Also we, design 3DTSP in the form of two objectives as cost and time with risk constraint. Here for an unkonwn problem as muti-objective, solid TSPs are modeled in different uncertain environments. The impreciseness in MOGA are of fuzzy, fuzzy extended and rough environment. Statistical tests are done for each case for the effectiveness of the proposed algorithms.

6.2 Model-6.1: An imprecise Multi-Objective Genetic Algorithm for uncertain Constrained Multi-Objective Solid Travelling Salesman Problem [1]

In this model, an imprecise Multi-Objective Genetic Algorithm (iMOGA) is developed to solve Constrained Multi-Objective Solid Travelling Salesman Problems (CMOSTSPs) in crisp, random, random-fuzzy, fuzzy-random and bi-random environments. In the proposed iMOGA, '3 - and 5 - level linguistic based age oriented selection', probabilistic selection and an adaptive crossover are used along with a new generation dependent mutation. In each environment, some sensitivity studies due to different risks/discomfort factors and other system parameters are presented. To test the efficiency, combining same size single objective problems from standard TSPLIB [162], the results of such multi-objective problems are obtained by the proposed algorithm, simple MOGA (Roulette wheel selection, cyclic crossover and random mutation), NSGA-II, MOEA-D/ACO and compared. Moreover, a statistical analysis (Analysis of Variance) is carried out to show the supremacy of the proposed algorithm.

6.2.1 Proposed iMOGA

Here a proposed algorithm, iMOGA using the fuzzy (3-level linguistic) and fuzzy extended (5-level linguistic) age based (FEA) selection, probabilistic selection, an adaptive crossover and a generation dependent mutation is developed.

[1] This portion is published in **Expert Systems With Applications, Elsevier** , 46(2016), 196-223, with title *An imprecise Multi-Objective Genetic Algorithm for uncertain Constrained Multi-Objective Solid Travelling Salesman Problem.*

Initially a randomly set of potential solutions is generated and then using proposed algorithm, we find out the Pareto optimal solutions until the termination criteria are encountered. The proposed iMOGA and its procedures are presented below:

(i) Representation:

Here a complete tour on N cities represents a solution. So an N dimensional integer vector X_i = (x_{i1}, x_{i2}, ..., x_{iN}) is used to represent a solution (path), where x_{i1}, x_{i2}, ..., x_{iN} represent N consecutive cities in a tour. Population size number M and i-th solution X_i = (x_{i1}, x_{i2}, ..., x_{iN}), where x_{i1}, x_{i2}, ..., x_{iN}, are randomly generated by random number generator between 1 to N maintaining the TSP conditions such as not repeating of cities (nodes) and also satisfying the constraints. Fitness are evaluated by summing the costs and times between the consecutive cities (nodes) of each solution (chromosome). The f(X_i) represents the i-th solution fitness in the solution space. Since the maximum population size is M, so M numbers of solutions (chromosomes) are generated randomly.

(ii) Selection:

Here three selection procedures are used for the selection of chromosomes. These are as follows:

(a). Fuzzy Set Based Age Dependent Selection

For the solution of an optimization problem, in the proposed iMOGA, the age of a chromosome is determined by a new mechanism based on weighted mean of their two objective values i.e. fitness values and then a **'fuzzy age based selection'** is applied. Here the age of each chromosome lie in a region of the common age represented by a fuzzy set using three linguistic expressions. These regions are termed as "young", "middle" and "old". So for the age of each chromosome, a linguistic value - young, middle or old is created. Now according to the age distributions of the members (in pair) of the mating pool, similar linguistic variables such as low, medium and high are generated for the said chromosomes to fix p_c's. Using the membership function of fuzzy set, the probability of crossover, p_c for each chromosome is assigned by the corresponding linguistic variables (cf Table 6.1).

Last et al. [88] and Roy et al. [147] improved the performance of GAs by providing a new fuzzy-based extension of the Life Time feature. They used a fuzzy logic controller (FLC) to adapt the crossover probability as a function of the chromosomes ages. These algorithms used three types of fuzzy classifications on the

Table 6.1: Fuzzy Based Linguistics

Chromosomes	Young	Middle	Old
Young	Low	Medium	Low
Middle	Medium	High	Medium
Old	Low	Medium	Low

basis of ages. Also, they consider the age of the chromosome in fuzzy environment. But, here we calculate the age differently which is described below and form a common fuzzy age. Next each chromosome age is compared with common fuzzy age to create the membership values. Then according to the Table 6.1, corresponding p_cs are generated which are also presented in Fig. 6.2.1. The general principle is that for both young and old individuals, the crossover probability is naturally low, while there is a certain age interval, where this probability is high. The concepts of young, old, and middle-aged are modeled as linguistic variables.

(a) Age formation

The above M such two-objective solutions have fitnesses represented by $f_1(x_i)$ and $f_2(x_i)$ of the i$-th$ chromosomes. Now $f(x_i)=\lambda f_1(x_i)+(1-\lambda)f_2(x_i)$, $\lambda \in$ rand [0,1]. At the time of initialization, each chromosome age is defined as null. Now in every generation, the age is counted using the mechanism in Equ. 4.48.

Now since age is calculated as crisp values, we construct the common fuzzy values from it as,

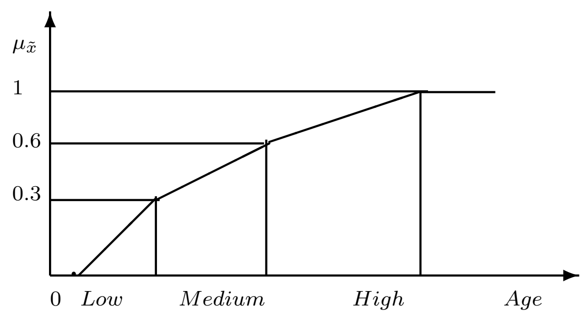
Fuzzy Age=(r_1*avg age, r_2*avg age, r_3*avg age),
where $r_1=\frac{Avg\ Age-Min\ Age}{Avg\ Age}$, $r_2=\frac{Max\ Age}{2}$, $r_3=\frac{Max\ Age-Avg\ Age}{Avg\ Age}$

For common fuzzy age (a, b, c) and ϵ (a small positive number given by the user), it is described as

$$Age = \begin{cases} Young & for\ a \leq age < b-\epsilon \\ Middle & \text{for } b-\epsilon \leq age \leq b+\epsilon \\ Old & \text{for } b+\epsilon < age \leq c \end{cases} \quad (6.1)$$

Table 6.2: Fuzzy Extended Based Linguistic

Chromosomes	Very Young	Young	Middle	Old	Very Old
Very Young	Very Low	Low	Medium	Low	Very Low
Young	Low	Low	High	Low	Very Low
Middle	Medium	High	Very High	High	Medium
Old	Low	Low	High	Low	Very Low
Very Old	Very Low	Very Low	Medium	Very Low	Very Low



$Fig.6.2.1 : Fuzzy\ age\ distribution\ of\ P_c.$

(b) Fuzzy Extended Age Based Selection:

As the mating selection enhances the exploitation of existing solutions and thus increases searching in more probable search regions, it will be more fruitful to divide the chromosomes, ages into more fuzzy classifications. To have more accurate classification, we make five classifications instead of above three and then, the region of common age is divided into very young, young, middle, old and very old. As before, combining the eligible parents, the very low, low, medium, high and very high linguistic variables are assigned for p$_c$'s of chromosomes. To achieve this, for the first time, membership function of fuzzy variable is divided and defined in the five regions which are shown in Equ. 6.2 and Fig. 6.2.2. Determined p_c values of the extended linguistics are also given below in the Fig 6.2.2.

The common fuzzy age (a,b,c) is extended to $0 \le a \le a_{11} \le a_{12} < b < b_{11} \le b_{12} \le c$ and is described as below,

$$Age = \begin{cases} Very\,Young & for\ a \le age \le a_{11} \\ Young & \text{for } a_{11} < age \le a_{12} \\ Middle & \text{for } a_{12} < age \le b_{11} \\ Old & \text{for } b_{11} < age \le b_{12} \\ Very\,Old & \text{for } b_{12} < aged \le c \end{cases} \quad (6.2)$$

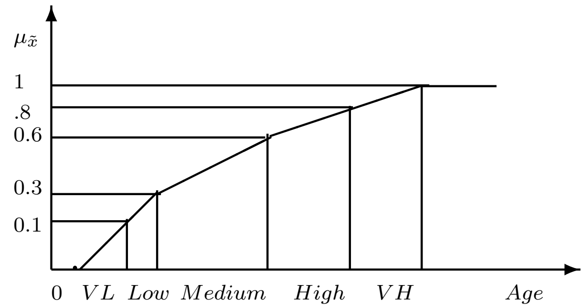


Fig.6.2.2 : Fuzzy extended age distribution of P_c.

Algorithm for Fuzzy extended set based selection

1. Set minimum age, maximum age
2. Evaluate the average fitness combining two objectives
3. **if** average fitness $>$ current fitness
4. age(x$_i$)=avg(age)+$\frac{k*(avgfit-f(X_i))}{(avgfit-minfit)}$
5. **else**
6. age(x$_i$)=$\frac{avg(age)}{2} + \frac{k*(f(X_i)-avgfit)}{(maxfit-avgfit)}$
7. **if** (age(x$_i$)$>$ maximum age)
8. age(x$_i$)= maximum age
9. **else if** (age(x$_i$)$<$ minimum age)
10. age(x$_i$)= minimum age
11. Determine average age
12. Determine common fuzzy age
13. Split Triangular Fuzzy Number in more regions
14. Developed linguistic variables very young, young, middle, old, very old
15. **for** each pair of parents **do**
16. Extended membership values based p$_c$ created
17. **End do**
18. End Algorithm.

The fuzzy age based selection algorithm as similar in the above algorithm.

(iii) Probabilistic Selection:

This part already given in section 4.2.1(c).

(iv) Adaptive Crossover:

At first, select two individuals (parents) from the matting pool, generate the

random number r $\in$[0,1]. If $r < p_c$ then select that population for first parent (say P_{r1}). Similarly choose the other parent (say P_{r2}).
Let these are P_{r1}: $a_1, a_2, ..., a_N, (v_1, v_2, ..., v_p)$
and P_{r2}: $s_1, s_2, ..., s_N, (v_1, v_2, ..., v_p)$

Here ($a_1, a_2, ..., a_N$) and ($s_1, s_2, ..., s_N$) are nodes within (1, 2, 3,..., N), these are numbers of cities. Then we choose a city randomly from 1 to N, say $a_i = s_k$(i=1, 2, ..., N), k=(1, 2,..., N). Then modify the first parents by placing a_i or s_k in the first place of P_{r1} and P_{r2}. Now the modified parents are given by

P_{r1}: $a_i, a_1, a_2, .., a_{i-1}, a_{i+1}, ...a_N, (v_1, v_2, ..., v_p)$
P_{r2}: $s_k, s_1, s_2, ., s_{k-1}, s_{k+1}, ..., s_N, (v_1, v_2, ..., v_p)$

Here the vehicle set are unchanged. To get the first child (Ch_1), placing a_i in the first place of Ch_1, then compare the adaptive weighted (say) $\mathbf{A=m_1*C_{i1}+m_2*T_{i1}}$, ($m_1$, m_2 are weight constants of cost and time respectively, C_{i1} and T_{i1} are the cost and time between the two node a_i to a_1, also $m_1, m_2 < 1$) between the next route a_i to a_1 and a_i to s_1. Minimum adaptive weight route be selected in Ch_1. The procedure is discussed in the section 4.3.1(c)(iii).

(v) Mutation:

(a) Generation Dependent Mutation:

Here model a new form of mutation mechanism where probability of mutation (p_m) is determined as follows

$$P_m=\frac{k}{\sqrt{(Current\ generation\ number)}},\ k\in[0,1].$$

So, here proposed mutation mechanism follows the real world demand and p_m decreases smoothly as generation increases.

(b) Mutation process:

Now for the particular node dependent problem like TSP, to mutate a chromosome $X = (x_1, x_2, ...x_N), (v_1, v_2, ...v_P)$, we find the number of mutated nodes as T= p_m* N, N=total number of nodes in chromosome. If $r < p_m$, r $\in$ rand [0,1], then corresponding chromosome is selected for mutation. Now two kinds of mutation process are presented below:

(i) Random Method: At first, we randomly generate two distinct integer x_i, x_j (say) between [1,N]. Then interchange x_i, x_j to get mutated solution which replaces the parent solution. This process is repeated until T times.

(ii) Fixed Method : In the selected chromosome X=$(x_1, x_2, ...x_N), (v_1, v_2, ...v_P)$, choose a consecutive $\frac{T}{2}$ nodes and interchange them. If T becomes odd, then similarly interchange the places of the solutions up to $\frac{T}{2}$+1 times. This new solution replace the parent solution.

Algorithm for generation dependent random mutation

1.**Start**
2. Set g=current generation number
3. p$_m$=$\frac{k}{sqrt(g)}$, k$\in$[0,1]
4. Determine T= p$_m$*N // total number of mutated node
5. **for**(i=0; i$<$ pop_size; i++)
6. r=rand(0,1)
7. **if**(r$<$ p$_m$)$\{$
8. Select current chromosome
9. a=rand[1,N]
10. b=rand[1,N]
11.   **if** (a==b)
12.   Goto step 9
13.   **for** (j=1; j<=N; j++) // N= total number of nodes
14.   **if** (x[j]==a)
15.   p=j;
16.   **if** (x[j]==b)
17.   q=j;
18.   x[p]=b;// replace a by b.
19.   x[q]=a; // replace b by a.
20.   **end for**
21. Repeat step-8 to 20 up to T times
22 . End if
23. **End for**
24. **End Algorithm**

Algorithm for generation dependent fixed location mutation

1.**Start**

2. Set g=current generation number
3. $p_m = \frac{k}{sqrt(g)}$, k∈[0,1]
4. Determine T= p_m*N // total number of mutated node
5. **for**(i=0; i< pop_size; i++)
6. r=rand(0,1)
7. **if**(r< p_m){
8. Select current chromosome
9. **for** (j=1; j< $\frac{T}{2}$; j++)
10. Exchange (x[j]=x[j+1])
11. **end for**
12. End if
13. **End for**
14. **End Algorithm**

(vi) Algorithm for Fuzzy age based GA:

Input: max_ gen, Population Size (pop_ size), Probability of Mutation (p_m), Problem Data (cost and risk matrices).

Output: Pareto set/font as optimum solutions.

1. **Start**
2. g ← 0 // g: iteration/generation number
3. **Initialize P**(**g**) // randomly generate initial population $P(g)$
4. **Evaluate f**(**P**(**g**)); //Evaluate fitness of each chromosome.
5. **while** (g ≤ max_gen) {
   // Selection Operation
6. **for** every chromosome {
7. Determine the age of each chromosome of P(g)
8. Create common fuzzy age
9. }
   // fuzzy set based selection
10. **for** every chromosome {
11. Three-fold linguistic developed // young, middle, old
12. Membership function used for each pair
13. p_c created for each chromosomes in P(g)

```
                    }
                  // fuzzy extended selection
                    for every chromosome {
                        Five-fold linguistic developed
                        Membership function as Equ. 3.8 used
                        p_c created for each chromosomes in P(g)
                    }
          // Crossover Operation according to subsection-2.3
                  Select the parents for crossover using p_c from matting pool
                    for each pair of parents {
                       Modify the parents;
                       Generate off springs according to subsection-2.3.1
                    }
                // generation dependent mutation
                      Generate p_m according given in section 2.4.1
                      Calculated T=p_m*N, N=Total number of nodes,
                      Select the off springs for mutation based on p_m
                  // random method
                      for selected chromosome {
                      Randomly exchange the nodes up to T times
                      }
                 // fixed location method
                      for selected chromosome {
                      Select the T numbers of nodes
                       Swap the nodes up to T/2, T/2 + 1 times as T even, odd
                      }
                      Store the new off springs into offspring set
           Reproduce a new P(g)
           Evaluate f(P(g));//evaluate the fitness of reproduce chromosome
               Store the local optimum and near optimum solutions
               g ← g+1
          } //endwhile
      Store the global optimum and near optimum results;
  End Algorithm.
```

(vii) Division of $P(T)$ into disjoint subsets having non-dominated solutions:

Now according to Deb et al. [36]), the procedure is given in section 2.1.5 used for create the disjoint subset.

(viii) To determine distance of a solution of subset F from other solutions:

According to Deb et al. [36], some modifications are made to evaluate the distance of Pareto solutions which are given as

Set n=number of solutions in F
For every $x \in F$ do
    $x_{distance} = 0$
End For
For every objective m do
    Sort F, in ascending order of magnitude of m^{th} objective.
    $F[1] = F[n] = M$, where M is a big quantity.
    For i=2 to n-1 do
        $F[i]_{distance} = F[i]_{distance} + (F[i+1].objm - F[i-1].objm)/(f_m^{max} - f_m^{average})$
    End For
End For

In the algorithm $F[i]$ represents i^th solution of F, $F[i].objm$ represent m^{th} objective value of $F[i]$. f_m^{max} and $f_m^{average}$ represents the maximum and average values of m^{th} objective function respectively.

(ix) Complexity Analysis:

MOGAs, that use non-dominated sorting and sharing are mainly criticized for their O(MN^3) complexity, but fast and elitist non-dominated sorting algorithm has O(MN^2) computational complexity where N is the popsize and M is the number of objectives. Here also the proposed iMOGA has the same O(MN^2) computational complexity.

6.2.2 Mathematical Formulation and Its crisp equivalence

Model 6.1A: Multi-Objective TSP with Risk/Discomfort Constraints:

In a classical Multi-Objective TSP (MOTSP), a salesman has to travel N cities at minimum cost and time. In this tour, salesman starts from a city, visit all the cities exactly once and comes back to the starting city using minimum cost and time. Here some risk/discomfort factors in travelling from one city to another are considered. The salesman should choose such a path in which a minimum risks/discomforts are involved i.e. a maximum risk factor for the entire tour is less than the permitted risk value. Let $c(i, j)$, $t(i, j)$ and $r(i, j)$ be the cost, time and risk/discomfort factor for travelling from i-th city to j-th city. Then the problem can be mathematically formulated as (Dantzig et al., [31]):

$$\left.\begin{aligned}
\text{Minimize} \quad Z &= \sum_{i \neq j} c(i, j) x_{ij} \\
\text{Minimize} \quad T &= \sum_{i \neq j} t(i, j) x_{ij} \\
\text{subject to} \quad & \sum_{i=1}^{N} x_{ij} = 1 \ \text{ for } j = 1, 2, ..., N \\
& \sum_{j=1}^{N} x_{ij} = 1 \ \text{ for } i = 1, 2, ..., N \\
& \sum_{i \in S}^{N} \sum_{j \in S}^{N} x_{ij} \leq |S| - 1, \forall S \subset P \\
& \sum_{i=1}^{N} \sum_{j=1}^{N} r(i, j) x_{ij} \leq r_{max} \\
where\ x_{ij} &\in \{0, 1\}, i, j = 1, 2.., N.
\end{aligned}\right\} \tag{6.3}$$

where P={1, 2, 3,.., N} set of nodes, x_{ij} is the decision variable and $x_{ij} = 1$ if the salesman travels from city-i to city-j, otherwise $x_{ij} = 0$ and r_{max} is the maximum permitted risk/discomfort factor that should be maintained for the entire tour to avoid unwanted situation. Then the above CMOTSP reduces to

$$\left.\begin{aligned}
& \text{determine a complete tour} \quad (x_1, x_2, ..., x_N, x_1) \\
\text{to minimize} \quad Z &= \sum_{i=1}^{N-1} c(x_i, x_{i+1}) + c(x_N, x_1) \\
\text{to minimize} \quad T &= \sum_{i=1}^{N-1} t(x_i, x_{i+1}) + t(x_N, x_1) \\
\text{subject to} \quad & \sum_{i=1}^{N-1} r(x_i, x_{i+1}) + r(x_N, x_1) \leq r_{max} \\
& where\ x_i \neq x_j, i, j = 1, 2..., N.
\end{aligned}\right\} \tag{6.4}$$

Model 6.1B: MOSTSP with Risk/Discomfort Constraints (CMOSTSP):

In a MOSTSP, a salesman has to travel N cities by choosing any one of the P types of conveyances available with minimum cost and time. Risk/discomfort factors in travelling from one city to another using different vehicles are different. The salesman should choose such paths and conveyances such that a maximum risk/discomfort level is not exceeded for the entire tour. Let $c(i, j, k)$ and $t(i, j, k)$ are cost and time for travelling from i-th city to j-th city using k-th type conveyance and $r(i, j, k)$ be the risk/discomfort factor in travelling from i-th city to j-th using k-th type conveyances. Then the salesman has to determine a complete tour (x_1, x_2, ...,x_N, x_1) and corresponding conveyance types (v_1, v_2, ..., v_P) to be used for the tour, where $x_i \in \{1, 2, ..N\}$ for $i = 1, 2, ..., N$, $v_i \in \{1, 2, ..P\}$ for $i = 1, 2, ..., N$ and all x_i's are distinct. Then the problem can be mathematically formulated as:

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ using any one available corresponding conveyance in each step from the vehicle types $(v_1, v_2, ..., v_P)$ so as

$$\left.\begin{array}{c}\text{to minimize} \quad Z = \sum_{i=1}^{N-1} c(x_i, x_{i+1}, v_i) + c(x_N, x_1, v_l), \\ \text{to minimize} \quad T = \sum_{i=1}^{N-1} t(x_i, x_{i+1}, v_i) + t(x_N, x_1, v_l), \\ \text{subject to} \quad \sum_{i=1}^{N-1} r(x_i, x_{i+1}, v_i) + r(x_N, x_1, v_l) \leq r_{max}, \\ where \ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\}\end{array}\right\} \tag{6.5}$$

where r_{max} is the maximum risk/discomfort factor that should be maintained by the salesman in the entire tour to avoid unwanted situation.

Model 6.1C: CMOSTSP in Random Environment (RaCMOSTSP):

In the problem in Equ. 6.5, if costs, times and risk/discomfort factors i.e, $\hat{c}(i, j, k)$, $\hat{t}(i, j, k)$ and $\hat{r}(i, j, k)$ respectively are random variables, and maximum risk/discomfort limit r_{max} also is random variables $\hat{r}_{max}$ then the Equ. 6.5 reduces to:

$$\left.\begin{array}{r}\text{minimize} \quad Z = \sum_{i=1}^{N-1} \hat{c}(x_i, x_{i+1}, v_i) + \hat{c}(x_N, x_1, v_l) \\ \text{minimize } T = \sum_{i=1}^{N-1} \hat{t}(x_i, x_{i+1}, v_i) + \hat{t}(x_N, x_1, v_l) \\ \text{subject to} \quad \sum_{i=1}^{N-1} \hat{r}(x_i, x_{i+1}, v_i) + \hat{r}(x_N, x_1, v_l) \leq \hat{r}_{max} \\ where\ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\}\end{array}\right\} \tag{6.6}$$

Now using Chance-constrained programming technique, the above model reduces to:

$$\left.\begin{array}{c}\text{to minimize} \quad Z = \sum_{i=1}^{N-1} \hat{c}(x_i, x_{i+1}, v_i) + \hat{c}(x_N, x_1, v_l) \\ \text{minimize } T = \sum_{i=1}^{N-1} \hat{t}(x_i, x_{i+1}, v_i) + \hat{t}(x_N, x_1, v_l) \\ \text{subject to} \quad P[\sum_{i=1}^{N-1} \hat{r}(x_i, x_{i+1}, v_i) + \hat{r}(x_N, x_1, v_l) \leq \hat{r}_{max}] \geq p_i, \\ where\ x_i \neq x_j, i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\}. \\ \text{Here } p_i\text{s are crisp values giving the levels of probability.}\end{array}\right\} \tag{6.7}$$

Here we consider all random variables as normal variate. Then the objective functions are also normal variate. Thus the problem is finally stated as:

$$\left.\begin{array}{c}\text{Minimize F(X)} \ = k_1 * E[\sum_{i=1}^{N-1} \hat{c}(x_i, x_{i+1}, v_i) + \hat{c}(x_N, x_1, v_l)] \\ +k_2 * \sqrt{(}X^T V X), \\ \text{Minimize T(X)} = k_3 * E[\sum_{i=1}^{N-1} \hat{t}(x_i, x_{i+1}, v_i) + \hat{t}(x_N, x_1, v_l)] \\ +k_4 * \sqrt{(}X^T V X), \\ \text{subject to} \quad \bar{h}_i + s_i \sqrt{(}Var(h_i) \leq 0, \ i = 1, 2, ...n \\ x_j \geq 0, j = 1, 2, ..., n \\ where\ x_i \neq x_j, \ i, j = 1, 2...N, \quad v_i, v_l \in \{1, 2.., or P\}, k_1, k_2, k_3, k_4 \geq 0\end{array}\right\} \tag{6.8}$$

$$\text{Here } \bar{h}_i = E[\sum_{i=1}^{N-1} \hat{r}(x_i, x_{i+1}, v_i) + \hat{r}(x_N, x_1, v_l)] - \bar{r}_{max_i},$$

where (k_1, k_3) and (k_2, k_4) are constants indicating the weights of mean and variance functions respectively, s_i is the tabulated value of the normal distribution.

Model 6.1D: CMOSTSP in Fuzzy Random Environment (FRCMOSTSP):

In the Equ. 6.5, if costs, times and risk/discomfort factors i.e, $\tilde{\hat{c}}(i,j,k)$, $\tilde{\hat{t}}(i,j,k)$ respectively are fuzzy random variables, and $\tilde{\hat{r}}(i,j,k)$ and maximum risk/discomfort limit r_{max} is also a fuzzy random variable $\tilde{\hat{r}}_{max}$ then the Equ. 6.5 reduces to:

$$\left.\begin{array}{l} \text{to minimize } \; Z = \sum_{i=1}^{N-1} \tilde{\hat{c}}(x_i, x_{i+1}, v_i) + \tilde{\hat{c}}(x_N, x_1, v_l) \\ \text{to minimize } \; T = \sum_{i=1}^{N-1} \tilde{\hat{t}}(x_i, x_{i+1}, v_i) + \tilde{\hat{t}}(x_N, x_1, v_l) \\ \text{subject to } \; \sum_{i=1}^{N-1} \tilde{\hat{r}}(x_i, x_{i+1}, v_i) + \tilde{\hat{r}}(x_N, x_1, v_l) \leq \tilde{\hat{r}}_{max} \\ where \; x_i \neq x_j, i, j = 1, 2...N, \;\; v_i, v_l \in \{1, 2.., or P\}. \end{array}\right\} \tag{6.9}$$

Above Equ. 6.9 can be reformulated as given, where the objective function $\sum_{i=1}^{N-1} \tilde{\hat{c}}(x_i, x_{i+1}, v_i) + \tilde{\hat{c}}(x_N, x_1, v_l) \leq F$, $\sum_{i=1}^{N-1} \tilde{\hat{t}}(x_i, x_{i+1}, v_i) + \tilde{\hat{}}(x_N, x_1, v_l) \leq T$, where F and T are given crisps, and equations evaluated using fuzzy random chance constrained programming technique.

to minimize F and T

$$\left.\begin{array}{l} \text{s.t. } \; Ch\{\sum_{i=1}^{N-1} \tilde{\hat{c}}(x_i, x_{i+1}, v_i) + \tilde{\hat{c}}(x_N, x_1, v_l) \leq F\}(\gamma) \geq \delta \\ Ch\{\sum_{i=1}^{N-1} \tilde{\hat{t}}(x_i, x_{i+1}, v_i) + \tilde{\hat{t}}(x_N, x_1, v_l) \leq T\}(\gamma_1) \geq \delta_1 \\ Ch \sum_{i=1}^{N-1} \tilde{\hat{r}}(x_i, x_{i+1}, v_i) + \tilde{\hat{r}}(x_N, x_1, v_l) \leq \tilde{\hat{r}}_{max}\}(\eta) \geq \theta \\ where \; x_i \neq x_j, i, j = 1, 2...N, \;\; v_i, v_l \in \{1, 2.., or P\}. \end{array}\right\} \tag{6.10}$$

Here the parameters $\gamma, \delta, \gamma_1, \delta_1, \theta, \eta$ are predetermined confidence levels in [0,1]. The above Equ. 6.10 is reformulated as

$$\left.\begin{array}{c}\text{minimize } \{\text{F, T}\}\\ \text{s.t } Ch\{\tilde{\hat{\tilde{C}}}x \leq F\}(\gamma) \geq \delta\\ Ch\{\tilde{\hat{\tilde{T}}}_1 x \leq T\}(\gamma_1) \geq \delta_1\\ Ch\{\tilde{\hat{\tilde{R}}}_1 x \leq \tilde{\hat{\tilde{R}}}_{max}\}(\eta) \geq \theta\\ x \in X\end{array}\right\} \quad (6.11)$$

where $\tilde{\hat{\tilde{C}}} = \sum_{i=1}^{N-1} \tilde{\hat{\tilde{c}}}(x_i, x_{i+1}, v_i) + \tilde{\hat{\tilde{c}}}(x_N, x_1, v_l)$, $\tilde{\hat{\tilde{T}}}_1 = \sum_{i=1}^{N-1} \tilde{\hat{\tilde{t}}}(x_i, x_{i+1}, v_i) + \tilde{\hat{\tilde{t}}}(x_N, x_1, v_l)$, $\tilde{\hat{\tilde{R}}}_1 = \sum_{i=1}^{N-1} \tilde{\hat{\tilde{r}}}(x_i, x_{i+1}, v_i) + \tilde{\hat{\tilde{r}}}_1(x_N, x_1, v_l)$, $\tilde{\hat{\tilde{R}}}_{max} = \tilde{\hat{\tilde{r}}}_{max}$, and x as a decision vectors.

It follows from section 3.13.9, the Equ. 6.11 is converted as follows using Probability Possibility measure

$$\left.\begin{array}{c}\text{minimize}\{F, T\}\\ \text{s.t. } Pr\{\omega | Pos\{\tilde{\hat{\tilde{C}}}x \leq F\} \geq \delta\} \geq \gamma\\ Pr\{\omega | Pos\{\tilde{\hat{\tilde{T}}}_1 x \leq T\} \geq \delta_1\} \geq \gamma_1\\ Pr\{\omega | Pos\{\tilde{\hat{\tilde{R}}}_1 x \leq \tilde{\hat{\tilde{R}}}_{max}\} \geq \theta\} \geq \eta\\ x \in X\end{array}\right\} \quad (6.12)$$

and the Probability Necessity measure form is given below

$$\left.\begin{array}{c}\text{minimize}\{F, T\}\\ \text{s.t. } Pr\{\omega | Nes\{\tilde{\hat{\tilde{C}}}x \leq F\} \geq \delta\} \geq \gamma\\ Pr\{\omega | Nes\{\tilde{\hat{\tilde{T}}}_1 x \leq T\} \geq \delta_1\} \geq \gamma_1\\ Pr\{\omega | Nes\{\tilde{\hat{\tilde{R}}}_1 x \leq \tilde{\hat{\tilde{R}}}_{max}\} \geq \theta\} \geq \eta\\ x \in X\end{array}\right\} \quad (6.13)$$

where $\gamma, \delta, \gamma_1, \delta_1, \eta, \theta \in [0, 1]$ are the predetermined confidence levels, Pos$\{.\}$ denotes possibility of the fuzzy events in $\{.\}$, and Pr$\{.\}$ denotes te probability of the random events in $\{.\}$, similarly for Nes$\{.\}$ denotes the necessity of the fuzzy events in $\{.\}$.

To find the crisp values of probability possibility and necessity model according the theorems 3.7, 3.8 and 3.9, the above model Equs.6.12 and 6.13 are converted as follows

$$\left.\begin{array}{c} \text{minimize F} = R^{-1}(\delta)\beta^{CT}x + d^{CT}x + \phi^{-1}(1-\gamma)\sqrt{(x^TV^Cx)} \\ \text{minimize T} = R^{-1}(\delta_1)\beta^{CT_1}x + d^{CT_1}x + \phi^{-1}(1-\gamma_1)\sqrt{(x^TV^{T_1}x)} \\ \text{s.t} \quad R^{-1}(\theta)\beta^{R_{max}} + L^{-1}(\theta)\alpha^{R_1T}x - (d^{R_1T}x - d^b) - \\ \phi^{-1}(\eta)\sqrt{(x^TV^{R_1}x + (\sigma^{R_{max}})^2)} \geq 0 \end{array}\right\} \quad (6.14)$$

and

$$\left.\begin{array}{c} \text{minimize F} = d^{CT}x - L^{-1}(1-\delta)\alpha^{CT}x \\ +\phi^{-1}(1-\gamma)\sqrt{(x^TV^Cx)} \\ \text{minimize T} = d^{CT_1}x - L^{-1}(1-\delta_1)\alpha^{CT_1}x \\ +\phi^{-1}(1-\gamma_1)\sqrt{(x^TV^{T_1}x)} \\ \text{s.t} \quad \phi^{-1}(1-\eta)\sqrt{(x^TV^{R_1}x + (\sigma^{R_{max}})^2)} - L^{-1}(1-\theta)\alpha^{R_{max}} \\ -R^{-1}(\theta)\beta^{R_1T}x + (d^{R_{max}} - d^{R_1T}x) \geq 0 \end{array}\right\} \quad (6.15)$$

Model 6.1E: CMOSTSP in Random-Fuzzy Environment (RFCMOSTSP):

In Equ 6.5, if cost, times and risk/discomfort factors i.e, $\hat{\tilde{c}}(i,j,k)$, $\hat{\tilde{t}}(i,j,k)$ and $\hat{\tilde{r}}(i,j,k)$ respectively are random-fuzzy variables, and maximum risk/discomfort limit r_{max} is also random-fuzzy variables $\hat{\tilde{r}}_{max}$, then the Equ. 6.5 reduces to:

$$\left.\begin{array}{c} \text{minimize} \quad Z = \sum_{i=1}^{N-1} \hat{\tilde{c}}(x_i, x_{i+1}, v_i) + \hat{\tilde{c}}(x_N, x_1, v_l) \\ \text{minimize} \quad T = \sum_{i=1}^{N-1} \hat{\tilde{t}}(x_i, x_{i+1}, v_i) + \hat{\tilde{t}}(x_N, x_1, v_l) \\ \text{subject to} \quad \sum_{i=1}^{N-1} \hat{\tilde{r}}(x_i, x_{i+1}, v_i) + \hat{\tilde{r}}(x_N, x_1, v_l) \leq \hat{\tilde{r}}_{max} \\ where \ x_i \neq x_j, i,j = 1,2...N, \quad v_i, v_l \in \{1,2..,or P\}. \end{array}\right\} \quad (6.16)$$

Above Equ. 6.16 can be reformulated as given below where the objective functions are

$\sum_{i=1}^{N-1} \hat{\tilde{c}}(x_i, x_{i+1}, v_i) + \hat{\tilde{c}}(x_N, x_1, v_l) \leq F_1$, F_1 is crisp.

$\sum_{i=1}^{N-1} \hat{\tilde{t}}(x_i, x_{i+1}, v_i) + \hat{\tilde{t}}(x_N, x_1, v_l) \leq T_1$, T_1 is crisp.

Now the Equ. 6.16 using section 3.13.15 defined as possibilistic and necessity chance constraint forms is given below

$$\left.\begin{array}{c} \text{minimize} \quad F_1 \ and \ T_1 \\ Pos\{Prob\{\sum_{i=1}^{N-1}\hat{\tilde{c}}(x_i,x_{i+1},v_i)+\hat{\tilde{c}}(x_N,x_1,v_l)\le F_1\}\ge\hat{\theta}_1^{obj}\}\ge\hat{h}_1^{obj} \\ Pos\{Prob\{\sum_{i=1}^{N-1}\hat{\tilde{t}}(x_i,x_{i+1},v_i)+\hat{\tilde{t}}(x_N,x_1,v_l)\le T_1\}\ge\hat{\theta}_2^{obj}\}\ge\hat{h}_2^{obj} \\ Nes\{Prob\{\sum_{i=1}^{N-1}\hat{\tilde{c}}(x_i,x_{i+1},v_i)+\hat{\tilde{c}}(x_N,x_1,v_l)\le F_1\}\ge\hat{\theta}_1^{obj}\}\ge\hat{h}_1^{obj} \\ Nes\{Prob\{\sum_{i=1}^{N-1}\hat{\tilde{t}}(x_i,x_{i+1},v_i)+\hat{\tilde{t}}(x_N,x_1,v_l)\le T_1\}\ge\hat{\theta}_2^{obj}\}\ge\hat{h}_2^{obj} \\ \text{s.t} \quad Pos\{Prob\{\sum_{i=1}^{N-1}\hat{\tilde{r}}(x_i,x_{i+1},v_i)+\hat{\tilde{r}}(x_N,x_1,v_l)\le\hat{\tilde{r}}_{max}\}\ge\hat{\theta}^{cst}\}\ge\hat{h}^{cst} \\ Nes\{Prob\{\sum_{i=1}^{N-1}\hat{\tilde{r}}(x_i,x_{i+1},v_i)+\hat{\tilde{r}}(x_N,x_1,v_l)\le\hat{\tilde{r}}_{max}\}\ge\hat{\theta}^{cst}\}\ge\hat{h}^{cst} \\ where \ x_i\ne x_j, i,j=1,2...N, \quad v_i,v_l\in\{1,2..,orP\}. \end{array}\right\} \quad (6.17)$$

The above Equ. 6.17 is equivalently written into

$$\left.\begin{array}{c} Pos\{Prob\{\hat{\tilde{C}}x\le F_1\}\ge\hat{\theta}_1^{obj}\}\ge\hat{h}_1^{obj} \\ Pos\{Prob\{\hat{\tilde{T}}x\le T_1\}\ge\hat{\theta}_2^{obj}\}\ge\hat{h}_2^{obj} \\ Nes\{Prob\{\hat{\tilde{C}}x\le F_1\}\ge\hat{\theta}_1^{obj}\}\ge\hat{h}_1^{obj} \\ Nes\{Prob\{\hat{\tilde{T}}x\le T_1\}\ge\hat{\theta}_2^{obj}\}\ge\hat{h}_2^{obj} \\ \text{subject to} \quad Pos\{Prob\{\hat{\tilde{R}}x\le\hat{\tilde{r}}_{max}\}\ge\hat{\theta}^{cst}\}\ge\hat{h}^{cst} \\ Nes\{Prob\{\hat{\tilde{R}}x\le\hat{\tilde{r}}_{max}\}\ge\hat{\theta}^{cst}\}\ge\hat{h}^{cst} \end{array}\right\} \quad (6.18)$$

where $\hat{\tilde{C}}=\sum_{i=1}^{N-1}\hat{\tilde{c}}_1(x_i,x_{i+1},v_i)+\hat{\tilde{c}}_1(x_N,x_1,v_l)$,
$\hat{\tilde{T}}=\sum_{i=1}^{N-1}\hat{\tilde{t}}_1(x_i,x_{i+1},v_i)+\hat{\tilde{t}}_1(x_N,x_1,v_l)$, $\hat{\tilde{R}}=\sum_{i=1}^{N-1}\hat{\tilde{r}}(x_i,x_{i+1},v_i)+\hat{\tilde{r}}(x_N,x_1,v_l)$.

The above Equ. 6.18 using section 3.13.15 is transformed to

$$\left.\begin{array}{c}\sum_{i=1}^{N}\{m_i^c - L*(\hat{h_{1i}}^{obj})\alpha_{1i}^c\}x_i + \Phi^{-1}(\hat{\theta}_1^{obj})\sqrt(x^tV^cx) \le F_1 \\ \sum_{i=1}^{N}\{m_i^t - L*(\hat{h_{2i}}^{obj})\alpha_{2i}^t\}x_i + \Phi^{-1}(\hat{\theta}_2^{obj})\sqrt(x^tV^tx) \le T_1 \\ \text{s.t.}\quad \sum_{i=1}^{N}\{m_i^R - L*(\hat{h_i}^{cst})\alpha_i^R\}x_i + \Phi^{-1}(\hat{\theta}^{cst})\sqrt(x^tV^Rx + (\sigma_i^r)^2) \\ \le m_i^r + L*(\hat{h}_i^{cst})\beta_i^r \\ \text{(using Possibility approach)}\end{array}\right\} \quad (6.19)$$

$$\left.\begin{array}{c}\sum_{i=1}^{N}\{m_i^c + L*(1-\hat{h_{1i}}^{obj})\beta_i^c\}x_i + \Phi^{-1}(\hat{\theta}_1^{obj})\sqrt(x^tV^cx) \le F_1 \\ \sum_{i=1}^{N}\{m_i^t + L*(1-\hat{h_{2i}}^{obj})\beta_i^t\}x_i + \Phi^{-1}(\hat{\theta}_2^{obj})\sqrt(x^tV^tx) \le T_1 \\ \text{s.t.}\quad \sum_{i=1}^{N}\{m_i^R + L*(1-\hat{h_i}^{cst})\beta_i^R\}x_i + \Phi^{-1}(\hat{\theta}^{cst})\sqrt(x^tV^Rx + (\sigma_i^r)^2) \\ \le m_i^r - L*(1-\hat{h}_i^{cst})\alpha_i^r \\ \text{(using Necessity approach)}\end{array}\right\} \quad (6.20)$$

Finally the above random-fuzzy models transformed into the crisp models as given below:

$$\left.\begin{array}{c}\text{Minimize } \mathrm{F}_1 = \sum_{i=1}^{N}\{m_i^c - L*(\hat{h_{1i}}^{obj})\alpha_i^c\}x_i + \Phi^{-1}(\hat{\theta}_1^{obj})\sqrt(x^tV^cx) \\ \text{Minimize } \mathrm{T}_1 = \sum_{i=1}^{N}\{m_i^t - L*(\hat{h_{2i}}^{obj})\alpha_{2i}^t\}x_i + \Phi^{-1}(\hat{\theta}_2^{obj})\sqrt(x^tV^tx) \\ \text{s.t.}\quad \sum_{i=1}^{N}\{m_i^R - L*(\hat{h_i}^{cst})\alpha_i^R\}x_i + \Phi^{-1}(\hat{\theta}^{cst})\sqrt(x^tV^Rx + (\sigma_i^r)^2) \\ \le m_i^r + L*(\hat{h}_i^{cst})\beta_i^r\end{array}\right\} \quad (6.21)$$

and

$$\left.\begin{array}{c}\text{Minimize } \mathrm{F}_1 = \sum_{i=1}^{N}\{m_i^c + L*(1-\hat{h_{1i}}^{obj})\beta_i^c\}x_i \\ +\Phi^{-1}(\hat{\theta}_1^{obj})\sqrt{(}x^tV^cx) \\ \text{Minimize } \mathrm{T}_1 = \sum_{i=1}^{N}\{m_i^t + L*(1-\hat{h_{2i}}^{obj})\beta_i^t\}x_i \\ +\Phi^{-1}(\hat{\theta}_2^{obj})\sqrt{(}x^tV^tx) \\ \text{subject to} \\ \sum_{i=1}^{N}\{m_i^R + L*(1-\hat{h_i}^{cst})\beta_i^R\}x_i + \Phi^{-1}(\hat{\theta}^{cst})\sqrt{(}x^tV^Rx + (\sigma_i^r)^2) \\ \leq m_i^r - L*(1-\hat{h}_i^{cst})\alpha_i^r \end{array}\right\} \quad (6.22)$$

where $\alpha_i^c, \alpha_i^R, \beta_i^c, \beta_i^R$ and β_i^r are predetermined given values. Again $\hat{h}_1^{obj}, \hat{h}_2^{obj}$, $\hat{h}_1^{cst}\hat{h}_2^{cst}$ are permissible possibility or necessity levels for the objectives and risk/discomfort constraints. Also $\hat{\theta}_1^{obj}, \hat{\theta}_2^{cst}$ are permissible probability levels for the objectives and constraints.

Model 6.1F: CMOSTSP in Bi-random Environment (BRCMOSTSP):

In Equ. 6.5, if the costs, times and risk/discomfort factors i.e, $\tilde{\tilde{c}}(i,j,k)$, $\tilde{\tilde{t}}(i,j,k)$ and $\tilde{\tilde{r}}(i,j,k)$ respectively are bi-random variables and maximum risk/discomfort limit r_{max} is also bi-random variables $\tilde{\tilde{r}}_{max}$, then the Equ. 6.5 reduces to:

$$\left.\begin{array}{rl}\text{minimize} & Z = \sum_{i=1}^{N-1}\tilde{\tilde{c}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{c}}(x_N, x_1, v_l) \\ \text{minimize} & T = \sum_{i=1}^{N-1}\tilde{\tilde{t}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{t}}(x_N, x_1, v_l) \\ \text{subject to} & \sum_{i=1}^{N-1}\tilde{\tilde{r}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{r}}(x_N, x_1, v_l) \leq \tilde{\tilde{r}}_{max} \\ & where\ x_i \neq x_j, i,j = 1,2...N,\ \ v_i, v_l \in \{1,2..,orP\}. \end{array}\right\} \quad (6.23)$$

Above Equ.6.23 can be reformulated with the objective functions as $\sum_{i=1}^{N-1}\tilde{\tilde{c}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{c}}(x_N, x_1, v_l) \leq F$, $\sum_{i=1}^{N-1}\tilde{\tilde{t}}(x_i, x_{i+1}, v_i) + \tilde{\tilde{t}}(x_N, x_1, v_l) \leq T$, where F and T are crisp, and equations evaluated using equilibrium chance constrained programming technique.

$$\left.\begin{array}{c}\text{to minimize } \{\text{F,T}\}\\ \text{subject to } Ch^e\{\sum_{i=1}^{N-1}\tilde{\tilde{c}}(x_i,x_{i+1},v_i)+\tilde{\tilde{c}}(x_N,x_1,v_l)\le F\}\ge\alpha_5\\ \text{subject to } Ch^e\{\sum_{i=1}^{N-1}\tilde{\tilde{t}}(x_i,x_{i+1},v_i)+\tilde{\tilde{t}}(x_N,x_1,v_l)\le T\}\ge\alpha_6\\ Ch^e\sum_{i=1}^{N-1}\tilde{\tilde{r}}(x_i,x_{i+1},v_i)+\tilde{\tilde{r}}(x_N,x_1,v_l)\le\tilde{\tilde{r}}_{max}\}\ge\beta_4\\ where\ x_i\ne x_j, i,j=1,2...N,\ \ v_i,v_l\in\{1,2..,or P\}.\end{array}\right\}\quad(6.24)$$

Here α, β are predetermined confidence levels.
Now the above Equ. 6.24 is reformulated as

$$\left.\begin{array}{c}\text{minimize } \{\text{F, T}\}\\ \text{s.t } Ch^e\{\tilde{\tilde{C}}x\le F\}\ge\alpha_5\\ Ch^e\{\tilde{\tilde{T}}x\le T\}\ge\alpha_6\\ Ch^e\{\tilde{\tilde{R}}x\le\tilde{\tilde{R}}_{max}\}\ge\beta_4\\ x\in D\end{array}\right\}\quad(6.25)$$

where $\tilde{\tilde{C}}=\sum_{i=1}^{N-1}\tilde{\tilde{c}}(x_i,x_{i+1},v_i)+\tilde{\tilde{c}}(x_N,x_1,v_l)$, $\tilde{\tilde{T}}=\sum_{i=1}^{N-1}\tilde{\tilde{t}}(x_i,x_{i+1},v_i)+\tilde{\tilde{t}}(x_N,x_1,v_l)$, $\tilde{\tilde{R}}=\sum_{i=1}^{N-1}\tilde{\tilde{r}}(x_i,x_{i+1},v_i)+\tilde{\tilde{r}}_1(x_N,x_1,v_l)$, $\tilde{\tilde{R}}_{max}=\tilde{\tilde{r}}_{max}$, and D is a fixed set that usually determined by a finite of inequalities involving functions of x.
It follows from Theorem 3.13.5 and 3.13.6 the Equ. 6.25 can be written as

$$\left.\begin{array}{c}\text{minimize } \{\text{F, T}\}\\ \text{subject to } Pr\{\omega\in\Omega|Pr\{\tilde{\tilde{C}}(\omega)x\le F\}\ge\alpha_5\}\ge\alpha_5\\ Pr\{\omega\in\Omega|Pr\{\tilde{\tilde{T}}(\omega)x\le T\}\ge\alpha_6\}\ge\alpha_6\\ Pr\{\omega\in\Omega|Pr\{\tilde{\tilde{R}}(\omega)x\le\tilde{\tilde{R}}_{max}\}\ge\beta_4\}\ge\beta_4\\ x\in D\end{array}\right\}\quad(6.26)$$

Finally the above problem using Lemmas in section 3.13.5 and 3.13.6 reduces to:

Determine a complete tour $(x_1, x_2, ..., x_N, x_1)$ and using any one available corresponding conveyance in each step from the vehicle types $(v_1, v_2, ..., v_P)$

$$\left.\begin{array}{c} \text{minimize F} = \mu^c x + \Phi^{-1}(\alpha_5)\sqrt(x^T V^c x) + \Phi^{-1}(\alpha_5)\sqrt(x^T V^{nc} x) \\ \text{minimize T} = \mu^t x + \Phi^{-1}(\alpha_6)\sqrt(x^T V^t x) + \Phi^{-1}(\alpha_6)\sqrt(x^T V^{nt} x) \\ \text{s.t } \mu^R x + \Phi^{-1}(\beta_4)\sqrt(x^T V^R x + (\sigma^{R_{max}})^2) + \Phi^{-1}(\beta_4)\sqrt(x^T V^{nR} x \\ +(\sigma^{R_{nmax}})^2) \leq \mu^{R_{max}}, \\ x \in D. \end{array}\right\} \quad (6.27)$$

Here α, β are given values. Again $\sigma^{R_{max}}$, $\sigma^{R_{nmax}}$, V^R, V^{nR}, V^c, V^{nc} are standard deviation and variances of maximum of risk/discomfort factors and costs in two fold randomness. Also Φ is the standard normal variate distributions.

Solution Procedures:

The deterministic forms of the uncertain CMOSTSPs given by Equ.6.5 for crisp CMOSTSP, Equ. 6.8 for RCMOSTSP in random environment, Equ. 6.14 and Equ. 6.15 for FRCMOSTSP in fuzzy random parameters, Equ. 6.21 and Equ. 6.22 for RFCMOSTSP in random fuzzy and Equ. 6.27 for BRCMOSTSP in birandom environment are solved by the proposed iMOGA, developed for this purpose in section 6.2.1.

6.2.3 Numerical Experiments

(i) Testing for iMOGA:

To judge the effectiveness and feasibility of the developed algorithm iMOGA, we have applied it on the standard two TSP problems from TSPLIB [162] with the combination of same size test problems. Table 6.3 gives the results of said multi-objective by iMOGA and the standard MOGA along with the comparison in terms of total cost and iterations and CPU time in minutes. Here classical MOGA is the combinations of RW-selection, cyclic crossover and random mutation, where as our proposed iMOGA is the combination fuzzy extended age based selection (FEA), adaptive crossover and generation dependent (GD) mutations.

(ii) Performance matrices:

To have a fair comparison, the Coverage (C) (Zitzler et al., [189]) and Inverted Generational Distance (IGD) (Zhang et al., [185]) metrics are used to access the

Table 6.3: Test combining Standard TSPLIB Problems by iMOGA

Instances	Single	Multi	iMOGA			MOGA		
			Cost	Iteration	Time	Cost	Iteration	Time
bays29 bayg29	2020 1610	-	4268	**132**	.14	4786	457	4.53
eil76 pr76	538 108159	-	111953	**216**	2.25	118447	874	7.21
kroA100 kroB100	21282 22141	49639 (Samanlioglu, [148])	49428	**234**	3.02	54658	679	6.45
kroA100 kroC100	21282 20749	50245 (Samanlioglu, [148])	49810	**276**	2.57	52754	734	6.57
kroB100 kroC100	22141 20749	-	48564	**342**	3.43	71368	751	7.23
kroB100 kroD100	22141 21294	-	52645	**423**	4.12	92743	932	9.54
kroD100 kroC100	21294 20749	-	49941	**678**	4.13	67894	876	9.02
kroA100 kroD100	20182 21294		50623	**564**	5.27	82347	829	9.39

performance of the two algorithms. The Coverage metric is used to compare the achieved non-dominated solutions.

$$C(A_1, A_2)= \frac{|\{\pi|\pi\in A_2, \exists\phi\in A_1: F(\phi)\prec F(\pi)\}|}{|A_2|}$$

where A_1, A_2 are the obtained non-dominated sets by two algorithms, $F(\phi) \prec F(\pi)$ denotes $F(\phi)$ dominates $F(\pi)$. $C(A_1, A_2)$ is not necessarily equal to 1-$C(A_2, A_1)$. If $C(A_1, A_2)$ is large and $C(A_2, A_1)$ is small, then A_1 is better than A_2 in a sense.

Lat A^* be a set of uniformly distributed Pareto optimal points in the PF. Let A be an approximation to the PF. The IGD metric is defined as follows,

$$IGD(A^*, A)=\frac{\sum d(v,A)}{|A^*|}$$

where d(v, A) is a minimum distance between v and any point in A, and $|A^*|$ is the cardinality of A^*. The IGD metric can measure both convergence and diversity. To have a low IGD value, A must be close to the PF and cannot miss any part of the whole PF. Here we combine the results obtained by all runs of all algorithms and find out the non-dominated solutions from the combination as the reference A^*.

Table 6.4 presents the statistical results of the Coverage and IGD metrics. It shows that iMOGA performs better than MOGA. The IGD represents both the diversity and convergence qualities of the final approximation. It can be seen that for all problems, the approximation obtained by iMOGA are better than other two algorithms.

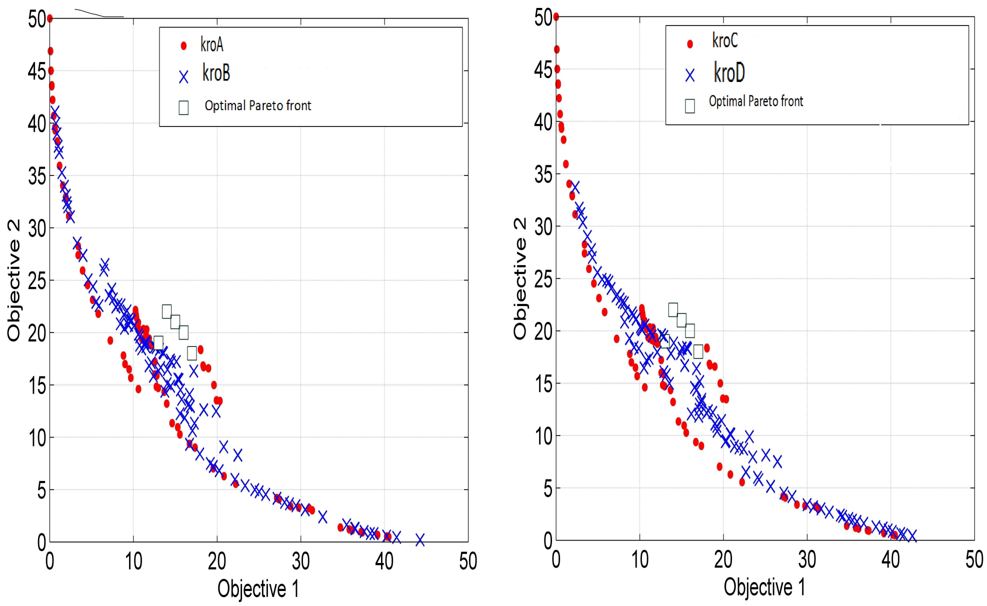


Figure 6.1: Pareto front kroAB and kroCD

Table 6.4: Results (Mean, SD) of iMOGA(A), MOEA/D-ACO(D), MOGA(S)

Instances	Coverage				IGD		
	C(A, D)	C(D, A)	C(A, S)	C(S, A)	A	D	S
kroAB100	0.625, 0.121	0.098, 0.068	0.997, 0.008	0.002, 0.002	1217.8, 384.32	2115.45, 237.41	12875.23, 487.16
kroAC100	0.567, 0.116	0.099, 0.074	0.967, 0.017	0.001, 0.018	1134.4, 367.45	1876.25, 272.59	13246.54, 376.82
kroAD100	0.653, 0.139	0.087, 0.051	0.974, 0.021	0.003, 0.005	1356.6, 246.56	1508.54, 229.13	12387.47, 512.23
kroBC100	0.598, 0.113	0.089, 0.087	0.982, 0.010	0.002, 0.002	1754.5, 364.75	2052.63, 373.52	14547.81, 631.27
kroBD100	0.703, 0.139	0.081, 0.057	0.985, 0.007	0.011, 0.003	1678.9, 302.56	1935.61, 307.18	14123.52, 565.87
kroCD100	0.634, 0.126	0.078, 0.072	0.975, 0.009	0.004, 0.001	1734.7, 267.83	2245.73, 337.82	15025.17, 579.63

(iii) Comparison iMOGA with other algorithms:

According to Lust et al., [107] [108], we compare iMOGA with two states of art algorithms as MOEA/D-ACO and 2PPLS. The quality indicators are hyper-volume **H** (to be maximized), the R measure (normalized between 0 and 1, to be maximized), the average distance D_1 and maximal distance D_2 (to be minimized). The results are considered 40 runs of each algorithm.

(iv) Different forms of iMOGA:

Moreover, for a particular test problem bayg29 and bays29, both standard MOGA and proposed iMOGA are used with different P_c's, P_m's and proposed P_s's. The obtained Pareto optimal solutions are presented in Tables 6.6 and 6.7.

Model 6.1A: Results of CMOTSP and CMOSTSP with Risk/Discomfort Constraint in Crisp Environment:

Table 6.5: Comparison with state-of-art-algorithms

Instances	Algorithm	H(10^8)	I_ϵ	R	D_1	D_2	Time(S)
kroAB100	iMOGA	283.56	1.005718	0.913348	0.572	3.487	127.68
	2PPLS	281.32	1.024810	0.883471	0.688	3.357	119.72
	MOEA-D/ACO	254.76	1.030837	0.901294	0.732	5.546	480.96
kroAC100	iMOGA	286.21	1.003182	0.914578	0.543	3.273	132.65
	2PPLS	282.11	1.014218	0.913760	0.677	3.758	122.76
	MOEA-D/ACO	276.31	1.025123	0.912365	0.665	11.816	528.73
kroAD100	iMOGA	281.45	1.004319	0.912486	0.503	4.891	117.39
	2PPLS	280.27	1.015827	0.906591	0.631	8.349	134.75
	MOEA-D/ACO	280.61	1.019564	0.902187	0.652	12.864	620.81
kroBC100	iMOGA	285.28	1.005507	0.917582	0.703	4.549	137.58
	2PPLS	284.81	1.005584	0.915273	0.727	8.756	141.82
	MOEA-D/ACO	281.37	1.034253	0.916479	0.726	10.278	532.78
kroBD100	iMOGA	283.73	1.032613	0.915423	0.563	4.923	141.82
	2PPLS	283.42	1.006489	0.915376	0.643	7.870	141.77
	MOEA-D/ACO	282.78	1.030743	0.915338	0.759	19.569	567.20
kroCD100	iMOGA	288.91	1.005075	0.914482	0.634	4.437	139.87
	2PPLS	286.01	1.023783	0.913276	0.689	10.392	147.32
	MOEA-D/ACO	284.67	1.025034	0.912787	0.705	21.548	589.13

Table 6.6: Comparison of iMOGA and MOGA

Algorithm	Selection	Crossover	Generation	p_c	p_m	p_s	Result
MOGA	Roulette Wheel	Cyclic	457	0.31	0.3	-	[2342, 1876]
MOGA	Probabilistic	Cyclic	432	0.31	0.3	-	
iMOGA	Probabilistic	Adaptive	256	0.4	0.3	-	
iMOGA	Probabilistic	Adaptive	276	0.44	0.3	-	
iMOGA	Probabilistic	Adaptive	163	-	0.3	0.3	
iMOGA	Age based	Adaptive	182	-	0.3	-	
iMOGA	Extended Age based	Adaptive	173	-	0.3	-	
iMOGA	Extended Age based	Adaptive	158	-	GD		
iMOGA	Extended Age based	Adaptive	146	-	GD	-	
iMOGA	Extended Age based	Adaptive	132	-	GD & Rand	-	

Table 6.7: Comparison of iMOGA for bayg29 and bays29

Algorithm	Selection	Crossover	Mutation	Generation	P_m	Result
iMOGA	**Fuzzy Age Based**	**Adaptive**	Simple	737	0.4	[2342, 1876]
				598	0.3	
				634	0.2	
			Random	356	0.4	
				265	0.3	
				273	0.2	
			Fixed	166	0.4	
				161	0.3	
				155	0.2	
			GD	**149**	**-**	
	Fuzzy Extended Age Based	**Adaptive**	Simple	664	0.4	
				564	0.3	
				605	0.2	
			Random	234	0.4	
				221	0.3	
				216	0.2	
			Fixed	164	0.4	
				158	0.3	
				150	0.2	
			GD	**132**	**-**	

Here we consider a deterministic CMOSTSP given by Equ. 6.7, whose costs, times and risk/discomfort matrices are given by Table 6.8. The problem is solved by iMOGA and the results are presented in Tables 6.9 and 6.10. Here, for CMOSTSP, we consider three types of conveyances. With the same data for the 1st conveyance, we solve the CMOTSP (with single conveyance) and the results are presented in Table 6.9.

For Table 6.9, we took maximum generation=1000 and max-pop size =100 and for Table 6.10, maximum generation=2000, and maximum popsize=150.

Table 6.8: Input Data: Crisp CMOSTSP (Model 6.1A)

	Crisp **Cost** Matrix(10×10) With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	35,36,27	18,39,30	20,33,34	30,21,62	6,23,8	15,36,47	27,38,19	40,31,42	20,31,42
2	35,26,17	∞	40,21,32	18,29,10	35,26,37	40,31,22	40,31,59	33,42,59	18,37,20	24,16,18
3	38,30,29	17,58,34	∞	12,25,14	42,25,46	35,36,34	19,11,8	32,33,25	30,19,41	30,22,33
4	28,20,11	10,22,14	17,8,29	∞	30,19,24	25,16,27	21,31,33	35,36,17	12,23,34	27,48,39
5	17,15,9	42,23,34	35,36,37	20,31,43	∞	30,21,42	45,16,27	30,31,13	19,10,8	28,26,7
6	15,6,7	30,21,29	5,26,28	8,9,12	28,29,40	∞	33,42,24	40,31,22	32,23,35	30,41,32
7	38,39,30	25,54,26	30,38,26	22,43,24	37,58,39	40,21,45	∞	10,41,13	32,33,35	20,15,26
8	40,41,23	25,6,17	32,53,45	40,21,42	35,36,47	25,16,5	40,22,43	∞	22,53,24	37,37,39
9	40,11,33	40,39,36	3,36,37	25,34,29	20,32,21	22,33,25	7,38,39	32,33,14	∞	28,19,26
10	18,27,29	30,21,32	28,19,30	20,31,22	11,33,22	32,12,34	37,28,39	40,41,33	30,51,33	∞

i/j	1	2	3	4	5	6	7	8	9	10
1	∞	15,16,17	28,19,20	30,13,14	20,31,12	62,13,68	25,16,27	17,28,39	30,21,22	30,21,22
2	15,16,27	∞	30,31,22	38,19,40	15,16,17	30,21,32	30,21,9	13,22,9	28,17,10	14,36,28
3	30,21,32	17,58,34	∞	12,25,14	42,25,46	35,36,34	19,11,8	32,33,25	30,19,41	30,22,33
4	28,20,11	10,22,14	17,8,29	∞	30,19,24	25,16,27	21,31,33	35,36,17	12,23,34	27,48,39
	Crisp **Time** Matrix(10 $\times$10) With Three Conveyances									
5	17,15,9	42,23,34	35,36,37	20,31,43	∞	30,21,42	45,16,27	30,31,13	19,10,8	28,26,7
6	25,26,37	20,31,19	55,16,18	61,58,55	18,19,20	∞	13,22,14	30,21,32	22,33,15	20,11,12
7	27,8,14	25,12,36	20,18,16	20,31,12	17,8,19	20,21,25	∞	30,21,33	22,13,15	30,25,16
8	38,19,40	15,16,17	28,19,20	30,13,14	20,31,12	62,13,68	25,16,27	∞	17,28,39	30,21,22
9	40,11,33	40,39,36	3,36,37	25,34,29	20,32,21	22,33,25	7,38,39	32,33,14	∞	28,19,26
10	28,17,19	20,31,12	18,39,20	30,11,18	31,33,22	32,12,34	37,28,39	40,41,33	30,51,33	∞
	Crisp **Risks/Discomforts** Matrix(10$\times$10) With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	.69,.68,.75	.84,.63,.7	.82,.7,.71	.72,.8,.42	.96,.79,.93	.87,.66,.55	.74,.42,.81	.41,.7,.59	.81,.7,.59
2	.67,.76,.84	∞	.61,.8,.7	.83,.73,.92	.67,.76,.65	.41,.71,.79	.41,.71,.43	.69,.6,.42	.83,.64,.81	.77,.85,.3
3	.63,.71,.73	.83,.44,.67	∞	.89,.76,.86	.59,.76,.55	.66,.65,.67	.83,.91,.94	.69,.68,.76	.71,.82,.6	.71,.79,.68
4	.73,.81,.9	.9,.78,.86	.84,.93,.72	∞	.71,.82,.77	.77,.86,.75	.81,.71,.69	.66,.65,.84	.89,.79,.77	.74,.53,.43
5	.84,.86,.92	.59,.78,.67	.66,.65,.64	.82,.71,.59	∞	.71,.81,.59	.57,.85,.74	.71,.7,.88	.82,.91,.93	.74,.75,.93
6	.85,.84,.93	.7,.8,.71	.95,.74,.72	.92,.91,.89	.73,.72,.61	∞	.69,.59,.77	.61,.71,.79	.69,.78,.66	.71,.6,.69
7	.63,.62,.71	.77,.47,.76	.71,.63,.76	.79,.59,.77	.66,.43,.62	.6,.79,.55	∞	.9,.6,.87	.69,.68,.66	.81,.87,.76
8	.61,.6,.78	.76,.95,.84	.69,.47,.56	.61,.81,.6	.67,.66,.55	.6,.85,.95	.61,.8,.59	∞	.79,.48,.77	.64,.64,.62
9	.61,.91,.71	.61,.62,.65	.97,.65,.64	.76,.77,.72	.81,.69,.73	.79,.68,.76	.94,.66,.63	.69,.68,.87	∞	.73,.82,.75
10	.83,.74,.72	.71,.8,.69	.73,.83,.72	.8,.69,.78	.89,.67,.78	.7,.9,.71	.64,.74,.22	.61,.59,.68	.71,.5,.67	∞

Table 6.9: Results of CMOTSP in Crisp (Model 6.1A)

Algorithm	Path	Value	R_{max}
	8-2-10-5-9-6-1-4-3-7	[124,147]	Without R_{max}
	8-2-10-5-9-6-1-4-3-7	[124, 147]	8.64
iMOGA	5-9-6-4-3-7-10-8-2-1	[130,126]	8.64
	6-8-2-10-4-3-7-9-1-5	[139, 110]	8.64
	4-8-2-10-5-9-6-1-3-7	[140, 104]	8.64
iMOGA	1-7-2-5-9-6-10-4-3-2	[167, 106]	8.5
MOGA	10-8-2-5-9-6-1-4-3-7	[207, 118]	8.5
iMOGA	8-5-9-6-1-4-3-7-2-10	[228, 109]	8.00
MOGA	1-2-5-10-4-3-7-9-6-8	[294, 132]	8.00
iMOGA	7-2-6-9-1-4-8-5-10-3	[242, 104]	8.00

Table 6.10: Results of CMOSTSP in Crisp (Model 6.1B)

Algorithm	Path(Vehicle)	Cost	Risk achieved	R_{max}
	1(1)-10(1)-5(2)-4(1)-2(1)-9(1)-3(2)-7(1)-8(3)-6(2)	[107, 142]	8.71	
	9(1)-7(1)-8(1)-6(2)-1(1)-3(1)-4(1)-2(1)-10(1)-5(3)	[131, 138]	8.50	8.75
	8(3)-2(2)-10(1)-9(1)-6(2)-4(2)-3(3)-7(2)-5(2)-1(3)	[141, 128]	8.50	
iMOGA	7(1)-8(1)-6(2)-1(1)-10(3)-5(2)-4(3)-2(3)-9(1)-3(2)	[144, 123]	8.19	8.75
MOGA	2(2)-9(1)-3(3)-7(3)-8(1)-6(2)-1(3)-10(2)-5(2)-4(1)	[190, 108]	8.73	8.75
	5(2)-4(3)-2(1)-9(3)-8(1)-6(3)-1(1)-3(2)-7(1)-10(1)	[151, 102]	7.92	
iMOGA	2(2)-10(3)-9(1)-3(1)-7(1)-8(1)-6(2)-1(1)-5(2)-4(1)	[165, 91]	7.79	8.00
	7(1)-5(2)-4(1)-2(3)-9(2)-3(3)-8(1)-6(2)-1(2)-10(2)	[240, 83]	7.75	

Model 6.1C: CMOSTSP with Risk/Discomfort Constraint in Random Environment (RaCMOSTSP):

Here we have taken the costs, times and risk/discomfort values as random for the CMOSTSP. Also we consider three types of conveyances. The random cost, time matrices for the CMOSTSP and random risk/discomfort matrix in the form of means and variances are given in Table 6.11. The Pareto optimum results of this CMOSTSP model for different values of k_1 and k_2 are obtained by iMOGA and presented in Table 6.12.

Model 6.1E: CMOSTSP with Risks/Discomforts Constraint in Random-Fuzzy Environment (RFCMOSTSP):

Here the costs, times and risk/discomfort factors are in random-fuzzy values for the CMOSTSP. Also we consider three types of conveyances. Assume that $\tilde{M}^c$ is a triangular fuzzy number. The random-fuzzy cost matrix for the CMOSTSP and corresponding random-fuzzy risk/discomfort matrix are presented in Table 6.13, where first part is a TFN (mean) and second part is a given variance presented in Table 6.13.

Here we took permissible probability levels $\hat{\theta}^{obj} = \hat{\theta}^{cst}$=0.94. We set L(x)=1-x, left and right spreads respectively $\alpha^c = m^c - \hat{h}^{obj}$, $\beta^c = m^c - 2 * \hat{h}^{obj}$, $\alpha^R = m^R - \hat{h}^{cst}$, $\beta^R = m^R - 2 * \hat{h}^{cst}$, $\alpha^r = m^r - \hat{h}^{cst}$, $\beta^r = m^r - 2 * \hat{h}^{cst}$.

Model 6.1D: CMOSTSP with risk/discomfort Constraint in Fuzzy Random Environment (FRCMOSTSP):

Here we have taken the costs, times and risk/discomfort as fuzzy random values for the CMOSTSP. Also we consider three types of conveyances. The extended operations on the basis of min-max cannot be directly applied to fuzzy numbers with discrete supports. So fuzzy numbers in LR-representation are used since computational effort in this case decreases very much . Assume that the costs are LR-type fuzzy random numbers as ($\hat{c}$, α, β) where $\hat{c}$ is a normal variate and α, β are respectively left and right spreads of the LR- fuzzy variables. Similarly time and risk/discomfort are taken as LR-type fuzzy random variables ($\hat{t}$, α, β) and ($\hat{r}$, α, β) where $\hat{t}$ and $\hat{r}$ are normal random variates and α, β are left and right spreads of the LR- fuzzy variables. The fuzzy random costs and times matrices for the CMOSTSP and corresponding fuzzy random risk/discomfort matrix are presented in Table 6.15.

Table 6.11: Input Data: RaCMOSTSP (Model 6.1C)

	Random **Cost** Matrix(10 ×10) for RCMOSTSP With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	(32,1.1) (37,1.21) (28,1.02)	(19,.9) (39,1.07) (30,1.11)	(21,1.02) (33,1.15) (35,1.17)	(30,1.01) (21,.98) (62,1.2)	(7,1.23) (23,1.02) (8,1.19)	(16,1.11) (36,1.03) (47,.97)	(28,1.04) (39,1.12) (19,1.18)	(41,1.12) (31,1.13) (42,1.03)	(21,1.02) (31,1.1) (43,1.01)
2	(35,1.12) (26,1.18) (17,1.13)	∞	(41,1.03) (21,1.17) (32,1.32)	(18,1.11) (29,1.12) (10,1.03)	(35,1.07) (26,1.2) (37,1.2)	(40,1.02) (31,1.2) (23,1.31)	(40,1.13) (30,1.15) (59,1.14)	(33,1.03) (42,1.21) (59,1.16)	(19,1.2) (37,1.13) (20,1.3)	(24,1.19) (16,1.12) (18,1.03)
3	(38,1.29) (30,1.13) (29,1.15)	(17,1.21) (58,1.43) (34,1.32)	∞	(12,1.25) (25,1.21) (14,1.11)	(42,1.23) (25,1.23) (46,1.24)	(35,1.21) (36,1.4) (34,1.12)	(19,1.13) (11,1.1) (8,1.3)	(32,1.1) (33,1.21) (25,1.16)	(30,1.11) (19,1.22) (41,1.41)	(30,1.21) (22,1.16) (33,1.33)
4	(28,1.14) (20,1.1) (10,1.31)	(10,1.2) (22,1.32) (14,1.2)	(18,1.21) (9,1.4) (29,1.31)	∞	(30,1.13) (19,1.15) (24,1.21)	(25,1.23) (16,1.12) (27,1.13)	(21,1.4) (31,1.4) (33,1.19)	(35,1.3) (36,1.2) (17,1.23)	(12,1.21) (23,1.31) (34,1.2)	(27,1.6) (48,1.2) (39,1.28)
5	(18,1.31) (15,1.2) (8,1.2)	(42,1.2) (23,1.31) (34,1.21)	(35,1.12) (36,1.41) (38,1.34)	(20,1.31) (13,1.31) (43,1.15)	∞	(30,1.21) (21,1.36) (41,1.5)	(45,1.16) (16,1.02) (27,1.31)	(30,1.24) (31,1.27) (13,1.02)	(19,1.34) (10,1.01) (8,1.04)	(28,1.42) (26,1.47) (27,1.21)
6	(15,1.31) (6,1.65) (7,1.27)	(29,1.15) (21,1.75) (29,1.15)	(4,1.32) (26,1.62) (28,1.72)	(8,1.41) (9,1.7) (12,1.04)	(28,1.61) (29,1.21) (39,1.37)	∞	(33,1.26) (42,1.31) (24,1.32)	(40,1.53) (31,1.32) (22,1.65)	(32,1.21) (23,1.34) (35,1.21)	(30,1.54) (41,1.52) (32,1.52)
7	(37,1.6) (39,1.43) (30,1.32)	(25,1.21) (53,1.6) (26,1.54)	(30,1.5) (38,1.71) (26,1.56)	(22,1.61) (43,1.31) (24,1.76)	(37,1.98) (58,1.21) (40,1.21)	(40,1.76) (21,1.65) (45,1.61)	∞	(10,1.31) (43,1.65) (13,1.21)	(33,1.54) (34,1.71) (36,1.37)	(20,1.04) (15,1.2) (26,1.6)
8	(41,1.27) (42,1.43) (23,1.15)	(26,1.43) (6,1.32) (17,1.23)	(32,1.34) (53,1.43) (45,1.17)	(40,1.21) (21,1.21) (42,1.31)	(35,1.53) (36,1.21) (47,1.32)	(25,1.53) (16,1.06) (5,1.03)	(40,1.27) (21,1.03) (43,1.04)	∞	(22,1.31) (53,1.62) (24,1.02)	(37,1.76) (36,1.78) (40,1.02)
9	(40,1.72) (11,1.21) (32,1.02)	(41,1.56) (39,1.56) (36,1.42)	(6,1.24) (36,1.42) (37,1.76)	(25,1.71) (34,1.57) (29,1.08)	(21,1.04) (32,1.3) (21,1.02)	(23,1.32) (33,1.06) (25,1.03)	(7,1.01) (38,1.02) (39,1.21)	(32,1.32) (33,1.76) (13,1.52)	∞	(28,1.41) (19,1.32) (26,1.72)
10	(17,1.51) (26,1.01) (29,1.21)	(30,1.31) (21,1.04) (32,1.92)	(28,1.15) (19,1.21) (30,1.72)	(20,1.72) (31,1.02) (22,1.51)	(11,1.82) (33,1.27) (22,1.19)	(32,1.52) (12,1.18) (34,1.17)	(38,1.02) (28,1.13) (39,1.16)	(41,1.62) (42,1.81) (33,1.21)	(31,1.52) (52,1.37) (32,1.15)	∞
	Random **Time** Matrix(10×10) for RCMOSTSP With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	(22,1.1) (17,1.21) (18,1.02)	(29,.9) (19,1.07) (20,1.11)	(31,1.02) (31,1.15) (15,1.17)	(20,1.01) (31,.98) (12,1.2)	(37,1.23) (33,1.02) (38,1.19)	(26,1.11) (16,1.03) (7,.97)	(18,1.04) (19,1.12) (39,1.18)	(11,1.12) (21,1.13) (22,1.03)	(31,1.02) (21,1.1) (23,1.01)
2	(15,1.12) (16,1.18) (37,1.13)	∞	(21,1.03) (21,1.17) (22,1.32)	(18,1.11) (19,1.12) (10,1.03)	(35,1.07) (16,1.2) (17,1.2)	(20,1.02) (21,1.2) (33,1.31)	(20,1.13) (20,1.15) (9,1.14)	(13,1.03) (22,1.21) (39,1.16)	(19,1.2) (17,1.13) (10,1.3)	(14,1.19) (16,1.12) (18,1.03)
3	(18,1.29) (20,1.13) (19,1.15)	(27,1.21) (8,1.43) (14,1.32)	∞	(12,1.25) (15,1.21) (34,1.11)	(12,1.23) (15,1.23) (6,1.24)	(25,1.21) (26,1.4) (14,1.12)	(39,1.13) (31,1.1) (38,1.3)	(12,1.1) (13,1.21) (15,1.16)	(20,1.11) (19,1.22) (21,1.41)	(20,1.21) (12,1.16) (23,1.33)
4	(18,1.14) (30,1.1) (40,1.31)	(30,1.2) (32,1.32) (24,1.2)	(38,1.21) (39,1.4) (19,1.31)	∞	(20,1.13) (39,1.15) (24,1.21)	(15,1.23) (46,1.12) (17,1.13)	(31,1.4) (21,1.4) (23,1.19)	(25,1.3) (16,1.2) (37,1.23)	(32,1.21) (23,1.31) (14,1.2)	(17,1.6) (8,1.2) (19,1.28)
5	(38,1.31) (35,1.2) (48,1.2)	(22,1.2) (33,1.31) (14,1.21)	(15,1.12) (16,1.41) (18,1.34)	(20,1.31) (13,1.31) (43,1.15)	∞	(20,1.21) (21,1.36) (21,1.5)	(5,1.16) (36,1.02) (17,1.31)	(20,1.24) (21,1.27) (33,1.02)	(29,1.34) (30,1.01) (38,1.04)	(28,1.42) (16,1.47) (27,1.21)
6	(35,1.31) (46,1.65) (37,1.27)	(19,1.15) (11,1.75) (19,1.15)	(44,1.32) (16,1.62) (18,1.72)	(8,1.41) (9,1.7) (12,1.04)	(18,1.61) (19,1.21) (29,1.37)	∞	(23,1.26) (22,1.31) (14,1.32)	(20,1.53) (21,1.32) (22,1.65)	(22,1.21) (23,1.34) (35,1.21)	(20,1.54) (21,1.52) (12,1.52)
7	(17,1.6) (19,1.43) (20,1.32)	(15,1.21) (3,1.6) (16,1.54)	(20,1.5) (18,1.71) (26,1.56)	(22,1.61) (43,1.31) (24,1.76)	(27,1.98) (8,1.21) (20,1.21)	(20,1.76) (11,1.65) (15,1.61)	∞	(20,1.31) (43,1.65) (13,1.21)	(33,1.54) (14,1.71) (36,1.37)	(10,1.04) (15,1.2) (16,1.6)
8	(31,1.27) (22,1.43) (33,1.15)	(16,1.43) (46,1.32) (37,1.23)	(12,1.34) (6,1.43) (5,1.17)	(40,1.21) (21,1.21) (42,1.31)	(15,1.53) (26,1.21) (17,1.32)	(15,1.53) (26,1.06) (35,1.03)	(10,1.27) (21,1.03) (13,1.04)	∞	(22,1.31) (33,1.62) (14,1.02)	(17,1.76) (16,1.78) (30,1.02)
9	(20,1.72) (31,1.21) (22,1.02)	(21,1.56) (19,1.56) (16,1.42)	(46,1.24) (26,1.42) (17,1.76)	(25,1.71) (34,1.57) (29,1.08)	(11,1.04) (22,1.3) (31,1.02)	(3,1.32) (23,1.06) (15,1.03)	(27,1.01) (28,1.02) (19,1.21)	(12,1.32) (13,1.76) (33,1.52)	∞	(18,1.41) (39,1.32) (16,1.72)

10	(27,1.51) (16,1.01) (19,1.21)	(20,1.31) (31,1.04) (22,1.92)	(18,1.15) (39,1.21) (20,1.72)	(20,1.72) (31,1.02) (22,1.51)	(21,1.82) (23,1.27) (12,1.19)	(12,1.52) (32,1.18) (14,1.17)	(28,1.02) (18,1.13) (19,1.16)	(21,1.62) (22,1.81) (13,1.21)	(11,1.52) (2,1.37) (22,1.15)	∞
	Random **Risks/Discomforts** Matrix(10×10) for RaCSTSP With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	(.62,1.1) (.54,1.21) (.28,1.02)	(.75,.9) (.53,1.07) (.64,1.11)	(.7,1.02) (.61,1.15) (.59,1.17)	(.66,1.01) (.78,.98) (.34,1.2)	(.87,1.23) (.71,1.02) (.88,1.19)	(.8,1.11) (.58,1.03) (.49,.97)	(.68,1.04) (.52,1.12) (.76,1.18)	(.5,1.12) (.64,1.13) (.55,1.03)	(.74,1.02) (.63,1.1) (.52,1.01)
2	(.6,1.12) (.65,1.18) (.79,1.13)	∞	(.54,1.03) (.74,1.17) (.63,1.32)	(.77,1.11) (.62,1.12) (.85,1.03)	(.6,1.07) (.68,1.2) (.58,1.2)	(.55,1.02) (.64,1.2) (.7,1.31)	(.54,1.13) (.66,1.15) (.35,1.14)	(.62,1.03) (.53,1.21) (.32,1.16)	(.76,1.2) (.58,1.13) (.73,1.3)	(.71,1.19) (.78,1.12) (.74,1.03)
3	(.58,1.29) (.64,1.13) (.66,1.15)	(.77,1.21) (.35,1.43) (.62,1.32)	∞	(.79,1.25) (.7,1.21) (.81,1.11)	(.54,1.23) (.745,1.23) (.49,1.24)	(.59,1.21) (.59,1.4) (.62,1.12)	(.76,1.13) (.85,1.1) (.86,1.3)	(.62,1.1) (.61,1.21) (.7,1.16)	(.66.11) (.76,1.22) (.52,1.41)	(.61,1.21) (.72,1.16) (.62,1.33)
4	(.65,1.14) (.76,1.1) (.84,1.31)	(.86,1.2) (.73,1.32) (.79,1.2)	(.78,1.21) (.9,1.4) (.65,1.31)	∞	(.66,1.13) (.79,1.15) (.71,1.21)	(.7,1.23) (.77,1.12) (.7,1.13)	(.77,1.4) (.63,1.4) (.63,1.19)	(.69,1.3) (.6,1.2) (.77,1.23)	(.82,1.21) (.71,1.31) (.59,1.2)	(.69,1.6) (.47,1.2) (.54,1.28)
5	(.8,1.31) (.8,1.2) (.88,1.2)	(.54,1.2) (.69,1.31) (.6,1.21)	(.6,1.12) (.6,1.41) (.56,1.34)	(.75,1.31) (.82,1.31) (.51,1.15)	∞	(.65,1.21) (.76,1.36) (.54,1.5)	(.5,1.16) (.8,1.02) (.68,1.31)	(.63,1.24) (.64,1.27) (.8,1.02)	(.76,1.34) (.84,1.01) (.86,1.04)	(.68,1.42) (.48,1.47) (.64,1.21)
6	(.8,1.31) (.89,1.65) (.85,1.27)	(.69,1.15) (.79,1.75) (.7,1.15)	(.89,1.32) (.76,1.62) (.65,1.72)	(.85,1.41) (.88,1.7) (.8,1.04)	(.7,1.61) (.68,1.21) (.53,1.37)	∞	(.63,1.26) (.55,1.31) (.73,1.32)	(.55,1.53) (.67,1.32) (.74,1.65)	(.63,1.21) (.72,1.34) (.7,1.21)	(.65,1.54) (.52,1.52) (.61,1.52)
7	(.55,1.6) (.57,1.43) (.66,1.32)	(.7,1.21) (.42,1.6) (.7,1.54)	(.67,1.5) (.59,1.71) (.71,1.56)	(.72,1.61) (.52,1.31) (.69,1.76)	(.62,1.98) (.37,1.21) (.54,1.21)	(.54,1.76) (.76,1.65) (.5,1.61)	∞	(.84,1.31) (.58,1.65) (.82,1.21)	(.62,1.54) (.62,1.71) (.6,1.37)	(.84,1.04) (.79,1.2) (.68,1.6)
8	(.55,1.23) (.55,1.43) (.72,1.15)	(.7,1.43) (.78,1.32) (.77,1.02)	(.65,1.34) (.42,1.43) (.5,1.32)	(.58,1.21) (.74,1.21) (.54,1.03)	(.59,1.53) (.6,1.21) (.48,1.05)	(.68,1.53) (.76,1.06) (.88,1.31)	(.57,1.27) (.72,1.03) (.52,1.38)	∞	(.72,1.31) (.44,1.62) (.61,1.73)	(.58,1.76) (.6,1.78) (.57,1.28)
9	(.54,1.72) (.84,1.21) (.57,1.02)	(.51,1.56) (.56,1.56) (.59,1.42)	(.88,1.24) (.6,1.42) (.6,1.76)	(.7,1.71) (.61,1.57) (.67,1.08)	(.72,1.04) (.67,1.3) (.75,1.02)	(.71,1.32) (.61,1.06) (.74,1.03)	(.87,1.01) (.62,1.02) (.58,1.21)	(.7,1.32) (.63,1.76) (.82,1.52)	∞	(.68,1.41) (.74,1.32) (.7,1.72)
10	(.8,1.51) (.7,1.01) (.64,1.21)	(.68,1.31) (.74,1.04) (.65,1.92)	(.69,1.15) (.55,1.21) (.66,1.72)	(.76,1.72) (.64,1.02) (.73,1.51)	(.8,1.82) (.61,1.27) (.74,1.19)	(.61,1.52) (.8,1.18) (.54,1.17)	(.58,1.02) (.68,1.13) (.58,1.16)	(.56,1.62) (.55,1.81) (.57,1.21)	(.63,1.52) (.42,1.37) (.6,1.15)	∞

Table 6.12: Results of RaCMOSTSP (Model 6.1C)

K1	K2	Algorithm	Path(Vehicle)	Costs & Time	R_{max}
0.5	0.5	iMOGA	10(1)-5(2)-9(2)-1(1)-6(1)-4(2)-3(3)-7(1)-8(2)-2(3)	[50.80, 39.3]	8.5
		AMOGA	5(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-6(2)-1(3)-7(3)	[54.32, 35.65]	8.5
		iMOGA	7(3)-4(2)-1(3)-5(1)-6(1)-2(2)-3(3)-10(1)-8(2)-9(3)	[56.60,33.20]	8.5
0.5	0.5	MOGA	5(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-6(2)-1(3)-7(3)	[64.32, 27.71]	8.5
0.4	0.6	iMOGA	10(1)-5(2)-9(2)-1(1)-6(1)-4(2)-3(3)-7(1)-8(2)-2(3)	[41.36, 48.63]	8.5.
0.6	0.4	iMOGA	10(1)-5(2)-9(2)-1(1)-6(1)-4(2)-3(3)-7(1)-8(2)-2(3)	[60.24, 26.28]	8.5
0.5	0.5	iMOGA	6(2)-4(3)-3(1)-5(1)-2(3)-7(1)-8(2)-2(1)-1(2)-9(2)	[72.2, 37.84]	8.25
0.5	0.5	MOGA	4(2)-10(3)-2(2)-5(3)-3(3)-9(1)-8(1)-6(2)-1(3)-7(3)	[84.17, 41.18]	8.0

Table 6.13: Input Data: RFCMOSTSP (Model 6.1E)

	Random-Fuzzy **Cost** Matrix(10 $\times$10) for RCMOSTSP With Three Conveyances				
i/j	1	2	3	4	5
1	∞	[(32,35,36),1.21] [(36,37,39),1.21] [(26,28,29),1.08]	[(17,19,20),.98] [(38,39,42),1.32] [(26,30,31),1.03]	[(17,21,22),1.76] [(31,33,34),1.16] [(33,35,36),1.23]	[(29,30,31),1.13] [(20,21,23),1.13] [(60,62,63),1.05]
2	[(34,35,38),1.34] [(22,26,27),1.12] [(14,17,19),1.54]	∞	[(40,41,44),1.42] [(18,21,22),1.14] [(27,32,33),1.36]	[(16,18,19),1.13] [(28,29,32),1.17] [(6,10,12),1.12]	[(32,35,37),1.45] [25,26,27),1.18] [(34,37,38),1.4]
3	[(36,38,39),1.18] [(29,30,32),1.41] [(28,29,32),1.72]	[(16,17,20),1.43] [(54,58,60),1.31] [(31,34,35),1.32]	∞	[(10,12,13),1.17] [(24,25,26),1.17] [(12,14,17),1.03]	[(40,42,45)1.54] [(23,25,26),1.02] [(45,46,48),1.13]
4	[(27,28,30),1.42] [(18,20,21),1.46] [(9,10,12),1.14]	[(9,10,11),1.17] [(19,22,23),1.32] [(12,14,15),1.17]	[(16,18,20),1.18] [(7,9,10),1.62] [(27,29,30),1.14]	∞	[(29,30,33),.9] [(17,19,20),1.54] [(23,24,25),1.76]
5	[(16,18,19,1.17] [(14,15,18),1.3] [(6,8,9),1.3]	[(41,42,44),1.17] [(21,23,24),1.3] [(32,34,37),1.3]	[(34,35,37),1.14] [(35,36,37),1.3] [(33,38,39),1.3]	[(17,20,21)1.2] [(12,13,14),1.38] [(40,43,44),1.16]	∞
6	[(13,15,16),1.3] [(5,6,8),1.3] [(5,7,8),1.3]	[(26,29,30),1.54] [(20,21,23),1.17] [(27,29,30),1.3]	[(4,4,6),1.17] [(25,26,27),1.41] [(27,28,30),1.3]	[(6,8,9),1.3]),1.13] [(7,9,11),1.2] [(10,12,13),1.24]	[(26,28,29),1.34] [(26,29,30),1.73] [(38,39,41),1.3]
7	[(36,37,39),1.71] [(37,39,40),1.43] [(28,30,32),1.43]	[(23,25,26),1.16] [(53,53,55),1.13] [(25,26,27),1.31]	[(27,30,32),1.3] [(37,38,39),1.3] [(24,26,27),.98]	[(21,22,24),1.3] [(40,43,44),1.17] [(23,24,25),1.3]	[(35,37,38),1.43] [(56,58,60),1.3] [(37,39,40),1.23]
8	[(39,41,42),1.37] [(41,42,43),1.14] [(20,23,24),1.46]	[(24,26,28),1.43] [(5,6,7),1.33] [(16,17,18),1.23]	[(30,32,33),1.54] [(52,53,54),1.22] [(43,45,46),1.79]	[(38,40,42),1.27] [(19,21,22),1.3] [(40,42,43),1.3]	[(34,35,37),1.3] [(34,36,37),1.25] [(46,47,48),1.3]
9	[(38,40,41),1.41] [(10,11,13),1.02] [(31,32,33),1.37]	[(39,41,42),1.21] [(38,39,40),1.28] [(34,36,37),1.11]	[(4,6,9),1.16] [(34,36,37),1.45] [(36,37,39),1.19]	[(23,25,26),1.3] [(33,34,36),1.3] [(28,29,30),1.3]	[(20,21,23),1.3] [(31,32,33),1.41] [(20,21,22),1.3]
10	[(15,17,18),1.12] [(25,26,28),1.13] [(25,29,30),1.2]	[(28,30,31),1.34] [(20,21,22),1.33] [(31,32,34),1.63]	[(26,28,29),1.32] [(18,19,20),1.23] [(28,30,32),1.13]	[(18,20,21),1.3] [(29,31,32),1.43] [(21,22,24),1.53]	[(9,11,12),1.47] [(32,33,34),1.63] [(20,22,24),1.37]
	Random-Fuzzy **Cost** Matrix(10 $\times$10) for RaCMOSTSP With Three Conveyances				
i/j	6	7	8	9	10
1	[(5,7,10),1.32] [(22,23,25),1.16] [(6,8,9),1.06]	[(15,16,18),.99) [(35,33,37),1.14] [(46,47,48),1.23]	[(25,28,29),1.1] [(37,39,43),1.11] [(16,19,20),1.9]	[(39,41,42),1.13] [(26,31,33),1.15] [(41,42,43)1.22]	[(20,22,23),1.12] [(30,31,34),1.09] [(42,43,45),1.41]
2	[(39,40,41),1.2] [(30,31,32),1.34] [(21,23,26),1.76]	[(39,40,42),1.67] [(29,30,32),1.32] [(57,59,60),1.33]	[(30,33,34),1.13] [(41,42,45),1.41] [(58,59,62),1.72]	[(17,19,22),1.16] [(36,37,38),1.3] [(17,20,21),1.8]	[(23,24,26),1.14] [(13,16,17),1.17] [(17,18,20),1.17]
3	[(33,35,36),1.13] [(34,36,39),1.13] [(33,34,35),1.5]	[(17,19,20),1.15] [(11,11,12),1.17] [(5,8,10),1.14]	[(30,32,33),1.98] [(30,33,34),1.07] [(24,25,27),1.53]	[(28,30,31),1.09] [(18,19,21),1.73] [(40,41,44),1.72]	[(29,30,31),1.31] [(19,22,23),1.32] [(32,33,35),1.36]
4	[(23,25,26),1.3] [(15,16,18),1.43] [(25,27,28),1.9]	[(19,21,22),.78] [(30,31,32),1.52] [(30,33,34),1.31]	[(33,35,36),1.7] [(32,36,38),1.15] [(16,17,18),1.7]	[(10,12,13),1.6] [(20,23,24),1.76] [(32,34,35),1.45]	[(24,27,29),1.65] [(47,48,49),1.17] [(37,39,40),1.76]
5	[(29,30,31),1.26] [(20,21,23),1.3] [(40,41,42),1.15]	[(42,45,46),1.23] [(14,16,18),1.3] [(25,27,27),1.54]	[(27,30,31),1.18] [(30,31,32),1.3] [(12,13,16),1.71]	[(18,19,22),1.3] [(8,10,11),1.3] [(7,8,9),1.3]	[(26,28,29),1.51] [(25,26,27),1.3] [(25,27,28),1.3]

6	∞	[(31,33,34),1.21] [(40,43,44),1.3] [(23,24,26),1.3]	[(39,40,42),1.3] [(30,31,31),1.3] [(20,22,23),1.3]	[(30,32,33),1.3] [(22,23,24),1.3] [(35,35,36),1.28]	[(28,30,31),1.3] [(40,41,42),1.47] [(30,32,34),1.3]
7	[(38,40,41),1.14] [(20,21,22),1.16] [(43,45,46),1.24]	∞	[(7,10,11),1.3] [(40,43,44),1.3] [(11,13,14),1.3]	[(31,33,34),1.3] [(33,34,35),1.45] [(34,36,37),1.3]	[(19,20,22),1.46] [(13,15,16),1.3] [(25,26,28),1.3]
8	[(23,25,26),1.3] [(15,16,18),1.3] [(4,5,6),1.3]	[(39,40,42),1.3] [(19,21,22),1.04] [(41,43,4),1.12]	∞	[(20,22,23),1.67] [(52,53,54),1.61] [(23,24,27),1.3]	[(35,37,38),1.3] [(35,36,38),1.3] [(39,40,41),1.15]
9	[(22,23,25),1.3] [(31,33,34),1.68] [(23,25,26),1.3]	[(5,7,8),1.17] [(36,38,39),1.3] [(38,39,41),1.3]	[(30,32,33),1.7] [(32,33,34),1.27] [(11,13,15),1.3]	∞	[(27,28,30),1.04] [(18,19,20),1.3] [(24,26,27),1.3]
10	[(30,32,34),1.49] [(10,12,13),1.41] [(33,34,35),1.57]	[(35,38,39),1.3] [(26,28,29),1.3] [(38,39,41),1.17]	[(40,41,43),1.23] [(41,42,43),1.3] [(30,33,34),1.15]	[(29,31,32),1.25] [(51,52,54),1.3] [(30,32,33),1.2]	∞
	Random-Fuzzy **Time** Matrix(10 $\times$10) for RCMOSTSP With Three Conveyances				
i/j	1	2	3	4	5
1	∞	[(12,15,16),1.21] [(16,17,19),1.21] [(16,18,19),1.08]	[(27,29,30),.98] [(18,19,12),1.32] [(16,20,21),1.03]	[(27,31,32),1.54] [(21,23,24),1.43] [(13,15,16),1.23]	[(19,20,21),1.13] [(30,31,33),1.13] [(5,6,7),1.05]
2	[(14,15,18),1.34] [(32,36,37),1.12] [(34,37,39),1.54]	∞	[(10,11,14),1.42] [(28,31,32),1.14] [(17,22,23),1.36]	[(36,38,39),1.13] [(18,19,22),1.17] [(46,51,52),1.12]	[(22,25,27),1.45] [15,16,17),1.18] [(31,17,18),1.4]
3	[(16,18,19),1.18] [(19,20,22),1.41] [(18,19,22),1.72]	[(26,27,30),1.43] [(4,8,10),1.31] [(11,14,15),1.32]	∞	[(20,22,23),1.17] [(34,35,36),1.17] [(32,34,37),1.03]	[(10,12,15)1.54] [(13,15,16),1.02] [(5,6,8),1.13]
4	[(17,18,20),1.42] [(28,30,31),1.46] [(49,50,52),1.14]	[(39,40,41),1.17] [(39,42,43),1.32] [(32,34,35),1.17]	[(26,28,30),1.18] [(47,49,50),1.62] [(37,39,40),1.14]	∞	[(19,20,23),.9] [(37,39,40),1.54] [(33,34,35),1.76]
5	[(36,38,39),1.17] [(14,15,18),1.3] [(56,58,59),1.3]	[(11,12,14),1.17] [(21,23,24),1.3] [(22,24,27),1.3]	[(14,15,17),1.14] [(35,36,37),1.3] [(23,28,29),1.3]	[(27,30,31)1.2] [(12,13,14),1.38] [(10,13,14),1.16]	∞
6	[(13,15,16),1.3] [(55,56,58),1.3] [(45,47,48),1.3]	[(26,29,30),1.54] [(30,31,33),1.17] [(17,19,20),1.3]	[(4,4,6),1.17] [(15,16,17),1.41] [(17,18,20),1.3]	[(6,8,9),1.3]),1.13] [(57,59,61),1.2] [(40,42,43),1.24]	[(26,28,29),1.34] [(16,19,20),1.73] [(18,19,21),1.3]
7	[(26,27,29),1.71] [(17,19,20),1.43] [(18,20,22),1.43]	[(13,15,16),1.16] [(3,5,6),1.13] [(15,16,17),1.31]	[(17,20,22),1.3] [(17,18,19),1.3] [(14,16,17),.98]	[(31,32,34),1.3] [(10,13,14),1.17] [(13,14,15),1.3]	[(25,27,28),1.43] [(6,8,9),1.3] [(17,19,20),1.23]
8	[(19,21,22),1.37] [(21,22,23),1.14] [(30,33,34),1.46]	[(14,16,18),1.43] [(45,46,47),1.33] [(26,27,28),1.23]	[(20,22,23),1.54] [(2,5,9),1.22] [(3,5,6),1.79]	[(18,19,22),1.27] [(39,41,42),1.3] [(10,1213),1.3]	[(14,15,17),1.3] [(24,26,27),1.25] [(6,7,8),1.3]
9	[(18,20,21),1.41] [(30,31,33),1.02] [(21,22,23),1.37]	[(9,11,12),1.21] [(18,19,20),1.28] [(14,16,17),1.11]	[(44,46,49),1.16] [(14,16,17),1.45] [(16,17,19),1.19]	[(13,15,16),1.3] [(23,24,26),1.3] [(18,19,20),1.3]	[(30,31,33),1.3] [(21,22,23),1.41] [(10,11,12),1.3]
10	[(35,37,38),1.12] [(35,36,38),1.13] [(15,19,20),1.2]	[(28,30,31),1.34] [(30,31,32),1.33] [(11,12,14),1.63]	[(26,28,29),1.32] [(38,39,30),1.23] [(18,20,22),1.13]	[(38,40,41),1.3] [(9,11,12),1.43] [(11,12,14),1.53]	[(49,51,52),1.47] [(12,13,14),1.63] [(10,12,15),1.37]
	Random-Fuzzy **Time** Matrix(10 $\times$10) for RCMOSTSP With Three Conveyances				
i/j	6	7	8	9	10
1	[(55,57,60),1.32] [(12,13,15),1.16] [(56,58,59),1.06]	[(35,36,38),.99) [(15,16,17),1.14] [(6,7,8),1.23]	[(15,18,19),1.1] [(17,19,23),1.11] [(36,39,30),1.9]	[(19,21,22),1.13] [(16,21,13),1.15] [(21,22,23)1.22]	[(30,32,33),1.12] [(10,11,14),1.09] [(22,23,25),1.41]
2	[(19,20,21),1.2] [(10,11,12),1.34] [(31,33,36),1.76]	[(19,10,12),1.67] [(39,40,42),1.32] [(7,9,10),1.33]	[(10,13,14),1.13] [(11,12,15),1.41] [(8,9,12),1.72]	[(37,39,42),1.16] [(16,17,18),1.3] [(27,30,31),1.8]	[(23,24,26),1.14] [(23,26,27),1.17] [(27,28,30),1.17]

3	[(13,15,16),1.13] [(14,16,19),1.13] [(13,14,15),1.5]	[(17,19,20),1.15] [(31,31,32),1.17] [(45,48,50),1.14]	[(20,22,23),1.98] [(10,13,14),1.07] [(14,15,17),1.53]	[(18,20,21),1.09] [(18,19,21),1.73] [(10,11,14),1.72]	[(19,20,21),1.31] [(19,22,23),1.32] [(12,13,15),1.36]
4	[(13,15,16),1.3] [(25,26,28),1.43] [(15,17,18),1.9]	[(19,21,22),.78] [(10,11,12),1.52] [(20,23,24),1.31]	[(13,15,16),1.7] [(12,16,18),1.15] [(36,37,38),1.7]	[(40,42,43),1.6] [(10,13,14),1.76] [(22,24,25),1.45]	[(24,27,29),1.65] [(7,8,9),1.17] [(17,19,20),1.76]
5	[(19,20,21),1.26] [(30,31,33),1.3] [(10,11,12),1.15]	[(2,5,6),1.23] [(24,26,28),1.3] [(15,17,17),1.54]	[(17,18,21),1.18] [(20,21,22),1.3] [(22,23,26),1.71]	[(18,19,22),1.3] [(38,40,41),1.3] [(27,28,29),1.3]	[(26,28,29),1.51] [(25,26,27),1.3] [(15,17,18),1.3]
6	∞	[(11,13,14),1.21] [(10,13,14),1.3] [(13,14,16),1.3]	[(19,20,22),1.3] [(10,11,12),1.3] [(10,12,13),1.3]	[(10,12,13),1.3] [(12,13,14),1.3] [(15,15,16),1.28]	[(18,20,21),1.3] [(10,11,12),1.47] [(10,12,14),1.3]
7	[(38,40,41),1.14] [(20,21,22),1.16] [(3,5,6),1.24]	∞	[(7,10,11),1.3] [(40,43,44),1.3] [(31,33,34),1.3]	[(31,33,34),1.3] [(33,34,35),1.45] [(14,16,17),1.3]	[(19,20,22),1.46] [(13,15,16),1.3] [(15,16,18),1.3]
8	[(13,15,16),1.3] [(25,26,28),1.3] [(44,45,46),1.3]	[(9,10,12),1.3] [(39,41,42),1.04] [(1,3,4),1.12]	∞	[(10,12,13),1.67] [(2,3,5),1.61] [(13,14,17),1.3]	[(15,17,18),1.3] [(15,16,18),1.3] [(9,10,11),1.15]
9	[(12,13,15),1.3] [(21,23,24),1.68] [(23,25,26),1.3]	[(35,37,38),1.17] [(16,18,19),1.3] [(18,19,21),1.3]	[(10,12,13),1.7] [(12,13,14),1.27] [(31,33,35),1.3]	∞	[(17,18,20),1.04] [(28,32,30),1.3] [(14,16,17),1.3]
10	[(20,22,24),1.49] [(30,32,33),1.41] [(13,14,15),1.57]	[(15,18,19),1.3] [(16,18,19),1.3] [(18,19,21),1.17]	[(20,21,23),1.23] [(3,4,6),1.3] [(10,13,14),1.15]	[(19,21,22),1.25] [(1,2,4),1.3] [(10,12,13),1.2]	∞

Random-Fuzzy **Risk/Discomfort** Matrix(10 ×10) for RCMOSTSP With Three Conveyances

i/j	1	2	3	4	5
1	∞	[(.69,.65,.61,1.12] [(.36,.37,.39),1.21] [(.26,.28,.29),1.08]	[(.72,.7,.68),.1.13 [(.38,.39,.42),1.32] [(.26,.30,.31),1.03]	[(.73,.71,.62),1.76] [(.31,.33,.34),1.16] [(.33,.35,.36),1.23]	[(.61,.30,.31),1.13] [(.20,.21,.23),1.13] [(.60,.62,.63),1.05]
2	[(.34,.35,.38),1.34] [(.22,.26,.27),1.12] [(.14,.17,.19),1.54]	∞	[(.40,.41,.44),1.42] [(.18,.21,.22),1.14] [(.27,.32,.33),1.36]	[(.16,.18,.19),1.13] [(.28,.29,.32),1.17] [(.6,.10,.12),1.12]	[(.32,.35,.37),1.45] [.25,.26,.27),1.18] [(.34,.37,.38),1.4]
3	[(.36,.38,.39),1.18] [(.29,.30,.32),1.41] [(.28,.29,.32),1.72]	[(.16,.17,.20),1.43] [(.54,.58,.61),1.31] [(.31,.34,.35),1.32]	∞	[(.10,.12,.13),1.17] [(.24,.25,.26),1.17] [(.12,.14,.17),1.03]	[(.40,.42,.45)1.54] [(.23,.25,.26),1.02] [(.45,.46,.48),1.13]
4	[(.27,.28,.30),1.42] [(.18,.20,.21),1.46] [(.9,.10,.12),1.14]	[(.9,.10,.11),1.17] [(.19.,22,.23),1.32] [(.12,.14,.15),1.17]	[(.16,.18,.20),1.18] [(.7,.9,.10),1.62] [(.27,.29,.30),1.14]	∞	[(.29,.30,.33),.9] [(.17,.19,.20),1.54] [(.23,.24,.25),1.76]
5	[(.16,.18,.19,1.17] [(14.,.15,.18),1.3] [(.6,.8,.9),1.3]	[(.41,.42,.44),1.17] [(.21,.23,.24),1.3] [(.32,.34,.37),1.3]	[(.34,.35,.37),1.14] [(.35,.36,.37),1.3] [(.33,.38,.39),1.3]	[(.17,.20,..21)1.2] [(.12,.13,.14),1.38] [(.40,.43,.44),1.16]	∞
6	[(.13,.15,.16),1.3] [(.5,.6,.8),1.3] [(.5,.7,.8),1.3]	[(.26,.29,.30),1.54] [(.20,.21,.23),1.17] [(.27,.29,.30),1.3]	[(.4,.4,.6),1.17] [(.25,.26,.27),1.41] [(.27,.28,.30),1.3]	[(.6,.8,.9),1.3]),1.13] [(.7,.9,.11),1.2] [(.10,.12,.13),1.24]	[(.26,.28,.29),1.34] [(.26,.29,.30),1.73] [(.38,.39,.41),1.3]
7	[(.36,.37,.39),1.71] [(.37,.39,.40),1.43] [(.28,.3,.32),1.43]	[(.23,.25,.26),1.16] [(.53,.53,.55),1.13] [(.25,.26,.27),1.31]	[(.27,.30,.32),1.3] [(.37,.38,.39),1.3] [(.24,.26,.27),.98]	[(.21,.22,.24),1.3] [(.4,.43,.44),1.17] [(.23,.24,.25),1.3]	[(.35,.37,.38),1.43] [(.56,.58,.60),1.3] [(.37,.39,.40),1.23]
8	[(.39,.41,.42),1.37] [(.41,.42,.43),1.14] [(.2,.23,.24),1.46]	[(.24,.26,.28),1.43] [(.5,.6,.7),1.33] [(.16,.17,.18),1.23]	[(.30,.32,.33),1.54] [(.52,.53,.54),1.22] [(.43,.45,.46),1.79]	[(.38,.40,.42),1.27] [(.19,.21,.22),1.3] [(.4,.42,.43),1.3]	[(.34,.35,.37),1.3] [(.34,.36,.37),1.25] [(.46,.47,.48),1.3]
9	[(.38,.40,.41),1.41] [(.1,.11,.13),1.02] [(.31,.32,.33),1.37]	[(.39,.41,.42),1.21] [(.38,.39,.4),1.28] [(.34,.36,.37),1.11]	[(.4,.6,.9),1.16] [(.34,.36,.37),1.45] [(.36,.37,.39),1.19]	[(.23,.25,.26),1.3] [(.33,.34,.36),1.3] [(.28,.29,.30),1.3]	[(.20,.21,.23),1.3] [(.31,.32,.33),1.41] [(.2,.21,.22),1.3]
10	[(.15,.17,.18),1.12] [(.25,.26,.28),1.13] [(.25,.29,.30),1.2]	[(.28,.30,.31),1.34] [(.2,.21,.22),1.33] [(.31,.32,.34),1.63]	[(.26,.28,.29),1.32] [(.18,.19,.20),1.23] [(.28,.30,.32),1.13]	[(.18,.20,.21),1.3] [(.29,.31,.32),1.43] [(.21,.22,.24),1.53]	[(.9,.11,.12),1.47] [(.32,.33,.34),1.63] [(.20,.22,.24),1.37]

Random-Fuzzy Risk/Discomfort Matrix(10 ×10) for RCMOSTSP With Three Conveyances

i/j	6	7	8	9	10
1	[(.5,.7,.10),1.32] [(.22,.23,.25),1.16] [(.6,.8,.9),1.06]	[(.15,.16,.18),.99) [(.35,.33,.37),1.14] [(.46,.47,.48),1.23]	[(.25,.28,.29),1.1] [(.37,.39,.43),1.11] [(.16,.19,.20),1.9]	[(.39,.41,.42),1.13] [(.26,.31,.33),1.15] [(.41,.42,.43)1.22]	[(.20,.22,.23),1.12] [(.30,.31,.34),1.09] [(.42,.43,.45),1.41]
2	[(.39,.40,.41),1.2] [(.30,.31,.32),1.34] [(.21,.23,.26),1.76]	[(.39,.40,.42),1.67] [(.29,.30,.32),1.32] [(.57,.59,.60),1.33]	[(.30,.33,.34),1.13] [(.41,.42,.45),1.41] [(.58,.59,.62),1.72]	[(.17,.19,.22),1.16] [(.36,.37,.38),1.3] [(.17,.20,.21),1.8]	[(.23,.24,.26),1.14] [(.13,.16,.17),1.17] [(.17,.18,.20),1.17]

3	[(.33,.35,.36),1.13] [(.34,.36,.39),1.13] [(.33,.34,.35),1.5]	[(.17,.19,.20),1.15] [(.11,.11,.12),1.17] [(.5,.8,.10),1.14]	[(.30,.32,.33),1.98] [(.3,.33,.34),1.07] [(.24,.25,.27),1.53]	[(.28,.30,.31),1.09] [(.18,.19,.21),1.73] [(.40,.41,.44),1.72]	[(.29,.30,.31),1.31] [(.19,.22,.23),1.32] [(.32,.33,.35),1.36]
4	[(.23,.25,.26),1.3] [(.15,.16,.18),1.43] [(.25,.27,.28),1.9]	[(.19,.21,.22),.78] [(.30,.31,.32),1.52] [(.30,.33,.34),1.31]	[(.33,.35,.36),1.7] [(.32,.36,.38),1.15] [(.16,.17,.18),1.7]	[(.10,.12,.13),1.6] [(.20,.23,.24),1.76] [(.32,.34,.35),1.45]	[(.24,.27,.29),1.65] [(.47,.48,.49),1.17] [(.37,.39,.40),1.76]
5	[(.29,.30,.31),1.26] [(.20,.21,.23),1.3] [(.40,.41,.42),1.15]	[(.42,.45,.46),1.23] [(.14,.16,.18),1.3] [(.25,.27,.27),1.54]	[(.27,.30,.31),1.18] [(.30,.31,.32),1.3] [(.12,.13,.16),1.71]	[(.18,.19,.22),1.3] [(.08,.01,.11),1.3] [(.07,.08,.09),1.3]	[(.26,.28,.29),1.51] [(.25,.26,.27),1.3] [(.25,.27,.28),1.3]
6	∞	[(.31,.33,.34),1.21] [(.40,.43,.44),1.3] [(.23,.24,.26),1.3]	[(.39,.40,.42),1.3] [(.30,.31,.31),1.3] [(.2,.22,.23),1.3]	[(.30,.32,.33),1.3] [(.22,.23,.24),1.3] [(.35,.35,.36),1.28]	[(.28,.3,.31),1.3] [(.4,.41,.42),1.47] [(.3,.32,.34),1.3]
7	[(.38,.4,.41),1.14] [(.20,.21,.22),1.16] [(.43,.45,.46),1.24]	∞	[(.07,.1,.11),1.3] [(.40,.43,.44),1.3] [(.11,.13,.14),1.3]	[(.31,.33,.34),1.3] [(.33,.34,.35),1.45] [(.34,.36,.37),1.3]	[(.19,.20,.22),1.46] [(.13,.15,.16),1.3] [(.25,.26,.28),1.3]
8	[(.23,.25,.26),1.3] [(.15,.16,.18),1.3] [(.04,.05,.06),1.3]	[(.39,.40,.42),1.3] [(.19,.21,.22),1.04] [(.41,.43,.4),1.12]	∞	[(.20,.22,.23),1.67] [(.52,.53,.54),1.61] [(.23,.24,.27),1.3]	[(.35,.37,.38),1.3] [(.35,.36,.38),1.3] [(.39,.40,.41),1.15]
9	[(.22,.23,.25),1.3] [(.31,.33,.34),1.68] [(.23,.25,.26),1.3]	[(.05,.07,.08),1.17] [(.36,.38,.39),1.3] [(.38,.39,.41),1.3]	[(.3,.32,.33),1.7] [(.32,.33,.34),1.27] [(.11,.13,.15),1.3]	∞	[(.27,.28,.30),1.04] [(.18,.19,.20),1.3] [(.24,.26,.27),1.3]
10	[(.30,.32,.34),1.49] [(.10,.12,.13),1.41] [(.33,.34,.35),1.57]	[(.35,.38,.39),1.3] [(.26,.28,.29),1.3] [(.38,.39,.41),1.17]	[(.40,.41,.43),1.23] [(.41,.42,.43),1.3] [(.30,.33,.34),1.15]	[(.29,.31,.32),1.25] [(.51,.52,.54),1.3] [(.30,.32,.33),1.2]	∞

Table 6.14: Results of RFCMOSTSP (Model 6.1E)

$\hat{h}^{obj}$	$\hat{h}^{cst}$	Algorithm	DM	Path(Vehicle)	Costs	R_{max}
0.95	0.95	iMOGA	PDM	3(1)-10(3)-2(3)-9(3)-7(3)-8(1)-6(2)-1(1)-5(2)-4(3)	[152.68,103.2]	8.5
			ODM	3(1)-10(3)-2(3)-9(3)-7(3)-8(1)-6(2)-1(1)-5(2)-4(3)	[144.2,121.3]	8.5
		iMOGA	PDM	5(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-6(2)-1(3)-7(3)	[156.5, 100.4]	8.5
			ODM	5(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-6(2)-1(3)-7(3)	[146.5, 112.3]	8.5
		iMOGA	PDM	10(2)-1(1)-9(2)-5(1)-7(2)-3(1)-4(2)-2(3)-8(1)-6(1)	[156.61, 110.7]	6.75
			ODM	10(2)-1(1)-9(2)-5(1)-7(2)-3(1)-4(2)-2(3)-8(1)-6(1)	[148.3, 124.1]	6.75
		MOGA	PDM	6(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-5(2)-1(3)-7(3)	[224.2, 117.3]	6.0
			ODM	6(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-5(2)-1(3)-7(3)	[162.4, 101.4]	6.0
0.95	0.7	iMOGA	PDM	6(2)-8(1)-7(1)-1(1)-10(2)-5(3)-2(3)-3(1)-4(2)-9(3)	[145.2, 132.7]	6.75
			ODM	6(2)-8(1)-7(1)-1(1)-10(2)-5(3)-2(3)-3(1)-4(2)-9(3)	[141.7, 138.2]	6.75
0.7	0.95	iMOGA	PDM	4(3)-7(1)-8(1)-6(2)-1(2)-5(1)-9(2)-2(1)-10(1)-3(1)	[164.9, 98.4]	6.5
			ODM	4(3)-7(1)-8(1)-6(2)-1(2)-5(1)-9(2)-2(1)-10(1)-3(1)	[154.1, 118.2]	6.5
0.8	0.75	iMOGA	PDM	3(1)-4(3)-2(3)-10(2)-5(3)-9(3)-8(1)-6(1)-1(3)-7(2)	[151.2, 120.9]1	6.0
			ODM	3(1)-4(3)-2(3)-10(2)-5(3)-9(3)-8(1)-6(1)-1(3)-7(2)	[147.3, 125.7]	6.0

Table 6.15: Input Data: FRCMOSTSP (Model 6.1D)

	Fuzzy Random **Cost** Matrix(10×10) for FRCSTSP With Three Conveyances				
i/j	1	2	3	4	5
1	∞	[(c,5,6),c∼ N(35,1)] [(c,3,3),c∼N(36,2)] [(c,2,2),c∼N(26,2)]	[(c,1,2),c∼N(17,2)] [(c,3,2),c∼N(38,2)] [(c,3,3),c∼N(26,2)]	[(c,1,2),c∼N(16,3)] [(c,3,4),c∼N(31,2)] [(c,5,6),c∼N(33,2)]	[(c,3,3),c∼N(29,2)] [(c,1,2),c∼N(20,2)] [(c,2,3),c∼N(60,4)]
2	[(c,3,3),c∼N(31,1)] [(c,2,2),c∼N(22,3)] [(c,1,1),c∼N(14,4)]	∞	[(c,4,4),c∼N(40,2)] [(c,2,2),c∼N(18,2)] [(c,2,3),c∼N(27,1)]	[(c,1,1),c∼N(16,2)] [(c,2,3),c∼N(28,2)] [(c,1,1),c∼N(6,3)]	[(c,3,3),c∼N(32,2)] [(c,2,2),c∼N(25,2)] [(c,3,4),c∼N(38,1)]
3	[(c,3,3),c∼N(36,2)] [(c,3,3),c∼N(26,3)] [(c,2,3),c∼N(28,2)]	[(c,1,2),c∼N(16,3)] [(c,5,6),c∼N(54,1)] [(c,3,3),c∼N(31,2)]	∞	[(c,1,1),c∼N(10,2)] [(c,2,2),c∼N(24,1)] [(c,1,1),c∼N(12,1)]	[(c,4,4)c∼N(40,1)] [(c,2,2),c∼N(26,3)] [(c,4,4),c∼N(45,5]
4	[(c,2,3),c∼N(26,1)] [(c,2,2),c∼N(18,2)] [(c,1,1),c∼N(9,2)]	[(c,1,1),c∼N(9,2)] [(c,2,2),c∼N(19,1)] [(c,1,1),c∼N(12,4)]	[(c,1,2),c∼N(16,4)] [(c,5,4),c∼N(7,2)] [(c,2,3),c∼N(27,2)]	∞	[(c,3,3),c∼N(29,3)] [(c,1,2),c∼N(17,3)] [(c,2,2),c∼N(23,4)]
5	[(c,1,1,c∼N(16,1)] [(c,1,1),c∼N(14,1)] [(c,1,1),c∼N(6,2)]	[(c,4,4),c∼N(41,3)] [(c,2,2),c∼N(22,2)] [(c,3,3),c∼N(32,3)]	[(c,3,3),c∼N(34,2)] [(c,6,7),c∼N(36,2)] [(c,3,3),c∼N(33,2)]	[(c,2,2),c∼N(17,2)] [(c,1,1),c∼N(12,2)] [(c,4,4),c∼N(38,3)]	∞
6	[(c,1,1),c∼N(15,1)] [(c,1,4),c∼N(6,2)] [(c,2,3),c∼N(6,2)]	[(c,2,3),c∼N(26,2)] [(c,2,2),c∼N(20,1)] [(c,2,3),c∼N(26,2)]	[(c,1,1),c∼N(5,5)] [(c,6,7),c∼N(36,2)] [(c,2,3),c∼N(26,3)]	[(c,2,3),c∼N(6,2)])] [c,1,1),c∼N(13,3)] [(c,1,1),c∼N(10,3)]	[(c,2,2),c∼N(26,1)] [(c,2,3),c∼N(26,2)] [(c,3,4),c∼N(38,4)]
7	[(c,3,3),c∼N(36,1)] [(c,3,4),c∼N(37,2)] [(c,3,3),c∼N(26,1)]	[(c,2,2),c∼N(36,3)] [(c,3,5),c∼N(53,4)] [(c,2,2),c∼N(25,2)]	[(c,3,3),c∼N(27,1)] [(c,3,3),c∼N(37,2)] [(c,2,2),c∼N(26,3)]	[(c,2,4),c∼N(20,2)] [(c,4,4),c∼N(43,1)] [(c,2,2),c∼N(20,1)]	[(c,3,3),c∼N(35,2)] [(c,5,4),c∼N(56,2)] [(c,3,4),c∼N(39,3)]
8	[(c,4,4),c∼N(39,2)] [(c,4,3),c∼N(41,2)] [(c,2,2),c∼N(20,2)]	[(c,2,2),c∼N(24,1)] [(c,1,1),c∼N(6,4)] [(c,1,1),c∼N(16,3)]	[(c,3,3),c∼N(30,3)] [(c,3,4),c∼N(53,1)] [(c,4,4),c∼N(40,2)]	[(c,4,4),c∼N(38,2)] [(c,2,2),c∼N(20,3)] [(c,4,3),c∼N(40,2)]	[(c,3,3),c∼N(34,3)] [(c,6,3),c∼N(32,2)] [(c,1,2),c∼N(43,1)]
9	[c,4,1),c∼N(38,2)] [(c,1,1),c∼N(10,4)] [(c,3,3),c∼N(31,2)]	[(c,4,4),c∼N(39,3)] [(c,3,4),c∼N(38,3)] [(c,3,3),c∼N(34,1)]	[(c,1,2),c∼N(4,2)] [(,3,3),c∼N(34,5)] [(c,3,3),c∼N(36,1)]	[(c,2,2),c∼N(23,2)] [(c,3,3),c∼N(33,3)] [(c,2,3),c∼N(28,1)]	[(c,1,3),c∼N(20,6)] [(c,3,3),c∼N(31,4)] [(c,2,2),c∼N(20,2)]
10	[(c,1,1),c∼N(15,2)] [(c,2,2),c∼N(25,1)] [(c,2,3),c∼N(25,2)]	[(c,3,3),c∼N(28,3)] [(c,2,2),c∼N(20,2)] [(c,3,3),c∼N(31,3)]	[(c,2,2),c∼N(28,3)] [(c,1,2),c∼N(18,3)] [(c,3,3),c∼N(28,2)]	[(c,2,2),c∼N(18,2)] [(c,3,2),c∼N(29,2)] [(c,2,2),c∼N(21,5)]	[(c,1,1),c∼N(9,2)] [(c,3,3),c∼N(32,2)] [(c,2,4),c∼N(20,4)]
	Fuzzy Random **Cost** Matrix(10 ×10) for RCSTSP With Three Conveyances				
i/j	6	7	8	9	10
1	[(c,1,1),c∼N(5,2)] [(c,2,2),c∼N(22,3)] [(c,1,2),c∼N(6,1)]	[(c,1,1),c∼N(15,1)) [(c,3,3),c∼N(35,3)] [(c,4,4),c∼N(46,6)]	[(c,2,3),c∼N(25,3)] [(c,3,4),c∼N(37,2)] [(c,1,2),c∼N(16,2)]	[(c,1,2),c∼N(39,3)] [(c,3,1),c∼N(26,4)] [(c,2,3),c∼N(41,2)]	[(c,2,2),c∼N(20,3)] [(c,1,3),c∼N(30,2)] [(c,4,5),c∼N(42,4)]
2	[(c,4,1),c∼N(39,2)] [(c,3,3),c∼N(30,1)] [(c,3,2),c∼N(30,1)]	[(c,1,2),c∼N(39,3)] [(c,3,3),c∼N(29,1)] [(c,5,6),c∼N(57,2)]	[(c,3,3),c∼N(30,2)] [(c,2,4),c∼N(41,2)] [(c,5,6),c∼N(58,1)]	[(c,1,2),c∼N(17,1)] [(c,3,8),c∼N(36,2)] [(c,2,2),c∼N(17,2)]	[(c,2,2),c∼N(23,2)] [(c,1,1),c∼N(13,2)] [(c,1,2),c∼N(17,3)]
3	[(c,3,3),c∼N(33,1)] [(c,3,3),c∼N(34,3)] [(c,4,5),c∼N(33,4)]	[(c,1,2),c∼N(17,3)] [(c,1,2),c∼N(11,4)] [(c,1,1),c∼N(5,1)]	[(c,3,3),c∼N(30,1)] [(c,3,3),c∼N(30,2)] [(c,2,2),c∼N(24,2)]	[(c,3,3),c∼N(28,2)] [(c,1,2),c∼N(18,1)] [(c,4,4),c∼N(40,4)]	[(c,3,3),c∼N(29,1)] [(c,2,2),c∼N(19,4)] [(c,3,3),c∼N(32,3)]
4	[(c,2,2),c∼N(23,2)] [(c,6,8),c∼N(15,1)] [(c,2,2),c∼N(25,3)]	[(c,1,2),c∼N(19,5)] [(c,1,3),c∼N(30,2)] [(c,3,3),c∼N(30,4)]	[(c,3,3),c∼N(33,1)] [(c,3,8),c∼N(32,5)] [(c,1,1),c∼N(16,3)]	[(c,2,3),c∼N(10,3)] [(c,2,4),c∼N(20,1)] [(c,3,5),c∼N(32,3)]	[(c,2,2),c∼N(24,4)] [(c,4,9),c∼N(47,2)] [(c,3,4),c∼N(37,2]
5	[(c,3,3),c∼N(29,1)] [(c,2,2),c∼N(20,2)] [(c,4,4),c∼N(40,2)]	[(c,5,6),c∼N(42,2)] [(c,1,1),c∼N(14,1)] [(c,2,2),c∼N(25,3)]	[(c,3,3),c∼N(27,1)] [(c,3,3),c∼N(30,4)] [(c,1,6),c∼N(12,1)]	[(c,1,2),c∼N(18,3)] [(c,1,1),c∼N(8,1)] [(c,8,9),c∼N(7,1)]	[(c,2,2),c∼N(26,1)] [(c,2,2),c∼N(25,1)] [(c,2,2),c∼N(25,1)]
6	∞	[(c,3,3),c∼N(31,1)] [(c,4,4),c∼N(40,1)] [(c,2,6),c∼N(23,1)]	[(c,4,2),c∼N(39,3)] [(c,3,3),c∼N(30,1)] [(c,2,2),c∼N(20,1)]	[(c,3,3),c∼N(30,1)] [(c,2,2),c∼N(22,1)] [(c,3,3),c∼N(35,1)]	[(c,3,3),c∼N(28,4)] [(c,4,4),c∼N(40,1)] [(c,3,3),c∼N(30,1)]
7	[(c,4,4),c∼N(38,1)] [(c,2,2),c∼N(20,1)] [(c,4,6),c∼N(43,1)]	∞	[(c,1,1),c∼N(7,1)] [(c,4,4),c∼N(40,1)] [(c,1,4),c∼N(11,1)]	[(c,3,3),c∼N(31,1)] [(c,3,3),c∼N(33,1)] [(c,3,7),c∼N(34,3)]	[(c,2,2),c∼N(19,1)] [(c,1,1),c∼N(13,1)] [(c,6,8),c∼N(25,2)]
8	[(c,2,2),c∼N(23,1)] [(c,1,1),c∼N(15,1)] [(c,1,2),c∼N(4,2)]	[(c,4,4),c∼N(39,1)] [(c,2,2),c∼N(19,1)] [(c,3,4),c∼N(41,4)]	∞	[(c,2,2),c∼N(20,1)] [(c,3,4),c∼N(52,1)] [(c,4,2),c∼N(23,1)]	[(c,3,3),c∼N(35,1)] [(c,6,8),c∼N(35,3)] [(c,4,4),c∼N(39,1)]
9	[(c,2,2),c∼N(22,1)] [(c,3,3),c∼N(31,1)] [(c,2,2),c∼N(23,1)]	[(c,1,3),c∼N(5,1)] [(c,3,3),c∼N(36,1)] [(c,3,4),c∼N(38,1)]	[(c,3,3),c∼N(30,1)] [(c,3,3),c∼N(32,1)] [(c,1,1),c∼N(11,2)]	∞	[(c,2,3),c∼N(27,1)] [(c,1,2),c∼N(18,1)] [(c,2,7),c∼N(24,3)]

10	[(c,3,3),c∼N(30,1)] [(c,1,1),c∼N(10,1)] [(c,3,3),c∼N(33,1)]	[(c,3,39),c∼N(3,1)] [(c,8,9),c∼N(26,1)] [(c,3,4),c∼N(38,1)]	[(c,4,3),c∼N(40,2)] [(c,4,4),c∼N(41,3)] [(c,3,3),c∼N(30,1)]	[(c,3,3),c∼N(29,1)] [(c,5,5),c∼N(51,5)] [(c,3,3),c∼N(30,1)]	∞
	Fuzzy Random **Time** Matrix(10×10) for FRCSTSP With Three Conveyances				
i/j	1	2	3	4	5
2 1	∞	[(c,2,4),c∼ N(25,1)] [(c,2,3),c∼N(12,2)] [(c,2,2),c∼N(16,2)]	[(c,2,3),c∼N(27,2)] [(c,4,2),c∼N(18,2)] [(c,3,3),c∼N(16,2)]	[(c,2,2),c∼N(26,3)] [(c,3,4),c∼N(21,2)] [(c,1,6),c∼N(13,2)]	[(c,2,3),c∼N(19,2)] [(c,1,2),c∼N(28,2)] [(c,2,3),c∼N(10,4)]
2	[(c,3,3),c∼N(11,1)] [(c,2,2),c∼N(32,3)] [(c,1,1),c∼N(34,4)]	∞	[(c,4,4),c∼N(10,2)] [(c,2,2),c∼N(48,2)] [(c,2,3),c∼N(17,1)]	[(c,1,1),c∼N(36,2)] [(c,2,3),c∼N(18,2)] [(c,1,1),c∼N(36,3)]	[(c,3,3),c∼N(12,2)] [(c,2,2),c∼N(25,2)] [(c,3,4),c∼N(18,1)]
3	[(c,3,3),c∼N(16,2)] [(c,3,3),c∼N(16,3)] [(c,2,3),c∼N(18,2)]	[(c,1,2),c∼N(26,3)] [(c,5,6),c∼N(4,1)] [(c,3,3),c∼N(21,2)]	∞	[(c,1,1),c∼N(30,2)] [(c,2,2),c∼N(34,1)] [(c,1,1),c∼N(32,1)]	[(c,4,4)c∼N(10,1)] [(c,2,2),c∼N(36,3)] [(c,4,4),c∼N(5,5]
4	[(c,2,3),c∼N(16,1)] [(c,2,2),c∼N(38,2)] [(c,1,1),c∼N(49,2)]	[(c,1,1),c∼N(39,2)] [(c,2,2),c∼N(39,1)] [(c,1,1),c∼N(32,4)]	[(c,1,2),c∼N(36,4)] [(c,5,4),c∼N(47,2)] [(c,2,3),c∼N(27,2)]	∞	[(c,3,3),c∼N(39,3)] [(c,1,2),c∼N(37,3)] [(c,2,2),c∼N(23,4)]
5	[(c,2,1,c∼N(36,1)] [(c,1,1),c∼N(34,1)] [(c,1,1),c∼N(36,2)]	[(c,3,4),c∼N(11,3)] [(c,2,2),c∼N(22,2)] [(c,3,3),c∼N(12,3)]	[(c,3,3),c∼N(14,2)] [(c,6,7),c∼N(16,2)] [(c,3,3),c∼N(13,2)]	[(c,2,2),c∼N(37,2)] [(c,1,1),c∼N(32,2)] [(c,4,4),c∼N(18,3)]	∞
6	[(c,1,1),c∼N(35,1)] [(c,1,4),c∼N(46,2)] [(c,2,3),c∼N(46,2)]	[(c,2,3),c∼N(26,2)] [(c,2,2),c∼N(30,1)] [(c,2,3),c∼N(26,2)]	[(c,1,1),c∼N(45,5)] [(c,6,7),c∼N(16,2)] [(c,2,3),c∼N(26,3)]	[(c,2,3),c∼N(46,2)])] [c,1,1),c∼N(33,3)] [(c,1,1),c∼N(30,3)]	[(c,2,2),c∼N(26,1)] [(c,2,3),c∼N(26,2)] [(c,3,4),c∼N(18,4)]
7	[(c,3,3),c∼N(16,1)] [(c,3,4),c∼N(17,2)] [(c,3,3),c∼N(26,1)]	[(c,2,2),c∼N(16,3)] [(c,3,5),c∼N(13,4)] [(c,2,2),c∼N(25,2)]	[(c,3,3),c∼N(27,1)] [(c,3,3),c∼N(17,2)] [(c,2,2),c∼N(26,3)]	[(c,2,4),c∼N(20,2)] [(c,3,4),c∼N(13,1)] [(c,2,2),c∼N(20,1)]	[(c,3,3),c∼N(15,2)] [(c,5,4),c∼N(6,2)] [(c,3,4),c∼N(19,3)]
8	[(c,4,4),c∼N(19,2)] [(c,4,3),c∼N(11,2)] [(c,2,2),c∼N(30,2)]	[(c,2,2),c∼N(24,1)] [(c,1,1),c∼N(46,4)] [(c,1,1),c∼N(36,3)]	[(c,3,3),c∼N(10,3)] [(c,3,4),c∼N(13,1)] [(c,4,4),c∼N(10,2)]	[(c,4,4),c∼N(18,2)] [(c,2,2),c∼N(20,3)] [(c,4,3),c∼N(10,2)]	[(c,3,3),c∼N(14,3)] [(c,6,3),c∼N(12,2)] [(c,1,2),c∼N(13,1)]
9	[c,4,1),c∼N(18,2)] [(c,1,1),c∼N(30,4)] [(c,3,3),c∼N(11,2)]	[(c,4,4),c∼N(19,3)] [(c,3,4),c∼N(18,3)] [(c,3,3),c∼N(14,1)]	[(c,1,2),c∼N(42,2)] [(,3,3),c∼N(14,5)] [(c,3,3),c∼N(16,1)]	[(c,2,2),c∼N(23,2)] [(c,3,3),c∼N(13,3)] [(c,2,3),c∼N(18,1)]	[(c,1,3),c∼N(20,6)] [(c,3,3),c∼N(11,4)] [(c,2,2),c∼N(30,2)]
10	[(c,1,1),c∼N(35,2)] [(c,2,2),c∼N(35,1)] [(c,2,3),c∼N(15,2)]	[(c,3,3),c∼N(28,3)] [(c,2,2),c∼N(30,2)] [(c,3,3),c∼N(11,3)]	[(c,2,2),c∼N(28,3)] [(c,1,2),c∼N(38,3)] [(c,3,3),c∼N(23,2)]	[(c,2,2),c∼N(38,2)] [(c,3,2),c∼N(29,2)] [(c,2,2),c∼N(19,5)]	[(c,1,1),c∼N(49,2)] [(c,3,3),c∼N(12,2)] [(c,2,4),c∼N(30,4)]
	Fuzzy Random **Time** Matrix(10 ×10) for RCMOSTSP With Three Conveyances				
i/j	6	7	8	9	10
1	[(c,1,1),c∼N(45,2)] [(c,2,2),c∼N(32,3)] [(c,1,2),c∼N(46,1)]	[(c,1,1),c∼N(35,1)) [(c,3,3),c∼N(15,3)] [(c,4,4),c∼N(6,6)]	[(c,2,3),c∼N(25,3)] [(c,3,4),c∼N(17,2)] [(c,1,2),c∼N(36,2)]	[(c,1,2),c∼N(19,3)] [(c,3,1),c∼N(16,4)] [(c,2,3),c∼N(11,2)]	[(c,2,2),c∼N(30,3)] [(c,1,3),c∼N(20,2)] [(c,4,5).c∼N(12,4)]
2	[(c,4,1),c∼N(19,2)] [(c,3,3),c∼N(20,1)] [(c,3,2),c∼N(20,3]	[(c,1,2),c∼N(19,3)] [(c,3,3),c∼N(19,1)] [(c,5,6),c∼N(7,2)]	[(c,3,3),c∼N(20,2)] [(c,2,4),c∼N(11,2)] [(c,5,6),c∼N(8,1)]	[(c,1,2),c∼N(37,1)] [(c,3,8),c∼N(16,2)] [(c,2,2),c∼N(37,2)]	[(c,2,2),c∼N(13,2)] [(c,1,1),c∼N(33,2)] [(c,1,2),c∼N(17,3)]
3	[(c,3,3),c∼N(13,1)] [(c,3,3),c∼N(14,3)] [(c,4,5),c∼N(13,4)]	[(c,1,2),c∼N(17,3)] [(c,1,2),c∼N(31,4)] [(c,1,1),c∼N(35,1)]	[(c,3,3),c∼N(30,1)] [(c,3,3),c∼N(10,2)] [(c,2,2),c∼N(14,2)]	[(c,3,3),c∼N(38,2)] [(c,1,2),c∼N(38,1)] [(c,4,4),c∼N(20,4)]	[(c,3,3),c∼N(39,1)] [(c,2,2),c∼N(39,4)] [(c,3,3),c∼N(12,3)]
4	[(c,2,2),c∼N(13,2)] [(c,6,8),c∼N(35,1)] [(c,2,2),c∼N(15,3)]	[(c,1,2),c∼N(39,5)] [(c,1,3),c∼N(20,2)] [(c,3,3),c∼N(20,4)]	[(c,3,3),c∼N(13,1)] [(c,3,8),c∼N(12,5)] [(c,1,1),c∼N(36,3)]	[(c,2,3),c∼N(30,3)] [(c,2,4),c∼N(30,1)] [(c,3,5),c∼N(12,3)]	[(c,2,2),c∼N(24,4)] [(c,4,9),c∼N(7,2)] [(c,3,4),c∼N(17,2]
5	[(c,3,3),c∼N(19,1)] [(c,2,2),c∼N(30,2)] [(c,4,4),c∼N(10,2)]	[(c,5,6),c∼N(2,1)] [(c,1,1),c∼N(34,1)] [(c,2,2),c∼N(25,3)]	[(c,3,3),c∼N(17,1)] [(c,3,3),c∼N(20,4)] [(c,1,6),c∼N(41,1)]	[(c,1,2),c∼N(38,3)] [(c,1,1),c∼N(38,1)] [(c,8,9),c∼N(37,1)]	[(c,2,2),c∼N(16,1)] [(c,2,2),c∼N(25,1)] [(c,2,2),c∼N(25,1)]
6	∞	[(c,3,3),c∼N(11,1)] [(c,4,4),c∼N(20,1)] [(c,2,6),c∼N(13,1)]	[(c,4,2),c∼N(19,3)] [(c,3,3),c∼N(20,1)] [(c,2,2),c∼N(10,1)]	[(c,3,3),c∼N(10,1)] [(c,2,2),c∼N(22,1)] [(c,3,3),c∼N(15,1)]	[(c,3,3),c∼N(18,4)] [(c,4,4),c∼N(10,1)] [(c,3,3),c∼N(10,1)]
7	[(c,4,4),c∼N(18,1)] [(c,2,2),c∼N(30,1)] [(c,4,6),c∼N(13,1)]	∞	[(c,1,1),c∼N(37,1)] [(c,4,4),c∼N(10,1)] [(c,1,4),c∼N(31,1)]	[(c,3,3),c∼N(11,1)] [(c,3,3),c∼N(13,1)] [(c,3,7),c∼N(14,3)]	[(c,2,2),c∼N(39,1)] [(c,1,1),c∼N(33,1)] [(c,6,8),c∼N(15,2)]
8	[(c,2,2),c∼N(13,1)] [(c,1,1),c∼N(35,1)] [(c,1,2),c∼N(43,2)]	[(c,4,4),c∼N(19,1)] [(c,2,2),c∼N(39,1)] [(c,3,4),c∼N(5,4)]	∞	[(c,2,2),c∼N(30,1)] [(c,3,4),c∼N(2,1)] [(c,4,2),c∼N(23,1)]	[(c,3,3),c∼N(15,1)] [(c,6,8),c∼N(15,3)] [(c,4,4),c∼N(19,1)]
9	[(c,2,2),c∼N(19,1)] [(c,3,3),c∼N(11,1)] [(c,2,2),c∼N(23,1)]	[(c,1,3),c∼N(35,1)] [(c,3,3),c∼N(16,1)] [(c,3,4),c∼N(18,1)]	[(c,3,3),c∼N(10,1)] [(c,3,3),c∼N(12,1)] [(c,1,1),c∼N(31,2)]	∞	[(c,2,3),c∼N(17,1)] [(c,1,2),c∼N(38,1)] [(c,2,7),c∼N(14,3)]

10	[(c,3,3),c∼N(10,1)] [(c,1,1),c∼N(40,1)] [(c,3,3),c∼N(13,1)]	[(c,3,3),c∼N(43,1)] [(c,8,9),c∼N(16,1)] [(c,3,4),c∼N(18,1)]	[(c,4,3),c∼N(10,2)] [(c,4,4),c∼N(11,3)] [(c,3,3),c∼N(10,1)]	[(c,3,3),c∼N(19,1)] [(c,5,5),c∼N(5,4)] [(c,3,3),c∼N(10,1)]	∞
	Fuzzy Random **risk/discomfort** Matrix(10×10) for FRCMOSTSP With Three Conveyances				
i/j	1	2	3	4	5
1	∞	(.69, .05, .01) (.36, .03, .03) (.26, .02, .02)	(.72, .07, .08) (.38, .03, .04) (.26, .03, .03)	(.73, .01, .02), (.31, .03, .03) (.33, .03, .03)	(.61, .03, .01) (.2, .02, .02) (.6, .06, .06)
2	(.34, .03, .03) (.22, .02, .02) (.14, .01, .01)	∞	(.4, .04, .04) (.18, .02, .02) (.27, .03, .02)	(.16, .01, .01) (.28, .02, .03) (.06, .01, .02)	(.32, .03, .03) (.25, .02, .02) (.34, .03, .03)
3	(.36, .03, .03) (.29, .03, .03) (.28, .02, .03)	(.16, .01, .02) (.54, .08, .01) (.31, .03, .03)	∞	(.1, .02, .03) (.24, .02, .02) (.12, .01, .01)	(.4, .04, .03) (.23, .02, .02) (.45, .04, .04)
4	(.27, .02, .03) (.18, .02, .02) (.9, .01, .01)	(.09, .01, .01) (.19, .02, .02) (.12, .01, .01)	(.16, .01, .02) (.7, .09, .01) (.27, .02, .03)	∞	(.29, .03, .03) (.17, .01, .02) (.23, .02, .02)
5	(.16, .01, .01) (.14, .01, .01) (.6, .01, .02)	(.41, .04, .04) (.21, .02, .02) (.32, .03, .02)	(.34, .03, .03) (.35, .03, .03) (.33, .03, .03)	(.17, .02, .02) (.12, .01, .01) (.4, .04, .03)	∞
6	(.13, .05, .01) (.5, .06, .08) (.5, .07, .03)	(.26, .02, .03) (.2, .02, .02) (.27, .02, .03)	(.4, .04, .03) (.25, .02, .02) (.27, .02, .03)	(.6, .08, .09) (.7, .09, .01) (.1, .01, .01)	(.26, .02, .02) (.26, .02, .03) (.38, .03, .01)
7	(.36, .03, .03) (.37, .03, .04) (.28, .03,.03)	(.23, .02, .02) (.53, .05, .05) (.25, .02, .02)	(.27, .03, .03) (.37, .03, .03) (.24, .02, .27)	(.21, .02, .02) (.4, .04, .04) (.23, .02, .021)	(.35, .03, .03) (.56, .05, .01) (.37, .03,.04)
8	(.39, .04, .04) (.41, .04, .04) (.2, .02, .02)	(.24, .02, .02) (.5, .01, .02) (.16, .01, .01)	(.3, .03, .03) (.52, .05, .05) (.43, .04, .04)	(.38, .04, .04) (.19, .02, .02) (.4, .04, .04)	(.34, .03, .03) (.34, .03, .031) (.46, .04, .04)
9	(.38, .04, .04) (.1, .01, .01) (.31, .03, .03)	(.39, .04, .04) (.38, .03, .04) (.34, .03, .03)	(.4, .01, .02) (.34, .03, .03) (.36, .03, .03)	(.23, .025, .02) (.33, .03, .03) (.28, .02, .03)	(.2, .021, .023) (.31, .03, .03) (.2, .02, .02)
10	(.15, .01, .01) (.25, .02, .02) (.25, .02, .03)	(.28, .03, .03) (.2, .02, .02) (.31, .03, .03)	(.26, .02, .02) (.18, .01, .02) (.28, .03, .03)	(.18, .02, .02) (.29, .03, .03) (.21, .02, .02)	(.9, .01, .01) (.32, .03, .03) (.2, .01, .01)
	Fuzzy Random **risk/discomfort** Matrix(10 ×10) for FRCMOSTSP With Three Conveyances				
i/j	6	7	8	9	10
1	(.5, .07, .01) (.22, .02, .02) (.6, .08, .09)	(.15, .01, .01) (.35, .03, .03) (.46, .07, .08)	(.25, .02, .02) (.37, .03, .04) (.16, .01, .02)	(.39, .04, .05) (.26, .03, .03) (.41, .04, .04)	(.2, .02, .03) (.3, .03, .03) (.42, .01, .04)
2	(.39, .04, .04) (.3, .03, .03) (.21, .02, .02)	(.39, .04, .04) (.29, .03, .03) (.57, .05, .06)	(.3, .03, .03) (.41, .04, .04) (.58, .05, .06)	(.17, .01, .02) (.36, .03, .03) (.17, .02, .02)	(.23, .02, .02) (.13, .01, .01) (.17, .01, .02)
3	(.33, .03, .03) (.34, .03, .03) (.33, .03, .03)	(.17, .01, .02) (.11, .01, .01) (.05, .01, .01)	(.3, .03, .04) (.3, .03, .03) (.24, .02, .02)	(.28, .03, .03) (.18, .01, .02) (.4, .04, .04)	(.29, .03, .03) (.19, .02, .02) (.32, .03, .03)
4	(.23, .02, .02) (.15, .01, .01) (.25, .02, .02)	(.19, .02, .02) [(.3, .03, .03) (.3, .03, .03)	(.33, .03, .03) (.32, .03, .03) (.16, .01, .01)	(.1, .01, .013) (.2, .02, .02) (.32, .03, .03)	(.24, .02, .029) (.47, .04, .04) (.37, .03, .04)
5	(.29, .03, .03) (.2, .02, .02) (.4, .04, .04)	(.42, .04, .04) (.14, .01, .02) (.25, .02, .02)	(.27, .03, .03) (.3, .03, .02) (.12, .02, .06)	(.18, .01, .02) (.08, .01, .01) (.07, .01, .01)	(.26, .02, .02) (.25, .02, .02) (.25, .02, .02)
6	∞	(.31, .03, .04) (.4, .04, .04) (.23, .02, .02)	(.39, .04, .04) (.3, .03, .031) (.2, .02, .02)	(.3, .03, .03) (.22, .02, .02) (.35, .03, .03)	(.28, .03, .03) (.4, .04, .04) (.3, .03, .03)
7	(.38, .04, .04) (.2, .02, .02) (.43, .04, .04)	∞	(.07, .001, .001) (.4, .04, .04) (.11, .01, .01)	(.31, .03, .03) (.33, .03, .03) (.34, .03, .03)	(.19, .02, .02) (.13, .01, .01) (.25, .02, .02)
8	(.23, .02, .02) (.15, .01, .01) (.04, .01, .01)	(.39, .04, .04) (.19, .02, .02) (.41, .03, .03)	∞	(.2, .02, .02) (.52, .05, .05) (.23, .02, .02)	(.35, .03, .03) (.35, .03, .03) (.39, .04, .04)
9	(.22, .02, .02) (.31, .03, .03) [(.23, .05, .06)	(.05, .07, .08) (.36, .03, .03) (.38, .03, .04)	(.3, .03, .03) (.32, .03, .03) (.11, .01, .01)	∞	(.27, .02, .03) (.18, .01, .02) (.24, .02, .02)
10	(.3, .03, .03) (.1, .012, .01) (.33, .02, .01)	(.35, .03, .03) (.26, .02,.03) (.38, .03, .04)	(.4, .04, .04) (.41, .04, .04) (.3, .01, .01)	(.29, .03, .03) (.51, .05, .05) (.3, .03, .03)	∞

Table 6.16: Results of FRCSTSP (Model 6.1D)

δ	θ	Algorithm	DM	Path(Vehicle)	Costs & Times	R_{max}
		iMOGA	PDM	4(2)-10(3)-2(3)-9(3)-7(3)-8(1)-6(3)-1(1)-5(2)-3(1)	[148.56, 102.43]	
			ODM	4(2)-10(3)-2(3)-9(3)-7(3)-8(1)-6(3)-1(1)-5(2)-3(1)	[140.13, 113.86]	
		iMOGA	PDM	6(3)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-5(2)-1(1)-7(2)	[151.21, 99.32]	8.5
			ODM	6(3)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-5(2)-113)-7(2)	[147.18, 104.51]	
0.9	0.9	iMOGA	PDM	1(3)-10(1)-9(2)-5(1)-7(2)-3(1)-4(2)-2(3)-8(1)-6(1)	[166.25, 94.73]	
			ODM	1(3)-10(1)-9(2)-5(1)-7(2)-3(1)-4(2)-2(3)-8(1)-6(1)	[151.31, 98.31]	6.75
		MOGA	PDM	6(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-5(2)-1(3)-7(3)	[169.21, 118.62]	
			ODM	6(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-5(2)-1(3)-7(3)	[162.45, 115.75]	6.0
0.96	0.7	iMOGA	PDM	3(2)-8(1)-7(1)-1(1)-10(2)-5(3)-2(3)-4(1)-6(2)-9(3)	[155.76, 124.84]	
			ODM	4(1)-8(3)-7(1)-1(1)-10(2)-5(3)-2(3)-3(1)-6(2)-9(3)	[142.18, 106.57]	6.75
0.79	0.9	iMOGA	PDM	5(3)-7(1)-8(1)-6(2)-1(2)-4(1)-9(2)-2(1)-10(1)-3(1)	[161.34, 97.43]	
			ODM	4(3)-7(1)-8(1)-6(2)-1(2)-5(1)-9(2)-2(1)-10(1)-3(1)	[164.13, 95.38]	6.5
0.85	0.75	iMOGA	PDM	1(3)-4(3)-2(3)-10(2)-5(3)-9(3)-8(1)-6(1)-3(3)-7(3)	[168.45, 100.37]	
			ODM	1(1)-4(3)-2(3)-10(2)-5(3)-9(3)-8(1)-6(1)-1(3)-7(2)	[146.93, 107.64]	6.0

Here we took permissible probability levels $\gamma = \eta = 0.9$, We set L(x)=1-x, left and right spreads are taken from given data set in the Table 6.15. DM means decision maker and optimistic DM (ODM), pessimistic DM (PDM). With these data, the FRCMOSTSP model is solved by iMOGA for different values of δ and θ and the optimum results are presented in Table 6.16.

Model 6.1F: CMOSTSP with Risk/Discomfort Constraint in Bi-random Environment (BRCMOSTSP):

Here we took the costs, times and risk/discomfort factor in bi-random values for the CMOSTSP. Also we consider three types of conveyances. We set two fold randomness of the given values in the form of mean and variances. The bi-random costs, times matrices for the CMOSTSP and corresponding bi-random risk/discomfort matrix are given in Table 6.17. For these data, Pareto optimum results obtained by iMOGA with different values of α and β are presented in Table 6.18.

Cpu Time Scale for BRCMOSTSP

Here we study the cpu time in seconds for different sizes of problems from n=10 to 50 in only bi-random environment. The parameters are choosen only for α=β=0.95, and the mean with SD are considered for iMOGA. The results are considered for 30 runs of each instances and given in Table 6.19.

Table 6.17: Input data: BRCMOSTSP (Model 6.1F)

	Bi-random **Cost** Matrix(10 ×10) for BRCSTSP With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	(32,1.1) (37,1.21) (28,1.02)	(19,.9) (39,1.07) (30,1.11)	(21,1.02) (33,1.15) (35,1.17)	(30,1.01) (21,.98) (62,1.2)	(7,1.23) (23,1.02) (8,1.19)	(16,1.11) (36,1.03) (47,.97)	(28,1.04) (39,1.12) (19,1.18)	(41,1.12) (31,1.13) (42,1.03)	(21,1.02) (31,1.1) (43,1.01)
2	(35,1.12) (26,1.18) (17,1.13)	∞	(41,1.03) (21,1.17) (32,1.32)	(18,1.11) (29,1.12) (10,1.03)	(35,1.07) (26,1.2) (37,1.2)	(40,1.02) (31,1.2) (23,1.31)	(40,1.13) (30,1.15) (59,1.14)	(33,1.03) (42,1.21) (59,1.16)	(19,1.2) (37,1.13) (20,1.3)	(24,1.19) (16,1.12) (18,1.03)
3	(38,1.29) (30,1.13) (29,1.15)	(17,1.21) (58,1.43) (34,1.32)	∞	(12,1.25) (25,1.21) (14,1.11)	(42,1.23) (25,1.23) (46,1.24)	(35,1.21) (36,1.4) (34,1.12)	(19,1.13) (11,1.1) (8,1.3)	(32,1.1) (33,1.21) (25,1.16)	(30,1.11) (19,1.22) (41,1.41)	(30,1.21) (22,1.16) (33,1.33)
4	(28,1.14) (20,1.1) (10,1.31)	(10,1.2) (22,1.32) (14,1.2)	(18,1.21) (9,1.4) (29,1.31)	∞	(30,1.13) (19,1.15) (24,1.21)	(25,1.23) (16,1.12) (27,1.13)	(21,1.4) (31,1.4) (33,1.19)	(35,1.3) (36,1.2) (17,1.23)	(12,1.21) (23,1.31) (34,1.2)	(27,1.6) (48,1.2) (39,1.28)
5	(18,1.31) (15,1.2) (8,1.2)	(42,1.2) (23,1.31) (34,1.21)	(35,1.12) (36,1.41) (38,1.34)	(20,1.31) (13,1.31) (43,1.15)	∞	(30,1.21) (21,1.36) (41,1.5)	(45,1.16) (16,1.02) (27,1.31)	(30,1.24) (31,1.27) (13,1.02)	(19,1.34) (10,1.01) (8,1.04)	(28,1.42) (26,1.47) (27,1.21)
6	(15,1.31) (6,1.65) (7,1.27)	(29,1.15) (21,1.75) (29,1.15)	(4,1.32) (26,1.62) (28,1.72)	(8,1.41) (9,1.7) (12,1.04)	(28,1.61) (29,1.21) (39,1.37)	∞	(33,1.26) (42,1.31) (24,1.32)	(40,1.53) (31,1.32) (22,1.65)	(32,1.21) (23,1.34) (35,1.21)	(30,1.54) (41,1.52) (32,1.52)
7	(37,1.6) (39,1.43) (30,1.32)	(25,1.21) (53,1.6) (26,1.54)	(30,1.5) (38,1.71) (26,1.56)	(22,1.61) (43,1.31) (24,1.76)	(37,1.98) (58,1.21) (40,1.21)	(40,1.76) (21,1.65) (45,1.61)	∞	(10,1.31) (43,1.65) (13,1.21)	(33,1.54) (34,1.71) (36,1.37)	(20,1.04) (15,1.2) (26,1.6)
8	(41,1.27) (42,1.43) (23,1.15)	(26,1.43) (6,1.32) (17,1.23)	(32,1.34) (53,1.43) (45,1.17)	(40,1.21) (21,1.21) (42,1.31)	(35,1.53) (36,1.21) (47,1.32)	(25,1.53) (16,1.06) (5,1.03)	(40,1.27) (21,1.03) (43,1.04)	∞	(22,1.31) (53,1.62) (24,1.02)	(37,1.76) (36,1.78) (40,1.02)
9	(40,1.72) (11,1.21) (32,1.02)	(41,1.56) (39,1.56) (36,1.42)	(6,1.24) (36,1.42) (37,1.76)	(25,1.71) (34,1.57) (29,1.08)	(21,1.04) (32,1.3) (21,1.02)	(23,1.32) (33,1.06) (25,1.03)	(7,1.01) (38,1.02) (39,1.21)	(32,1.32) (33,1.76) (13,1.52)	∞	(28,1.41) (19,1.32) (26,1.72)
10	(17,1.51) (26,1.01) (29,1.21)	(30,1.31) (21,1.04) (32,1.92)	(28,1.15) (19,1.21) (30,1.72)	(20,1.72) (31,1.02) (22,1.51)	(11,1.82) (33,1.27) (22,1.19)	(32,1.52) (12,1.18) (34,1.17)	(38,1.02) (28,1.13) (39,1.16)	(41,1.62) (42,1.81) (33,1.21)	(31,1.52) (52,1.37) (32,1.15)	∞

	Bi-random **Time** Matrix(10×10) for BRCSTSP With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	(12,1.1) (17,1.21) (18,1.02)	(39,.9) (19,1.07) (10,1.11)	(21,1.02) (13,1.15) (15,1.17)	(10,1.01) (31,.98) (2,1.2)	(37,1.23) (33,1.02) (38,1.19)	(26,1.11) (16,1.03) (7,.97)	(18,1.04) (19,1.12) (29,1.18)	(11,1.12) (11,1.13) (2,1.03)	(13,1.02) (11,1.1) (3,1.01)
2	(15,1.12) (16,1.18) (37,1.13)	∞	(11,1.03) (31,1.17) (12,1.32)	(28,1.11) (29,1.12) (30,1.03)	(15,1.07) (26,1.2) (17,1.2)	(10,1.02) (11,1.2) (13,1.31)	(10,1.13) (10,1.15) (9,1.14)	(13,1.03) (12,1.21) (9,1.16)	(29,1.2) (17,1.13) (30,1.3)	(14,1.19) (36,1.12) (28,1.03)
3	(18,1.29) (10,1.13) (29,1.15)	(27,1.21) (5,1.43) (14,1.32)	∞	(22,1.25) (15,1.21) (34,1.11)	(12,1.23) (15,1.23) (6,1.24)	(15,1.21) (16,1.4) (14,1.12)	(29,1.13) (31,1.1) (38,1.3)	(12,1.1) (13,1.21) (15,1.16)	(20,1.11) (39,1.22) (11,1.41)	(20,1.21) (22,1.16) (13,1.33)
4	(18,1.14) (30,1.1) (20,1.31)	(30,1.2) (32,1.32) (34,1.2)	(28,1.21) (39,1.4) (19,1.31)	∞	(10,1.13) (39,1.15) (14,1.21)	(15,1.23) (36,1.12) (17,1.13)	(11,1.4) (11,1.4) (13,1.19)	(15,1.3) (16,1.2) (27,1.23)	(32,1.21) (23,1.31) (14,1.2)	(17,1.6) (8,1.2) (19,1.28)
5	(38,1.31) (35,1.2) (28,1.2)	(4,1.2) (32,1.31) (14,1.21)	(15,1.12) (16,1.41) (18,1.34)	(30,1.31) (23,1.31) (3,1.15)	∞	(10,1.21) (31,1.36) (4,1.5)	(15,1.16) (36,1.02) (17,1.31)	(10,1.24) (11,1.27) (23,1.02)	(29,1.34) (20,1.01) (38,1.04)	(28,1.42) (16,1.47) (17,1.21)
6	(25,1.31) (36,1.65) (37,1.27)	(19,1.15) (21,1.75) (19,1.15)	(44,1.32) (16,1.62) (18,1.72)	(48,1.41) (39,1.7) (32,1.04)	(18,1.61) (19,1.21) (19,1.37)	∞	(13,1.26) (12,1.31) (14,1.32)	(10,1.53) (11,1.32) (12,1.65)	(12,1.21) (13,1.34) (15,1.21)	(10,1.54) (11,1.52) (12,1.52)
7	(17,1.6) (19,1.43) (10,1.32)	(15,1.21) (3,1.6) (16,1.54)	(10,1.5) (18,1.71) (12,1.56)	(22,1.61) (3,1.31) (14,1.76)	(17,1.98) (8,1.21) (10,1.21)	(10,1.76) (11,1.65) (5,1.61)	∞	(30,1.31) (13,1.65) (33,1.21)	(13,1.54) (14,1.71) (16,1.37)	(10,1.04) (25,1.2) (16,1.6)
8	(11,1.27) (12,1.43) (23,1.15)	(16,1.43) (36,1.32) (27,1.23)	(12,1.34) (5,1.43) (5,1.17)	(10,1.21) (21,1.21) (12,1.31)	(15,1.53) (16,1.21) (7,1.32)	(25,1.53) (26,1.06) (52,1.03)	(10,1.27) (11,1.03) (3,1.04)	∞	(12,1.31) (5,1.62) (24,1.02)	(17,1.76) (16,1.78) (10,1.02)
9	(10,1.72) (31,1.21) (12,1.02)	(11,1.56) (19,1.56) (16,1.42)	(62,1.24) (16,1.42) (17,1.76)	(25,1.71) (14,1.57) (19,1.08)	(21,1.04) (12,1.3) (21,1.02)	(23,1.32) (13,1.06) (25,1.03)	(37,1.01) (18,1.02) (19,1.21)	(12,1.32) (13,1.76) (23,1.52)	∞	(28,1.41) (39,1.32) (16,1.72)
10	(27,1.51) (16,1.01) (19,1.21)	(10,1.31) (11,1.04) (12,1.92)	(18,1.15) (39,1.21) (10,1.72)	(21,1.72) (13,1.02) (12,1.51)	(31,1.82) (13,1.27) (12,1.19)	(12,1.52) (32,1.18) (14,1.17)	(18,1.02) (18,1.13) (19,1.16)	(11,1.62) (2,1.81) (13,1.21)	(11,1.52) (5,1.37) (12,1.15)	∞

	Bi-random **Risk/Discomfort** Matrix(10×10) for BRCSTSP With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	(.32,1.1) (.37,1.21) (.28,1.02)	(.19,.9) (.39,1.07) (.30,1.11)	(.21,1.02) (.33,1.15) (.35,1.17)	(.30,1.01) (.21,.98) (.62,1.2)	(.07,1.23) (.23,1.02) (.08,1.19)	(.16,1.11) (.36,1.03) (.47,.97)	(.28,1.04) (39,1.12) (.19,1.18)	(.41,1.12) (.31,1.13) (.42,1.03)	(.21,1.02) (.31,1.1) (43,1.01)
2	(.35,1.12) (.26,1.18) (.17,1.13)	∞	(.41,1.03) (.21,1.17) (.32,1.32)	(.18,1.11) (.29,1.12) (.10,1.03)	(.35,1.07) (.26,1.2) (.37,1.2)	(.40,1.02) (.31,1.2) (.23,1.31)	(.40,1.13) (.30,1.15) (.59,1.14)	(.33,1.03) (.42,1.21) (.59,1.16)	(.19,1.2) (.37,1.13) (.20,1.3)	(.24,1.19) (.16,1.12) (.18,1.03)
3	(.38,1.29) (.30,1.13) (.29,1.15)	(.17,1.21) (.58,1.43) (.34,1.32)	∞	(.12,1.25) (.25,1.21) (.14,1.11)	(.42,1.23) (.25,1.23) (.46,1.24)	(.35,1.21) (.36,1.4) (.34,1.12)	(.19,1.13) (.11,1.1) (.08,1.3)	(32,1.1) (.33,1.21) (.25,1.16)	(.30,1.11) (.19,1.22) (.41,1.41)	(.30,1.21) (.22,1.16) (.33,1.33)
4	(.28,1.14) (.20,1.1) (.10,1.31)	(.10,1.2) (.22,1.32) (.14,1.2)	(.18,1.21) (.09,1.4) (.29,1.31)	∞	(.30,1.13) (.19,1.15) (.24,1.21)	(.25,1.23) (.16,1.12) (.27,1.13)	(.21,1.4) (.31,1.4) (.33,1.19)	(.35,1.3) (.36,1.2) (.17,1.23)	(.12,1.21) (.23,1.31) (.34,1.2)	(.27,1.6) (.48,1.2) (.39,1.28)
5	(.18,1.31) (.15,1.2) (.08,1.2)	(.42,1.2) (.23,1.31) (.34,1.21)	(.35,1.12) (.36,1.41) (.38,1.34)	(.20,1.31) (.13,1.31) (.43,1.15)	∞	(.30,1.21) (.21,1.36) (.41,1.5)	(.45,1.16) (.16,1.02) (.27,1.31)	(.30,1.24) (.31,1.27) (.13,1.02)	(.19,1.34) (.10,1.01) (.08,1.04)	(.28,1.42) (.26,1.47) (.27,1.21)
6	(.15,1.31) (.06,1.65) (.07,1.27)	(.29,1.15) (.21,1.75) (.29,1.15)	(.04,1.32) (.26,1.62) (.28,1.72)	(.08,1.41) (.09,1.7) (.12,1.04)	(.28,1.61) (.29,1.21) (.39,1.37)	∞	(.33,1.26) (.42,1.31) (.24,1.32)	(40,1.53) (.31,1.32) (.22,1.65)	(.32,1.21) (.23,1.34) (.35,1.21)	(.30,1.54) (.41,1.52) (.32,1.52)
7	(.37,1.6) (.39,1.43) (.30,1.32)	(.25,1.21) (.53,1.6) (.26,1.54)	(.30,1.5) (.38,1.71) (.26,1.56)	(.22,1.61) (.43,1.31) (.24,1.76)	(.37,1.98) (.58,1.21) (.40,1.21)	(.40,1.76) (.21,1.65) (.45,1.61)	∞	(.10,1.31) (.43,1.65) (.13,1.21)	(.33,1.54) (.34,1.71) (.36,1.37)	(.20,1.04) (.15,1.2) (.26,1.6)
8	(.41,1.23) (.42,1.43) (.23,1.15)	(.26,1.43) (.06,1.32) (.17,1.02)	(.32,1.34) (.53,1.43) (.45,1.32)	(.40,1.21) (.21,1.21) (.42,1.03)	(.35,1.53) (.36,1.21) (.47,1.05)	(.25,1.53) (.16,1.06) (.05,1.31)	(.40,1.27) (.21,1.03) (.43,1.38)	∞	(.22,1.31) (.53,1.62) (.24,1.73)	(.37,1.76) (.36,1.78) (.40,1.28)
9	(.40,1.72) (.11,1.21) (.32,1.02)	(.41,1.56) (.39,1.56) (.36,1.42)	(.06,1.24) (.36,1.42) (.37,1.76)	(.25,1.71) (.34,1.57) (.29,1.08)	(.21,1.04) (.32,1.3) (.21,1.02)	(.23,1.32) (.33,1.06) (.25,1.03)	(.07,1.01) (.38,1.02) (.39,1.21)	(.32,1.32) (.33,1.76) (.13,1.52)	∞	(.28,1.41) (.19,1.32) (.26,1.72)
10	(.17,1.51) (.26,1.01) (.29,1.21)	(.30,1.31) (.21,1.04) (.32,1.92)	(.28,1.15) (.19,1.21) (.30,1.72)	(.20,1.72) (.31,1.02) (.22,1.51)	(.11,1.82) (.33,1.27) (.22,1.19)	(.32,1.52) (.12,1.18) (.34,1.17)	(.38,1.02) (.28,1.13) (.39,1.16)	(.41,1.62) (.42,1.81) (.33,1.21)	(.31,1.52) (.52,1.37) (.32,1.15)	∞

Table 6.18: Results of BRCMOSTSP (Model 6.1F)

α	β	Algorithm	Path(Vehicle)	Costs & Times	R_{max}
0.95	0.95	iMOGA	2(2)-10(3)-3(3)-9(3)-7(3)-8(1)-6(2)-1(1)-5(2)-4(3)	[56.31, 32.43]	9.5
		MOGA	5(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-6(2)-1(3)-7(3)	[59.61, 30.54]	
0.8	0.9	iMOGA	8(1)-6(2)-1(2)-9(1)-3(1)-4(2)-2(2)-10(1)-5(3)-7(3)	[58.45, 33.76]	8.75
		MOGA	3(1)-4(3)-2(3)-10(2)-5(3)-9(3)-8(1)-6(1)-1(3)-7(2)	[71.59, 27.56]	
0.7	0.9	iMOGA	7(2)-8(1)-6(2)-1(1)-10(2)-5(3)-2(3)-3(1)-4(2)-9(3)	[59.48, 30.23]	8.5
		MOGA	10(2)-1(1)-9(2)-5(1)-7(2)-3(1)-4(2)-2(3)-8(1)-6(1)	[64.54, 26.28]	
0.75	0.75	iMOGA	3(2)-7(1)-8(1)-6(2)-1(2)-5(1)-9(2)-2(1)-10(1)-4(1)	[63.42, 27.5]	8.0
		MOGA	1(3)-10(2)-8(1)-6(1)-9(1)-2(1)-7(1)-5(3)-3(1)-4(1)	[65.21, 25.43]	
0.95	0.75	iMOGA	3(1)-4(3)-2(3)-10(2)-5(3)-9(3)-8(1)-6(1)-1(3)-7(2)	[57.79, 32.78]	7.5
		MOGA	5(1)-10(3)-2(2)-4(1)-3(3)-9(1)-8(1)-6(2)-1(3)-7(3)	[72.49, 34.31]	

Table 6.19: CPU time for BRCMOSTSP (Model 6.1F)

Instances	iMOGA		MOGA	
Cities	Mean	SD	Mean	SD
10	151.45	3.81	273.71	22.56
20	341.47	4.13	465.91	30.91
30	568.75	7.69	531.32	28.64
40	697.32	9.57	861.81	31.43
50	863.51	10.43	976.54	31.79

Table 6.20: Mean and Variance of the diversity metric

Algorithm	SCH	FON	POL	KUR	ZDT1	ZDT2	ZDT3	ZTD4
NSGA-II	0.043246	0.357124	0.623163	0.021271	0.365122	0.535073	0.117382	0.414373
	0.00135	0.017217	0.021836	0.035364	0.015125	0.032531	0.032163	0.012172
MOGA	0.052943	0.453495	0.310266	0.231765	0.3548221	0.213267	0.076296	0.197286
	0.002769	0.036234	0.006362	0.003368	0.001513	0.003537	0.003164	0.023154
iMOGA	0.032356	0.241782	0.296783	0.015386	0.25372	0.232735	0.321785	0.414315
	0.001928	0.002651	0.014362	0.003057	0.0010283	0.002319	0.001201	0.001013

Table 6.21: Mean and Variance of the convergence metric

Algorithm	SCH	FON	POL	KUR	ZDT1	ZDT2	ZDT3	ZTD4
NSGA-II	0.003287	0.023942	0.020341	0.003275	0.125612	0.003562	0.001382	0.014317
	0.000156	0.013231	0.001367	0.001364	0.000512	0.000538	0.002162	0.013183
MOGA	0.003162	0.024534	0.010261	0.004763	0.025481	0.013248	0.002373	0.021928
	0.001505	0.005413	0.000721	0.001421	0.001639	0.000456	0.002036	0.000218
iMOGA	0.002354	0.001734	0.006711	0.000153	0.025372	0.023276	0.00178	0.012318
	0.000092	0.0006513	0.004362	0.000543	0.000284	0.000339	0.000251	0.001312

6.2.4 Statistical Test and Sensitivity Analyses

Performance Measure for iMOGA:

Unlike in single objective optimization, there are two goals in a bi-objective optimization problem. The first goal is to achieve the convergence to the Pareto optimal set and second one is to preserve the diversity in solutions of the given Pareto optimal set. Here two performance matrices following Deb et al., [37] are obtained for the multi-objective optimization algorithms and given in Tables 6.20 and 6.21.

To show the performance of the proposed iMOGA, we used it for some standard multi-objective test functions (Deb, [36]; He et al., [65]). Here each function is compared with the Pareto optimal solutions of proposed iMOGA. For all experiments with every test function, we set the parameters as described above and the experimental results are presented in Tables 6.20 and 6.21. In Table 6.21, we compare the mean and standard deviation (δ) of the convergence metric used by (Deb et al. [36]) for NSGA-II, classical MOGA and proposed iMOGA. This table demands that proposed iMOGA gives better results in the case of mean and standard deviation of the convergence metric. Again from the Table 6.20, we find out the diversity metrics using the same parameters against three algorithms NSGA-II, MOGA and iMOGA. From the Table 6.20, it is observed that proposed

Table 6.22: ANOVA: Number of win for different algorithms

Problem	SCH	FON	POL	KUR	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
iMOGA	79	87	74	82	77	86	90	85	78
NSGA-II	67	76	66	78	63	68	73	69	71
MOGA	61	51	64	66	61	59	69	57	58

algorithm gives better results except in some few cases.

Efficiency Test for iMOGA with other algorithms by ANOVA:

Some standard test problems are solved using the developed algorithm iMOGA. Different parametric values of iMOGA, used for this purpose, are given below:

Here for three algorithms- iMOGA, NSGA-II and Classical MOGA, Pop−size=100 and Maxgen=2000.

The algorithm is tested against a list of standard test functions of crisp valued benchmark problems (Deb et al. [36]). Results are obtained for these standard problems and number of wins for 100 runs of the algorithms- iMOGA, NSGA and MOGA are presented in Table 6.22. To compare the efficiency of the developed algorithm, another two established heuristic technique NSGA-II and classical MOGA are used against these standard test functions and their results (number of wins for 100 runs) are obtained.

Hence, when a set of algorithms are compared, the common statistical method for testing the differences between more than two related samples i.e. ANOVA test is used. Different steps of this ANOVA are as follows. For statistical comparison of the results (obtained by these three algorithms), i.e., for sample of runs for the algorithms (number of wins for 100 runs), the ANOVA procedure is performed.

For calculation of different steps of ANOVA easily, we subtract 60 (with out lose of generality) from each numbers and the Table 6.22 reduces to the Table 6.23.

Here, total sample size of each algorithm is equal and say, I=9 and number of algorithm is J=3. Mean of the sample means, $\overline{\overline{X}}$=10.85.

Critical F values, $F_{0.05(2,24)} \approx 3.4$. As the compared F (from Table 6.24) is higher (38.36) than the critical F value (3.4) for 0.05 level of significance, it may

Table 6.23: ANOVA: Subtracted table from Table 6.22

Problem	SCH	FON	POL	KUR	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	Mean
X_1	19	27	14	22	17	26	30	25	18	$\overline{X}_1$=22
X_2	7	16	6	18	3	7	13	8	11	$\overline{X}_2$=9.89
X_3	1	-9	4	6	1	-1	9	-3	-2	$\overline{X}_3$=0.67

Table 6.24: ANOVA summary table

Source of variation	Sum of square	df	Mean of square	F
Between groups	SS_B=2059.89	J-1=2	$MS_B=\frac{SS_B}{J-1}$=1029.94	$\frac{MS_B}{MS_W}$=38.36
Within groups	SS_W=644.33	J(I-1)=24	$MS_W=\frac{SS_W}{J(I-1)}$=26.85	
Total	SS_T=2704.22	IJ-1=26		

be inferred that there is a significant differences between the groups. When F ratio is found to be significant in an ANOVA with more than two groups, it should be followed by a multiple comparison test to find which group- means differ significantly from each other. Scheffe's multiple comparison F- test is done for this purpose to find out whether iMOGA and NSGA-II and/or iMOGA and MOGA are significant. For the first pair i.e., for iMOGA and MOGA, calculated F value is given by F=$\frac{(\overline{X}_1-\overline{X}_3)^2}{MS_W(\frac{1}{I}+\frac{1}{J})}$=38.12. Similarly, for the second pair i.e., for iMOGA and NSGA-II, calculated F= 12.28. As both calculated F values are greater than the tabulated value (3.4), there is significant difference between iMOGA and classical MOGA and also iMOGA and NSGA-II. From Table 6.23, it is observed that the mean ($\overline{X}_1$) of X_1 is higher than the other two means ($\overline{X}_2$ and $\overline{X}_3$). Significant differences between the algorithms are observed (discussion already is given above) and therefore, it can be concluded that iMOGA is better compared to the other two algorithms.

6.2.5 Discussion

To validate the feasibility and effectiveness of the proposed algorithm, we have applied the iMOGA on some standard TSP combination, (problems are taken from TSPLIB [162]. The proposed algorithm is the combination of fuzzy, fuzzy extended based selections, probabilistic selection, adaptive crossover and generation depended mutation which was implemented in C++ with 150 chromosomes and 2000 iterations in maximum.

For Pareto optimal solutions, Table 6.3 shows the comparisons between MOGA and iMOGA for some standard TSP problems. It is seen that the number of iterations is less in iMOGA than classical MOGA, where the classical MOGA is the combinations of RW selection, cyclic crossover and random mutation. Here we consider the multi-objective standard TSP from TSPLIB [162] combining the same sizes problems. Again Table 6.3 asserts the effectiveness of the proposed algorithm with respect to CPU time. In Table 6.6, we survey the importances of different parameters and operators in proposed iMOGA. It indicates that for the Pareto optimal solution of the combination of **bayg29** and **bays29**, the algorithm navigates the sample space better with generation dependent mutation. In this case, Pareto optimal results are obtained quickly by 132 iterations only. Here also, iMOGA performs better than the classical MOGA.

In Table 6.8, we consider 10×10 crisp costs and times, risk/discomfort matrices for a CSTSP. The Pareto optimal results are presented in Table 6.9 for only CMOTSP considering single conveyance for the given data in Table 6.9. It is observed that CMOTSP without any total risk factor as a goal gives the lowest minimum cost and time and as the total risk/discomfort decreases, total cost as well as time increases. It is realistically true in our day-to-day life. For a particular value of risk/discomfort factor, some near Pareto optimum results along with the Pareto optimum one are presented. Due to some reasons, if the TS fails to implement the optimum result, he/she may choose the most feasible near Pareto optimum solution. Again, we formulated a CMOSTSP with three conveyances i.e. ($10\times10\times3$) costs, times and risks/discomforts matrices presented in Table 6.8. Along each route, the corresponding conveyance is in parentheses. The Pareto optimum results of CMOSTSP are given in Table 6.10. Here also as total risk/discomfort goes down, the corresponding travelling cost and time increases. A ($10\times10\times3$) RCMOSTSP is presented in Table 6.11 where all costs, times and risk/discomfort factors along with the targeted total risk/discomfort are random variates. The Pareto optimum results are presented in Table 6.12. As expected, as the risk goes down, corresponding costs and times compromise each other and go up. For random-fuzzy CMOSTSP, random-fuzzy input data and Pareto optimum results are presented in Tables 6.13 and 6.14 respectively. Here, costs, times and risk/discomfort factors are L-L fuzzy numbers. For a fixed θ =0.88, results by possibility and necessity approaches are given and as before, optimistic

(Possibilistic) representation gives better result (less cost, less time) than the pessimistic (Necessity) one. Again fuzzy random input data are given in Table 6.15 with the costs, times and risk factor, where, means are as random variables (standard normal variate) with right and left spreads of the fuzzy variables. The results presented in Table 6.16 show that Pareto optimal solution gives costs and times w.r. to risk factor as per our expectations. Similarly for bi-random costs, times and risk/discomfort factors are presented in Table 6.17, Pareto optimum results are obtained with different probability levels-α and β for multi-objectives (cost and time) and constraint (risk/discomfort factors) and presented in Table 6.18. In all cases, the near Pareto optimum solutions along with optimal one are available. Also iMOGA gives better results than the classical MOGA.

6.3 Model-6.2 A Rough Multi-Objective Genetic Algorithm for uncertain Constrained Multi-Objective Solid Travelling Salesman Problem

This model addresses a Rough Multi-Objective Genetic Algorithm (R-MOGA) to solve Constrained Multi-Objective Solid Travelling Salesman Problems (CMOSTSPs) in rough, fuzzy rough and random rough environments. In the proposed R-MOGA, '3 - and 5 - level linguistic based rough age oriented selection', 'adaptive crossover' are used along with a new generation dependent mutation. Here we model the CMOSTSP with travelling costs and times as two objectives and a constraint for route risk/discomfort factors. The costs, times and risk/discomfort are rough, fuzzy rough and random rough in nature. The above model is illustrated by using empirical data and a statistical analysis (Analysis of Variance) is carried out to show the supremacy of the proposed algorithm.

6.3.1 Proposed R-MOGA

Here proposed algorithm for R-MOGA using the rough (3-level linguistic) age based and rough extended (5-level linguistic) age based (REA) selection strategies, an adaptive crossover and a generation dependent mutation are presented. Initially a randomly set of potential solutions is generated and then using proposed algorithm, we find out the Pareto optimal solutions until the termination criteria are encountered. The proposed R-MOGA and its procedures are presented below:

(i) Representation:

Here a complete tour on N cities represents a solution. So an N dimensional integer vector $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$ is used to represent a solution (path), where $x_{i1}, x_{i2}, ..., x_{iN}$ represent N consecutive cities in a tour. Population size number M and i-th solution $X_i = (x_{i1}, x_{i2}, ..., x_{iN})(v_{i1}, v_{i2}, ..., v_{iP})$, where $x_{i1}, x_{i2}, ..., x_{iN}$, and $v_{i1}, v_{i2}, ..., v_{iP}$ are randomly generated by random number generator between 1 to N and 1 to P (vehicle set) respectively maintaining the TSP conditions such as not repeating of cities (nodes) and also satisfying the constraints. Fitness are evaluated by summing the costs and times between the consecutive cities (nodes) of each solution (chromosome). The solution $f(X_i)$ represents the i-th solution fitness in the solution space. Since the maximum population size is

M, so M numbers of solutions (chromosomes) are generated randomly.

(ii) Rough set based Selection:

This part is given in section 4.5.1.

(iii) Rough Extended Age Based Selection:

It is given in section 4.6.1.

(iii) Crossover:

This part is given in section 4.3.1(c)(iii).

(iv) Generation Dependent Mutation:

This part given in section 6.2.1.

(v) Algorithm for Rough age based GA:

Input: max_ gen, pop_ size, Max_age, Min_ age, Problem Data (cost matrix, risk matrix).

Output: The optimum and near optimum solutions.

1. **Start**
2. g ← 0 // g: iteration/generation number
3. $\mathbf{Initialize\ P(g)}$ // randomly generate initial population $P(g)$
4. $\mathbf{Evaluate\ f(P(g))}$; //Evaluate fitness of each chromosome of P(g).
5. $\mathbf{while(g \leq max\_gen})$
6. Evaluate the average fitness
7. **if** average fitness > current fitness
8. age(x_i)=avg(age)+$\frac{k*(avgfit-f(X_i))}{(avgfit-minfit)}$
9. **else**
10. age(x_i)=$\frac{avg(age)}{2} + \frac{k*(f(X_i)-avgfit)}{(maxfit-avgfit)}$
11. **if**$(age(x_i) > maximumage)$
12. age(x_i)= maximum age
13. **else if** (age(x_i)< minimum age)
14. age(x_i)= minimum age
15. Determine average age
16. Determine common rough age
17. Switch (Choice)
18. Case I:// **RSGA-I**
    - (a). Developed linguistic variables young, middle, old
    - (b). **for** each pair of parents **do**
    - (c). Trust based p_c created
    - (d). **end for**

19. Case-II:// **RSGA-II**
(a). Developed variables very young, young, middle, old, very old
(b). **for** each pair of parents **do**
(c). Extended trust based p_c created
(d). **end for**// end switch
20. **for** i=1 to Pop Size//**min-point crossover**
21. Choose pair of chromosomes according to p_c
22. Randomly generate node between 1 to N (say a_r)
23. Replace a_r at first place of each parents chromosomes
24. Determine min-point value of each corresponding node
25. **for** j=1 to N
26. Compare min-point value
27. Check the existence of corresponding node in child
28. Concatenated node to the child (offspring)
29. **end for**
30. Replace a_r at end place of each parents chromosomes
31. Compare min-point value from end of the each corresponding nodes
32. **for** j=1 to N
33. Compare min-point value
34. Check the existence of corresponding node in child
35. Concatenated node to the child (offspring)
36. **end for**
37. Replace the child's in offspring's set
38. **end for**
39. Switch (Choice) // **Mutation**
40. Case-I(**simple**):
(a). **for** i=0 to pop_size
(b). Select chromosome depending p_m
(c). Randomly select two different nodes between [1,N]
(d). Replace the places of the selected two nodes
(e). **end for**
41. Case-II(**variable**):
(a). $p_m=\frac{k}{\sqrt{g}}$, $k\in[0,1]$
(b). Determine T= p_m*N // total number of mutated node
(c). **for** i=0 to pop_size

(d). Select chromosome depending p_m
(e). **for** j=1 to T //**Type -I**
(f). Randomly select two different nodes between [1,N]
(g). Replace the places of the selected two nodes
(h). **end for**
(i). **end for**

42. Case-III(**variable**):
(a). $p_m = \frac{k}{\sqrt{g}}$, k∈[0,1]
(b). Determine T= p_m*N
(c). **for** i=0 to pop_size
(d). Select chromosome depending p_m
(e). **for** j=1 to $\frac{T}{2}$ or $(\frac{T}{2}+1)$// T even or odd(**Type-II**)
(f). Replace the places of the any two nodes
(g). **end for**
(h). **end for**

43. Store the new off springs into offspring set
44. **Reproduce a new P(g)**
45. **Evaluate f(P(g))**;//evaluate the fitness of reproduce chromosome
46. Store the local optimum and near optimum solutions
47. g ← g+1
48. endwhile
49. Store the global optimum and near optimum results
50. End Algorithm.

(vi) Division of $P(T)$ into disjoint subsets having non-dominated solutions:
This part is already discussed in section 6.2.1.(vii).

(vii) To determine distance of a solution of subset F from other solutions:
This part is discussed in section 6.2.1.(viii).

(viii) Termination Criteria:

RSGA-I (Rough set based) and RSGA-II (Rough extended set based) algorithms are terminated if any one of the following conditions is satisfied (which over is earlier):

(a) the best solution does not improve within 20 consecutive generations

(b) number of generations reaches user defined iterations (generations).

The same termination criteria are used for other algorithms used in this investigation.

(ix) Complexity analysis:

MOGAs, that use non-dominated sorting and sharing are mainly criticized for their O(MN^3) complexity, but fast and elitist non-dominated sorting algorithm has O(MN^2) computational complexity where N is the popsize and M is the number of objectives. Here also the proposed R-MOGA has the same O(MN^2) computational complexity.

6.3.2 Mathematical Formulation and Its crisp equivalence

Model 6.2A: Multi-Objective TSP with Risk/Discomfort Constraints (CMOTSP):
The model 6.2A is described previously in Equ. 6.4.

Model 6.2B: MOSTSP with Risk/Discomfort Constraints (CMOSTSP):
This model 9B is given in Equ. 6.5.

Model 6.2C: CMOSTSP in Rough Environment (RCMOSTSP):

In the Equ. 6.5, if costs, times and risk/discomfort factors are rough variables, i.e, $\hat{c}(i,j,k)$, $\hat{t}(i,j,k)$ and $\hat{r}(i,j,k)$ respectively and maximum risk/discomfort limit r_{max} is crisp. then the Equ. 6.5 reduces to:

$$\left.\begin{array}{c}\text{minimize } \ Z=\sum_{i=1}^{N-1}\hat{c}(x_i,x_{i+1},v_i)+\hat{c}(x_N,x_1,v_l)\\ \text{minimize } T=\sum_{i=1}^{N-1}\hat{t}(x_i,x_{i+1},v_i)+\hat{t}(x_N,x_1,v_l)\\ \text{subject to } \ \sum_{i=1}^{N-1}\hat{r}(x_i,x_{i+1},v_i)+\hat{r}(x_N,x_1,v_l)\le r_{max}\\ where\ x_i\neq x_j, i,j=1,2...N,\ \ v_i,v_l\in\{1,2..,orP\}\end{array}\right\}\tag{6.28}$$

(the parameters with on the top represent rough quantities)

The above equations written as

$$\left.\begin{array}{c}\text{Minimize } \ \hat{Z}=\hat{C}(x,v)\\ \text{Minimize } \ \hat{T}=\hat{T}(x,v)\\ \text{subject to } \hat{R}(x,v)\le R_{max}\end{array}\right\}\tag{6.29}$$

where $\hat{C} = \sum_{i=1}^{N-1} \hat{c}(x_i, x_{i+1}, v_i) + \hat{c}(x_N, x_1, v_l)$, $\hat{T} = \sum_{i=1}^{N-1} \hat{t}(x_i, x_{i+1}, v_i) + \hat{t}(x_N, x_1, v_l)$, $\hat{R} = \sum_{i=1}^{N-1} \hat{r}(x_i, x_{i+1}, v_i) + \hat{r}_1(x_N, x_1, v_l)$, $R_{max} = r_{max}$, and $\hat{C}$=([a,b],[c,d]), $\hat{T}$=($[t_2, t_3], [t_1, t_4]$), $\hat{R}$=($[R_2], [R_3], [R_1, R_4]$) (say) are rough variables.

Now using trust measure, the above model reduces to:

$$\left.\begin{array}{c} \text{minimize} \quad Z_1, T_1 \\ Tr\{\hat{C}(x,v) \leq Z_1\} \geq \alpha \\ Tr\{\hat{T}(x,v) \leq T_1\} \geq \beta \\ Tr\{\hat{R}(x,v) \leq R_{max}\} \geq \eta \\ where\ x_i \neq x_j, i,j = 1,2...N, \ \ v_i, v_l \in \{1,2..,or P\}. \end{array}\right\} \tag{6.30}$$

Thus the above model are transformed as $minimize\{Z_1, T_1\}$

$$Z_1 = \begin{cases} c + 2\alpha(d-c), & \text{if } c \leq Z_1 \leq a \\ \frac{c(b-a)+a(d-c)+2\alpha(d-c)(b-a)}{d-c+b-a} & \text{if } a \leq Z_1 \leq b \\ c + (d-c)(2\alpha - 1) & \text{if } b \leq Z_1 \leq d \\ d & \text{if } d \leq Z_1 \end{cases} \tag{6.31}$$

and

$$T_1 = \begin{cases} t_1 + 2(t_4 - t_1)\beta, & \text{if } t_1 \leq T_1 \leq t_2 \\ \frac{t_1(t_3-t_2)+t_2(t_4-t_1)+2\beta(t_4-t_1)(t_3-t_2)}{t_4-t_1+t_3-t_2} & \text{if } t_2 \leq T_1 \leq t_3 \\ t_1 + (t_4 - t_1)(2\beta - 1) & \text{if } t_3 \leq T_1 \leq t_4 \\ t_4 & \text{if } t_4 \leq T_1 \end{cases} \tag{6.32}$$

s.t.

$$R_{max} \geq \begin{cases} R_1 + 2(R_4 - R_1)\eta, & \text{if } R_1 \leq R_{max} \leq R \\ \frac{R_1(R_3-R_2)+R_2(R_4-R_1)+2\eta(R_4-R_1)(R_3-R_2)}{R_4-R_1+R_3-R_2} & \text{if } R_2 \leq R_{max} \leq R_3 \\ R_1 + (R_4 - R_1)(2\eta - 1) & \text{if } R_3 \leq R_{max} \leq R_4 \\ R_4 & \text{if } R_4 \leq R_{max} \end{cases} \tag{6.33}$$

Here α, β and η are predetermined confidence levels.

Model 6.2D: CMOSTSP in Fuzzy Rough Environment (FRCMOSTSP):

In the Equ. 6.5, if costs, times and risk/discomfort factors are fuzzy rough variables, i.e, $\tilde{\hat{c}}(i,j,k)$, $\tilde{\hat{t}}(i,j,k)$ and $\tilde{\hat{r}}(i,j,k)$ respectively and maximum risk/discomfort limit r_{max} is also fuzzy rough variables $\tilde{\hat{r}}_{max}$, then the Equ. 6.5 reduces to:

$$\left.\begin{array}{c}
\text{to minimize } \; Z = \sum_{i=1}^{N-1} \tilde{\hat{c}}(x_i, x_{i+1}, v_i) + \tilde{\hat{c}}(x_N, x_1, v_l) \\
\text{to minimize } \; T = \sum_{i=1}^{N-1} \tilde{\hat{t}}(x_i, x_{i+1}, v_i) + \tilde{\hat{t}}(x_N, x_1, v_l) \\
\text{subject to } \; \sum_{i=1}^{N-1} \tilde{\hat{r}}(x_i, x_{i+1}, v_i) + \tilde{\hat{r}}(x_N, x_1, v_l) \leq \tilde{\hat{r}}_{max} \\
where\ x_i \neq x_j, i, j = 1, 2...N, \;\; v_i, v_l \in \{1, 2.., or P\}.
\end{array}\right\} \quad (6.34)$$

Above Equ. 6.34 can be reformulated, where the objective function are

$$\sum_{i=1}^{N-1} \tilde{\hat{c}}(x_i, x_{i+1}, v_i) + \tilde{\hat{c}}(x_N, x_1, v_l) \leq F,$$

$\sum_{i=1}^{N-1} \tilde{\hat{t}}(x_i, x_{i+1}, v_i) + \tilde{\hat{}}(x_N, x_1, v_l) \leq T$, where F and T are given crisps, and equations evaluated using FRCCMOP according to theorem 3.12 and Equ. 3.40 in section 3.13.14.

$$\left.\begin{array}{c}
\text{to minimize F and T} \\
\text{s.t. } Ch\{\sum_{i=1}^{N-1} \tilde{\hat{c}}(x_i, x_{i+1}, v_i) + \tilde{\hat{c}}(x_N, x_1, v_l) \leq F\}(\alpha) \geq \beta \\
Ch\{\sum_{i=1}^{N-1} \tilde{\hat{t}}(x_i, x_{i+1}, v_i) + \tilde{\hat{t}}(x_N, x_1, v_l) \leq T\}(\alpha_1) \geq \beta_1 \\
Ch \sum_{i=1}^{N-1} \tilde{\hat{r}}(x_i, x_{i+1}, v_i) + \tilde{\hat{r}}(x_N, x_1, v_l) \leq \tilde{\hat{r}}_{max}\}(\alpha_2) \geq \beta_2 \\
where\ x_i \neq x_j, i, j = 1, 2...N, \;\; v_i, v_l \in \{1, 2.., or P\}.
\end{array}\right\} \quad (6.35)$$

Here the parameters $\alpha, \beta, \alpha_1, \beta_1, \alpha_2, \beta_2$, are predetermined confidence levels in [0,1].

The above Equ. 6.35 is reformulated as

$$\left.\begin{array}{c}
\text{minimize \{F, T\}} \\
\text{s.t } Ch\{\tilde{\hat{C}}x \leq Z\}(\alpha) \geq \beta \\
Ch\{\tilde{\hat{T}}_1 x \leq T\}(\alpha_1) \geq \beta_1 \\
Ch\{\tilde{\hat{R}}_1 x \leq \tilde{\hat{R}}_{max}\}(\alpha_2) \geq \beta_2 \\
x \in X
\end{array}\right\} \quad (6.36)$$

where $\tilde{\hat{C}} = \sum_{i=1}^{N-1} \tilde{\hat{c}}(x_i, x_{i+1}, v_i) + \tilde{\hat{c}}(x_N, x_1, v_l)$, $\tilde{\hat{T}}_1 = \sum_{i=1}^{N-1} \tilde{\hat{t}}(x_i, x_{i+1}, v_i) + \tilde{\hat{t}}(x_N, x_1, v_l)$, $\tilde{\hat{R}}_1 = \sum_{i=1}^{N-1} \tilde{\hat{r}}(x_i, x_{i+1}, v_i) + \tilde{\hat{r}}_1(x_N, x_1, v_l)$, $\tilde{\hat{R}}_{max} = \tilde{\hat{r}}_{max}$, and X is a fixed set that usually determined by a finite of inequalities involving functions of x as a decision vectors.

It follows from section 3.13.14, the Equ. 6.36 is converted as follows using Trust Possibility measure

$$\left.\begin{array}{c} \text{minimize}\{F, T\} \\ \text{s.t. } Tr\{\lambda | Pos\{\tilde{\hat{C}}x \le Z\} \ge \beta\} \ge \alpha \\ Tr\{\lambda | Pos\{\tilde{\hat{T}}_1 x \le T\} \ge \beta_1\} \ge \alpha_1 \\ Tr\{\lambda | Pos\{\tilde{\hat{R}}_1 x \le \tilde{\hat{R}}_{max}\} \ge \beta_2\} \ge \alpha_2 \\ x \in X \end{array}\right\} \quad (6.37)$$

and the Probability Necessity measure form as given below

$$\left.\begin{array}{c} \text{minimize}\{F, T\} \\ \text{s.t. } Tr\{\omega | Nes\{\tilde{\hat{C}}x \le F\} \ge \beta\} \ge \alpha \\ Tr\{\omega | Nes\{\tilde{\hat{T}}_1 x \le T\} \ge \beta_1\} \ge \alpha_1 \\ Tr\{\omega | Nes\{\tilde{\hat{R}}_1 x \le \tilde{\hat{R}}_{max}\} \ge \beta_2\} \ge \alpha_2 \\ x \in X \end{array}\right\} \quad (6.38)$$

where $\alpha, \beta, \alpha_1, \beta_1, \alpha_2, \beta_2 \in (0, 1]$ are the predetermined confidence levels, Pos$\{.\}$ denotes possibility of the fuzzy events and Tr$\{.\}$ denotes the trust measures of the rough events in $\{.\}$.

To find the crisp values of trust possibility model according in section 3.13.14 the above model Equ. 6.37 is transformed as follows: minimizes $\{F, T\}$

$$F = \begin{cases} c + 2\alpha(d-c) - L^{-1}(\beta)\gamma^{cT}, & \text{if } c \le W \le a \\ \frac{c(b-a)+a(d-c)+2\alpha(d-c)(b-a)}{d-c+b-a} - L^{-1}(\beta)\gamma^{cT} & \text{if } a \le W \le b \\ c + (d-c)(2\alpha - 1) - L^{-1}(\beta)\gamma^{cT} & \text{if } b \le W \le d \\ d - L^{-1}(\beta)\gamma^{cT} & \text{if } d \le W \end{cases} \quad (6.39)$$

$$T = \begin{cases} t_1 + 2\alpha_1(t_4 - t_1) - L^{-1}(\beta_1)\gamma^{tT}, & \text{if } t_1 \leq S \leq t_2 \\ \frac{t_1(t_3-t_2)+t_2(t_4-t_1)+2\alpha_1(t_4-t_1)(t_3-t_2)}{t_4-t_1+t_3-t_2} - L^{-1}(\beta_1)\gamma^{tT} & \text{if } t_2 \leq S \leq t_3 \\ t_1 + (t_4 - t_1)(2\alpha - 1) - L^{-1}(\beta_1)\gamma^{tT} & \text{if } t_3 \leq S \leq t_4 \\ t_4 - L^{-1}(\beta_1)\gamma^{tT} & \text{if } t_4 \leq S \end{cases} \tag{6.40}$$

s.t

$$R_{max} \geq \begin{cases} R_1 + 2(R_4 - R_1)\alpha_2, & \text{if } R_1 \leq R_{max} \leq R_2 \\ \frac{R_1(R_3-R_2)+R_2(R_4-R_1)+2\alpha_2(R_4-R_1)(R_3-R_2)}{R_4-R_1+R_2-R_2} & \text{if } R_2 \leq R_{max} \leq R_2 \\ R_1 + (R_4 - R_1)(2\alpha_2 - 1) & \text{if } R_2 \leq R_{max} \leq R_4 \\ R_4 & \text{if } R_4 \leq R_{max} \end{cases} \tag{6.41}$$

where R_{max}=$R^{-1}(\beta_2)\delta^{r_{max}} + L^{-1}(\beta_2)\gamma^{rT}$, $W = Z + L^{-1}(\beta)\gamma^{cT}$ and S= $T + L^{-1}(\beta_1)\gamma^{tT}$. Here $\delta^{r_{max}}$ and S=γ^{rT} are the right and left spread of LR fuzzy numbers. Also reference functions L, R :[0,1]→ [0,1] with L(1)=R(1)=0 and L(0)=R(0)=1 are non-increasing continuous functions.

Model 6.2E: CMOSTSP in Random-Rough Environment (RRCMOSTSP):

In the Equ. 6.5, if costs, times and risk/discomfort factors are random-rough variables, i.e, $\hat{\tilde{c}}(i,j,k)$, $\hat{\tilde{t}}(i,j,k)$ and $\hat{\tilde{r}}(i,j,k)$ respectively and maximum risk/discomfort limit r_{max} is also a random-rough variable $\hat{\tilde{r}}_{max}$, then the Equ.6.5 reduces to:

$$\left.\begin{aligned} \text{minimize } \; Z &= \sum_{i=1}^{N-1} \tilde{\bar{c}}(x_i, x_{i+1}, v_i) + \tilde{\bar{c}}(x_N, x_1, v_l) \\ \text{minimize } \; T &= \sum_{i=1}^{N-1} \tilde{\bar{t}}(x_i, x_{i+1}, v_i) + \tilde{\bar{t}}(x_N, x_1, v_l) \\ \text{subject to } \; & \sum_{i=1}^{N-1} \tilde{\bar{r}}(x_i, x_{i+1}, v_i) + \tilde{\bar{r}}(x_N, x_1, v_l) \leq \tilde{\bar{r}}_{max} \\ where \; x_i \neq x_j, i, j &= 1, 2...N, \;\; v_i, v_l \in \{1, 2.., or P\}. \end{aligned}\right\} \tag{6.42}$$

Above Equ.6.42 can be reformulated as given below where the objective functions are

$\sum_{i=1}^{N-1} \tilde{\bar{c}}(x_i, x_{i+1}, v_i) + \tilde{\bar{c}}(x_N, x_1, v_l) \leq Z_1$, Z_1 being a crisp quantity.

$\sum_{i=1}^{N-1} \tilde{\bar{t}}(x_i, x_{i+1}, v_i) + \tilde{\bar{t}}(x_N, x_1, v_l) \leq T_1$, T_1 is a crisp quantity.

Now the Equ. 6.42 , using section 3.13.17, defined as RRCMOSTSP is given below

$$\left.\begin{array}{c} \text{minimize} \quad Z_1 \ and \ T_1 \\ Tr\{Pr\{\sum_{i=1}^{N-1}\hat{\tilde{c}}(x_i,x_{i+1},v_i)+\hat{\tilde{c}}(x_N,x_1,v_l)\leq Z_1\}\geq\beta\}\geq\alpha \\ Tr\{Pr\{\sum_{i=1}^{N-1}\hat{\tilde{t}}(x_i,x_{i+1},v_i)+\hat{\tilde{t}}(x_N,x_1,v_l)\leq T_1\}\geq\beta_1\}\geq\alpha_1 \\ \text{s.t} \quad Tr\{Pr\{\sum_{i=1}^{N-1}\hat{\tilde{r}}(x_i,x_{i+1},v_i)+\hat{\tilde{r}}(x_N,x_1,v_l)\leq\hat{\tilde{r}}_{max}\}\geq\beta_2\}\geq\alpha_2 \\ where \ x_i\neq x_j, i,j=1,2...N, \quad v_i,v_l\in\{1,2..,orP\}. \end{array}\right\} \quad (6.43)$$

$\alpha,\beta,\alpha_1,\beta_1,\alpha_2,\beta_2\in(0,1]$ are predetermined confidence levels. The above Equ. 6.43 is equivalently written into according to section 3.13.18 as below Thus the above model transformed as $minimize\{Z_1,T_1\}$

$$Z_1=\begin{cases} c+2\alpha(d-c)+\phi^{-1}(\beta)\sqrt{x^TV^cx}, & \text{if } c\leq R\leq a, \\ \frac{c(b-a)+a(d-c)+2\alpha(d-c)(b-a)}{d-c+b-a}+\phi^{-1}(\beta)\sqrt{x^TV^cx} & \text{if } a\leq R\leq b \\ c+(d-c)(2\alpha-1)+\phi^{-1}(\beta)\sqrt{x^TV^cx} & \text{if } b\leq R\leq d \\ d+\phi^{-1}(\beta)\sqrt{x^TV^cx} & \text{if } d\leq R \end{cases} \quad (6.44)$$

and

$$T_1=\begin{cases} t_1+2\alpha_1(t_4-t_1)+\phi^{-1}(\beta_1)\sqrt{x^TV^tx}, & \text{if } t_1\leq Q\leq t_2 \\ \frac{t_1(t_3-t_2)+t_2(t_4-t_1)+2\alpha_1(t_4-t_1)(t_3-t_2)}{t_4-t_1+t_3-t_2}+\phi^{-1}(\beta)\sqrt{x^TV^tx} & \text{if } t_2\leq Q\leq t_3 \\ t_1+(t_4-t_1)(2\alpha_1-1)+\phi^{-1}(\beta_1)\sqrt{x^TV^tx} & \text{if } t_3\leq Q\leq t_4 \\ t_4+\phi^{-1}(\beta_1)\sqrt{x^TV^cx} & \text{if } t_4\leq Q \end{cases} \quad (6.45)$$

s.t.

$$R_{max}\geq\begin{cases} R_1+2(R_4-R_1)\alpha_2, & \text{if } R_1\leq R_{max}\leq R \\ \frac{R_1(R_3-R_2)+R_2(R_4-R_1)+2\alpha_2(R_4-R_1)(R_3-R_2)}{R_4-R_1+R_3-R_2} & \text{if } R_2\leq R_{max}\leq R_3 \\ R_1+(R_4-R_1)(2\alpha_2-1) & \text{if } R_3\leq R_{max}\leq R_4 \\ R_4 & \text{if } R_4\leq R_{max} \end{cases} \quad (6.46)$$

where R_{max}= - $\phi^{-1}(\beta_2)\sqrt{x^TV^Rx+(\sigma^{rmax})^2}$, R=$Z_1$+$\phi^{-1}(\beta)\sqrt{x^TV^cx}$ and Q=$T_1-\phi^{-1}(\beta_1)\sqrt{x^TV^tx}$. Here α, β, α_1, β_1, α_2, β_2 are predetermined confidence levels. Again σ^{rmax}, V^c, V^t, V^R are standard deviation and variances of costs, times

Table 6.25: Test TSPLIB Problems by R-MOGA

Instances	Single	Multi	R-MOGA			MOGA		
			Cost	Iteration	Time	Cost	Iteration	Time
bays29 bayg29	2020 1610	-	2270 1839	**169**	.23	2834 2487	473	4.71
eil76 pr76	538 108159	-	759 124247	**238**	2.31	936 156721	718	6.54
kroa100 krob100	21282 22141	49639 [148]	49428	**198**	2.52	53745	567	5.58
kroa100 kroc100	21282 20749	50245[148]	50292	**159**	1.46	51634	673	6.12
krob100 kroc100	22141 20749	-	24582 21390	**246**	2.49	26156 23689	638	5.47

and maximum of risk/discomfort factors which we assume that all are standard normal variates with known mean and variances. Also Φ is the standard normal variate distributions.

Solution Procedures:

The deterministic forms of the uncertain RCMOSTSPs given by Eqs.6.31, 6.32 and 6.33 for rough environment, again Eqs. 6.39, 6.40 and 6.41 for FRC-MOSTSP in fuzzy rough environment, Eqs.6.44, 6.45 and 6.46 for RRCMOSTSP in random rough environment are solved by the proposed R-MOGA, developed for this purpose in the section.

6.3.3 Numerical Experiments

Testing for R-MOGA:

To judge the effectiveness and feasibility of the developed algorithm R-MOGA, we have applied it on the standard two TSP problems from TSPLIB [162] with the combination of same size test problems. Table 6.25 gives the results of along with the standard MOGA comparison in terms of total cost and iterations and CPU time in minutes. Here classical MOGA is the combinations of RW-selection, cyclic crossover and random mutation, where as our proposed R-MOGA is the combinations rough extended age based selection (REA), adaptive crossover and generation dependent (GD) mutations.

Moreover, for a particular test problem kora100 and korc100, both standard MOGA and proposed R-MOGA are used with different P_c's, P_m's. The obtained Pareto optimal solutions are presented in Tables 6.26 and 6.27.

Table 6.26: Comparison of R-MOGAs and MOGA

Algorithm	Selection	Crossover	Generation	p_c	p_m	p_s	Result
MOGA-I	Roulette Wheel	Cyclic	423	0.31	0.3	-	[25528, 24764]
MOGA-II	Probabilistic	Cyclic	402	0.31	0.3	-	
MOGA-III	Probabilistic	Adaptive	356	0.4	0.3	-	
MOGA-III	Probabilistic	Adaptive	376	0.44	0.3	-	
MOGA-IV	Probabilistic	Adaptive	363	-	0.3	0.3	
R-MOGA-I	Rough Age based	Adaptive	282	-	0.3	-	
R-MOGA-II	REA	Adaptive	193	-	0.3	-	
R-MOGA-III	REA	Adaptive	159	-	GD	-	

Table 6.27: Comparison of Different operators of R-MOGAs

Algorithm	Selection	Crossover	Mutation	Generation	P_m	Result
R-MOGA	**Rough Age Based**	**Adaptive**	Simple	582	0.4	[25528, 24764]
				542	0.3	
				597	0.2	
			Random	248	0.4	
				226	0.3	
				276	0.2	
			Fixed	193	0.4	
				147	0.3	
				168	0.2	
			GD	**127**	**-**	

Model 6.2B: Results of CMOTSP and CMOSTSP with Risk/Discomfort Constraint in Crisp Environment:

Here we consider a deterministic CMOSTSP given by Equ. 6.32, whose costs, times and risk/discomfort matrices are given by Table 6.28. The problem is solved by R-MOGA and the results are presented in Tables 6.29 and 6.30. Here, for the CMOSTSP, we consider three types of conveyances. With the same data for the 1st conveyance, we solve the CMOTSP (with single conveyance) and the results are presented in Table 6.29.

For Table 6.29, we took maximum generation=1000 and max-popsize =100 and Table 6.31 maximum generation=2000, and maximum popsize=150.

Model 6.2C: CMOSTSP with Risk/Discomfort Constraint in Rough Environment (RCMOSTSP):

R-MOGA	**Rough Extended Age Based**	**Adaptive**	Simple	567	0.4	[25528, 24764]
				451	0.3	
				479	0.2	
			Random	211	0.4	
				109	0.3	
				136	0.2	
			Fixed	157	0.4	
				102	0.3	
				131	0.2	
			GD	**92**	**-**	

Table 6.28: Input Data: Crisp CMOSTSP (Model 6.2B)

	Crisp **Cost** Matrix(10 × 10) With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	35,36,27	18,39,30	20,33,34	30,21,62	6,23,8	15,36,47	27,38,19	40,31,42	20,31,42
2	35,26,17	∞	40,21,32	18,29,10	35,26,37	40,31,22	40,31,59	33,42,59	18,37,20	24,16,18
3	38,30,29	17,58,34	∞	12,25,14	42,25,46	35,36,34	19,11,8	32,33,25	30,19,41	30,22,33
4	28,20,11	10,22,14	17,8,29	∞	30,19,24	25,16,27	21,31,33	35,36,17	12,23,34	27,48,39
5	17,15,9	42,23,34	35,36,37	20,31,43	∞	30,21,42	45,16,27	30,31,13	19,10,8	28,26,7
6	15,6,7	30,21,29	5,26,28	8,9,12	28,29,40	∞	33,42,24	40,31,22	32,23,35	30,41,32
7	38,39,30	25,54,26	30,38,26	22,43,24	37,58,39	40,21,45	∞	10,41,13	32,33,35	20,15,26
8	40,41,23	25,6,17	32,53,45	40,21,42	35,36,47	25,16,5	40,22,43	∞	22,53,24	37,37,39
9	40,11,33	40,39,36	3,36,37	25,34,29	20,32,21	22,33,25	7,38,39	32,33,14	∞	28,19,26
10	18,27,29	30,21,32	28,19,30	20,31,22	11,33,22	32,12,34	37,28,39	40,41,33	30,51,33	∞
	Crisp **Time** Matrix(10 ×10) With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	15,16,17	28,19,20	30,13,14	20,31,12	62,13,68	25,16,27	17,28,39	30,21,22	30,21,22
2	15,16,27	∞	30,31,22	38,19,40	15,16,17	30,21,32	30,21,9	13,22,9	28,17,10	14,36,28
3	30,21,32	17,58,34	∞	12,25,14	42,25,46	35,36,34	19,11,8	32,33,25	30,19,41	30,22,33
4	28,20,11	10,22,14	17,8,29	∞	30,19,24	25,16,27	21,31,33	35,36,17	12,23,34	27,48,39
5	17,15,9	42,23,34	35,36,37	20,31,43	∞	30,21,42	45,16,27	30,31,13	19,10,8	28,26,7
6	25,26,37	20,31,19	55,16,18	61,58,55	18,19,20	∞	13,22,14	30,21,32	22,33,15	20,11,12
7	27,8,14	25,12,36	20,18,16	20,31,12	17,8,19	20,21,25	∞	30,21,33	22,13,15	30,25,16
8	38,19,40	15,16,17	28,19,20	30,13,14	20,31,12	62,13,68	25,16,27	∞	17,28,39	30,21,22
9	40,11,33	40,39,36	3,36,37	25,34,29	20,32,21	22,33,25	7,38,39	32,33,14	∞	28,19,26
10	28,17,19	20,31,12	18,39,20	30,11,18	31,33,22	32,12,34	37,28,39	40,41,33	30,51,33	∞
	Crisp **Risks/Discomforts** Matrix(10×10) With Three Conveyances									
i/j	1	2	3	4	5	6	7	8	9	10
1	∞	.69,.68,.75	.84,.63,.7	.82,.7,.71	.72,.8,.42	.96,.79,.93	.87,.66,.55	.74,.42,.81	.41,.7,.59	.81,.7,.59
2	.67,.76,.84	∞	.61,.8,.7	.83,.73,.92	.67,.76,.65	.41,.71,.79	.41,.71,.43	.69,.6,.42	.83,.64,.81	.77,.85,.3
3	.63,.71,.73	.83,.44,.67	∞	.89,.76,.86	.59,.76,.55	.66,.65,.67	.83,.91,.94	.69,.68,.76	.71,.82,.6	.71,.79,.68
4	.73,.81,.9	.9,.78,.86	.84,.93,.72	∞	.71,.82,.77	.77,.86,.75	.81,.71,.69	.66,.65,.84	.89,.79,.77	.74,.53,.43
5	.84,.86,.92	.59,.78,.67	.66,.65,.64	.82,.71,.59	∞	.71,.81,.59	.57,.85,.74	.71,.7,.88	.82,.91,.93	.74,.75,.93
6	.85,.84,.93	.7,.8,.71	.95,.74,.72	.92,.91,.89	.73,.72,.61	∞	.69,.59,.77	.61,.71,.79	.69,.78,.66	.71,.6,.69
7	.63,.62,.71	.77,.47,.76	.71,.63,.76	.79,.59,.77	.66,.43,.62	.6,.79,.55	∞	.9,.6,.87	.69,.68,.66	.81,.87,.76
8	.61,.6,.78	.76,.95,.84	.69,.47,.56	.61,.81,.6	.67,.66,.55	.6,.85,.95	.61,.8,.59	∞	.79,.48,.77	.64,.64,.62
9	.61,.91,.71	.61,.62,.65	.97,.65,.64	.76,.77,.72	.81,.69,.73	.79,.68,.76	.94,.66,.63	.69,.68,.87	∞	.73,.82,.75
10	.83,.74,.72	.71,.8,.69	.73,.83,.72	.8,.69,.78	.89,.67,.78	.7,.9,.71	.64,.74,.22	.61,.59,.68	.71,.5,.67	∞

Table 6.29: Results of CMOTSP in Crisp (Model 6.2B)

Algorithm	Path	Value	R_{max}
R-MOGA-III	10-2-8-5-9-6-1-4-3-7	[110,132]	Without R_{max}
	8-2-10-5-3-6-7-4-9-1	[110, 132]	8.75
	7-9-6-4-3-5-10-8-2-1	[121,124]	8.75
	5-3-2-10-4-8-7-9-1-6	[134, 115]	8.75
	3-8-2-10-5-9-6-1-4-7	[140, 106]	8.75
R-MOGA-III	2-7-1-5-9-6-10-4-3-2	[167, 100]	8.5
MOGA	9-8-2-5-10-6-1-4-3-7	[207, 117]	8.5
R-MOGA-II	3-5-9-6-1-4-8-7-2-10	[228, 106]	8.00
MOGA	10-2-5-1-4-3-7-9-6-8	[315, 129]	8.00
R-MOGA-II	6-2-7-9-1-4-8-5-10-3	[237, 102]	8.00

Table 6.30: Results of CMOSTSP in Crisp (Model 6.2B)

Algorithm	Path(Vehicle)	Cost	Risk achieved	R_{max}
R-MOGA-III	2(3)-10(2)-5(3)-4(1)-1(2)-9(3)-3(2)-7(1)-8(3)-612)	[112, 137]	8.69	8.75
	6(2)-7(1)-8(1)-9(3)-1(1)-3(1)-4(1)-2(1)-10(1)-5(3)	[130, 123]	8.50	
	6(1)-2(2)-10(1)-9(1)-8(1)-4(2)-3(3)-7(2)-5(2)-1(3)	[147, 121]	8.50	
	2(1)-8(1)-6(2)-1(1)-10(3)-5(2)-4(3)-7(3)-9(1)-3(2)	[153, 117]	8.26	8.75
MOGA-I	1(2)-9(1)-3(3)-7(3)-8(2)-6(1)-2(3)-10(2)-5(2)-4(1)	[200, 128]	8.71	8.75
R-MOGA-III	3(2)-4(3)-2(1)-9(3)-8(1)-6(3)-1(1)-5(2)-7(1)-10(1)	[149, 101]	7.87	8.00
	1(2)-10(3)-9(1)-3(1)-7(1)-8(1)-6(2)-2(1)-5(2)-4(1)	[176, 96]	7.69	
	6(1)-5(2)-4(3)-2(2)-9(2)-3(1)-8(1)-7(1)-1(2)-10(2)	[245, 87]	7.7	

Table 6.31: Input Data: RCMOSTSP (Model 6.2C)

	Rough **Cost** Matrix(10 ×10) for RCMOSTSP With Three Conveyances							
i/j	1	2	3	4	5	6	7	8
1	∞	([29,30],[27,32]) ([35,37],[34,39]) ([24,25],[23,28])	([13,15],[12,17]) ([36,37][34,39]) ([29,30],[27,31])	([20,21],[18,22]) ([31,33],[30,34]) ([29,30],[28,35])	([28,29],[26,31]) ([19,20],[18,21]) ([58,59],[57,62])	([23,26],[21, 27]) ([21,23],[20,25]) ([7,8],[6,10])	([15,16],[13,17]) ([34,36],[32,37]) ([44,46],[43,47])	([26,28],[23,29]) ([37,38],[35,39]) ([17,18],[16,20])
2	([33,34],[33,35] ([23,24],[22,26]) ([15,16],[14,17])	∞	([38,39],[37,41]) ([20,21],[19,22]) ([29,30],[28,32])	([15,16],[14,18]) ([28,29],[27,30]) ([9,10],[8,11])	([33,34],[32,35]) ([25,26],[24,27]) ([33,35],[32.37])	([39,40],[37,41]) ([28,29],27,31]) ([2,22],[20,23])	([39,40],[38,41]) ([29,30],[28,31]) ([57,59],[56,61])	([32,33],[31,34]) ([40,41],[39,42]) ([54,55],[53,59])
3	([34,35],[33,38]) ([28,29],[27,30]) ([28,29],[27,30])	([15,17],[13,18]) ([54,56],[53,58]) ([30,31],[29,34])	∞	([11,12],10,13]) ([22,24],[21,25]) ([13,14],[11,15])	([39,40],[37,42]) ([23,24],[22,25]) ([44,45],[43,46])	([33,35],[32,36]) ([33,34],[31,36]) ([32,33],31,34])	([18,19],[17,20]) ([10,11],[9,13]) ([7,8],[6,10])	([29,32],[28,33]) ([32,33],[31,30]) ([23,25],[22,26])
4	([26,28],[25,29]) ([17,18],[16,20]) ([9,10],[8,11])	([9,10],[8,11]) ([19,20],[18,22]) ([14,15],[13,17])	([15,16],[14,18]) ([8,9],[7,10]) ([27,29],[26,30])	∞	([28,30],[27,31]) ([18,19],[17,20]) ([22,23],[21,24])	([23,25],[22,26]) ([14,16],[13,17]) ([25,27],[26,28])	([19,21],[18,22]) ([30,31],[29,33]) ([31,33],[30,34])	([33,35],[32,36]) ([33,34],[32,36]) ([15,16],[14,17])
5	([15,17],[14,18]) ([13,15],[12,16]) ([6,7],[5,8])	([39,40],[38,42]) ([21,23],[20,24]) ([31,34],[30,35])	([33,35],[32,36]) ([33,34],[32,36]) ([35,37],[34,38])	([18,19],[17,20]) ([11,13],[10,14]) ([42,43],[41,44])	∞	([29,30],[28,32]) ([20,21],,[19,22]) ([40,41],[39,43])	([43,44],[42,45]) ([15,16],[13,17]) ([25,27],[24,28])	([28,29],[27,30]) ([29,30],[27,31]) ([12,13],[11.14])
6	([15,16],14,18]) ([6,7].[5,8]) ([7,8],[6,10])	([27,28],[26,29]) ([21,22],[20,23]) ([28,29],[27,30])	([4,6],[3,8]) ([25,26],[24,27]) ([26,28],[25,29])	([6,7],[5,8]) ([7,9],[6,10]) ([11,12],[10,15])	([26,27],[28,30]) ([27,29],[26,30]) ([38,39],[37,40])	∞	([32,33],[30,34]) ([41,42],[40,44]) ([23,24],[22,25])	([39,40],[38,42]) ([29,31],[28,30]) ([21,22],[20,25])
7	([33,34],[35,37]) ([36,39],[35,40]) ([28,30],[27,31])	([25,26],[23,28]) ([48,49],[47,53]) ([25,26],[23,27])	([28,29],[27,30]) ([37,38],[36,39]) ([24,25],[23,26])	([21,22],[20,23]) ([40,43],[39,44]) ([23,24],[22,25])	([36,37],[35,38]) ([55,56],[54,58]) ([39,40],[38,41])	([39,40],[38,42]) ([20,21],[19,25]) ([43,44],[42,45])	∞	([8,9],[7,10]) ([39,40],[38,43]) ([11,13],[10,14])
8	([39,40],[37,41]) ([41,42],[40,44]) ([22,23],[21,24])	([23,25],[22,26]) ([5,6],[4,7]) ([15,17],[14,18])	([29,32],[28,33]) ([49,53],[48,54]) ([44,45],[43,47])	([38,40],[37,41]) ([19,21],[18,22]) ([39,40],[38,42])	([35,36],[33,38]) ([33,36],[31,37]) ([45,47],[44,48])	([23,25],[22,27]) ([13,16],[12,18]) ([5,6],[4,7])	([40,41],[39,42]) ([20,21],[19,22]) ([41,43],[39,44])	∞
	Rough **Time** Matrix(10×10) for RCMOSTSP With Three Conveyances							
i/j	1	2	3	4	5	6	7	8
1	∞	([20,21],[19,22]) ([16,17],[15,18]) ([16,18],[15,19])	([]27,29],[26,30]) ([17,19],[16,20]) ([18,20],[17,21])	([29,30],[28,31]) ([28,31],[27,33]) ([13,15],[12,17])	([17,19],[16,21]) ([29,30],[27,31]) ([11,12],[10,14])	([33,37],[32.38]) ([29,30],[28,33]) ([31,33],[30,38])	([25,26],[24,27]) ([14,5],[13,16]) ([7,8],[6,9])	([17,18],[16,19]) ([17,19],[16,22]) ([36,38],[34,39])
2	([15,16],[13,17]) ([13,14],[12,16]) ([34,35],[33,37])	∞	([20,21],[19,23]) ([20,21],[19,23]) ([20,22],[19,23])	([17,18],[16,20]) ([18,19],[17,20]) ([9,10],[8,11])	([33,34],[32,35]) ([14,16],[13,15]) ([13,14],[15,17])	([19,20],[18,21]) ([20,21],[19,23]) ([32,33],[30,35])	([18,20],[17,21]) ([19,20],[18,22]) ([8,9],[7,11])	([12,13],[11,15]) ([21,22],[19,23]) ([36,38],[33,39])
3	([18,19],[17,20]) ([19,20],[18,22]) ([19,20],[18,22])	([26,27],[25,29]) ([7,8],[6,10]) ([13,14],[12,15])	∞	([11,12],[10,15]) ([11,15],[10,17]) ([33,34],[31,35])	([10,12],[9,13]) ([13,14],[12,15]) ([5,6],[4,7])	([23,25],[22,27]) ([23,25],[21,26]) ([11,14],[10,16])	([33,39],[31,35]) ([28,30],[27,31]) ([34,35],[33,38])	([11,12],[10,14]) ([11,13],[10,12]) ([14,15],[13,17])
4	([18,19],[17,20]) ([28,30],[27,32]) ([39,40],[38,42])	([29,30],[27,31]) ([29,32],[27,33]) ([23,24],[21,26])	([33,38],[32,39]) ([34,35],[33,39]) ([18,19],[17,20])	∞	([18,19],[16,20]) ([33,37],[32,39]) ([23,24],[22,26])	([15,16].[13,18]) ([43,45],[42,46]) ([13,17],[12,18])	([30,31],[29,32]) ([20,21],[18,22]) ([21,23],[20,25])	([24,25],[22,27]) ([15,16],[14,17]) ([33,36],[31,40])
5	([37,38],[36,39]) ([33,35],[32,36]) ([43,44],[40,48])	([22,23],[20,25]) ([31,33],[30,34]) ([14,15],[13,17])	([13,15],[12,17]) ([15,16],[13,18]) ([18,19],[17,20])	([18,20],[17,21]) ([11,13],[10,14]) ([41,43],[40,44])	∞	([18,20],[17,21]) ([20,21],[18,23]) ([20,21],[19,24])	([4,5],[3,7]) ([33,35],[30,37]) ([17,18],[16,17])	([18,20],[17,21]) ([20,21],[19,23]) ([31,32],[30,33])
6	([33,34],[31,35]) ([43,46],[40,47]) ([33,34],[31,37])	([18,19],[17,22]) ([11,13],[10,14]) ([18,19],[17,20])	([41,42],[40,44]) ([15,16],[13,17]) ([16,18],[15,19])	([8,9],[7,10]) ([8,9],[7,10]) ([11,12],[10,14])	([16,17],[15,18]) ([17,18],[16,20]) ([21,23],[20,29])	∞	([21,22],[20,23]) ([21,22],[20,23]) ([11,14],[10,15])	([19,20],[18,22]) ([19,20],[18,21]) ([21,22],[20,24])
7	([13,17],[12,18]) ([18,19],[17,20]) ([19,20],[18,21])	([15,17],[13,19]) ([2,3],[1,6]) ([14,16],[13,17])	([19,20],[18,21]) ([17,18],[16,19]) ([23,26],[21,24])	([20,22],[19,24]) ([42,43],[41,44]) ([22,25],[21,26])	([23,27],[22,26]) ([7,8],[6,9]) ([19,20],[18,23])	([19,20],[18,21]) ([10,11],[9,12]) ([14,15],[13,17])	∞	([18,20],[17,21]) ([41,43],[40,45]) ([11,13],[10,15])

Here we have taken the costs, times and risk/discomfort values as rough for the CMOSTSP. Also we consider three types of conveyances. The rough cost, time matrices for the CMOSTSP corresponding random risk/discomfort matrix are given in Table 6.31. The Pareto optimum results of this CMOSTSP model for different values of risk are obtained by R-MOGA and presented in Table 6.32.

Model 6.2D: CMOSTSP with Risks/Discomforts Constraint in Fuzzy Rough Environment (FRCMOSTSP):

Here the costs, times and risks/discomforts are in fuzzy rough values for the CMOSTSP. Also we consider three types of conveyances. Assume that ξ is a rough variable with corresponding values of Table 6.31 and in the Table 6.33 its fuzzy values are given for cost, time and risk values.

8	([29,31],[28,33]) ([22,23].[21,24]) ([31,33],[30,34])	([14,16],[13,18]) ([43,44],[41,46]) ([35,37],[34,39])	([12,13],[11,14]) ([6,7],[5,8]) ([4,5],[3,8])	([32,37],[31,40]) ([20,21],[19.22]) ([42,43],[40,44])	([13,15],[12,17]) ([23,26],[20,27]) ([13,17],[12,18])	([15,16],[13,17]) ([23,26],[21,27]) ([33,35],[31,37])	([10,11],[9,14]) ([20,21],[19,22]) ([11,13],[10,15])	∞
	Rough **Risks/Discomforts** Matrix(10×10) for RCSTSP With Three Conveyances							
i/j	1	2	3	4	5	6	7	8
1	∞	([.56,.58],[.55,.62]) ([.52,.54],[.51,.55]) ([.25,.27],[.23,.28])	([.71,.73],[.7,.75]) ([.51,.53],[.5,.56]) ([.63,.64],[.61,.67])	([.68,.69],[.67,.7]) ([.57,.6],[.54,.61]) ([.54,.56],[.53,.59])	([.62,.64],[.61,.66]) ([.71,.73],[.7,.78]) ([.31,.33],[.3,.34])	([.81,.83],[.8,.87]) ([.69,.71],[.68,.73]) ([.81,.83],[.8,.88])	([.76,.77],[.75,.8]) ([.55,.58],[.53,.59]) ([.47,.49],[.45,.5])	([.67,.68],[.66,.69]) ([.48,.52],[.47,.54]) ([.65,.66],[.64,.68])
2	([.54,.55],[.53,.6]) ([.64,.65],[.61,.67]) ([.72,.73],[.7,.79])	∞	([.51,.52],[.5,.54]) ([.71,.72],[.69,.74]) ([.61,.63],[.6,.64])	([.7,71],[.67,.77]) ([.6,.62],[.57,.63]) ([.76,.77],[.74,.85])	([.63,.64],[.61,.6]) ([.61,.62],[.58,.68]) ([.55,.56],[.52,.58])	([.53,.55],[.51,.56]) ([.6,.63],[.57,.64]) ([.66,.67],[.65,.7])	([.5,.51],[.52,.54]) ([.61,.62],[.6,.66]) ([.33,.35],[.31,.36])	([.6,.62],[.57,.63]) ([.51,.52],[.5,.53]) ([.27,.29],[.26,.32])
3	([.55,.56],[.53,.58]) ([.6,.62],[.59,.64]) ([.61,.63],[.6,.66])	([.71,.72],[.7,.77]) ([.33,.34],[.31,.35]) ([.6,62],[.57,.64])	∞	([.76,.77],[.75,.79]) ([.67,.7],[.66,.72]) ([.76,.77],[.73,.8])	([.53,.54],[.51,.54]) ([.7,.71],[.69,.74]) ([.43,.44],[.42,.49])	([.53,.54],[.5,.59]) ([.55,.56],[.53,.59]) ([.6,.62],[.57,.58])	([.71,.72],[.7,.74]) ([.81,.83],[.79,.85]) ([.83,.84],[.81,.86])	([.57,.59],[.56,.62]) ([.6,.61],[.59,.63]) ([.68,.67],[.66,.67])
4	([.61,.62],[.6,.65]) ([.73,.74],[.7,.76]) ([.76,.78],[.74,.84])	([.81,.82],[.79,.85]) ([.71,.73],[.7,.74]) ([.76,.77],[.75,.79])	([.77,.78],[.76,.79]) ([.85,.87],[.83,.9]) ([.63,.65],[.62,.66])	∞	([.65,.66],[.63,.67]) ([.73,.74],[.7,.79]) ([.67,.68],[.65,.71])	([.68,.69],[.67,.7]) ([.71,.72],[.73,.77]) ([.66,.69,],[.64,.7])	([.71,.73],[.7,.77]) ([.6,.63],[.56,.65]) ([.61,.63],[.6,.66])	([.67,.68],[.64,.69]) ([.56,.58],[.55,.6]) ([.71,.73],[.7,.77])
5	([.76,.77],[.74,.8]) ([.76,.77],[.73,.8]) ([.83,.84],[.8,.88])	([.52,.54],[.51,.55]) ([.67,.68,],[.65,.69]) ([.56,.58],[.55,.6])	([.56,.57],[.55,.6]) ([.56,.58],[.55,.6]) ([.51,.53],[.5,56])	([.73,.75],[.72,.76]) ([.78,.79],[.76,.82]) ([.49,.51],[.47,.52])	∞	([.63,.65],[.6,.66]) ([.73,.74],[.7,.76]) ([.53,.54],[.5,.55])	([.47,.48],[.44,.5]) ([.76,.78],[.75,.8]) ([.67,.68][.66,.69])	([.61,.63],[.58,.64]) ([.63,.64],[.62,.66]) ([.74,.76],[.73,.8])
6	([.78,.8],[.7,.81]) ([.87,.89],[.85,.9]) ([.8,.81],[.78,.85])	([.67,.68],[.59,.69]) ([.77,.79],[.76,.8]) ([.67,.7],[.66,.71])	([.86,.88],[.83,.89]) ([.73,.74],[.72,.76]) ([.63,.65],[.6,.66])	([.8,.85],[.78,.88]) ([.83,.88],[.81,.89]) ([.76,.8],[.73,.81])	([.69,.7],[.66,.71]) ([.67,.68],[.66,.7]) ([.51,.53],[.5,.54])	∞	([.6,.63],[.56,.64]) ([.53,.55],[.51,.56]) ([.73,.74],[.7,.75])	([.51,.55],[.5,.56]) ([.61,.63],[.6,.67]) ([.7,.78])
7	([.55,.56],[.5,.57]) ([.55,.57],[.53,.6]) ([.63,64],[.6,.67])	([.66,.67],[.65,.68]) ([.41,.42],[.4,43]) ([.68,.7],[.66,.71])	([.63,.64],[.62,.67]) ([.56,.59],[.55,.6]) ([.67,.68],[.66,.71])	([.71,.72],[.69,.75]) ([.49,.52],[.47,.56]) ([.65,.69],[.64,.7])	([.61,.62],[.6,.63]) ([.33,.37],[.31,.39]) ([.49,.54],[.48,.55])	([.5,.54],[.49,.56]) ([.73,.76],[.71,.77]) ([.5,.56],[.49,.57])	∞	([.78,.79],[.77,.84]) ([.55,.56],[.54,.58]) ([.79,.8],[.77,.82])
8	([.51,.55],[.5,.56]) ([.51,.55],[.5,.56]) ([.7,.72],[.69,.73])	([.67,.7],[.65,.71]) ([.73,.74],[.7,.78]) ([.73,.74],[.7,.77])	([.63,.65],[.6,.66]) ([.41,.42],[.39,.43]) ([.49,.5],[.47,.52])	([.57,.58],[.55,.6]) ([.7,.71],[.67,.72]) ([.52,.54],[.5,.55])	([.56,.57],[.55,.59]) ([.56,.6],[.55,.57]) ([.45,.48],[.43,.49])	([.66,.68],[.65,.69]) ([.76,.77],[.75,.78]) ([.87,.88],[.86,.89])	([.56,.57],[.55,.59]) ([.71,.72],[.7,.74]) ([.5,.52],[.49,.55])	∞

Table 6.32: Results of RCMOSTSP (Model 6.2C)

α	β	η	Algorithm	Path(Vehicle)	Costs & Time	R_{max}
.9	.9	.9	R-MOGA-III	5(2)-1(1)-6(1)-4(2)-3(3)-7(1)-8(2)-2(3)	[62.5, 41.7]	8.5
			MOGA-I	5(1)-2(2)-4(1)-3(3)-8(1)-6(2)-1(3)-7(3)	[65.25, 45.65]	8.5
			R-MOGA-III	7(3)-4(2)-1(3)-5(1)-6(1)-2(2)-3(3)-8(2)	[76.60, 33.20]	8.5
.95	.95	.95	MOGA-I	5(1)-2(2)-4(1)-3(3)-8(1)-6(2)-1(3)-7(3)	[84.32, 68.5]	8.5
.95	.95	0.95	R-MOGA-III	5(2)1(1)-6(1)-4(2)-3(3)-7(1)-8(2)-2(3)	[73.5, 43.3]	8.5.
.97	.97	.97	R-MOGA-III	5(2)-1(1)-6(1)-4(2)-3(3)-7(1)-8(2)-2(3)	[80.24, 26.28]	8.5
.9	.9	.9	R-MOGA-III	6(2)-4(3)-3(1)-5(1)-7(1)-8(2)-2(1)-1(2)	[71.5, 36.4]	8.25
.9	.9	.9	MOGA-I	4(2)-2(2)-5(3)-3(3)-8(1)-6(2)-1(3)-7(3)	[84.17, 71.8]	8.0

Table 6.33: Input Data: FRCMOSTSP (Model 6.2D)

	Fuzzy Rough **Cost** Matrix(10 ×10) for FRCMOSTSP With Three Conveyances							
i/j	1	2	3	4	5	6	7	8
1	∞	$(\xi$-5, ξ, ξ+5) $(\xi$-4, ξ, ξ+4) $(\xi$-6, ξ, ξ+6)	$(\xi$-6, ξ, ξ+6) $(\xi$-7, ξ, ξ+7) $(\xi$-5, ξ, ξ+5)	$(\xi$-3, ξ, ξ+3) $(\xi$-2, ξ, ξ+2) $(\xi$-3, ξ, ξ+3)	$(\xi$-9, ξ, ξ+9) $(\xi$-5, ξ, ξ+5) $(\xi$-2, ξ, ξ+2)	$(\xi$-5, ξ, ξ+5) $(\xi$-10, ξ, ξ+10) $(\xi$-13, ξ, ξ+13)	$(\xi$-6, ξ, ξ+6) $(\xi$-11, ξ, ξ+11) $(\xi$-4, ξ, ξ+4)	$(\xi$-11, ξ, ξ+11) $(\xi$-3, ξ, ξ+3) $(\xi$-6, ξ, ξ+6)
2	$(\xi$-8, ξ, ξ+8) $(\xi$-7, ξ, ξ+7) $(\xi$-4, ξ, ξ+4)	∞	$(\xi$-7, ξ, ξ+7) $(\xi$-6, ξ, ξ+6) $(\xi$-1, ξ, ξ+1)	$(\xi$-3, ξ, ξ+3) $(\xi$-2, ξ, ξ+2) $(\xi$-5, ξ, ξ+5)	$(\xi$-4, ξ, ξ+4) $(\xi$-2, ξ, ξ+2) $(\xi$-3, ξ, ξ+3)	$(\xi$-12, ξ, ξ+12) $(\xi$-10, ξ, ξ+10) $(\xi$-12, ξ, ξ+12)	$(\xi$-9, ξ, ξ+9) $(\xi$-9, ξ, ξ+9) $(\xi$-10, ξ, ξ+10)	$(\xi$-1, ξ, ξ+1) $(\xi$-14, ξ, ξ+14) $(\xi$-3, ξ, ξ+3)

3	$(\xi-4, \xi, \xi+4)$ $(\xi-16, \xi, \xi+16)$ $(\xi-3, \xi, \xi+3)$	$(\xi-7, \xi, \xi+7)$ $(\xi-12, \xi, \xi+12$ $(\xi-11, \xi, \xi+11)$	∞	$(\xi-2, \xi, \xi+2)$ $(\xi-2, \xi, \xi+2)$ $(\xi-13, \xi, \xi+13)$	$(\xi-1, \xi, \xi+1)$ $(\xi-5, \xi, \xi+5)$ $(\xi-17, \xi,\xi+17)$	$(\xi-10, \xi, \xi+10)$ $(\xi-10, \xi, \xi+10)$ $(\xi-1, \xi, \xi+1)$	$(\xi-12, \xi, \xi+12$ $(\xi-14, \xi, \xi+14)$ $(\xi-6, \xi, \xi+6)$	$(\xi-13, \xi, \xi+13)$ $(\xi-2, \xi, \xi+2)$ $(\xi-8, \xi, \xi+8)$
4	$(\xi-2, \xi, \xi+)$ $(\xi-4, \xi, \xi+4)$ $(\xi-6, \xi, \xi+6)$	$(\xi-3, \xi, \xi+63$ $(\xi-7, \xi, \xi+7)$ $(\xi-5, \xi, \xi+5)$	$(\xi-3, \xi, \xi+3)$ $(\xi-12, \xi, \xi+12)$ $(\xi-3, \xi, \xi+3)$	∞	$(\xi-11, \xi, \xi+11)$ $(\xi-5, \xi, \xi+5)$ $(\xi-7, \xi, \xi+7)$	$(\xi-13, \xi, \xi+13)$ $(\xi-1, \xi, \xi+1)$ $(\xi-12, \xi, \xi+12)$	$(\xi-11, \xi, \xi+11)$ $(\xi-13, \xi, \xi+13)$ $(\xi-15, \xi, \xi+15)$	$(\xi-1, \xi, \xi+1)$ $(\xi-14, \xi, \xi+14)$ $(\xi-12, \xi, \xi+12)$
5	$(\xi-5, \xi, \xi+5)$ $(\xi-4, \xi, \xi+4)$ $(\xi-6, \xi, \xi+6)$	$(\xi-6, \xi, \xi+6)$ $(\xi-7, \xi, \xi+7)$ $(\xi-5, \xi, \xi+5)$	$(\xi-3, \xi, \xi+3)$ $(\xi-2, \xi, \xi+2)$ $(\xi-3, \xi, \xi+3)$	$(\xi-5, \xi, \xi+5)$ $(\xi-5, \xi, \xi+5)$ $(\xi-7, \xi, \xi+7)$	∞	$(\xi-1, \xi, \xi+1)$ $(\xi-10, \xi, \xi+10)$ $(\xi-12, \xi, \xi+12)$	$(\xi-14, \xi, \xi+14)$ $(\xi-3, \xi, \xi+3)$ $(\xi-15, \xi, \xi+15)$	$(\xi-11, \xi, \xi+11)$ $(\xi-13, \xi, \xi+13)$ $(\xi-12, \xi, \xi+12)$
6	$(\xi-6, \xi, \xi+6)$ $(\xi-5, \xi, \xi+5)$ $(\xi-11, \xi, \xi+11)$	$(\xi-7, \xi, \xi+7)$ $(\xi-6, \xi, \xi+6)$ $(\xi-17, \xi, \xi+17)$	$(\xi-13, \xi, \xi+13)$ $(\xi-2, \xi, \xi+2)$ $(\xi-5, \xi, \xi+5)$	$(\xi-14, \xi, \xi+14)$ $(\xi-3, \xi, \xi+3)$ $(\xi-3, \xi, \xi+3)$	$(\xi-2, \xi, \xi+2)$ $(\xi-4, \xi, \xi+4)$ $(\xi-2, \xi, \xi+2)$	∞	$(\xi-6, \xi, \xi+6)$ $(\xi-12, \xi, \xi+12)$ $(\xi-1, \xi, \xi+1)$	$(\xi-5, \xi, \xi+5)$ $(\xi-4, \xi, \xi+4)$ $(\xi-12, \xi, \xi+12)$
7	$(\xi-15, \xi, \xi+15)$ $(\xi-4, \xi, \xi+4)$ $(\xi-6, \xi, \xi+6)$	$(\xi-6, \xi, \xi+6)$ $(\xi-7, \xi, \xi+7)$ $(\xi-5, \xi, \xi+5)$	$(\xi-3, \xi, \xi+3)$ $(\xi-2, \xi, \xi+2)$ $(\xi-3, \xi, \xi+3)$	$(\xi-2, \xi, \xi+2)$ $(\xi-5, \xi, \xi+5)$ $(\xi-7, \xi, \xi+7)$	$(\xi-11, \xi, \xi+11)$ $(\xi-12, \xi, \xi+12)$ $(\xi-1, \xi, \xi+1)$	$(\xi-12, \xi, \xi+12)$ $(\xi-4, \xi, \xi+4)$ $(\xi-8, \xi, \xi+8)$	∞	$(\xi-1, \xi, \xi+1)$ $(\xi-7, \xi, \xi+7)$ $(\xi-4, \xi, \xi+4)$
8	$(\xi-8, \xi, \xi+8)$ $(\xi-5, \xi, \xi+5)$ $(\xi-4, \xi, \xi+4)$	$(\xi-7, \xi, \xi+7)$ $(\xi-6, \xi, \xi+6)$ $(\xi-7, \xi, \xi+7)$	$(\xi-3, \xi, \xi+3)$ $(\xi-2, \xi, \xi+2)$ $(\xi-5, \xi, \xi+5)$	$(\xi-4, \xi, \xi+4)$ $(\xi-2, \xi, \xi+2)$ $(\xi-13, \xi, \xi+13)$	$(\xi-10, \xi, \xi+10)$ $(\xi-10, \xi, \xi+10)$ $(\xi-4, \xi, \xi+4)$	$(\xi-10, \xi, \xi+10)$ $(\xi-10, \xi, \xi+10)$ $(\xi-4, \xi, \xi+4)$	$(\xi-10, \xi, \xi+10)$ $(\xi-11, \xi, \xi+11)$ $(\xi-4, \xi, \xi+4)$	∞
	Fuzzy Rough **Time** Matrix(10 $\times$10) for FRCMOSTSP With Three Conveyances							
i/j	1	2	3	4	5	6	7	8
1	∞	$(\xi-1, \xi, \xi+1)$ $(\xi-4, \xi, \xi+4)$ $(\xi-2, \xi, \xi+2)$	$(\xi-2, \xi, \xi+2)$ $(\xi-7, \xi, \xi+7)$ $(\xi-3, \xi, \xi+3)$	$(\xi-3, \xi, \xi+3)$ $(\xi-2, \xi, \xi+2)$ $(\xi-4, \xi, \xi+4)$	$(\xi-19, \xi, \xi+19)$ $(\xi-5, \xi, \xi+5)$ $(\xi-6, \xi, \xi+6)$	$(\xi-15, \xi, \xi+15)$ $(\xi-1, \xi, \xi+1)$ $(\xi-6, \xi, \xi+6)$	$(\xi-3, \xi, \xi+3)$ $(\xi-11, \xi, \xi+11)$ $(\xi-8, \xi, \xi+8)$	$(\xi-6, \xi, \xi+6)$ $(\xi-5, \xi, \xi+5)$ $(\xi-7, \xi, \xi+7)$
2	$(\xi-3, \xi, \xi+3)$ $(\xi-6, \xi, \xi+6)$ $(\xi-5, \xi, \xi+5)$	∞	$(\xi-4, \xi, \xi+4)$ $(\xi-2, \xi, \xi+2)$ $(\xi-2, \xi, \xi+2)$	$(\xi-2, \xi, \xi+2)$ $(\xi-1, \xi, \xi+1)$ $(\xi-6, \xi, \xi+6)$	$(\xi-1, \xi, \xi+1)$ $(\xi-3, \xi, \xi+3)$ $(\xi-3, \xi, \xi+3)$	$(\xi-3, \xi, \xi+3)$ $(\xi-11, \xi, \xi+11)$ $(\xi-7, \xi, \xi+7)$	$(\xi-4, \xi, \xi+4)$ $(\xi-4, \xi, \xi+4)$ $(\xi-3, \xi, \xi+3)$	$(\xi-5, \xi, \xi+5)$ $(\xi-3, \xi, \xi+3)$ $(\xi-2, \xi, \xi+2)$
3	$(\xi-1, \xi, \xi+1)$ $(\xi-4, \xi, \xi+4)$ $(\xi-6, \xi, \xi+6)$	$(\xi-2, \xi, \xi+2)$ $(\xi-7, \xi, \xi+7)$ $(\xi-5, \xi, \xi+5)$	∞	$(\xi-4, \xi, \xi+4)$ $(\xi-2, \xi, \xi+2)$ $(\xi-3, \xi, \xi+3)$	$(\xi-5, \xi, \xi+5)$ $(\xi-5, \xi, \xi+5)$ $(\xi-7, \xi, \xi+7)$	$(\xi-3, \xi, \xi+3)$ $(\xi-11, \xi, \xi+11)$ $(\xi-10, \xi, \xi+10)$	$(\xi-12, \xi, \xi+12)$ $(\xi-2, \xi, \xi+2)$ $(\xi-1, \xi, \xi+1)$	$(\xi-9, \xi, \xi+9)$ $(\xi-11, \xi, \xi+11)$ $(\xi-4, \xi, \xi+4)$
4	$(\xi-7, \xi, \xi+7)$ $(\xi-4, \xi, \xi+4)$ $(\xi-2, \xi, \xi+2)$	$(\xi-7, \xi, \xi+7)$ $(\xi-7, \xi, \xi+7)$ $(\xi-7, \xi, \xi+7)$	$(\xi-3, \xi, \xi+3)$ $(\xi-2, \xi, \xi+2)$ $(\xi-5, \xi, \xi+5)$	∞	$(\xi-12, \xi, \xi+12)$ $(\xi-5, \xi, \xi+5)$ $(\xi-3, \xi,\xi+3)$	$(\xi-10, \xi, \xi+10)$ $(\xi-12, \xi, \xi+12)$ $(\xi-4, \xi, \xi+4)$	$(\xi-1, \xi, \xi+1)$ $(\xi-4, \xi, \xi+4)$ $(\xi-6, \xi, \xi+6)$	$(\xi-11, \xi, \xi+11)$ $(\xi-7, \xi, \xi+7)$ $(\xi-12, \xi, \xi+12)$
5	$(\xi-5, \xi, \xi+5)$ $(\xi-2, \xi, \xi+2)$ $(\xi-10, \xi, \xi+10)$	$(\xi-6, \xi, \xi+6)$ $(\xi-7, \xi, \xi+7)$ $(\xi-5, \xi, \xi+5)$	$(\xi-3, \xi, \xi+3)$ $(\xi-2, \xi, \xi+2)$ $(\xi-3, \xi, \xi+3)$	$(\xi-4, \xi, \xi+4)$ $(\xi-8, \xi, \xi+8)$ $(\xi-7, \xi, \xi+7)$	∞	$(\xi-3, \xi, \xi+3)$ $(\xi-3, \xi, \xi+3)$ $(\xi-5, \xi, \xi+5)$	$(\xi-4, \xi, \xi+4)$ $(\xi-11, \xi, \xi+11)$ $(\xi-6, \xi, \xi+6)$	$(\xi-6, \xi, \xi+6)$ $(\xi-2, \xi, \xi+2)$ $(\xi-7, \xi, \xi+7)$
6	$(\xi-9, \xi, \xi+9)$ $(\xi-4, \xi, \xi+4)$ $(\xi-4, \xi, \xi+4)$	$(\xi-7, \xi, \xi+7)$ $(\xi-6, \xi, \xi+6)$ $(\xi-7, \xi, \xi+7)$	$(\xi-3, \xi, \xi+3)$ $(\xi-2, \xi, \xi+2)$ $(\xi-5, \xi, \xi+5)$	$(\xi-4, \xi, \xi+4)$ $(\xi-3, \xi, \xi+3)$ $(\xi-3, \xi, \xi+3)$	$(\xi-11, \xi, \xi+11)$ $(\xi-12, \xi, \xi+12)$ $(\xi-1, \xi, \xi+1)$	∞	$(\xi-1, \xi, \xi+1)$ $(\xi-4, \xi, \xi+4)$ $(\xi-11, \xi, \xi+11)$	$(\xi-3, \xi, \xi+3)$ $(\xi-5, \xi, \xi+5)$ $(\xi-9, \xi, \xi+9)$
7	$(\xi-2, \xi, \xi+2)$ $(\xi-4, \xi, \xi+4)$ $(\xi-6, \xi, \xi+6)$	$(\xi-3, \xi, \xi+3)$ $(\xi-7, \xi, \xi+7)$ $(\xi-5, \xi, \xi+5)$	$(\xi-4, \xi, \xi+4)$ $(\xi-2, \xi, \xi+2)$ $(\xi-3, \xi, \xi+3)$	$(\xi-6, \xi, \xi+6)$ $(\xi-5, \xi, \xi+5)$ $(\xi-7, \xi, \xi+7)$	$(\xi-5, \xi, \xi+5)$ $(\xi-12, \xi, \xi+12)$ $(\xi-5, \xi, \xi+5)$	$(\xi-7, \xi, \xi+7)$ $(\xi-4, \xi, \xi+4)$ $(\xi-2, \xi, \xi+2)$	∞	$(\xi-2, \xi, \xi+2)$ $(\xi-7, \xi, \xi+7)$ $(\xi-1, \xi, \xi+1)$
8	$(\xi-8, \xi, \xi+8)$ $(\xi-5, \xi, \xi+5)$ $(\xi-12, \xi, \xi+12)$	$(\xi-7, \xi, \xi+7)$ $(\xi-6, \xi, \xi+6)$ $(\xi-7, \xi, \xi+7)$	$(\xi-3, \xi, \xi+3)$ $(\xi-2, \xi, \xi+2)$ $(\xi-5, \xi, \xi+5)$	$(\xi-4, \xi, \xi+4)$ $(\xi-2, \xi, \xi+2)$ $(\xi-3, \xi, \xi+3)$	$(\xi-5, \xi, \xi+5)$ $(\xi-1, \xi, \xi+1)$ $(\xi-2, \xi, \xi+2)$	$(\xi-6, \xi, \xi+6)$ $(\xi-11, \xi, \xi+11)$ $(\xi-5, \xi, \xi+5)$	$(\xi-7, \xi, \xi+7)$ $(\xi-10, \xi, \xi+10)$ $(\xi-1, \xi, \xi+1)$	∞
	Fuzzy Rough **Risk** Matrix(10 $\times$10) for FRCMOSTSP With Three Conveyances							
i/j	1	2	3	4	5	6	7	8
1	∞	$(\xi-.15, \xi, \xi+.15)$ $(\xi-.14, \xi, \xi+.14)$ $(\xi-.36, \xi, \xi+.36)$	$(\xi-.16, \xi, \xi+.16)$ $(\xi-.1, \xi, \xi+.1)$ $(\xi-.15, \xi, \xi+.15)$	$(\xi-.13, \xi, \xi+.13)$ $(\xi.08, \xi, \xi+.08)$ $(\xi-.13, \xi, \xi+.13)$	$(\xi-.19, \xi, \xi+.19)$ $(\xi-.25, \xi, \xi+.25)$ $(\xi-.05, \xi, \xi+.05)$	$(\xi-.17, \xi, \xi+.17)$ $(\xi-.11, \xi, \xi+.11)$ $(\xi-.1, \xi, \xi+.1)$	$(\xi-.1, \xi, \xi+.1)$ $(\xi-.18, \xi, \xi+.18)$ $(\xi-.17, \xi, \xi+.17)$	$(\xi-.12, \xi, \xi+.12)$ $(\xi-.21, \xi, \xi+.21)$ $(\xi-.21, \xi, \xi+.21)$
2	$(\xi-.18, \xi, \xi+.18)$ $(\xi-.25, \xi, \xi+.25)$ $(\xi-.04, \xi, \xi+.04)$	∞	$(\xi-.17, \xi, \xi+.17)$ $(\xi-.06, \xi, \xi+.06)$ $(\xi-.07, \xi, \xi+.07)$	$(\xi-.13, \xi, \xi+.13)$ $(\xi-.12, \xi, \xi+.12)$ $(\xi-.05, \xi, \xi+.05)$	$(\xi-.14, \xi, \xi+.14)$ $(\xi-.23, \xi, \xi+.23)$ $(\xi-.03, \xi, \xi+.03)$	$(\xi-.1, \xi, \xi+.1)$ $(\xi-.1, \xi, \xi+.1)$ $(\xi-.1, \xi, \xi+.1)$	$(\xi-.16, \xi, \xi+.16)$ $(\xi-.11, \xi, \xi+.11)$ $(\xi-.11, \xi, \xi+.11)$	$(\xi-.21, \xi, \xi+.21)$ $(\xi-.14, \xi, \xi+.14)$ $(\xi-.4, \xi, \xi+.4)$
3	$(\xi-.15, \xi, \xi+.15)$ $(\xi-.14, \xi, \xi+.14)$ $(\xi-.06, \xi, \xi+.06)$	$(\xi-.16, \xi, \xi+.16)$ $(\xi-.17, \xi, \xi+.17)$ $(\xi-.05, \xi, \xi+.05)$	∞	$(\xi-.13, \xi, \xi+.13)$ $(\xi-.12, \xi, \xi+.12)$ $(\xi-.03, \xi, \xi+.03)$	$(\xi-.12, \xi, \xi+.12)$ $(\xi-.15, \xi, \xi+.15)$ $(\xi-.07, \xi, \xi+.07)$	$(\xi-.1, \xi, \xi+.1)$ $(\xi-.1, \xi, \xi+.1)$ $(\xi-.1, \xi, \xi+.1)$	$(\xi-.21, \xi, \xi+.21)$ $(\xi-.22, \xi, \xi+.22)$ $(\xi-.2, \xi, \xi+.2)$	$(\xi-.03, \xi, \xi+.03)$ $(\xi-.03, \xi, \xi+.03)$ $(\xi-.3, \xi, \xi+.3)$
4	$(\xi-.17, \xi, \xi+.17)$ $(\xi-.16, \xi, \xi+.16)$ $(\xi-.12, \xi, \xi+.12)$	$(\xi-.07, \xi, \xi+.07)$ $(\xi-.06, \xi, \xi+.06)$ $(\xi-.07, \xi, \xi+.07)$	$(\xi-.13, \xi, \xi+.13)$ $(\xi-.12, \xi, \xi+.12)$ $(\xi-.05, \xi, \xi+.05)$	∞	$(\xi-.12, \xi, \xi+.12)$ $(\xi-.21, \xi, \xi+.21)$ $(\xi-.03, \xi,\xi+.03)$	$(\xi-.1, \xi, \xi+.1)$ $(\xi-.1, \xi, \xi+.1)$ $(\xi-.12, \xi, \xi+.12)$	$(\xi-.2, \xi, \xi+.2)$ $(\xi-.13, \xi, \xi+.13)$ $(\xi-.13, \xi, \xi+.13)$	$(\xi-.11, \xi, \xi+.11)$ $(\xi-.12, \xi, \xi+.12)$ $(\xi-.3, \xi, \xi+.3)$
5	$(\xi-.15, \xi, \xi+.15)$ $(\xi-.14, \xi, \xi+.14)$ $(\xi-.16, \xi, \xi+.16)$	$(\xi-.16, \xi, \xi+.16)$ $(\xi-.17, \xi, \xi+.17)$ $(\xi-.15, \xi, \xi+.15)$	$(\xi-.13, \xi, \xi+.13)$ $(\xi-.12, \xi, \xi+.12)$ $(\xi-.13, \xi, \xi+.13)$	$(\xi-.15, \xi, \xi+.15)$ $(\xi-.05, \xi, \xi+.05)$ $(\xi-.17, \xi, \xi+.17)$	∞	$(\xi-.1, \xi, \xi+.1)$ $(\xi-.1, \xi, \xi+.1)$ $(\xi-.1, \xi, \xi+.1)$	$(\xi-.21, \xi, \xi+.21)$ $(\xi-.01, \xi, \xi+.01)$ $(\xi-.11, \xi, \xi+.11)$	$(\xi-.12, \xi, \xi+.12)$ $(\xi-.31, \xi, \xi+.31)$ $(\xi-.31, \xi, \xi+.31)$
6	$(\xi-.18, \xi, \xi+.18)$ $(\xi-.15, \xi, \xi+.15)$ $(\xi-.04, \xi, \xi+.04)$	$(\xi-.17, \xi, \xi+.17)$ $(\xi-.06, \xi, \xi+.06)$ $(\xi-.07, \xi, \xi+.07)$	$(\xi-.13, \xi, \xi+.13)$ $(\xi-.2, \xi, \xi+.2)$ $(\xi-.05, \xi, \xi+.05)$	$(\xi-.04, \xi, \xi+.04)$ $(\xi-.12, \xi, \xi+.12)$ $(\xi-.03, \xi, \xi+.03)$	$(\xi-.1, \xi, \xi+.1)$ $(\xi-.1, \xi, \xi+.1)$ $(\xi-.1, \xi, \xi+.1)$	∞	$(\xi-.11, \xi, \xi+.11)$ $(\xi-.13, \xi, \xi+.13)$ $(\xi-.11, \xi, \xi+.11)$	$(\xi-.21, \xi, \xi+.21)$ $(\xi-.14, \xi, \xi+.14)$ $(\xi-.12, \xi, \xi+.12)$
7	$(\xi-.15, \xi, \xi+.15)$ $(\xi-.14, \xi, \xi+.14)$ $(\xi-.06, \xi, \xi+.06)$	$(\xi-.06, \xi, \xi+.06)$ $(\xi-.07, \xi, \xi+.07)$ $(\xi-.15, \xi, \xi+.15)$	$(\xi-.3, \xi, \xi+.3)$ $(\xi-.02, \xi, \xi+.02)$ $(\xi-.13, \xi, \xi+.13)$	$(\xi-.15, \xi, \xi+.15)$ $(\xi-.05, \xi, \xi+.05)$ $(\xi-.07, \xi, \xi+.07)$	$(\xi-.1, \xi, \xi+.1)$ $(\xi-.1, \xi, \xi+.1)$ $(\xi-.1, \xi, \xi+.1)$	$(\xi-.21, \xi, \xi+.21)$ $(\xi-.1, \xi, \xi+.1)$ $(\xi-.11, \xi, \xi+.11)$	∞	$(\xi-.4, \xi, \xi+.4)$ $(\xi-.15, \xi, \xi+.15)$ $(\xi-.2, \xi, \xi+.2)$
8	$(\xi-.08, \xi, \xi+.08)$ $(\xi-.15, \xi, \xi+.15)$ $(\xi-.04, \xi, \xi+.04)$	$(\xi-.17, \xi, \xi+.17)$ $(\xi-.16, \xi, \xi+.16)$ $(\xi-.07, \xi, \xi+.07)$	$(\xi-.13, \xi, \xi+.13)$ $(\xi-.12, \xi, \xi+.12)$ $(\xi-.05, \xi, \xi+.05)$	$(\xi-.04, \xi, \xi+.04)$ $(\xi-.02, \xi, \xi+.02)$ $(\xi-.03, \xi, \xi+.03)$	$(\xi-.13, \xi, \xi+.13)$ $(\xi-.21, \xi, \xi+.21)$ $(\xi-.04, \xi, \xi+.04)$	$(\xi-.21, \xi, \xi+.21)$ $(\xi-.21, \xi, \xi+.21)$ $(\xi-.14, \xi, \xi+,14)$	$(\xi-.12, \xi, \xi+.12)$ $(\xi-.14, \xi, \xi+.14)$ $(\xi-.13, \xi, \xi+.13)$	∞

Table 6.34: Results of FRCMOSTSP (Model 6.2D)

α	α_1	α_2	β	β_1	β_2	Algorithm	ODM	Path(Vehicle)	Costs	R_{max}
						R-MOGA-I	ODM	3(1)-2(3)-7(3)-8(1)-6(2)-1(1)-5(2)-4(3)	[166.32, 132.6]	
						R-MOGA-III	ODM	3(4)-2(3)-7(3)-8(2)-6(2)-1(1)-5(2)-4(2)	[151.3, 126.7]	8.5
						R-MOGA-I	ODM	5(1)-2(2)-4(1)-3(3)-8(1)-6(2)-1(3)-7(3)	[143.5, 97.5]	
						R-MOGA-III	ODM	5(3)-2(2)-4(3)-3(3)-8(1)-6(2)-1(3)-7(1)	[134.5, 112.3]	
.9	.9	.9	.9	.9	.9	R-MOGA-I	ODM	2(2)-3(1)-7(2)-5(1)-4(2)-1(3)-8(1)-6(1)	[143.7, 121.2]	
						R-MOGA-III	ODM	1(2)-7(1)-5(2)-3(1)-4(2)-2(3)-6(1)-8(1)	[118.3, 132.2]	7.5
						MOGA	ODM	6(1)-2(2)-4(1)-3(3)-8(1)-5(2)-1(3)-7(3)	[211.2, 101.5]	7.0
						R-MOGA-III	ODM	6(1)-2(2)-4(1)-3(3)-8(1)-5(2)-1(3)-7(3)	[151.4, 110.2]	
.95	.95	.95	.95	.95	.95	R-MOGA-I	ODM	6(2)-8(1)-7(1)-1(1)-5(3)-2(3)-3(1)-4(2)	[145.2, 132.7]	7.5
						R-MOGA-III	ODM	6(2)-8(1)-7(1)-1(1)-5(3)-2(3)-3(1)-4(2)	[141.7, 138.2]	6.75
.97	.97	.97	.97	.97	.97	R-MOGA-I	ODM	4(3)-7(1)-8(1)-6(2)-1(2)-5(1)-2(1)-3(1)	[175.4, 108.4]	
						R-MOGA-III	ODM	4(3)-7(1)-8(1)-6(2)-1(2)-5(1)-2(1)-3(1)	[163.7, 107.3]	6.5
.99	.99	.99	.99	.99	.99	R-MOGA-I	ODM	3(1)-4(3)-2(3)-5(3)-8(1)-6(1)-1(3)-7(2)	[161.2, 123.1]	
						R-MOGA-III	ODM	3(1)-4(3)-2(3)-5(3)-8(1)-6(1)-1(3)-7(2)	[132.3, 112.2]	6.0

We set L(x)=1-x, as references function where right spreads are respectively δ^{cT}, δ^{tT}, δ^{rT} for cost ,time and risk values and left spreads are γ^{cT} and γ^{tT} for cost and time are given.

Model 6.2E: CMOSTSP in Random rough Environment (RRCMOSTSP):

Here we have taken the costs, times and risk/discomfort as random rough values for the CMOSTSP. Also we consider three types of conveyances. We assume that the costs, times and risks as maintaining normal distribution with exception as rough variables. Now c$\sim$N(ξ,σ) where expectation ξ is a rough variable find in the corresponding position on Table 6.31 and the normal distributed values given below in Table 6.35. Also σ is the corresponding standard deviation. The random rough costs, times matrices for the CMOSTSP and corresponding random rough risk/discomfort matrix are represented in Table 6.35.

Here we took permissible probability levels β = β_1 =β_2= 0.9, With these data, The RRCMOSTSP model is solved by R-MOGA for different values of αs and βs and the optimum results are presented in Table 6.36.

6.3.4 Statistical test and Sensitivity Analyses

Performance Measure for R-MOGAs:

Unlike in single objective optimization, there are two goals in a bi-objective optimization problem. The first goal is to achieve the convergence to the Pareto optimal set and second one is preserve the diversity in solutions of the given

Table 6.35: Input Data: RRCMOSTSP (Model 6.2E)

	Random Rough **Cost** Matrix(10×10) for RRCSTSP With Three Conveyances							
i/j	1	2	3	4	5	6	7	8
1	∞	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)]$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,3)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,4)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,4)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,4)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,4)$
2	$c\sim N(\xi,1)$ $c\sim N(\xi,3)$ $c\sim N(\xi,4)$	∞	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,1)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	$c\sim N(\xi,2)]$ $c\sim N(\xi,2)$ $c\sim N(\xi,1)$	$c\sim N(\sim,2)$ $c\sim N(\xi,3)$ $c\sim N(\xi,1)$	$c\sim N(\xi,1)$ $c\sim N(\xi,3)$ $c\sim N(\xi,6)$	$c\sim N(\xi,3)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$
3	$c\sim N(\xi,2)$ $c\sim N(\xi,3)$ $c\sim N(\xi,2)$	$c\sim N(\xi,3)$ $c\sim N(\xi,1)$ $c\sim N(\xi,2)$	∞	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,1)$	$c\sim N(\xi,1)$ $c\sim N(\xi,3)$ $c\sim N(\xi,5$	$c\sim N(\xi,1)$ $c\sim N(\xi,1)$ $c\sim N(\xi,3)$	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,2)$	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$
4	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,4)$	$c\sim N(\xi,4)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	∞	$c\sim N(\xi,3)$ $c\sim N(\xi,3)$ $c\sim N(\xi,4)$	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,5)$	$c\sim N(\xi,3)$ $c\sim N(\xi,6)$ $c\sim N(\xi,4)$	$c\sim N(\xi,3)$ $c\sim N(\xi,2)$ $c\sim N(\xi,6)$
5	$c\sim N(\xi,1)$ $c\sim N(\xi,1)$ $c\sim N(\xi,2)$	$c\sim N(\xi,3)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	∞	$c\sim N(\xi,3)$ $c\sim N(\xi,1)$ $c\sim N(\xi,7)$	$c\sim N(\xi,2)$ $c\sim N(\xi,8)$ $c\sim N(\xi,1)$	$c\sim N(\xi,2)$ $c\sim N(\xi,4)$ $c\sim N(\xi,4)$
6	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,2)$	$c\sim N(\xi,5)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	$c\sim N(\xi,2)$ $c\sim N(\xi,3)$ $c\sim N(\xi,3)$	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,4)$	∞	$c\sim N(\xi,5)$ $c\sim N(\xi,9)$ $c\sim N(\xi,8)$	$c\sim N(\xi,6)$ $c\sim N(\xi,4)$ $c\sim N(\xi,5)$
7	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,1)$	$c\sim N(\xi,3)$ $c\sim N(\xi,4)$ $c\sim N(\xi,2)$	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,1)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	$c\sim N(\xi,9)$ $c\sim N(\xi,4)$ $c\sim N(\xi,7)$	∞	$c\sim N(\xi,6)$ $c\sim N(\xi,6)$ $c\sim N(\xi,7)$
8	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,1)$ $c\sim N(\xi,4)$ $c\sim N(\xi,3)$	$c\sim N(\xi,3)$ $c\sim N(\xi,1)$ $c\sim N(\xi,2)$	$c\sim N(\xi,2)$ $c\sim N(\xi,3)$ $c\sim N(\xi,2)$	$c\sim N(\xi,3)$ $c\sim N(\xi,2)$ $c\sim N(\xi,1)$	$c\sim N(\xi,8)$ $c\sim N(\xi,7)$ $c\sim N(\xi,3)$	$c\sim N(\xi,9)$ $c\sim N(\xi,6)$ $c\sim N(\xi,5)$	∞
	Random Rough **Time** Matrix(10×10) for RRCSTSP With Three Conveyances							
i/j	1	2	3	4	5	6	7	8
1	∞	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)]$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,3)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,4)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,4)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,4)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,4)$
2	$c\sim N(\xi,1)$ $c\sim N(\xi,3)$ $c\sim N(\xi,4)$	∞	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,1)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	$c\sim N(\xi,2)]$ $c\sim N(\xi,2)$ $c\sim N(\xi,1)$	$c\sim N(\sim,2)$ $c\sim N(\xi,3)$ $c\sim N(\xi,1)$	$c\sim N(\xi,1)$ $c\sim N(\xi,3)$ $c\sim N(\xi,6)$	$c\sim N(\xi,3)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$
3	$c\sim N(\xi,2)$ $c\sim N(\xi,3)$ $c\sim N(\xi,2)$	$c\sim N(\xi,3)$ $c\sim N(\xi,1)$ $c\sim N(\xi,2)$	∞	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,1)$	$c\sim N(\xi,1)$ $c\sim N(\xi,3)$ $c\sim N(\xi,5$	$c\sim N(\xi,1)$ $c\sim N(\xi,1)$ $c\sim N(\xi,3)$	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,2)$	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$
4	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,4)$	$c\sim N(\xi,4)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	∞	$c\sim N(\xi,3)$ $c\sim N(\xi,3)$ $c\sim N(\xi,4)$	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,5)$	$c\sim N(\xi,3)$ $c\sim N(\xi,6)$ $c\sim N(\xi,4)$	$c\sim N(\xi,3)$ $c\sim N(\xi,2)$ $c\sim N(\xi,6)$
5	$c\sim N(\xi,1)$ $c\sim N(\xi,1)$ $c\sim N(\xi,2)$	$c\sim N(\xi,3)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	∞	$c\sim N(\xi,3)$ $c\sim N(\xi,1)$ $c\sim N(\xi,7)$	$c\sim N(\xi,2)$ $c\sim N(\xi,8)$ $c\sim N(\xi,1)$	$c\sim N(\xi,2)$ $c\sim N(\xi,4)$ $c\sim N(\xi,4)$
6	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,2)$	$c\sim N(\xi,5)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	$c\sim N(\xi,2)$ $c\sim N(\xi,3)$ $c\sim N(\xi,3)$	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,4)$	∞	$c\sim N(\xi,5)$ $c\sim N(\xi,9)$ $c\sim N(\xi,8)$	$c\sim N(\xi,6)$ $c\sim N(\xi,4)$ $c\sim N(\xi,5)$
7	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,1)$	$c\sim N(\xi,3)$ $c\sim N(\xi,4)$ $c\sim N(\xi,2)$	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,1)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	$c\sim N(\xi,9)$ $c\sim N(\xi,4)$ $c\sim N(\xi,7)$	∞	$c\sim N(\xi,6)$ $c\sim N(\xi,6)$ $c\sim N(\xi,7)$
8	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,1)$ $c\sim N(\xi,4)$ $c\sim N(\xi,3)$	$c\sim N(\xi,3)$ $c\sim N(\xi,1)$ $c\sim N(\xi,2)$	$c\sim N(\xi,2)$ $c\sim N(\xi,3)$ $c\sim N(\xi,2)$	$c\sim N(\xi,3)$ $c\sim N(\xi,2)$ $c\sim N(\xi,1)$	$c\sim N(\xi,8)$ $c\sim N(\xi,7)$ $c\sim N(\xi,3)$	$c\sim N(\xi,9)$ $c\sim N(\xi,6)$ $c\sim N(\xi,5)$	∞
	Random Rough **Risk** Matrix(10×10) for RRCSTSP With Three Conveyances							
i/j	1	2	3	4	5	6	7	8
1	∞	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)]$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,3)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,4)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,4)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,4)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,4)$
2	$c\sim N(\xi,1)$ $c\sim N(\xi,3)$ $c\sim N(\xi,4)$	∞	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,1)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	$c\sim N(\xi,2)]$ $c\sim N(\xi,2)$ $c\sim N(\xi,1)$	$c\sim N(\sim,2)$ $c\sim N(\xi,3)$ $c\sim N(\xi,1)$	$c\sim N(\xi,1)$ $c\sim N(\xi,3)$ $c\sim N(\xi,6)$	$c\sim N(\xi,3)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$

3	$c\sim N(\xi,2)$ $c\sim N(\xi,3)$ $c\sim N(\xi,2)$	$c\sim N(\xi,3)$ $c\sim N(\xi,1)$ $c\sim N(\xi,2)$	∞	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,1)$	$c\sim N(\xi,1)$ $c\sim N(\xi,3)$ $c\sim N(\xi,5$	$c\sim N(\xi,1)$ $c\sim N(\xi,1)$ $c\sim N(\xi,3)$	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,2)$	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$
4	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,4)$	$c\sim N(\xi,4)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	∞	$c\sim N(\xi,3)$ $c\sim N(\xi,3)$ $c\sim N(\xi,4)$	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,5)$	$c\sim N(\xi,3)$ $c\sim N(\xi,6)$ $c\sim N(\xi,4)$	$c\sim N(\xi,3)$ $c\sim N(\xi,2)$ $c\sim N(\xi,6)$
5	$c\sim N(\xi,1)$ $c\sim N(\xi,1)$ $c\sim N(\xi,2)$	$c\sim N(\xi,3)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	∞	$c\sim N(\xi,3)$ $c\sim N(\xi,1)$ $c\sim N(\xi,7)$	$c\sim N(\xi,2)$ $c\sim N(\xi,8)$ $c\sim N(\xi,1)$	$c\sim N(\xi,2)$ $c\sim N(\xi,4)$ $c\sim N(\xi,4)$
6	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,2)$	$c\sim N(\xi,5)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	$c\sim N(\xi,2)$ $c\sim N(\xi,3)$ $c\sim N(\xi,3)$	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,4)$	∞	$c\sim N(\xi,5)$ $c\sim N(\xi,9)$ $c\sim N(\xi,8)$	$c\sim N(\xi,6)$ $c\sim N(\xi,4)$ $c\sim N(\xi,5)$
7	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,1)$	$c\sim N(\xi,3)$ $c\sim N(\xi,4)$ $c\sim N(\xi,2)$	$c\sim N(\xi,1)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	$c\sim N(\xi,2)$ $c\sim N(\xi,1)$ $c\sim N(\xi,1)$	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,3)$	$c\sim N(\xi,9)$ $c\sim N(\xi,4)$ $c\sim N(\xi,7)$	∞	$c\sim N(\xi,6)$ $c\sim N(\xi,6)$ $c\sim N(\xi,7)$
8	$c\sim N(\xi,2)$ $c\sim N(\xi,2)$ $c\sim N(\xi,2)$	$c\sim N(\xi,1)$ $c\sim N(\xi,4)$ $c\sim N(\xi,3)$	$c\sim N(\xi,3)$ $c\sim N(\xi,1)$ $c\sim N(\xi,2)$	$c\sim N(\xi,2)$ $c\sim N(\xi,3)$ $c\sim N(\xi,2)$	$c\sim N(\xi,3)$ $c\sim N(\xi,2)$ $c\sim N(\xi,1)$	$c\sim N(\xi,8)$ $c\sim N(\xi,7)$ $c\sim N(\xi,3)$	$c\sim N(\xi,9)$ $c\sim N(\xi,6)$ $c\sim N(\xi,5)$	∞

Table 6.36: Results of RRCSTSP(Model 6.2E)

α	α_1	α_2	Algorithm	Path(Vehicle)	Costs & Times	R_{max}
.9	.9	.9	R-MOGA-I	4(2)-2(3)-7(3)-8(1)-6(3)-1(1)-5(2)-3(1)	[148.56, 102.43]	8.5
			R-MOGA-III	4(2)-2(3)-7(3)-8(1)-6(3)-1(1)-5(2)-3(1)	[140.13, 113.86]	
			R-MOGA-I	6(3)-2(2)-4(1)-3(3)-8(1)-5(2)-1(1)-7(2)	[151.21, 99.32]	
			R-MOGA-III	6(3)-2(2)-4(1)-3(3)-8(1)-5(2)-113)-7(2)	[147.18, 104.51]	
			R-MOGA-I	1(3)-5(1)-7(2)-3(1)-4(2)-2(3)-8(1)-6(1)	[166.25, 94.73]	6.75
			R-MOGA-III	1(3)-5(1)-7(2)-3(1)-4(2)-2(3)-8(1)-6(1)	[151.31, 98.31]	
			MOGA	6(1)-2(2)-4(1)-3(3)-8(1)-5(2)-1(3)-7(3)	[169.21, 118.62]	6.0
			R-MOGA-III	6(1)-2(2)-4(1)-3(3)-8(1)-5(2)-1(3)-7(3)	[162.45, 115.75]	
.95	.95	.95	R-MOGA-I	3(2)-8(1)-7(1)-1(1)-5(3)-2(3)-4(1)-6(2)	[155.76, 124.84]	6.75
			R-MOGA-III	4(1)-8(3)-7(1)-1(1)-5(3)-2(3)-3(1)-6(2)	[142.18, 106.57]	
.99	.99	.99	R-MOGA-I	5(3)-7(1)-8(1)-6(2)-1(2)-4(1)-2(1)-3(1)	[161.34, 97.43]	6.5
			R-MOGA-III	4(3)-7(1)-8(1)-6(2)-1(2)-5(1)-2(1)-3(1)	[164.13, 95.38]	
.95	.95	.95	R-MOGA-I	1(3)-4(3)-2(3)-5(3)-8(1)-6(1)-3(3)-7(3)	[168.45, 100.37]	6.0
			R-MOGA-III	1(1)-4(3)-2(3)-5(3)-8(1)-6(1)-1(3)-7(2)	[146.93, 107.64]	

Table 6.37: Mean and Variance of the diversity metric

Algorithm	SCH	FON	POL	KUR	ZDT1	ZDT2	ZDT3	ZTD4
NSGA-II	0.031223	0.32725	0.423164	0.024210	0.345122	0.535073	0.117382	0.414373
	0.00135	0.017217	0.021836	0.035364	0.015125	0.032531	0.032163	0.012172
MOGA-I	0.052943	0.453495	0.310266	0.231765	0.3548221	0.213267	0.076296	0.197286
	0.002769	0.036234	0.006362	0.003368	0.001513	0.003537	0.003164	0.023154
R-MOGA-III	0.030145	0.220041	0.206712	0.010234	0.23412	0.182748	0.2917521	0.319317
	0.001081	0.001918	0.010366	0.002619	0.0010003	0.001891	0.000902	0.001001

Table 6.38: Mean and Variance of the convergence metric

Algorithm	SCH	FON	POL	KUR	ZDT1	ZDT2	ZDT3	ZTD4
NSGA-II	0.003287	0.023942	0.020341	0.003275	0.125612	0.003562	0.001382	0.014317
	0.000156	0.013231	0.001367	0.001364	0.000512	0.000538	0.002162	0.013183
MOGA-I	0.003162	0.024534	0.010261	0.004763	0.025481	0.013248	0.002373	0.021928
	0.001505	0.005413	0.000721	0.001421	0.001639	0.000456	0.002036	0.000218
R-MOGA-III	0.002001	0.001241	0.005113	0.000114	0.02271	0.020157	0.00117	0.007319
	0.000071	0.0004522	0.003267	0.000457	0.000168	0.000301	0.000191	0.001147

Pareto optimal set. Here two performance matrices according Deb et al. [36] are obtained for the multi objective optimization algorithms.

To show the performance of the proposed R-MOGA-III, we used it for some standard multi objective test functions according to Deb et al. [36] and He et al. [65]. Here the test functions are same as in Deb et al. [36] and He et al. [65], each function is compared with the Pareto optimal solutions of proposed R-MOGA-III. For all instances of test functions, we set the parameters as described above and the experimental results are presented in Tables 6.37 and 6.38. In Table 6.38, we compare the mean and standard deviation (δ) of the convergence metric used by Deb et al. [36] for NSGA-II, classical MOGA and proposed R-MOGA-III. This table demands that proposed R-MOGA-III gives better results in the case of mean and standard deviation of the convergence metric. Again from the Table 6.37, we find out the diversity metrics according to Deb et al. [36] using the same parameters against three algorithms NSGA-II, MOGA and R-MOGA-III. From the Table 6.37, it is observed that proposed algorithm gives better results except some few cases.

Efficiency Test for R-MOGAs with other algorithms by ANOVA:

Some standard test problems are solved using the developed algorithm R-

Table 6.39: ANOVA: Number of win for different algorithms

Problem	SCH	FON	POL	KUR	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
R-MOGA-III	81	86	78	85	71	88	91	70	82
NSGA-II	67	76	66	78	63	68	73	69	71
MOGA	61	51	64	66	61	59	69	57	58

Table 6.40: ANOVA: Subtracted table from Table 6.39

Problem	SCH	FON	POL	KUR	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	Mean
X_1	21	26	18	25	11	28	31	20	22	$\overline{X}_1$=22.44
X_2	7	16	6	18	3	8	13	9	11	$\overline{X}_2$=10.11
X_3	1	-9	4	6	1	-1	9	-3	-2	$\overline{X}_3$=0.67

MOGAs. Different parametric values of R-MOGAs, used for this purpose, are given below:

Here for three algorithms R-MOGA-III, NSGA-II and Classical MOGA, Popsize=100 and Maxgen=2000.

As multi objective standard TSP (test problems) not available in the literature, the algorithm is tested against a list of standard test functions of crisp valued benchmark problems [36]. Results obtained for these standard problems and number of wins for 100 runs of the algorithms R-MOGA-III, NSGA-II and MOGA are presented in Table-19. To compare the efficiency of the developed algorithm, another two established heuristic technique NSGA-II (developed by Deb et al. [36] and used by Changder et al. [24] and classical MOGA are used against these standard test functions and their results (number of wins for 100 runs) are obtained.

For statistical comparison of the results (obtained by these three algorithms), i.e., sample of runs for the algorithms (number of wins for 100 runs), the ANOVA procedure is performed. When a set of algorithms are compared, the common statistical method for testing the differences between more than two related sample means is the repeated-measures ANOVA. Different steps of this ANOVA are as follows.

For calculation of different steps of ANOVA easily, we subtract 60 (with out lose of generality) from each numbers and the Table 6.39 reduces as given below.

Table 6.41: ANOVA: Summary table (data taken from Table 6.40

Source of variation	Sum of square	df	Mean of square	F
Between groups	SS_B=2144.54	J-1=2	$MS_B=\frac{SS_B}{J-1}$=1072.27	$\frac{MS_B}{MS_W}$=41.66
Within groups	SS_W=617.68	J(I-1)=24	$MS_W=\frac{SS_W}{J(I-1)}$=25.73	
Total	SS_T=2762.22	IJ-1=26		

Here, total sample size of each algorithm is equal and say, it I=9 and number of algorithm is say, J=3. Mean of the sample means, $\overline{\overline{X}}$=11.07.

Critical F values, $F_{0.05(2,24)} \approx 3.4$. As the compared F (form Table-19) is higher than the critical F values for 0.05 level of significance, it may be inferred that there is a significant differences between the groups. When F ratio is found to be significant in an ANOVA with more than two groups, it should be followed by a multiple comparison test to find which group means differ significantly from each other. Scheffe's multiple comparison F- test is done for this purpose to find out whether R-MOGA-III & NSGA-II and/or R-MOGA-III & MOGA are significant. For the first pair i.e., for R-MOGA-III & MOGA, calculated F value is given by $F=\frac{(\overline{X}_1-\overline{X}_3)^2}{MS_W(\frac{1}{I}+\frac{1}{J})}$=41.44. Similarly, for the second pair i.e., for R-MOGA-III and NSGA-II, calculated F= 13.3. As both calculated F values are greater than the tabulated values (3.4), there is significant difference between R-MOGA-III & classical MOGA and also R-MOGA-III & NSGA-II. From Table-19 it is observed that the mean ($\overline{X}_1$) of X_1 is higher than the other two means ($\overline{X}_2$ and $\overline{X}_3$). Significant differences between the algorithms are observed (discussion already given) and therefore, it can be concluded that R-MOGA-III is better compared to the other two algorithms.

6.3.5 Discussion

To validate the feasibility and effectiveness of the proposed algorithm, we have applied the R-MOGAs on some standard TSP combinations, problems are taken from TSPLIB [162]. The proposed algorithm is the combination of rough extended based selection, adaptive crossover and generation depended mutation which was implemented in C++ with 200 chromosomes and 2000 iterations in maximum.

To find out the Pareto optimal solution, Table 6.25 shows the comparisons between MOGA and R-MOGA for the some standard TSP problems. It is seen that the number of iterations is less in R-MOGA than classical MOGA, where the classical MOGA are the combinations of RW selection, cyclic crossover and random mutations. Here we consider the multi objective standard TSP from TSPLIB combining the same sizes problems. Again Table 6.25 asserts that the effectiveness of the proposed algorithm with respect to CPU time. In Table 6.26, we survey the importances of different parameters and operators in proposed R-MOGA. It indicates that for the Pareto optimal solution of the standard TSP, combination of **bayg29** and **bays29**, and Pareto optimal solutions show that it navigate the sample space better for generation dependent mutation. In this case, Pareto optimal results are obtained quickly by 169 iterations only. Here also, R-MOGA performs better than the classical MOGA.

In Table 6.28, we consider 10×10 crisp costs and times, risk/discomfort matrices for a CSTSP. The Pareto optimal results are presented in Table 6.29 for only CMOTSP considering single conveyance for the given data in Tables 6.28. It is observed that CMOTSP without any total risk factor as a goal gives the lowest minimum cost and time and as the total risk/discomfort decreases, total cost as well as time increases. It is realistically true in our day-to-day life. For a particular value of risk/discomfort, some near Pareto optimum results along with the Pareto optimum one are presented. Due to some reasons, if the TS fails to implement the optimum results, he/she may be to achieve the most feasible near Pareto optimum solution. Again, we formulated a CMOSTSP with the three conveyances i.e. ($10\times10\times3$) costs, times and risk/discomfort matrices and the data are presented in Table 6.28. Along each route, the corresponding conveyance is in parentheses. Next the Pareto optimum results of CMOSTSP are given in Table 6.30. Here also as total risk/discomfort goes down, the corresponding travelling cost and time increases. A ($10\times10\times3$) RCMOSTSP is presented in Table 6.31 where all costs, times and risk/discomfort factors are rough variable. The Pareto optimum results are presented in Table 6.32. As expected, as the risk goes down, corresponding costs and times compromise each other and go up. For fuzzy rough CMOSTSP, fuzzy rough input data and Pareto optimum results are presented in Table 6.33. Here, costs, times and risk/discomfort factors are L-R fuzzy numbers. For a fixed β (0.9), results in possibility approaches are given and as before, optimistic (Possibilistic) representation gives better result (less cost, less time) presents. Again random rough input data are given in Table 6.35 with the costs, times and risk factor, where we have taken, mean as random variables (standard normal variate) with expectation is the rough variables with known standard deviation. The results presented in Table 6.36, show that Pareto optimal solution gives costs and times w.r. to risk factor smooth movements as realistic work. In all cases, the near Pareto optimum solutions and Pareto optimum solution are available. Also R-MOGAs gives better results than the classical MOGA.

6.4 Conclusion

In this chapter, two MOGAs, called imprecise MOGA (iMOGA) and Rough MOGA (R-MOGA) are proposed and illustrated in CMOSTSP formulated in different environments. Both the multi-objective algorithms are also tested with some test problems from TSPLIB [162] and compared with classical MOGA and NSGA-II. In iMOGA, fuzzy age based and fuzzy extended age based selection operators are used where as for R-MOGA, a rough age, rough extended age based with adaptive crossover are used along with generation dependent mutations. Such CMOSTSPs are here formulated with crisp, rough, random, fuzzy-random, random-fuzzy, bi-random, fuzzy rough and random rough costs, times and risk/discomfort levels and solved by the proposed algorithms. Here, development of multi objective algorithms are in general form and it can be applied in other discrete problems such as network optimization, graph theory, solid transportation problems, vehicle routing, VLSI chip design, etc. In spite of the better results by proposed algorithms, there is a lot of scope for variation in iMOGA and R-MOGAs, specially with respect to the CMOSTSPs. In three dimensional TSPs with conveyances, we have assigned a conveyance arbitrarily during each crossover and mutation for the optimum selection of the routes. This is a limitation of the present CMOSTSPs. The formulated CMOSTSPs can be extended to include some more features /restrictions such as 4DTSP to include/omit some specific routes, time windows, multi-TS, profit maximization, etc.

.

Part IV

Summary And Future Research Scope

Chapter 7

Summary and Future Research Scope

In this dissertation, main objectives are (i) to develop/modify some evolutionary methods, specially Genetic Algorithm (GA), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO), (ii) to develop some hybrid evolutionary methods connecting GA, ACO and PSO and (iii) to formulate some new uncertain (random and imprecise) singe/ multi-objective TSP problems and to solve them using the developed evolutionary methods (GAs) and hybrid evolutionary methods.

Here, for the first time, constrained single/multi-objective some 3D- and 4D-TSPs have been formulated in crisp, fuzzy, random, bi-random, bi-fuzzy, rough, bi-rough, fuzzy-rough, fuzzy-random, random-fuzzy, random-rough, etc, environments. Some uncertain risk/safety constraints along the routes and time constraints for the tour are also imposed. These virgin problems have been solved by developed evolutionary and hybrid evolutionary methods. Restrictions on vehicles and paths are imposed in 3D- and 4D- TSPs respectively. In this thesis, in developing different GAs, nine (9) different types of selections such as probabilistic selection, probability of selection parameter, fuzzy age based, fuzzy -extended age based, rough age based (3, 5 and 7 point scale), rough pheromone based, three (3) types of crossover such as adaptive crossover, comparison crossover, min-point crossover and five (5) types of mutations such as nodes oriented, generation dependent (two types), fixed point location and random mutations are used. Also two (2) types of hybrid evolutionary algorithms such as ACO-GA and ACO-PSO-GA have been developed and used. Here, in ACO, rough pheromone has been defined and used. In PSO, swap sequence based updatation of velocities and positions has been used. To the best of my knowledge, none developed and used these operators before.

In total, nine (9) virgin constrained TSP models have been formulated and solved.

These new algorithms have been tested against standard problems from TSPLIB to establish the efficiency of the developed techniques. The results from these methods are statistically tested. The statistical tests include Mean, SD, error analysis, ANOVA, Firedman Test and Post hoc paired comparison.

The thesis has been divided into four parts. Part-I contains three chapters-chapter-1 contains Introduction, chapter-2 preliminaries of algorithms and chapter-3 different uncertainties. Part-II contains two chapters where chapter-4 and chapter-5 contains single objective constraints respectively 3D-TSP and 4D-TSP problems solved by single (GA) and hybrid evolutionary (ACO-GA and ACO-PSO-GA) techniques. In chapter-6, multi-objective TSP problems are formulated and solved by developed evolutionary algorithm. All these TSP models are formulated in uncertain environments.

In chapter-7, Part-IV summary and future extensions along with bibliography and index are given.

Limitations and Future Extension

The present investigation has been confined to the development of different types of GA, ACO,PSO and two hybridization of GA, ACO and PSO in uncertain environments. There are other several evolutionary algorithms such as Artificial Bees Colony (ABC), Tabu search, etc., which can also be extended in uncertain (random and imprecise) environments and can be used tio solve the Np-hard TSP, CSP, VRP, etc., problems.

In three dimensional TSPs with conveyances, we have assigned a conveyance arbitrarily during each crossover and mutation for the optimum selection of the routes. This is a limitation of the present CSTSPs. The formulated CSTSPs can be extended to include some more features /restrictions such as 4DTSP to include/omit some specific routes, time windows, multi-TS, profit maximization, etc. In the literature, these several types of TSPs such as multiple TSP, MAx TSP etc., which can be developed in different uncertain environments not dealt in this thesis and solved by the proposed algorithms.

Part V

Bibliography and Index

Bibliography

[1] Ahuja, R. K., Orlin, J. B., Tiwari, A., *A greedy genetic algorithm for the quadratic assignment problem*, Computers & Operations Research, 27(2000), 917-934.

[2] Albanyrak, M., Allahverdi, N., *Development of a new mutation operator to solve the TSP by genetic Algorithm*, Expert Systems with Application, 38(3)(2011), 1313-1320.

[3] Applegate, D., Bixby, R. E., Chvatal, V., Cook, W., *On the solution of traveling salesman problems*, Documenta Mathematica Journal der Deutschen Mathematiker-Vereinigung, International Congress of Mathematicians, 645-656, 1998.

[4] Applegate, D., Bixby, R. E., Chvatal, V., Cook, W., *TSP cuts outside the template paradigm*, Computer sciences, Rutgers University, 2000.

[5] Applegate, D., Bixby, R. E., Chvatal, V., Cook, W., *Implementing the Dantzig Fulkerson -Johnson Algorithm for Large Traveling Salesman Problems*, Mathematical Programming 97, pp. 91-153, 2003.

[6] Bai, J., Yang, G-K. Chen, Y-W. Hu, L-S, Pan, C-C., *A model induced max-min ant colony optimization for asymmetric travelling salesman problem*, Applied Soft Computing, 13(2013), 1365-1375.

[7] Balcombe, K. G., *Model selection using information criteria and genetic algorithms*,Computational Economics, 25(2005), 207-228.

[8] Bellman, R. E., Kalaba, L., Zadeh, L.A., *Abstraction and pattern classification*, Journal of Mathematical Analysis and Applications, 13(1966), 1-7.

[9] Bellman, R. E., Zadeh, L.A., *Decision making in a fuzzy environment*, Management Science, 17(4)(1970), 141-164.

[10] Bektas, T., *The multiple traveling salesman problem: an overview of formulations and solution procedures*, OMEGA: The International Journal of Management Science, 34(3)(2006), 209-219.

[11] Bertsimas, D. J., Simchi-Levi, D., *A new generation of vehicle routing research:Robust algorithms, addressing uncertainty*. Operations Research, 44(2)(1996), 286-304.

[12] Beyer, H-G., Schwefel, H-P., *Evolution strategies: A comprehensive introduction*. Natural Computing, 1(1)(2002), 3-52.

[13] Bhandari, D., Murthy, C. A., Pal, S. K., *Variance as a Stopping Criterion for Genetic Algorithms with Elitist Model*, Fundamenta Informarticae, 120(2)(2012), 145-164.

[14] Bianchi, L., Dorigo, M., Gambardella, L. M., *Ant colony optimization approach to the probabilistic travelling salesman problem, PPSN VII, LNCS*, 2439(2002), 883-892.

[15] Birge, J. R., Louveaux, F., *Introduction to stochastic programming* (2nd ed.), New York/Dordrecht/Heidelberg/London: Springer, (2011).

[16] Bit, A. K., Biswal, M. P., Alam, S. S., *Fuzzy programming approach to multi-objective solid transportation problem*, Fuzzy Sets and Systems, 57(1993), 183-194.

[17] Bit, A. K., *Fuzzy programming with hyperbolic membership functions for multi-objective capacitated solid transportation problem*, The Journal of Fuzzy Mathematics, 13(2)(2005), 373-385.

[18] Bitran, G. R., *Linear multiple objective problems with interval coefficients*, Management science, 26(7)(1980), 694-706.

[19] Bock, F., *Mathematical programming solution of traveling salesman examples*, In R.L. Graves and P. Wolfe, editors, Recent Advances in Mathematical Programming. McGraw-Hill, New York, 1963.

[20] Bonyadi, M. R., Li, X., Michalewicz, Z., *A hybrid particle swarm with a time-adaptive topology for constrained optimization*, Swarm and Evolutionary Computation, 18(2014), 22-37.

[21] Chang, C. B., Mao, C-P., *A modified ant colony system for solving the travelling salesman problem with time windows*, Mathematical and Computer Modeling, 46(2007), 1225-1235.

[22] Chang, T.-S., Wan, Y.-W., Ooi, W. T., *A stochastic dynamic traveling salesman problem with hard time windows*, European Journal of Operational Research, 198(3)(2009), 748-759.

[23] Changdar, C., Maiti, M. K., Maiti, M., *A Constrained solid TSP in fuzzy environment:two heuristic approaches, Iranian Journal of Fuzzy System*, 10(1)(2013), 1-28.

[24] Changdar, C., Mahapatra, G. S, Pal, R. K., *An efficient genetic algorithm for multi-objective solid travelling salesman problem under fuzziness*, Swarm and Evolutionary Computation, 15(2014), 27-37.

[25] Charnes, A., Cooper, W. W., *Chance Constrained Programming, Management Science*, 6(1959), 73-79.

[26] Chaudhuri1, A., De, K., *A Study of Traveling Salesman Problem Using Fuzzy Self Organizing Map*, Traveling Salesman Problem, Theory and Applications, Prof. Donald Davendra (Ed.), ISBN: 978-953- 307-426-9, InTech.

[27] Che, S. M., Ohiem, C. Y., *Solving TSP based on the genetic simulated annealing ant colony system with PSO*, Expert Systems with Application, 38 (12)(2012), 14439-14450.

[28] Chen, S. H., Hsieh, C. H., *Optimization of fuzzy production inventory model under fuzzy parameters*, Proceedings of the Fifth Joint Conference on Information Science, (JCIS'2000), 1(2000), 1098-1101.

[29] Clerc, M., Kennedy, J., *The particle swarm: explosion stability and convergence in a multi-dimensional complex space*, IEEE Transaction on Evolutionary Computation, 6(1) (2002), 58-73.

[30] Cormen, T. H., Leiserson, C. E., Rivest, R. L., *Introduction to Algorithms*: MIT Press, 1990.

[31] Dantizg, G. B., Fulkerson, D. R., Johnson, S. M., *Solution of a large scale travelling salesman problem*, Operation Research, 2(1954), 393-410.

[32] Dantzig, G. B., *Linear Programming and Extensions*, Princeton University Press, New Jersey (1963).

[33] Das, N., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M., Basu, D. K., *A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application*, Applied Soft Computing, 12(5)(2012), 1592-1606.

[34] Das, N., Acharya, K., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M., *A novel GA-SVM based multistage approach for recognition of handwritten Bangla compound characters*, Proceedings of the ICISDIA-2012, Visakhapatnam, India, January 2012(Springer), 145-152.

[35] Deb, K., *Multi-objective Optimization Using Evolutionary Algorithms*, U.K, Wiley(2001).

[36] Deb, K., *A fast and elitist multi objective genetic algorithm:NSGA-II*, IEEE Trans-actions on Evolutionary Computation, 6(2)(2002), 182-197.

[37] Deb, K., Tiwari, S., *Omni optimizer: A generic evolutionary algorithm for single and multi-objective optimization*, European Journal Of Operational Research, 185(3)(2006), 1062-1087.

[38] De Jong, K. A., Fogel, D. B., Schwefel, H. P., *A history of evolutionary computation*, in Handbook of Evolutionary Computation, T. Bck, D. B. Fogel, and Z. Michalewicz, Eds. Oxford, UK: IOP Oxford University Press, 1997, pp. A2.3:1-11.

[39] Derrac, J., Garcia, S., Molina, D., Herrera, F., *A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms*, Swarm and Evolutionary Computation, 1(2011), 3-18.

[40] Dong, G., Guo, N., Tickle, K., *Solving the travelling salesman problem using cooperative genetic ant systems*, Expert Systems and Applications, 39 (2012), 5006-5011.

[41] Dorigo, M., *Optimization, Learning and Natural Algorithms*, Ph D thesis, Politecnico di Milano, 1992.

[42] Dorigo, M., Caro, G. D., *Ant Colony Optimization: A New Meta-Heuristic*, In Proceedings of The IEEE Congress on Evolutionary Computation, 2(1999), 1477.

[43] Dorigo, M., Gambardella, L. M., *Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman problem*, IEEE Transactions on Evolutionary Computation, 1(1)(1997), 53-66.

[44] Dubois, D., Prade, H., *Fuzzy sets and systems, Theory and applications*, Academic Press (1980).

[45] Dubois, D., Prade, H., *The three semantics of fuzzy sets*, Fuzzy Sets and Systems, 90(1997), 141-150.

[46] Durillo, J. J., Nebro, A. J., Luma, F., Alba, E., *On the effect of steady state selection sme in Multi-objective genetic algorithm*, Proc. 5th. International conference EMO, Nantco, France, 183-197.

[47] Elaoud, S., Teghem, J., Loukil, T., *Multiple crossover genetic algorithm for the multi- objective travelling salesman problem*, Electronics notes in Discrete Mathematics, 36(2010), 939-946.

[48] Enigin, O., Ceran, G., Yilmaz, M. K., *An efficient GA for hybrid flow shop scheduling with multiprocessor task problem*, Applied Soft Computing, 11(3)(2011), 3056-3065.

[49] Falkenauer, E., Delchambre, A., *A genetic algorithm for bin packing and line balancing*, Proc. of the IEEE 1992 Int. Conf. on Robotics and Automation, pp. 1186-1192, 1992.

[50] Fdez, J. A., Alcal, R., Gacto, M. J., Herrera, F., *Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms*, Fuzzy sets and systems, 160(2009), 905-921.

[51] Feng, H. M., Liao, K. L., *Hybrid evolutionary fuzzy learning scheme in the applications of traveling salesman problems*, Information Sciences, 270(2014), 204-225.

[52] Filippi, C., Stevanato, E., *Approximation schemes for bi-objective combinatorial optimization and their application to the TSP with profits*, Computers & Operations Research, 40(2013), 2418-2428.

[53] Focacci, F., Lodi, A., Milano, M., *A hybrid exact algorithm for the TSPTW*, Inform Journal on Computing, 14(4)(2002), 403-417.

[54] Fogel, L. J., Owens, A. J., Walsh, M. J., *Artificial intelligence through simulated evolution*, New York: John Wiley and Sons, Inc., 1966.

[55] Fogel, D. B., *An Evolutionary Approach to the Traveling Salesman Problem*, Biological Cybernetics, 60(1988), 139-144.

[56] Fogel, D. B., *Empirical Estimation of the Computation Required to Reach Approximate Solutions to the Traveling Salesman Problem Using Evolutionary Programming*, Proc. of the Second Annual Conf. on Evolutionary Programming, pp. 56-61, 1993.

[57] Friedberg, R. M., *A Learning Machine: Part I*,IBM J. Research and Development, 2(1958), 2-13.

[58] Friedberg, R. M., Dunham, B., North, J. H., *A Learning Machine: Part II*, IBM J. Research and Development,, vol. 3(1959), pp. 183-191. bibitemFuruhashi2001 Furuhashi, T., *Fusion of fuzzy/neuo/evolutionary computing for knowledge acquisition*, IEEE Proceedings, 89(9): 2001.

[59] Gen, M., Cheng, R., *Genetic algorithms and engineering optimization*, NY: John Wiley & Sons, 2000.

[60] Ghodratnama, A., Tavakkoli-Moghaddam, R., Azaron, A., *A fuzzy possibilistic bi-objective hub covering problem considering production facilities, time horizons and transporter vehicles*, International Journal of Advanced Manufacturing Technology, 66(2013), 187-206.

[61] Goldberg, D., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, MA, USA, 1989.

[62] Goodriich, M. T., Tamassia, R. *Algorithm Design*, New York, USA: John Wiley and Sons, 2002.

[63] Grimaldi, E. A., Grimacia, F., Mussetta, M., Pirinoli, P., Zich, R. E, *A new hybrid genetical swarm algorithm for electromagnetic optimization*, In Proceedings of International Conference on Computational Electromagnetics and its Applications, Beijing, China, 157-160, (2004).

[64] Grzegorzewski, P., Nearest interval approximation of a fuzzy number, Fuzzy Sets and Systems 130(2002), 321-330.

[65] He, G., Gao, J., *A novel weight-based immune genetic algorithm for multi-objective optimization problems*, Advances in Neural Networks, Part-II, Springer -Verlag, Heidelberg, ISNN2009,(2009), 500-509.

[66] Hitchcock, F. L., *The distribution of a product from several sources to numerous localities*, Journal of Mathematical Physics, 20(1941), 224-230.

[67] Hoffman, A. J., Wolfe, P., History. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, D. B. Shmoys, editors, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley, Chichester, 1985.

[68] Holland, H. J., *Adaptation in Natural and Artificial Systems*, University of Michigan, 1975.

[69] Horn, J., Nafpliotis, N., Goldberg, D. E., *A niched pareto genetic algorithm for multi-objective optimization*, In Z. Michalewicz, editor, Proceedings of the First IEEE Conference on Evolutionary Computation, 82-87, Piscataway, New Jersey, IEEE Press, 1994.

[70] Huang, K. Y., *An enhance classification method comparing a GA, rough set theory and modified PBMF-index function*, Applied Soft Computing, 12(10)(2012), 46-63.

[71] Ida, K., Gen, M., Li, Y., *Neural networks for solving multi criteria solid transportation problem*, Computers & Industrial Engineering, 31(1996), 873-877.

[72] Infanger, G., *Stochastic programming*, New York/Dorderecht/ Heidelberg/London: Springer, 2011.

[73] Jang, J. S., Sun, C., *Neuro-fuzzy modeling and control*, Proceedings of IEEE 83(3)(1995), 378-406.

[74] Jaillet, P., *A priori solution of a traveling salesman problemin which a random subset of the customers are visited*. Operations Research, 36(6)(1988), 929-936.

[75] Johnson, D. S., Papadimitriou, C. H., *Computational complexity, in The Traveling Salesman Problem*, Chichester: Wiley, 1985, 37-85.

[76] Jula, H., Dessouky, M., Ioannou, P. A., *Truck route planning in non-stationary stochastic networks with time windows at customer location*, IEEE Transactions on Intelligent Transportation Systems, 7(1)(2006), 51-52.

[77] Kao, E., *A preference order dynamic program for a stochastic travelling salesman problem*, Operations Research, 26(1978), 1033-1045.

[78] Karmakar, S., Mahato, S. K., Bhunia, A.K., *Interval oriented multi-section techniques for global optimization*, Journal of Computational and Applied Mathematics, 224(2)(2009), 476-491.

[79] Karmakar, S., Bhunia, A. K., *A Comparative study of Different Order Relations of Intervals* , Reliable Computing, 16(2012), 360-369.

[80] Katagiri, H., Uno, T., Kato, K., Tsuda, H., Tsubaki, H., *Random fuzzy multi-objective linear programming: Optimization of possibilistic value at risk (pVaR)*, Expert Systems with Applications, 40(1)(2013), 563-574.

[81] Kennedy, J., Eberhart, R., *Particle swarm optimization*, Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 1(1995), 1942-1945.

[82] Kennedy, J., *The particle swarm: social adaptation of knowledge*, In Proceedings of IEEE International Conference on Evolutionary Computation, Indianapolis, IN, 1997, 303-308.

[83] Koc, C., Bekta, T., Jabali, O., Laporte, G., *A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows*, Computers & Operations Research, 64(2015), 11-27.

[84] Koulinas, G., Kotsikas, L., Anagnostopoulos, K., *A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem*, Information Sciences, 277(2014), 680-693.

[85] Koza, J. R., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Cambridge, MA, USA: MIT Press, 1992.

[86] Koza, J. R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D. B., Garzon, M. H., Goldberg, D. E., Iba, H., Riolo, R. L., *Genetic Programming 1998*: Proceedings of the Third Annual Conference, Evolutionary Computation, IEEE Transactions on, 3(1999), 159-161.

[87] Lai, Y. J., Hwang, C. L., *Fuzzy multiple objective decision making*, Springer-Verlag, Heidelberg, 1994.

[88] Last, M., Eyal, S., *A fuzzy-based lifetime extension of genetic algorithms*, Fuzzy Sets and Systems, 149 (2005), 131-147.

[89] Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., Shmoys, D. B., *The traveling salesman problem: G. E. Re Guided tour of combinatorial optimization*, Wiley and Sons, New York, 1985.

[90] Li, H., Zhang, Q., *Multi-objective optimization problems with complicated Pareto sets MOEA/D and NSGA-II*, IEEE Transactions Evolutionary Computation, 3(2009), 284-302.

[91] Little, J. D. C., Murty, K. G., Sweeney, D. W., Karel, C., *An algorithm for the Travelling Salesman Problem*, Operation Research, 11(6)(1963), 972-989.

[92] Liu, B., Iwamura, K. B., *Chance constraint programming with fuzzy parameters*, Fuzzy Sets and Systems, 94 (1998), 227-237.

[93] Li, Y., Ida, K., Gen, M., Kobuchi, R., *Neural network approach for multicriteria solid transportation problem*, Computers & Industrial Engineering, 33(1997), 465-468.

[94] Li, Y., Ida, K., Gen, M., *Improved Genetic Algorithm for Solving Multi-objective Solid Transportation Problem with Fuzzy Numbers*, Computers & Industrial Engineering, 33 (1997), 589-592.

[95] Liaw, C. F., A hybrid genetic algorithm for the open shop scheduling problem, European Journal of Operational Research 124(1)(2000), 28-42

[96] Liu, B., Iwamura, K., *Chance constrained programming with fuzzy parameters*, Fuzzy Sets and Systems, 94(1998), 227-237.

[97] Liu, B., Iwamura, K. B., *A note on chance constrained programming with fuzzy coefficients*, Fuzzy Sets and Systems, 100(1998), 229-233.

[98] Li, L., Lai, K. K., *A fuzzy approach to the multi-objective transportation problem*, Computers & Operation Research, 27(2000), 43-57.

[99] Li, J., Pan, Y., *Erratum to: A hybrid discrete particle swarm optimization algorithm for solving fuzzy job shop scheduling problem*, The International Journal of Advanced Manufacturing Technology, 66(2013), 597-600.

[100] Liu, B., *Theory and Practice of Uncertain Programming*, Physica-Verlag, Heidelberg, 2002.

[101] Liu, Y. K., Liu, B., *A class of fuzzy ramdom optimization: expected value models*, Information Science, 155(2003), 89-102.

[102] Liu, Y. K., Liu, B., *Fuzzy random variable: a scalar expected value operator*, Fuzzy Optimization and Decision Making, 2(2)(2003), 143-160.

[103] Liu, B., *Uncertainty theory*, Berlin Heidelberg-USA, New York: Springer-Verlag, 2004.

[104] Liu, Y. K., Liu,B., *On minimum-risk problems in fuzzy random decision systems*, Computers & Operation Research, 32(2005), 257-283.

[105] Liu, B., *Uncertainty theory (2nd Ed.)* Berlin: Springer, 2007.

[106] Lo, C. C., Chang, W. H., *A multiobjective hybrid genetic algorithm for the capacitated multipoint network design problem*, IEEE Transactions on Systems, Man and Cybernetics - Part B, 30(3)(2000), 461-470.

[107] Lust, T., Jaszkiewicz, A., *Speedup techniques for solving large-scale bi objective TSP*, Computers & Operations Research, 37(2010), 521-533.

[108] Lust, T., Teghan, J., *Two phase Pareto local search for bi-objective travelling salesman problem*, Journal of Heuristics, 16(3)(2010), 475-510.

[109] Ma, H., Simon, D., Fei, M., Shu, M., Chen, Z., *Hybrid biogeography-based evolutionary algorithms*, Engineering Application of Artificial Intelligence, 30(2014), 213-224.

[110] Machado, P., Tavares, J., Pereira, F. B., Costa, E., *Vehicle Routing Problem: Doing it the Evolutionary Way*, Proc. Genetic and Evolutionary Computation Conference (GECCO), pp. 690, 2002.

[111] Majumder, S., Bhunia, A. K., *Genetic algorithm for asymmetric traveling salesman problem with imprecise travel times*, Journal of Computational and Applied Mathematics, 235(9)(2011), 3063-3078.

[112] Mahalanobis, P. C., *A sample survey of the acreage under jute in Bengal*, Sankhya, 4(1940), 511-530,

[113] Marinakis, Y., Marinakii, M., *A Hybrid Multi-Swarm Particle Swarm Optimization algorithm for the Probabilistic Traveling Salesman Problem*, Computers & Operations Research, 37(3)(2010), 432-442.

[114] Menger, K., *Das botenproblem*, Ergebnisse Eines Mathematischen Kolloquiums, 2(1932), 11-12.

[115] Mendel, J. M., John, R. I. B., *Type-2 Fuzzy Sets Made Simple*, IEEE Transactions on Fuzzy Systems, 10(2)(2002), 117-127.

[116] Mestria, M., Ochi, L. S., Martins, S. L., *GRASP with path relinking for the symmetric Euclidean clustered traveling salesman problem*, Computers & Operations Research, 40(12)(2013), 3218-3229.

[117] Michalewicz, Z., *Genetic Algorithms + Data Structure = Evolution programs*, Springer Verlag, Berlin, 1992.

[118] Miranda- Bront, J., Mendez- Diaz, I., Zabala, P., *Facts and valid inequalities for time dependent travelling salesman problem*, Europian Journal of Operational Research, 236(2014), 891-902.

[119] Moore, R., *Interval Analysis*, Prentice-Hall, Englewood Cliffs,NJ (1966).

[120] Moon, C. K., Choi, J., Seo, Y. G., *An efficient genetic algorithm for the traveling salesman problem with precedence constraints*, European Journal of Operational Research, 140(2002), 606-617.

[121] Mousa, A. A., El-Shorbagy, M. A., Abd-El-Wahed, W. F., *Local search based hybrid particle swarm optimization algorithm for multiobjective optimization*, Swarm and Evolutionary Computation, 3(2012), 1-14.

[122] Nag, K., Pal, T., Pal, N. R., *ASMiGA: An Archive- Based Strategy- State Micro Genetic Algorithm*, IEEE Transactions on Cybernetics, 45 (1)(2015), 40-51.

[123] Nagata, V., Soler, D., *A new GA for asymmetric TSP*, Expert Systems with Application, 39(10)(2012), 8947-8953.

[124] Neungmatcha, W., Sethanan, K., Gen M. and S. Theerakulpisut, *Adaptive genetic algorithm for solving sugarcane loading stations with multi-facility services problem*, Computers and Electronics in Agriculture, 98(10)(2013), 85-99.

[125] Oberlin, P., Rathinam, S., Darbha, S., *A transformation for a heterogeneous, multi-depot, multiple traveling salesman problem*, In Proceedings of the American Control Conference, St. Louis, June 10 - 12, 2009, 1292-1297.

[126] Oliver, I. M., Smith, D. J., Holland, J. R., *A study of permutation crossover operators on the traveling salesman problem*, Proc. 2nd International Conference on Genetic Algorithms and their Applications, pp. 224-230, 1987.

[127] ORSOC, *A brief History of TSP*, vol. 2005: The Operations Research Society, UK, 2006.

[128] Paquete, L., Stutzle, T., *Design and analysis of stochastic local search for the multi-objective traveling salesman problem*, Computers & Operations Research, 36(9)(2009), 2619-2631.

[129] Pardalos, P., Rendl, F., Wolkowicz, H., *The quadratic assignment problem: A survey and recent developments*, Proceedings of the Discrete Mathematics

and Theoretical Computer Science (DIMACS) Workshop on Quadratic Assignment Problems, American Mathematical Society, pp. 1-41, 1994.

[130] Parker, R. G., Rardin, R. L., *Guaranteed performance heuristics for the bottleneck traveling salesman problem*, Operations Research Letters, 2(6)(1984), 269-272.

[131] Psychas, I. D., Delimpasi, E., Marinakis, Y., *Hybrid evolutionary algorithms for the Multiobjective Traveling Salesman Problem*, Expert Systems with Applications, 42(22)(2015), 8956-8970.

[132] Pawlak, Z., *Rough sets,* International Journal of Information and Computer Sciences, 11(5)(1982), 341-356.

[133] Pedrycz, A., Dong, F., Hirota, K., *Nonlinear mappings in problem solving and their PSO-based development*, Information Sciences, 181 (2011), 4112-4123.

[134] Pedrycz, W., *Granular Computing: Analysis and Design of Intelligent Systems*, CRC Press/Francis Taylor, Boca Raton (2013).

[135] Peng, J., Liu, B., *Fuzzy random programming with equilibrium chance constraints, Information Sciences*, 170(2005), 363-395.

[136] Peng, J., Liu, B., *Bi-random variables and bi-random programming*, Computers & Industrial Engineering, 53(2007), 433-453.

[137] Petersen, H. L., Madsen, O. B. G., *The double travelling salesman problem with multiple stack - Formulation and heuristic solution approaches*, European Journal of Operational Research, 198(2009), 339-347.

[138] Prodhon, C., *A hybrid evolutionary algorithm for the periodic location-routing problem*, European Journal of Operational Research, 210(2)(2011), 204-212.

[139] Puri, M. L., Ralescu, D., *Fuzzy random variables*, Journal of Mathematics Analysis Application, 114(1986), 409-422.

[140] Queyranne, M., Wang, Y., *Hamiltonian path and symmetric travelling salesman polytopes*, Math. Program. Ser. A, 58:89-110, 1993.

[141] Queyranne, M., Wang, Y., *Symmetric inequalities and their composition for asymmetric travelling salesman polytopes*, Math. Oper. Res., 20:838-863, 1995.

[142] Rakke, J. G., Christiansen, M., Fagerholt, K. Laportei, G., *The Traveling Salesman Problem with Draft Limits*, Computers and Operations Research, 39(9)(2012), 2161-2167.

[143] Reinelt, G., *TSPLIB - A Traveling Salesman Problem Library*, ORSA, Journal on Computing, 3(1991), 376-384.

[144] Rechenberg, I., Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, Stuttgart, Fromman-Holzboog, 1973.

[145] Renato, A., Silva, V. da., Ochi, L. S., *An efficient hybrid algorithm for the Traveling Car Renter Problem*, Expart Systems With Applications, 64(2016), 132-140.

[146] Rubio, L., Sen, M. D. l., Longstaff, A. P., Fletcher, S., *Model-based expert system to automatically adapt milling forces in Pareto optimal multi-objective working points*, Expert Systems with Applications, 40(2013), 2312-2322.

[147] Roy, A., Kar, S., Maiti, M., *A production inventory model with stock dependent demand incorporating learning and inflationary effect in a random planning horizon: A fuzzy genetic algorithm with varying population size approach*, Computers & Industrial Engineering, 57(2009), 1324-1335.

[148] Samanlioglu, F., Ferrel Jr, W. G., Kurz, M. E., *A memetic random-key genetic algorithm for a symmetric multi-objective travelling salesman problem*, Computer & Industrial Engineering, 55(2008), 439-449.

[149] Samarghandi, H., ElMekkawy, T. Y., *A genetic algorithm and particle swarm optimization for no-wait flow shop problem with separable setup times and makespan criterion*, The International Journal of Advanced Manufacturing Technology, 61(2012), 1101-1114.

[150] Sanchez, E., Squillero, G., Tonda, A., *Industrial Applications of Evolutionary Algorithms(34)*, of Intelligent Systems Reference Library, Springer, 2012.

[151] Schwefel, H. P., Numerische Optimierung von Computer modellen mittels der Evolutionsstrategie, Basel, Birkhaeuser, 1977.

[152] Sepideh, F., *Solving traveling salesman problem by using a fuzzy multi objective linear programming*, African Journal of Mathematics and Computer Science Research, 4(11)(2011), 339-349.

[153] Shi, X. H., Liang, Y. C., Lee, H. P., Lu, C., Wang, L. M., *An improved GA and a novel PSO-GA-based hybrid algorithm*, Information Processing Letters, 93(5)(2005), 255-261

[154] Sibeyn, J. F., *The Parallel Maxflow Problem is easy for almost all Graphs*, vol. 2006: citeseer.ist.psu.edu/sibeyn97parallel.html, 1997.

[155] Sinha, A., Goldberg, D. E., *A Survey of hybrid genetic and evolutionary algorithms*, ILLIGAL Technical Report 2003, 2003.

[156] Slowinski, R., Stefanowski, D., *Rough classification in incomplete information systems,* Mathematical and Computer Modelling, 12(1989), 1347-1357.

[157] Somasundaram, P., Lakshmiramanan, R., Kuppusamy, K., *Hybrid algorithm based on EP and LP for security constrained economic dispatch problem*, Electric Power Systems Research, 76(1-3)(2005), 77-85.

[158] Tan, K. C., Yu, Q., Heng, C. M., Lee, T. H., *Evolutionary computing for knowledge discovery in medical diagnosis*, Artificial Intelligence in Medicine, 27(2)(2003), 129-154.

[159] Tao, Z., Xu, J., *A class of rough multiple objective programming and its application to solid transportation problem*, Information Sciences, 188(2012), 215-235.

[160] Tiwari, S., Adel, G., Deb, K., *AMGA2; Improving the performance of the archive based micro-genetic algorithm for multi-objective optimization*, Engineering Optimization, 43(4)(2011), 371-401.

[161] Tseng, L. Y., Liang, S. C., *A hybrid metaheuristic for the quadratic assignment problem*, Computational Optimization and Applications, 34(1)(2005), 85-113.

[162] *TSPLIB http://comopt.ifi.uniidelberg.de/software/TSPLIB95/.*

[163] Turing, A., *On computable numbers with an application to the entscheidnungs problem*, Proc. London Math. Soc., 42 (1936), 230-265.

[164] Ursani, Z., Essam, D., Cornforth, D., Stocker, R., *Localized genetic algorithm for vehicle routing problem with time windows*, Applied Soft Computing, 11(8)(2011), 5375-5390.

[165] Vazquez, M., Whitley, D., *A hybrid genetic algorithm for the quadratic assignment problem*, In Proceedings of Genetic and Evolutionary Computation Conference (GECCO), Morgan Kaufmann, San Mateo, CA, 135-142, 2000.

[166] Vignaux, G. A., Michalewicz, Z. A., *Genetic algorithm for the linear transpotation problem*, IEEE Transactions on Systems, Man. and Cybernetics, 21(1991), 445-452.

[167] Wang, X., He, Y., Dong, L., Zhao, H., *Particle swarm optimization for determining fuzzy measures from data*, Information Sciences, 181(2011), 4230-4252.

[168] Wang, Y., *An Approximate method to compute a sparse graph for travelling salesman problem*, Expert Systems with Application, (2015) In press.

[169] Whitley, L. D., Starkweather, T., Fuquay, D. A., *Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operator*, Proc. 3rd International Conference on Genetic Algorithms, 133-140, 1989.

[170] Xing, L. N., Chen, Y-W., Yang, K-W., Hou, F. Shen, X-S., Cai, H-P., *A hybrid approach combining an improved genetic algorithm and optimization strategies for the asymmetric travelling salesman problem*, Artificial Intelligence, 21 (2008), 1370-1380.

[171] Xu, J., Zhou, X., *Fuzzy Like Multiple-Objective Decision Making*, Springer- Verlag, Berlin, 2009.

[172] Xu, J., Yao, L., *A Class of multi-objective linear programming models with random rough coefficients*, Mathematical And Computer Modelling, 49(2009), 189-206.

[173] Xu, J., Zhao, L., *A multi-objective decision-making model with fuzzy rough coefficients and its application to the inventory problem*, Information Sciences, 180(2010), 679-696.

[174] Xu, J., Zhou, X., *Fuzzy-Like Multiple Objective Decision Making*, Springer, 2011.

[175] Xu, J., Tao, Z., *A class of multi-objective equilibrium chance maximization model with twofold random phenomenon and its application to hydropower station operation*, Mathematics and Computers in Simulation, 83(1)(2012), 11-33.

[176] Yuan, S., Skinner, B., Huang, S., Liu, D., *A new crossover approaches for solving the multiple travelling salesman problem using genetic algorithms*, European Journal of Operational Research, 228(2013), 72-82.

[177] Zadeh, L. A., *From circuit theory to system theory*, Proceedings of Radio Engineering, 50(1962), 856-865.

[178] Zadeh. L. A., *Fuzzy sets*, Information and Control, 8(1965), 338-353.

[179] Zadeh, L. A., *Quantitative fuzzy semantics*, Information Sciences, 3(1971), 177-200.

[180] Zadeh, L. A., *The concept of linguistic variable and its application to approximate reasoning*, Memorandum ERL-M 411 Berkeley, 1973.

[181] Zadeh, L. A., *The concept of a linguistic variable and its application to Appromximate Resoning -I*, Information Sciences, 8(1975), 199-249.

[182] Zadeh, L. A., *Fuzzy sets as a basis for a theory of possibility*, Fuzzy Sets and Systems, 1(1978), 3-28.

[183] Zadeh, L. A., *Fuzzy Logic and Soft Computing: Issues, Contentions and Perspectives*, In Proc. of IIZUKA'94: Third Int. Conf. on Fuzzy Logic, Neural Nets and Soft Computing, 1-2(1994), Iizuka, Japan.

[184] Zadeh, L. A., *Some reflection on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems*, Soft Computing A Fusion of Foundations, Methodologies and Applications, 2(1)(1998), 23-25.

[185] Zhang, Q., Zhou, A., Jin, Y., *RMMEDA: A regularity model based multi objective estimation of distribution algorithm*, IEEE Trans.Evol.Comput., 12(1)(2008), 41-63.

[186] Zhang, G., Liu, M., Li, J., Ming, W., Shao, X., Huang, Y., *Multi-objective optimization for surface grinding process using a hybrid particle swarm optimization algorithm*, The International Journal of Advanced Manufacturing Technology, 71, 9-12 (2014), 1861-1872.

[187] Zhou, A., Gao, F., Zhang, G., *A decomposition based estimation of distribution algorithm for multi-objective travelling salesman problem*, Computers and Mathematics with Applications, 66(2011), 1857-1868.

[188] Zimmermann, H. J., *Fuzzy programming and linear programming with several objective functions*, Fuzzy Sets and Systems, 1(1978), 45-55.

[189] Zitler, E., Thiele, L., Laumanns, M., Fonseca, C. M., da Fonseca, V. G., *Performance assessment of multi objective optimizers: Ananalysis and review*, IEEE Trans.Evol.Comput., 7(2)(2003), 117-132.

[190] Zmuda, M. A., Rizki, M. M., Tamburino, L. A, *Hybrid evolutionary learning for synthesizing multi-class pattern recognition systems*, Applied Soft Computing, 2(4)(2003), 269-282.

Index

Computers & Industrial Engineering 83 (2015) 273–296

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A Modified Genetic Algorithm for solving uncertain Constrained Solid Travelling Salesman Problems ☆

Samir Maity [a,*], Arindam Roy [b], Manoranjan Maiti [c]

[a] Dept. of Computer Science, Vidyasagar University, Medinipur, 721102 WB, India
[b] Dept. of Computer Science, P.K. College, Contai, Purba Medinipur, 721401 WB, India
[c] Dept. of Applied Mathematics, Vidyasagar University, Medinipur, 721102 WB, India

ARTICLE INFO

Article history:
Received 20 May 2014
Received in revised form 12 January 2015
Accepted 18 February 2015
Available online 9 March 2015

Keywords:
STSP
CSTSP
Probabilistic selection
Comparison crossover
Modified Genetic Algorithm (MGA)

ABSTRACT

In this paper, a Modified Genetic Algorithm (MGA) is developed to solve Constrained Solid Travelling Salesman Problems (CSTSPs) in crisp, fuzzy, random, random-fuzzy, fuzzy-random and bi-random environments. In the proposed MGA, for the first time, a new 'probabilistic selection' technique and a 'comparison crossover' are used along with conventional random mutation. A Solid Travelling Salesman Problem (STSP) is a Travelling Salesman Problem (TSP) in which, at each station, there are a number of conveyances available to travel to another station. Thus STSP is a generalization of classical TSP and CSTSP is a STSP with constraints. In CSTSP, along each route, there may be some risk/discomfort in reaching the destination and the salesman desires to have the total risk/discomfort for the entire tour less than a desired value. Here we model the CSTSP with traveling costs and route risk/discomfort factors as crisp, fuzzy, random, random-fuzzy, fuzzy-random and bi-random in nature. A number of benchmark problems from standard data set, TSPLIB are tested against the existing Genetic Algorithm (with Roulette Wheel Selection (RWS), cyclic crossover and random mutation) and the proposed algorithm and hence the efficiency of the new algorithm is established. In this paper, CSTSPs are illustrated numerically by some empirical data using this algorithm. In each environment, some sensitivity studies due to different risk/discomfort factors and other system parameters are presented.

1. Introduction

The TSP was first formulated as a mathematical problem in 1930 and became increasingly popular after 1950. It is one of the most intensively studied problems in optimization even in recent years. A TSP is to find a possible tour along which a Travelling Salesman (TS) visits each city exactly once for a given list of cities and back to the starting city, so that total cost spent/distance covered is minimal. TSP is a well-known NP-hard combinatorial optimization problem (Lawler, Lenstra, Rinnooy Kan, & Shmoys, 1985). Different types of TSPs have been solved by researchers during last two decades. These are TSPs with time windows (Focacci, Lodi, & Milano, 2002), stochastic TSP (Chang, Wan, & Tooi, 2009), double TSP (Petersen & Madsen, 2009), asymmetric TSP (Majumder & Bhunia, 2011; Mestria, Ochi, & Martins, 2013), TSP with precedence constraints (Moon, Ki, Choi, & Seo, 2002; Rakke, Christiansen, Fagerholt, & Laportei, 2012), etc.

In TSP, it is assumed that a TS travels from one city to another using only one conveyance. But in real life, a set of conveyances may be available at each city. In that case, a TS has to design his/her tour for minimum cost maintaining the TSP conditions and using the suitable conveyances at different cities. This problem is called Solid Travelling Salesman Problem (STSP). Traveling cost from one city to another city depends on the types of conveyances, condition of roads, geographical areas, weather condition at the time of the travel, etc., so there always prevail some uncertainties/vagueness. For this reason it is better to model the costs by uncertain parameters as fuzzy, random, random-fuzzy, bi-random and fuzzy random values. To analyses the large scale/amount of data throughout a long time interval, we observe that the data values are fluctuating over a period of time/year/session etc. So, for the decision making problem, twofold random phenomena is well suited/realistic approach. Also since TS may use different conveyances to travel along different routes, there may be corresponding some risk/discomfort factors, which depend on the condition of roads, types and conditions of vehicles, law and order condition

☆ This manuscript was processed by Area Editor Mitsuo Gen.

* Corresponding author. Mobile: +91 9874512894.
E-mail addresses: maitysamir13@gmail.com (S. Maity), royarindamroy@yahoo.com (A. Roy), mmaiti2005@yahoo.com (M. Maiti).

http://dx.doi.org/10.1016/j.cie.2015.02.023
0360-8352/

Expert Systems With Applications 46 (2016) 196–223

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

An imprecise Multi-Objective Genetic Algorithm for uncertain Constrained Multi-Objective Solid Travelling Salesman Problem

Samir Maity[a,*], Arindam Roy[b], Manoranjan Maiti[c]

[a] Department of Computer Science, Vidyasagar University, Medinipur, WB 721102, India
[b] Department of Computer Science, P.K. College, Contai, Purba Medinipur, WB 721401, India
[c] Department of Applied Mathematics, Vidyasagar University, Medinipur, WB 721102, India

ARTICLE INFO

Keywords:
CMOSTSP
Fuzzy set based selection
Adaptive crossover
Generation dependent mutation
iMOGA

ABSTRACT

In this paper, an imprecise Multi-Objective Genetic Algorithm (iMOGA) is developed to solve Constrained Multi-Objective Solid Travelling Salesman Problems (CMOSTSPs) in crisp, random, random-fuzzy, fuzzy-random and bi-random environments. In the proposed iMOGA, '3- and 5-level linguistic based age oriented selection', 'probabilistic selection' and an 'adaptive crossover' are used along with a new generation dependent mutation. In each environment, some sensitivity studies due to different risk/discomfort factors and other system parameters are presented. To test the efficiency, combining same size single objective problems from standard TSPLIB, the results of such multi-objective problems are obtained by the proposed algorithm, simple MOGA (Roulette wheel selection, cyclic crossover and random mutation), NSGA-II, MOEA-D/ACO and compared. Moreover, a statistical analysis (Analysis of Variance) is carried out to show the supremacy of the proposed algorithm.

1. Introduction

Genetic algorithms (GAs) are robust search algorithms that use the operations of natural genetics to find the optimum through a search space. Recently, GAs have been used to solve several single and multi-objective decision making problems. In multi-objective optimization techniques (MOOTs), a Pareto Front (PF) is generated and an optimum solution set should be very close to the true PF. But, the above two goals are conflicting for the fixed number of functions, evaluations as the first property requires intensive search over a particular region of the search space and the second one for the uniform search of the whole region. Thus MOOTs make a trade-off between exploration and exploitation. The first real implication of multi-objective evolutionary algorithm (vector evaluated GA or VEGA) was suggested by David Schaffer in 1984. Then Goldberg suggested to implement domination principle in evolutionary algorithm (EA). Realizing the potential of a good multi-objective evolutionary algorithm (MOEA) (Deb, 2001; Rubio, Sen, Longstaff, & Fletcher, 2013) which can be derived from Goldberg's suggestions, researchers developed different versions of MOEAs such as multi-objective GAs (MOGAs), Niched Pareto GAs (NPGAs) (Horn, Nafpliotis, & Goldberg, 1994), non-dominated sorting GAs (NSGAs) (Deb, 2002), hybrid scatter search like MOGA by Durillo, Nebro, Luma, and Alba (2009), decomposition -based MOAs like MOiA/D-DE (Li & Zhang, 2009), archive-based micro GAs like AMGA2 (Tiwari, Adel, & Deb, 2011), etc. In AMGA2, a modified definition of crowding distance for the generation of mating pool has been presented. Recently, an archived-based steady-state micro genetic algorithm (ASMiGA) has been developed with new environmental selection and mating selection strategies (Nag, Pal, & Pal, 2015).

TSP is a well-known NP-hard combinatorial optimization problem (Lawler, Lenstra, Rinnooy Kan, & Shmoys, 1985). Different types of TSPs have been solved by the researchers during last two decades. These are TSPs with time windows (Focacci, Lodi, & Milano, 2002), stochastic TSP (Chang, Wan, & Tooi, 2009), double TSP (Petersen & Madsen, 2009), asymmetric TSP (Majumder & Bhunia, 2011), TSP with precedence constraints (Moon, Ki, Choi, & Seo, 2002). Wang (2015) proposed an approximate method on sparse graph for TSP, Nagata and Soler (2012) developed a new GA for asymmetric TSP, Che and Ohiem (2012) considered genetic simulated annealing ant colony systems with PSO to solve TSP, Dong, Guo, and Tickle (2012) proposed a cooperative GA for general TSP, Albanyrak and Allahverdi (2011) developed a new mutation operator to solve TSP by GA, Xu and Tao (2012) solved multi-objective problem with power station operation, Elaoud, Teghem, and Loukil (2010) proposed multiple crossover and mutation operators with dynamic selection scheme in MOGA for multi-objective TSP (MOTSP), Lust and Teghan (2010) presented two-phase Pareto local search (2PPLS) for bi objective TSP, Filippi and Stevanato (2013) considered a Pareto ϵ approximation named as ABE

* Corresponding author. Tel.: +91 9874512894.
E-mail addresses: maitysamir13@gmail.com, samirmca13@gmail.com (S. Maity), royarindamroy@yahoo.com (A. Roy), mmaiti2005@yahoo.co.in (M. Maiti).

http://dx.doi.org/10.1016/j.eswa.2015.10.019
0957-4174/