



UNIVERSITY OF NOTTINGHAM

DOCTORAL THESIS

**An integrated model of Supplier
Selection and Inventory Planning using
Fuzzy Logic and Multi-objective
Evolutionary Algorithms**

Author:

Seda ÖZDEMİR

Supervisor:

Prof. Robert JOHN

*Thesis submitted to the University of Nottingham
for the degree of Doctor of Philosophy*

July 2017

Abstract

Supplier selection and inventory planning are critical and challenging tasks in supply chain management. There are many studies on both topics and many solution techniques have been proposed dealing with each problem separately. In this thesis, we present a two-stage integrated approach to the supplier selection and inventory planning. In the first stage, in order to get a risk value of each supplier, suppliers are evaluated based on various criteria derived from cost, quality, service and delivery using Interval Type-2 Fuzzy Sets (IT2FSs). In the second stage, the information of supplier rank is fed into an inventory model built to cover the effect of suppliers on the total cost of a supply chain.

The proposed model is formulated as single, multi and many-objective optimisation problems, respectively. Firstly, we generated a set of new instances based on a real world problem. Twenty four problem instances are provided as a benchmark for the community. Various metaheuristics, including MOSA (Multi-objective Simulated Annealing) as a single point based search algorithm, NSGA-II (Non-dominated Sorting Genetic Algorithm-II), SPEA2 (Strength Pareto Evolutionary Algorithm 2), IBEA (Indicator Based Evolutionary Algorithm) as population based multi-objective algorithms and NSGA-III (Non-dominated Sorting Genetic Algorithm-III) as a many objective algorithm are applied to those integrated supply chain management problem instances.

It is a well-known fact that parameter setting is crucial for an improved performance of a metaheuristic. Hence, in order to use each algorithm at its best, we employed the experimental design methodology of Taguchi orthogonal arrays for parameter tuning detecting the best setting for each algorithm. A comparative analysis of multi-objective metaheuristics is provided to find the best performing approach in the second stage. The experimental results show that the proposed two-stage approach is indeed capable of solving the integrated supply chain management problem successfully. NSGA-III as a population based technique outperforms the single point based search approach of Simulated Annealing which aggregates multiple objectives into a single objective and other three population based techniques, NSGA-II, SPEA2 and IBEA. Within the population based approaches, NSGA-II performs the best as a multi-objective algorithm when excluding NSGA-III as a many objective algorithm.

Acknowledgements

I am deeply grateful to Professor Robert John and Dr. Ender Özcan, my supervisors who generously rendered help and encouragement during the process of this thesis. Whatever I have accomplished in pursuing this undertaking is due to their guidance and thoughtful advices. To them my thanks.

I would like to thank the examiners, Professor Shengxiang Yang and Dr Rong Qu, for their valuable comments.

I would like to express my warmest thanks to my parents and my husband. My parents have always respected my choices and given me unconditional love, support and encouragement through all my life. My husband, Hasan, who had to go through difficult time to help me fulfil my dream, is so kind, patient, loving and caring and has provided tremendous support for my graduate study.

Finally, many thanks are owed to my friends, Sumeyra Cihangiroglu Akbey, Eylul Hanim Bas, Rukiye Samci Karadeniz, Ilyas Karadeniz, Tayyibe Erten and Aytul Gokce for their unconditional support throughout my academic career and life in general.

Seda Özdemir
University of Nottingham
April, 2017

Contents

Abstract	i
Acknowledgements	ii
Contents	iii
List of Figures	vi
List of Tables	viii
List of Abbreviations	xi
1 Introduction	1
1.1 Research Motivation and Aims	1
1.2 Research Contributions	3
1.3 Publications	4
1.4 Structure of the Thesis	5
2 Fuzzy Logic and Supply Chain Management	7
2.1 Fuzzy Logic	7
2.1.1 Brief Review of Type-1 Fuzzy Sets and Systems	8
2.1.2 Basic Concepts of Interval Type-2 Fuzzy Sets and Systems .	10
2.2 Supply Chain Management	14
2.2.1 Supplier Selection	16
2.2.2 Inventory Planning	22
2.3 Fuzzy Logic in SCM	25
2.4 Summary	27
3 Multi-objective Optimisation	28
3.1 Basic Concepts and Terminology	28
3.2 Performance Metrics	31
3.2.1 Generational Distance	32
3.2.2 Inverted Generational Distance	32
3.2.3 Hypervolume	32

3.3	Scalarisation Techniques in Multi-objective Optimisation	34
3.3.1	The Weighted-sum Scalarisation	34
3.3.2	The Tchebycheff Scalarisation	34
3.4	Metaheuristics	35
3.4.1	Simulated Annealing	36
3.4.2	Evolutionary Algorithm	40
3.4.3	Parameter Tuning	41
3.5	Multi-objective Evolutionary Algorithms	42
3.5.1	Vector Evaluated Genetic Algorithm	44
3.5.2	Multi-Objective Genetic Algorithm	45
3.5.3	Niched-Pareto Genetic Algorithm	45
3.5.4	Non-dominated Sorting Genetic Algorithm	46
3.5.5	Pareto-Archived Evolution Strategy	50
3.5.6	Strength Pareto Evolutionary Algorithm	50
3.5.7	Indicator-based Evolutionary Algorithm	52
3.5.8	Summary of Multi-objective Optimisation Algorithms	53
3.6	Existing Tools for Multi-objective Optimisation	57
3.7	Multi-objective optimisation in SCM	58
3.8	Summary	62
4	Integrated Supply Chain Management Problem	63
4.1	Problem Description	63
4.1.1	Stage One: Ranking of Suppliers	63
4.1.1.1	Membership Functions	64
4.1.1.2	Proposed Method for Ranking Suppliers	65
4.1.2	Stage Two: Supplier Selection Aware Inventory Planning	66
4.1.3	Problem Formulation	67
4.1.4	Problem Instances	71
4.2	Summary	73
5	Single Objective Algorithm for the Integrated Problem	74
5.1	Stage One: Ranking of Suppliers	74
5.1.1	Effect of the fuzzy membership functions shape on the result	76
5.1.2	Developing approaches with IT2FSs	78
5.2	Stage Two: Supplier Selection Aware Inventory Planning	80
5.2.1	Multi-objective Simulated Annealing	83
5.2.2	MOSA with certain weight settings	84
5.2.2.1	Experimental Setup	84
5.2.2.2	Experimental Results	85
5.2.3	MOSA with random weight settings	88
5.2.3.1	Experimental Setup	88
5.2.3.2	Parameter Tuning	89
5.2.3.3	Experimental Results	91

5.3	Summary	97
6	Multi-objective Evolutionary Algorithms for the Integrated Problem	99
6.1	Multi-objective Evolutionary Algorithms	99
6.1.1	Experimental Setup	100
6.1.2	Results from Ranking of Suppliers	101
6.1.3	Parameter Tuning of NSGA-II, SPEA2 and IBEA	101
6.1.4	Experimental Results	104
6.2	Many-objective Evolutionary Algorithm	110
6.2.1	Experimental Setup	110
6.2.2	Parameter Tuning of NSGA-III	111
6.2.3	Experimental Results	112
6.3	A Comparative Analysis of Approaches	115
6.4	Summary	119
7	Conclusion	121
7.1	Summary of Work	122
7.2	Research Contributions	123
7.3	Future Work	124
	References	127

List of Figures

2.1	Type-1 fuzzy logic system [Mendel, 2001].	10
2.2	Type-2 fuzzy logic system [Karnik and Mendel, 1999].	11
2.3	The upper trapezoidal membership function $\tilde{A}_i^U$, the lower trapezoidal membership function $\tilde{A}_i^L$ of the IT2FS $\tilde{A}$ [Gong, 2013].	13
3.1	Pareto optimality in the objective space (left) and the possible relations of solutions in the objective space (right) [Zitzler, 1999]. . .	29
3.2	Hypervolume of a solution set A	33
4.1	The Membership functions of the linguistic importance weight, (a) The type-1 MFs [Ordoobadi, 2009] and (b) The interval type-2 MFs.	64
4.2	The membership functions of the linguistic performance rate, (a) The type-1 MFs [Ordoobadi, 2009] and (b) The interval type-2 MFs.	65
4.3	The criteria and sub-criteria used for selection of suppliers [Ordoobadi, 2009].	67
4.4	Representation of systems consisting of a fixed number of suppliers, manufacturing plants and customers, namely $2 \times 2 \times 2$, $3 \times 2 \times 3$, $3 \times 3 \times 3$ and $5 \times 5 \times 5$, from which 24 instances are derived.	72
5.1	Behaviour of the SA algorithm in a single run for each scenario using a Tchebycheff scalarisation.	86
5.2	TR vs TC from 100 runs (each with a random weight setting) of the SA algorithm based on weighted sum (on the left) and Tchebycheff (on the right) scalarisations. Circles indicate the pareto front.	87
5.3	TR vs TC for five runs (a run for each scenario except Scn-6) of the SA algorithm based on the weighted sum approach.	87
5.4	Main effects plot with mean rank values based on 3 instances in MOSA based on weighted sum (on the left) and Tchebycheff (on the right) scalarisations, P1: Initial temperature, P2: Absolute temperature.	90

5.5	TR vs TC from 30 runs (each with a random weight setting) of the SA algorithm based on weighted sum (on the left), Tchbycheff (in the middle) and weighted sum versus Tchbycheff (on the right) scalarisations. Circles show the pareto front. The first row represents a $2 \times 2 \times 2$ (supplier,manufacturing plant,customer in a sequence) system, in the second row, a $3 \times 2 \times 3$ system is indicated, the third row shows a $3 \times 3 \times 3$ system and the last row depicts a $5 \times 5 \times 5$ system. For all graphs, instances generated by certain parameters are used.	96
6.1	Main effects plot with mean rank values in three multi-objective algorithms	104
6.2	Final pareto-front plots for NSGA-II, SPEA2 and IBEA based on TR vs TC from a sample run on (a) Inst1 and (b) Inst19.	107
6.3	Main effects plot with mean rank values in NSGA-III	112
6.4	Results of NSGA-III in parallel coordinate for Inst19; green lines show trade-off solutions among objectives and black line indicates the knee solution.	114
6.5	Two solutions of NSGA-III highlighted in parallel coordinate; Trade-off among objectives for Inst19	114
6.6	Risk vs Cost objectives' results of NSGA-III in parallel coordinate for Inst19	116
6.7	Pareto-front plots for NSGA-III, NSGA-II and $MOSA_{W_s}$ based on TR vs TC from a sample run on Inst1	119

List of Tables

3.1	Related work on Simulated Annealing algorithms range from single objective to multi-objective optimisation problems	39
3.2	Components of well-known multi-objective optimisation algorithms	54
3.3	Fitness assignment and diversity mechanism of well-known multi-objective algorithms	55
4.1	Linguistic weights of the attributes represented by type-1 fuzzy sets and its corresponding interval type-2 fuzzy sets (LT: Linguistic terms).	64
4.2	Linguistic performance rates represented by type-1 fuzzy sets and its corresponding interval type-2 fuzzy sets (LT: Linguistic terms). .	65
4.3	Notation for Decision Variables [Turk et al., 2015]	68
4.4	Notation used for the formulation of the problem [Turk et al., 2015]	68
4.5	Characteristics of the problem instances (S: the number of suppliers, P: the number of manufacturing plants, C: the number of customers, CO: configuration)	72
4.6	Configuration of problem instance parameters-I	72
4.7	Configuration of problem instance parameters-II	72
5.1	The importance of criteria, L: Low importance, M: Moderate importance, H: High importance, VH: Very high importance	75
5.2	The performance of each supplier in terms of selected criteria for three suppliers, EX: Excellent, VG: Very good, G: Good, P: Poor .	75
5.3	Crisp scores and supplier rankings Ordoobadi [2009]	75
5.4	Linguistic weights of the attributes represented by type-1 fuzzy sets	77
5.5	Linguistic performance rates represented by type-1 fuzzy sets	77
5.6	Ranking of suppliers results for three aproaches	77
5.7	Their corresponding interval type-2 fuzzy sets for linguistic weights of attributes, LT: Linguistic terms, L: Low importance, M: Moderate importance, H: High importance, VH: Very high importance . .	78
5.8	Their corresponding interval type-2 fuzzy sets for linguistic performance rates, LT: Linguistic terms, EX: Excellent, VG: Very good, G: Good, P: Poor	79
5.9	The weighted trapezoidal interval type-2 sets for criteria	79
5.10	Suppliers' performance ratings with respect to the selection criteria	81

5.11	Found Scores for Suppliers by Ordoobado's work	81
5.12	Ranking of suppliers results for three approaches	82
5.13	The performance of each supplier in terms of selected criteria for five suppliers	82
5.14	Rank and Risk Values of Suppliers for each Instance	82
5.15	Best and average results in terms of TR , TC obtained from 30 runs for each scenario for Weighted sum and Tchebycheff scalarisation .	86
5.16	Parameter setting options used during the parameter tuning of MOSA	89
5.17	Average rank for MOSA algorithm, with a particular parameter configuration, Init. temp.:Initial temperature, Abs. temp.: Abso- lute temperature.	89
5.18	Parameter setting options used during the parameter tuning of three MOEAs, Init. temp.:Initial temperature, Abs. temp.: Absolute temperature, Max. evaluation: Maximum evaluation, Ns : Number of suppliers.	90
5.19	Performance metrics of two objectives supply chain problem with default settings	92
5.20	Performance metrics of two objectives supply chain problem with tuned parameters	93
5.21	The average percentage improvement of metrics on 24 instances when tuned parameters are used	94
5.22	The summary of SA results	95
5.23	Comparison of two MOSA approaches for two-objective supply chain problem in terms of different cost results which compose the total cost	97
6.1	L^{16} Taguchi orthogonal arrays design	102
6.2	Average rank for three multi-objective algorithms, with a particular parameter configuration based on the L^{16} Taguchi orthogonal array	103
6.3	ANOVA test results for dismissing the contribution of each param- eter for MOEAs in terms of percent contribution	103
6.4	Tuned Parameters of three MOEAs	103
6.5	Performance comparison of NSGA-II, SPEA2 and IBEA for the two-objective supply chain management problem based on three metrics	106
6.6	Comparison of three MOEAs for two-objective supply chain man- agement problem in terms of different cost results which compose the total cost	109
6.7	Average rank of NSGA-III, with a particular parameter configura- tion based on the L^{16} Taguchi orthogonal array	112
6.8	ANOVA test results for dismissing the contribution of each param- eter for NSGA-III in terms of percent contribution	112

6.9	Objective wise results of NSGA-III. TR: Total Risk, TC: Total Cost, TBC: Total Batch Cost, TPC: Total Production Cost, TTC: Total Transportation Cost, TSC: Total Stock-out Cost, THC: Total Holding Cost.	115
6.10	A comparison of single ($MOSA_{Ws}$), multi (NSGA-II) and many objective (NSGA-III) evolutionary algorithms based on three performance metrics	117
6.11	Performance comparison between NSGA-II and NSGA-III	118

List of Abbreviations

AGP	Archimedean Goal Programming
AHP	Analytic Hierarchy Process
AMOSA	Archived Multi-objective Simulated Annealing
ANOVA	Analysis of Variance
ANP	Analytic Network Process
CA	Cluster Analysis
CBR	Case Based Reasoning
DEA	Data Envelopment Analysis
DTLZ	Deb-Thiele-Laumanns-Zitzler
EA	Evolutionary Algorithm
FFEs	Fractional Factorial Experiments
FLS	Fuzzy Logic System
HV	Hypervolume
GA	Genetic Algorithm
GD	Generational Distance
GRASP	Greedy Randomized Adaptive Search Procedure
IBEA	Indicator-based Evolutionary Algorithm
IGD	Inverted Generational Distance
ISRMS	Intelligent Supplier Relationship Management System
IT2FL	Interval Type-2 Fuzzy Logic
IT2FS	Interval Type-2 Fuzzy Set
MADM	Multiple Attribute Decision Making
MAUT	Multi Attribute Utility Technique

MFs	Membership Functions
MIP	Mixed Integer Programming
MOABC	Multi-objective Artificial Bee Colony
MOEA	Multi-objective Evolutionary Algorithm
MOEA/D	Multi-objective Evolutionary Algorithm Based on Decomposition
MOGA	Multi-Objective Genetic Algorithm
MOLIP	Multi-objective Location Inventory Problem
MOMGA	Multi-Objective Messy Genetic Algorithm
MOSA	Multi-objective Simulated Annealing
NPGA	Niched-Pareto Genetic Algorithm
NSGA	Non-dominated Sorting Genetic Algorithm
NSGA-II	Non-dominated Sorting Genetic Algorithm-II
NSGA-III	Non-dominated Sorting Genetic Algorithm-III
PAES	Pareto-Archived Evolution Strategy
PDMOSA	Pareto Domination based Multi-objective Simulated Annealing
SA	Simulated Annealing
SBX	Simulated Binary Crossover
SC	Supply Chain
SCM	Supply Chain Management
SPEA	Strength Pareto Evolutionary Algorithm
SPEA2	Strength Pareto Evolutionary Algorithm 2
TOPSIS	Technique for Order Preference by Similarity to Ideal Solution
TS	Tabu Search
T1FL	Type-1 Fuzzy Logic
T2FL	Type-2 Fuzzy Logic
VEGA	Vector Evaluated Genetic Algorithm
WFG	Walking Fish Group
WMOSA	Weight based Multi-objective Simulated Annealing
ZDT	Zitzler-Deb-Thiele

*To my husband, Hasan for his endless love, support and
encouragement*

Chapter 1

Introduction

1.1 Research Motivation and Aims

Supply chain management (SCM) is the management of material flows from the procurement of basic raw materials to final product delivery considering information flows in supply chains, material flows and long-term relations between customers and suppliers [Thomas and Griffin, 1996]. Supply chain management comprises of planning and management of all activities which add value to products and services for the ultimate customer [Setak et al., 2012]. A number of components are involved in supply chain management, such as service level management, order and demand management, supplier management and inventory management, etc.

Supply chain management is critical to achieve sustainable competitive advantage for a company. One major aspect of supply chain management is to select suppliers which can support the success of a company meeting expectations of the company. Another one is to plan and control inventory through the whole network from suppliers to customers, balancing material flows among the entire processes of a supply chain effectively.

In the literature, there are many studies on supplier selection and inventory management and many solution techniques have been proposed dealing with each problem separately. The motivation behind this study is that there has been a

lack of extensive research in the field of the integrated supply chain management problem of supplier selection and inventory management.

Moreover, supply chain management decisions are made under conflicting criteria for different objectives such as maximising customer service while minimising the total cost of a supply chain. Multiple-criteria decision making has become an important activity in supply chain management. Multiple-criteria decision making methods include, i) multiple attribute decision making methods such as Analytic Hierarchy Process (AHP) [Saaty, 1987], Analytic Network Process (ANP) [Sarkis and Talluri, 2002], Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [Hwang and Yoon, 1981]; ii) Fuzzy Set Theory [Zadeh, 1965]; iii) multiple objective decision making such as multi-objective evolutionary optimisation methods [Kahraman, 2008]. This research here combines interval type-2 fuzzy sets and multi-objective evolutionary algorithms in one approach in order to tackle the integrated supply chain management problem.

The main aim of this research is to deal with supplier selection and inventory management problems by building an optimisation-based integrated supply chain management model. In order to solve the integrated supply chain management problem, the formulation of the problem informs the selection of optimisation algorithms and formulation of the objectives. Firstly, each supplier performance is assessed based on different criteria such as cost, quality, service and delivery, and interval type-2 fuzzy sets are used to obtain the risk value of each supplier whilst considering their rank. This information is fed into an inventory model built to capture the impact of supplier risk on the total cost of a supply chain. Secondly, the proposed model is formulated as single, multi and many-objective optimisation problems, respectively, and different algorithms (MOSA (Multi-objective Simulated Annealing) as a single objective algorithm, NSGA-II (Non-dominated Sorting Genetic Algorithm-II), SPEA2 (Strength Pareto Evolutionary Algorithm 2), IBEA (Indicator Based Evolutionary Algorithm) as multi-objective algorithms and NSGA-III (Non-dominated Sorting Genetic Algorithm-III) as a many-objective algorithm) are used to solve it. Finally, a comparative analysis of algorithms is provided based on their performance on the integrated problem to find the best performing approach.

1.2 Research Contributions

The crucial contributions of the thesis can be briefly summarised as flows:

- In the literature, there are many studies on supplier selection and inventory management problems, and many solution methods have been proposed dealing with each problem separately. To the authors' knowledge, this is one of the first studies in which the integrated problem of supplier selection and inventory planning has been investigated as a multi-objective problem. First, an effective model to rank suppliers using interval type-2 fuzzy sets which are capable of coping with quantitative and qualitative factors in a supplier selection problem is presented. And then the rank of suppliers is combined with components of inventory management problems to achieve an appropriate integrated model.
- The multi-objective nature of the integrated supply chain management problem is tested and confirmed on a simple single problem instance. The problem is formulated as single, multi and many-objective optimisation problem, respectively. Correspondingly, various metaheuristics are presented to solve them, including a single objective optimisation algorithm, MOSA, three multi-objective evolutionary algorithms, NSGA-II, SPEA2 and IBEA, and a many-objective optimisation algorithm, NSGA-III.
- Parameters of evolutionary algorithms have an essential impact on the performance of algorithms. Identifying the optimal parameter settings for an evolutionary algorithm is also one of the challenging tasks. Parameters of each algorithm used in this study are tuned using Taguchi orthogonal arrays to detect the best setting for each algorithm. The best combination of these parameters are used to reach solutions with better quality.
- The problem instances used in this study are provided as a benchmark along with our implementation of the integrated supply chain management problem for future research. The problem instances are made publically available from https://www.researchgate.net/publication/311873391_SCMproblem with DOI : 10.13140/RG.2.2.21820.64643.

1.3 Publications

The following publications have been produced as a result of this research presented in this thesis:

- Seda Türk, Robert John and Ender Özcan (2014), Interval Type-2 Fuzzy Sets in Supplier Selection, The 14th annual workshop on computational intelligence (UKCI), pp. 1-7. [conference]

This publication introduces a model for solving a supplier selection problem using interval type-2 fuzzy sets. In addition, the influence of the membership function shape on the results has been investigated. Based on the findings, the approach is used to evaluate five suppliers in a real-world problem instance in this thesis. Chapter 5.1 is based on the work presented in this publication. This paper is available from <http://ieeexplore.ieee.org/document/6930168/> with DOI : 10.1109/UKCI.2014.6930168.

- Seda Türk, Simon Miller, Ender Özcan and Robert John, A simulated annealing approach to supplier selection aware inventory planning, Evolutionary Computation (CEC) 2015 IEEE Congress on, pp. 1799-1806, 2015. [conference]

In this publication, we describe an integrated two-stage fuzzy embedded approach to deal with both supplier selection and inventory planning of a supply chain problem. We apply a simulated annealing approach to the problem balancing the trade-off between supply chain operational cost and supplier risk. Chapter 5.2 covers the findings of this paper and an extension of this work with regard to the weight setting procedure. This paper is available from <http://ieeexplore.ieee.org/abstract/document/7257105/> with DOI : 10.1109/CEC.2015.7257105.

- Seda Türk, Ender Özcan and Robert John, Multi-objective optimisation in inventory planning with supplier selection, Expert Systems With Applications, 78 . pp. 51-63, 2017 [journal] [Impact Factor: 2.981]

This journal paper extends the previous work and investigates three multi-objective meta-heuristic algorithms to solve the problem with an attempt to detect the best performing approach. Chapters 4 and part of Chapter 6.1 are based on the work presented in this publication. It is noted that the results

and the performance analysis of the multi-objective algorithms which have been published in this paper are revised with respect to the global pareto set in this thesis. This paper is available from <http://doi.org/10.1016/j.eswa.2017.02.014>.

An abstract entitled “Inventory Planning With Supplier Selection Using Interval Type-2 Fuzzy Sets” was also presented at Association of European Operational Research Societies (EURO) conference, 2015 in Glasgow.

1.4 Structure of the Thesis

This section provides an overview of the contents of chapters of this thesis.

- **Chapter 1:**

This chapter introduces the thesis topic and relevant concepts. It also provides the aim of the thesis, as well as the scientific contributions and papers published during the study.

- **Chapter 2:**

In the second chapter, a literature review of fuzzy logic is presented, giving the fundamental definitions and applications. Also, a review of supply chain management and a related work covering fuzzy logic in supply chain management are provided.

- **Chapter 3:**

This chapter provides a literature review of multi-objective optimisation, covering an overview of concepts and meta-heuristics. In addition, a related work covering multi-objective optimisation in supply chain management is given with regards to supplier selection and inventory management.

- **Chapter 4:**

A problem description of a two-stage fuzzy based optimisation approach proposed in this study is presented explaining each stage separately with providing information required to deal with both supplier selection and inventory planning of a supply chain.

- **Chapter 5:**

Chapter 5 presents implementations of the integrated SCM problem as a single objective problem in a two-stage approach employing interval type-2 fuzzy sets and Simulated Annealing to deal with both supplier selection and inventory planning of a supply chain problem. Two different scalarisation methods, weighted sum and Tchbycheff are used to convert the problem into a scalar problem. A comparison between the performance of scalarisation methods is given using different weight setting scenario. Parameters of the Simulated Annealing algorithm are tuned and the influence of parameters on results is analysed comparing performance of default setting SA to performance of SA with tuned parameter.

- **Chapter 6:**

Four population based meta-heuristic algorithms are used to solve the problem as a multi and a many-objective optimisation problem with an attempt to obtain the best performing approach. First, the problem is taken into account as a multi-objective problem and is tackled by three Multi-objective Evolutionary Algorithms (MOEAs), NSGA-II, SPEA2 and IBEA. Then the same problem is considered as a many-objective optimisation problem treating each cost factor as a separate objective and is solved by NSGA-III. A comparative analysis of approaches used solving the problem in this study is given.

- **Chapter 7**

Chapter 7 presents a summary of the work undertaken in this research along with recommendations for future work.

Chapter 2

Fuzzy Logic and Supply Chain Management

This chapter provides background material on fuzzy logic and a literature review on the application of fuzzy logic to SCM. Also a literature review on relevant approaches to SCM is provided.

2.1 Fuzzy Logic

In many problems, knowledge can be comprised of *objective knowledge* used for problem formulations in a mathematical approach and *subjective knowledge* which demonstrates linguistic information. Generally in mathematical models, subjective knowledge is overlooked [Ross, 2004]. However, Zadeh [1965] introduced fuzzy set theory which allows varying degrees of membership values in a given set to consider both objective and subjective knowledge. Zadeh also proposed type-2 fuzzy sets which provide additional degrees of freedom to model those kind of problems which type-1 fuzzy sets are not capable of handling adequately [Zadeh, 1975]. Mendel and John [2002] stated that there are several sources of uncertainties in type-1 fuzzy logic systems due to the following reasons:

1. Meanings of the words can be varied from different people to people. Thus, vagueness can be revealed in provided rules.

2. When a group of experts have different ideas to define the consequent of the same problem linguistically, uncertainties could arise.
3. The input activating a type-1 fuzzy logic system may be noisy, and therefore imprecise.
4. The data used for parameter tuning of a T1FL system could be noisy.

There are several advantages that type-2 fuzzy sets have in order to cope with these types of uncertainty. Firstly, because of its fuzzy membership functions, a type-2 fuzzy set can deal with numerical and linguistic uncertainties effectively. Secondly, type-2 fuzzy sets with fuzzy membership functions provide a wider coverage of uncertainties while using less rules [Hagras, 2007]. Finally, the extra dimension increases the performance of a type-2 fuzzy logic system while a type-1 fuzzy logic system cannot achieve the same result using the same number of membership functions [Wu and Tan, 2005].

However, type-2 fuzzy logic has some issues. Type-2 fuzzy sets are difficult to apply and understand. Especially, type-reduction and defuzzification processes have been challenging. Because of the computational complexity of type-2 fuzzy sets, Greenfield et al. [2012] stated that most researchers have concentrated on interval type-2 fuzzy models. In addition, according to Mendel et al. [2006], interval type-2 fuzzy sets are quite practical due to their manageable computational complexity. In this section, we define interval type-2 fuzzy sets and associated concepts. For completeness, before discussing interval type-2 fuzzy sets, a review of type-1 fuzzy sets which type-2 fuzzy sets build upon will be given briefly.

2.1.1 Brief Review of Type-1 Fuzzy Sets and Systems

Definition 2.1.1. [Mendel, 2001] A type-1 fuzzy set A in the universe of discourse X can be represented by a type-1 membership function $\mu_A : X \rightarrow [0, 1]$ shown as follows:

$$A = \{(x, \mu_A(x)) \mid \forall x \in X\} \quad (2.1)$$

where $0 \leq \mu_A(x) \leq 1$.

Garibaldi and John [2003] point out that choosing membership functions of linguistic terms has played an essential role in fuzzy systems to capture the opportunities to best represent human knowledge. A variety of membership functions with differing shapes, such as triangle, trapezoidal, Gaussian, have been previously proposed and more can be found in Ross [2004], Zhao and Bose [2002]. This work considers trapezoidal membership functions.

Definition 2.1.2. [Ross, 2004] A trapezoidal type-1 fuzzy set can be defined by four parameters as a, b, c, d where $a < b < c < d$ with a given value x . Assume that the trapezoidal curve is a function of a vector x and depends on four scalar parameters a, b, c and d , as given by;

$$\text{trapezoidal}(x; a, b, c, d) = \begin{cases} 0 & \text{for } x \leq a \\ \frac{x-a}{b-a} & \text{for } a \leq x \leq b \\ 1 & \text{for } b \leq x \leq c \\ \frac{d-x}{d-c} & \text{for } c \leq x \leq d \\ 0 & \text{for } d \leq x \end{cases} \quad (2.2)$$

Definition 2.1.3. [Mendel et al., 2014] A linguistic variable is a linguistic expression of words where fuzzy systems characterise the words in the universe of discourse where the variable is defined. For instance, the performance of suppliers is a linguistic variable defined using linguistic terms as ‘poor’, ‘good’, ‘very good’ and ‘excellent’. Each term is treated as a fuzzy set to model the problem using membership functions.

Definition 2.1.4. [DuBois, 1997] A fuzzy set can be combined by operations, *Union*, *Intersection* and *Complement*. All operations are described by their membership functions as follows:

$$\text{Union: } \mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) = \max[\mu_A(x), \mu_B(x)] \quad (2.3)$$

$$\text{Intersection: } \mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) = \min[\mu_A(x), \mu_B(x)] \quad (2.4)$$

$$\text{Complement: } \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (2.5)$$

where fuzzy sets, A and B , their membership functions (MFs), $\mu_A(x)$ and $\mu_B(x)$ and the symbol $\vee$ represents the maximum operator and $\wedge$ depicts the minimum

operator. $\mu_{A \cup B}(x)$ and $\mu_{A \cap B}(x)$ can be explained more generally by using t-conorms and t-norms.

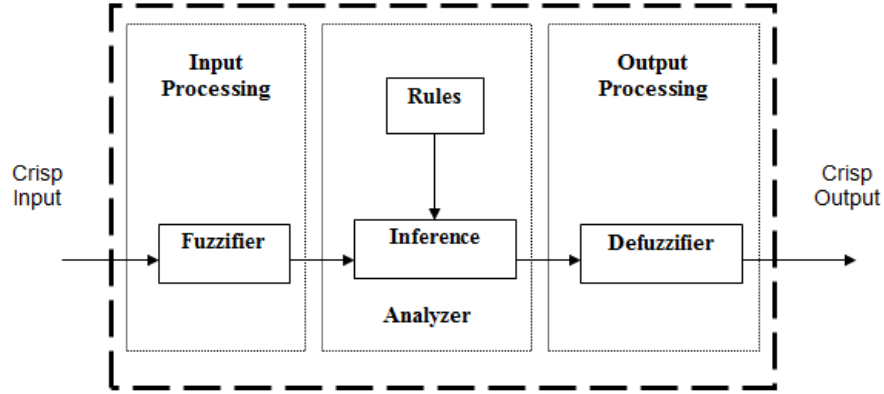


FIGURE 2.1: Type-1 fuzzy logic system [Mendel, 2001].

Traditional fuzzy logic systems are composed of four components, rules, fuzzification process, fuzzy inference engine and defuzzification process as shown in Figure 2.1. But in this research, we only used the defuzzification process. Defuzzification is the conversion of a fuzzy set to a precise single value as opposed to a fuzzy set. Although, there are several defuzzifiers commonly used in the literature such as centroid, centre-of-sums, height, modified height and centre-of-set [Mendel, 2001], the centroid is used in the study. *Centroid* is a type of defuzzifier which determines the centre of gravity (centroid) using the union operation (a t-conorm) to combine the result of type-1 fuzzy sets as an output set and discretize the output set into a single value. The centroid defuzzifier is given as:

$$m_c(x) = \frac{\sum_{i=1}^N m_i \mu_A(m_i)}{\sum_{i=1}^N \mu_A(m_i)} \quad (2.6)$$

where A is a fuzzy set with associated i^{th} membership function $\mu_A(m_i)$ and the fuzzy set has been discretised into N points.

2.1.2 Basic Concepts of Interval Type-2 Fuzzy Sets and Systems

A fuzzy logic system explained using at least one type-2 fuzzy set is called a type-2 fuzzy logic system (FLS). The structure of a type-2 FLS is demonstrated in

Figure 2.2. As seen in Figure 2.2, a type-2 FLS comprises fuzzifier, inference, rules and output processing. In other words, a type-2 FLS is very similar to a type-1 FLS. The only difference is that the defuzzifier process of a type-1 FLS is replaced by the output processing consists of type-reduction followed by defuzzification [Karnik and Mendel, 1999]. In a type-2 FLS, an operation which gives a type-1 fuzzy set from a type-2 fuzzy set is called type-reduction rather than defuzzification. After type-reduction, to find a crisp number, the defuzzification which produces a crisp output from the fuzzy set is applied [Karnik and Mendel, 1998]. In this section, rather than explaining each of block shown in Figure 2.2, only selected relevant definitions and operations of type-2 FLSs are briefly introduced from Mendel et al. [2006].

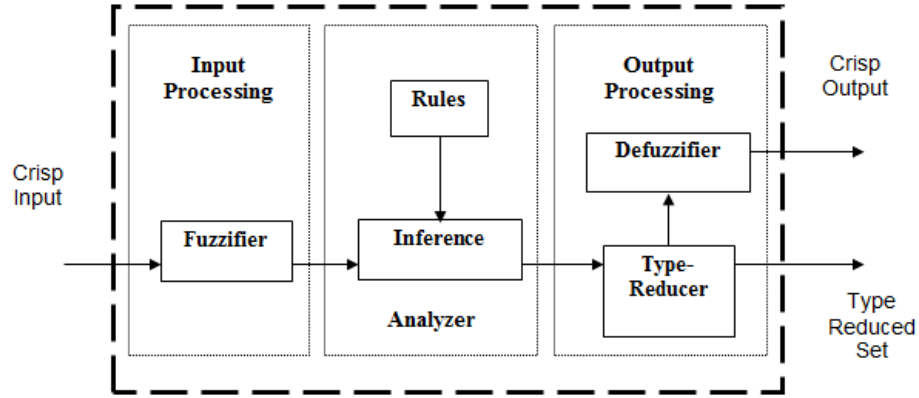


FIGURE 2.2: Type-2 fuzzy logic system [Karnik and Mendel, 1999].

In Definition 2.1.1, for a type-1 fuzzy set A , the membership grade (the degree of membership or membership function) of $x \in X$ in A is shown as $\mu_A(x)$, which is a crisp number in $[0, 1]$. On the other hand, for a type-2 fuzzy set $\tilde{A}$ in X , the membership grade of $x \in X$ in $\tilde{A}$ is $\mu_{\tilde{A}}$ which is a type-1 fuzzy set in $[0, 1]$. Thus, a type-2 membership function $\mu_{\tilde{A}}$ is not a crisp number, it is the support of a fuzzy set $\tilde{A}$, J_x in $[0, 1]$ shown as $J_x \subseteq [0, 1]$ [Karnik and Mendel, 1999].

Definition 2.1.5. [Mendel et al., 2006] A type-2 fuzzy set $\tilde{A}$ in the universe of discourse X can be represented by a type-2 membership function $\mu_{\tilde{A}}$ shown as follows:

$$\tilde{A} = \{(x, u), \mu_{\tilde{A}}(x, u) \mid \forall x \in X, \forall u \in J_x \subseteq [0, 1]\} \quad (2.7)$$

where $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$. As the primary membership function is between 0 and 1, a type-2 fuzzy set $\tilde{A}$ can be expressed as:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u) \quad J_x \subseteq [0, 1] \quad (2.8)$$

where $\int \int$ denotes a union over all admissible x and u .

Definition 2.1.6. [Mendel et al., 2006] Let $\tilde{A}$ be a type-2 fuzzy set in the universe of discourse X represented by the type-2 membership function $\mu_{\tilde{A}}$. If all $\mu_{\tilde{A}}(x, u) = 1$ for $\forall x \in X$ and $u \in J_x \subseteq [0, 1]$, then $\tilde{A}$ is called an interval type-2 fuzzy set, shown as the following:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} 1 / (x, u) \quad J_x \subseteq [0, 1] \quad (2.9)$$

where $J_x \subseteq [0, 1]$, i.e.

Definition 2.1.7. [Mendel et al., 2006] The IT2FS can be considered as a particular case of type 2 fuzzy set, where the upper and lower membership functions are both Type-1 membership functions, respectively. As an example, a trapezoidal IT2FS $\tilde{A}_i$ for all $x \in X$ represented by;

$$\begin{aligned} \tilde{A}_i = (\tilde{A}_i^U, \tilde{A}_i^L) = & ((a_{i1}^u, a_{i2}^u, a_{i3}^u, a_{i4}^u; h_1(\tilde{A}_i^U), h_2(\tilde{A}_i^U)), \\ & (a_{i1}^l, a_{i2}^l, a_{i3}^l, a_{i4}^l; h_1(\tilde{A}_i^L), h_2(\tilde{A}_i^L))) \end{aligned} \quad (2.10)$$

where $h_j(\tilde{A}_i^U)$ and $h_j(\tilde{A}_i^L)$ for $1 \leq j \leq 2$ depict membership values of the corresponding elements $a_{i(j+1)}^u$ and $a_{i(j+1)}^l$, respectively [Gong, 2013]. The height of each constituent membership function is not explicitly defined as it is assumed to be equal to 1. Figure 2.3 indicates a trapezoidal T2FS which comprises of the upper and the lower membership functions, $\tilde{A}_i^U$, $\tilde{A}_i^L$ where the corresponding elements of each membership function are defined as $a_{i1}^u, a_{i2}^u, a_{i3}^u, a_{i4}^u$ and $a_{i1}^l, a_{i2}^l, a_{i3}^l, a_{i4}^l$, respectively. The height of each constituent membership function is shown as $h_1(\tilde{A}_i^U)$, $h_2(\tilde{A}_i^U)$ for the upper membership function $\tilde{A}_i^U$, and $h_1(\tilde{A}_i^L)$, $h_2(\tilde{A}_i^L)$ for the lower membership function $\tilde{A}_i^L$.

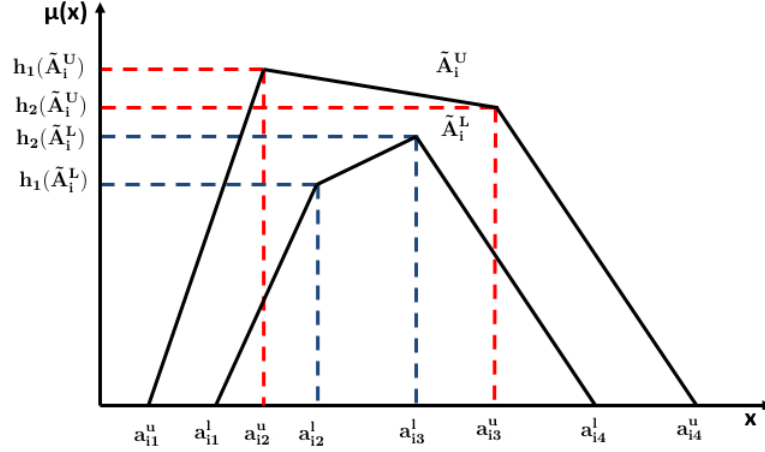


FIGURE 2.3: The upper trapezoidal membership function $\tilde{A}_i^U$, the lower trapezoidal membership function $\tilde{A}_i^L$ of the IT2FS $\tilde{A}$ [Gong, 2013].

Definition 2.1.8. [Mendel et al., 2006] Algebraic operations used in this work are addition, multiplication, and they are defined as follows for the trapezoidal interval type-2 fuzzy sets $\tilde{A}_1$ and $\tilde{A}_2$;

$$\begin{aligned} \tilde{A}_1 = (\tilde{A}_1^U, \tilde{A}_1^L) = & ((a_{11}^u, a_{12}^u, a_{13}^u, a_{14}^u; h_1(\tilde{A}_1^U), h_2(\tilde{A}_1^U)), \\ & (a_{11}^l, a_{12}^l, a_{13}^l, a_{14}^l; h_1(\tilde{A}_1^L), h_2(\tilde{A}_1^L))) \end{aligned} \quad (2.11)$$

$$\begin{aligned} \tilde{A}_2 = (\tilde{A}_2^U, \tilde{A}_2^L) = & ((a_{21}^u, a_{22}^u, a_{23}^u, a_{24}^u; h_1(\tilde{A}_2^U), h_2(\tilde{A}_2^U)), \\ & (a_{21}^l, a_{22}^l, a_{23}^l, a_{24}^l; h_1(\tilde{A}_2^L), h_2(\tilde{A}_2^L))) \end{aligned} \quad (2.12)$$

The addition operation is:

$$\begin{aligned} \tilde{A}_1 \oplus \tilde{A}_2 = & (\tilde{A}_1^U, \tilde{A}_1^L) \oplus (\tilde{A}_2^U, \tilde{A}_2^L) \\ = & ((a_{11}^u + a_{21}^u, a_{12}^u + a_{22}^u, a_{13}^u + a_{23}^u, a_{14}^u + a_{24}^u; \\ & \min(h_1(\tilde{A}_1^U), h_1(\tilde{A}_2^U)), \min(h_2(\tilde{A}_1^U), h_2(\tilde{A}_2^U))), \\ & (a_{11}^l + a_{21}^l, a_{12}^l + a_{22}^l, a_{13}^l + a_{23}^l, a_{14}^l + a_{24}^l; \\ & \min(h_1(\tilde{A}_1^L), h_1(\tilde{A}_2^L)), \min(h_2(\tilde{A}_1^L), h_2(\tilde{A}_2^L)))) \end{aligned} \quad (2.13)$$

The multiplication operation is:

$$\begin{aligned}
\tilde{A}_1 \otimes \tilde{A}_2 &= (\tilde{A}_1^U, \tilde{A}_1^L) \otimes (\tilde{A}_2^U, \tilde{A}_2^L) \\
&= ((a_{11}^u \times a_{21}^u, a_{12}^u \times a_{22}^u, a_{13}^u \times a_{23}^u, a_{14}^u \times a_{24}^u; \\
&\quad \min(h_1(\tilde{A}_1^U), h_1(\tilde{A}_2^U)), \min(h_2(\tilde{A}_1^U), h_2(\tilde{A}_2^U))), \\
&\quad (a_{11}^l \times a_{21}^l, a_{12}^l \times a_{22}^l, a_{13}^l \times a_{23}^l, a_{14}^l \times a_{24}^l; \\
&\quad \min(h_1(\tilde{A}_1^L), h_1(\tilde{A}_2^L)), \min(h_2(\tilde{A}_1^L), h_2(\tilde{A}_2^L))).
\end{aligned} \tag{2.14}$$

Multiplying a fuzzy set by a constant k is defined as follows:

$$\begin{aligned}
k\tilde{A}_1 &= k(\tilde{A}_1^U, \tilde{A}_1^L) \\
&= ((ka_{11}^u, ka_{12}^u, ka_{13}^u, ka_{14}^u; h_1(\tilde{A}_1^U), h_2(\tilde{A}_1^U)), \\
&\quad (ka_{11}^l, ka_{12}^l, ka_{13}^l, ka_{14}^l; h_1(\tilde{A}_1^L), h_2(\tilde{A}_1^L))).
\end{aligned} \tag{2.15}$$

2.2 Supply Chain Management

In this thesis, a two-stage integrated supply chain management problem which considers the supplier selection and inventory management is addressed. In the first stage, suppliers are ranked based on various criteria, including cost, delivery, service and product quality using interval type-2 fuzzy sets to get the risk value of each supplier whilst considering their ranks. In the following stage, an inventory model which includes information achieved in the first stage is built to capture the impact of supplier risk on the total cost of a supply chain. For this reason, in this section, the fundamental definitions of SCM, together with related literature review are given.

Management is on the verge of a major breakthrough in understanding how industrial company success depends on the interactions between the flows of information, materials, money, manpower, and capital equipment. The way these five flow systems interlock to amplify one another and to cause change and fluctuation will form the basis for anticipating the effects of decisions, policies, organizational forms, and investment choices. [Forrester, 1958, p.37]

Forrester [1961] introduced the concept of a supply chain derived from the definition of management [Forrester, 1958]. There is no clear agreement among researchers about the definition of SCM. Cooper et al. [1997] defined SCM as “an integrative philosophy to manage the total flow of a distribution channel from supplier to the ultimate user”. According to Christopher [1992], SCM is a network of organisations that are involved, through upstream and downstream linkages, in the different processes and activities that generate value in the form of products and services in the hands of the ultimate consumer. Keah et al. [1998] explained that SCM encompasses materials/supply management from the supply of basic raw materials to final product (and possible recycling and re-use). SCM focuses on how companies utilise their suppliers in terms of processes, technology and capacity to enhance the competitive advantage. It is a management philosophy that extends traditional *intra-enterprise* activities by bringing trading partners together with the common goal of optimisation and efficiency. Saunders [1997] mentioned that “they focus on the external environment of an organisation, with the boundaries of the latter defined conventionally in terms of an entity identified legally as a company or some other form of business unit”. Despite the fact that there are a number of definitions of the term SCM [Mentzer et al., 2001], a widely approved definition of supply chain management is given as follows [Vitasek, 2010, p.116]:

Supply chain management encompasses the planning and management of all activities involved in sourcing, procurement, conversion, and all logistics management activities. Importantly, it also includes the coordination and collaboration with channel partners, which can be suppliers, intermediaries, third-party service providers, and customers. In essence, supply chain management integrates supply and demand management within and across companies. Supply chain management is an integrating function with primary responsibility for linking major business functions and business processes within and across companies into a cohesive and high-performing business model. It includes all the logistics management noted above, as well as manufacturing operations, and it drives coordination of processes and activities with and across marketing, sales, product design, finance and information technology.

From these selected definitions, basically SCM is the management of material flows from suppliers to customers through production processes considering information flows among the whole processes of supply chains with the aim to provide long-term relations between customers and suppliers [Thomas and Griffin, 1996]. Supply chain management is the integration of all facilities which add value for buyers from the procurement of raw materials to the delivery of the end product [Setak et al., 2012].

SCM comprises six fundamental elements highlighted in each definition of SCM given above: service level management, order and demand management, production management, supply management, distribution management, integrated SCM planning and execution [Cappello et al., 2008]. In order to achieve competitive advantages for those involved, two key aspects considered in this study are i) working with a small number of reliable suppliers, and ii) efficient allocation of resources within the supply chain through inventory planning.

2.2.1 Supplier Selection

Choosing a supplier can impact on the cost and the quality of products. Evaluation and selection of suppliers is a critical and essential issue in a supply chain. In light of this, selection of proper suppliers whilst taking into consideration supplier evaluation is of ongoing interest to practitioners as well as researchers [Turk et al., 2015]. Katsikeas et al. [2004] pointed out that the purchasing operation requires a list of decision criteria determined to examine supplier performance. In their work, it is found that based on the survey sample of 237 distributors, four main criteria on 20 purchasing decision criteria such as reliability, competitive pricing, service support and technological capability are identified as the most important factors to achieve and maintain the right performance in the competitive environment. In addition, Ho et al. [2010] provided an overview of 78 articles published in period 2000 - 2008 focusing on the methods used, the criteria selected for evaluation of vendors and the effectiveness of the methods. A thorough review of previous studies indicated that a variety of exact and (meta)heuristic solution methods, such as, mathematical programming and genetic algorithms, have been used to evaluate and select suppliers [Ho et al., 2010]. Some of those methods treat the supplier selection problem as a Multiple Attribute Decision Making (MADM) problem and

take the requirements of decision makers into account as well during the solution process. There are various evaluation criteria commonly used for supplier selection, which range from the quality of product and service to production capacity [Setak et al., 2012]. Moreover, the supplier selection problem is not based on precise measures, but often on vague and imprecise terms. To sum up, a number of methods have been successfully applied to the supplier selection problems. The most prominent of these methods are:

- 1) *Multi Attribute Decision Making (MADM)* is about how to make decisions in the discrete decision spaces and focuses on how to choose one among different pre-determined alternatives. MADM methods consist of several approaches such as Analytic Hierarchy Process (AHP), Analytic Network Process (ANP), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) and Multi Attribute Utility Technique (MAUT). In a MADM problem, a number of alternative suppliers are chosen based upon constraints, preference and priorities of decision makers [Rao, 2007].

Saaty [1987] introduced the AHP method which relies on a series of pair-wise comparisons with regard to the judgements of decision makers. The AHP method has been used in a wide variety of applications in the literature. Ghodsypour and O'Brien [1998] proposed an integrated AHP and linear programming method to evaluate both quantitative and qualitative factors considering a trade-off between them with difference preference attitudes. In addition, Kahraman et al. [2003] proposed a fuzzy AHP method to handle a multi-criteria supplier selection problem. In this paper, they extend the traditional view of supplier selection including a fuzzy method. Preferences about the importance of criteria are specified using linguistic variables with overcoming difficulties of being a complex environment. Tahriri et al. [2008] stated that supplier selection is a complex problem formed by qualitative and quantitative criteria. In order to find the best vendor, a trade-off between these criteria has been essential to supplier selection problems. In order to handle it, an AHP method is performed and developed in this study. Based on results, development of a multi-criteria decision model for measuring supplier performance is an effective way to identify and evaluate vendors with regard to tangible and intangible factors of the problem. Galankashi et al.

[2016] presented a fuzzy AHP model and implemented that in the automotive industry to select the best supplier.

In addition, Sarkis and Talluri [2002] developed an AHP method and proposed an extended version of AHP named as Analytic Network Process. The difference between these two methods is that the ANP follows a network structure rather than a hierarchical process. In the literature, Gencer and Gürpınar [2007] presented an ANP implementation in an electronic company and showed that the ANP provides a flexible decision process.

Another MADM method is TOPSIS which combines both tangible and intangible attributes and compares them with regard to the distance from the ideal solution [Hwang and Yoon, 1981]. Chen [2000] drew attention to the necessity of improvement of TOPSIS by combining TOPSIS with fuzzy set theory. Based on the results found in this study, fuzzy set theory helps to deal with the supply chain problem. Wang et al. [2009] addressed the supplier selection evaluation related to a fuzzy hierarchical TOPSIS, which uses a simplified parameterized metric distance. The aim of the approach is to achieve a desirable improvement on Chen's Fuzzy TOPSIS. Junior et al. [2014] applied fuzzy AHP and fuzzy TOPSIS methods to a supplier selection problem and compared the two models with regard to changes of criteria, efficiency and computational complexity. Furthermore, Gong [2013] presented an IT2FS approach to a Fuzzy Multiple Attribute Group Decision Making problem to select the best supplier among three potential vendors.

The MAUT is another MADM method which can effectively handle both deterministic and stochastic decision environments. Sanayei et al. [2008] proposed an integrated method considering both a MAUT and linear programming to rate suppliers and choose the best supplier. Shaik and Abdul-Kader [2011] addressed a green supplier selection problem considering price, quality, lead time and flexibility using the MAUT in order to evaluate green suppliers.

- 2) *Mathematical Programming:* Various type of mathematical programming methods have been used for supplier selection problems. As mentioned in MADM methods, Ghodsypour and O'Brien [1998] proposed an integrated AHP and linear programming approach. A fuzzy linear programming model is used to examine the behaviour of a supply chain in the work of Zarandi et al. [2007]. Ghodsypour and O'Brien [2001] built a mixed integer non-linear

programming model for a multiple sourcing problem in order to minimise the total purchasing cost by determining the optimum allocation of products to suppliers. Karpak et al. [2001] used a goal programming approach to improve the way to choose the best suppliers assessing trade-offs among three goals: cost, quality and delivery reliability. Also a fuzzy multi-objective linear model is used to cope with imprecision in supplier selection [Amid et al., 2006].

In addition, data envelopment analysis (DEA) introduced by Charnes et al. [1978] is a non-parametric approach to measure the relative performance of complex entities which comprises qualitative and quantitative factors. There are numerous applications used the DEA approach for selecting suppliers [Ho et al., 2010]. Weber [1996] applied the DEA method for a baby food manufacturer in order to measure the performance of suppliers. In order to improve the power of DEA, it is integrated with other methods such as AHP [Ramanathan, 2007], multi-objective optimisation [Weber et al., 2000], neural network [Çelebi and Bayraktar, 2008]. Setak et al. [2012] noted that the amount of research using DAE methods has reduced since 2006, although DEA has played an essential role in the literature.

- 3) *Artificial Intelligence Methods* can handle complexity and ambiguity using their functions that learn from experts or cases happened in the past. In the literature, there are several works about supplier selection using artificial intelligence methods such as cluster analysis (CA), case based reasoning (CBR) and neural networks.

Cluster analysis is a statistical approach which classifies a number of items into a number of groups (clusters) by using attribute scores determined such as the minimum differences between items within a cluster [de Boer et al., 2001]. Che and Wang [2010] pointed out that selecting suppliers after cluster analysis helps to eliminate undesirable suppliers which could not meet the customer demand. In addition, Choy et al. [2002] proposed an integrated customer-supplier relationship model using the case based reasoning technology to predict and to assess suppliers and to verify the accuracy of the result. Choy et al. [2003a] designed an intelligent supplier relationship management system (ISRMS) with the CBR technology and implemented that in Honeywell Consumer Product Limited (Hong Kong). It is clearly seen

that there is a significant decrease in the percentage of delay in delivery, the amount of unacceptable units while there is an essential increase in customer satisfaction. Moreover, Choy et al. [2003b] developed this system using both CBR and artificial neural networks approaches to choose and benchmark a list of suppliers selected. They found that there is an important reduction in the duration of order allocation and the selection of potential suppliers.

- 4) *Fuzzy Set Theory*: Ambiguity and vagueness in qualitative factors of supplier selection problems have an effect on obtaining a realistic selection of suppliers. Numerous applications of supply selection problems are solved using fuzzy set theory to overcome uncertainties in supplier selection models.

Jain et al. [2004] presented an evolutionary fuzzy-based approach to evaluate the performance of suppliers in the presence of both qualitative and quantitative data. In this study, an evolutionary fuzzy system was used to find a set of acceptable rules for several cases with regard to the performance of vendors. This research indicated that the genetic algorithm (GA) led to the evolution of fuzzy rules and the successful execution of the expert system depends on not only fuzzy rules but also the length of the chromosome in the evolutionary fuzzy system.

Chen et al. [2006] tackled supplier selection problems which comprise of uncertain and imprecise data. Although, there have been a number of methods which consider numerical data to deal with supplier selection problems, Chen et al. [2006] handled these problems with an approach based on linguistic assessments. They addressed supplier selection problems through a fuzzy method which is able to capture linguistic assessments effectively. Results of the study showed that this method is an effective way to provide more objective information about suppliers and to evaluate them in a SC system. Sarkar and Mohapatra [2006] drew attention to the necessity of reducing the number of vendors in order to achieve a strong buyer-supplier relationship. This article aimed to reduce the number of suppliers considering performance and capacity of suppliers and used a fuzzy approach in order to overcome imprecise information. Strengths and weaknesses of each supplier were listed and suppliers were classified in terms of sustainability using the performance information of each supplier.

Amid et al. [2006] addressed a fuzzy multi-objective linear model to overcome vagueness and imprecision in a supplier selection problem. The performance of suppliers on cost, quality, delivery and service are defined as objectives of the model. Different importance weights of each criterion are used to analyse their effects on suppliers and results showed that the varying importance of criteria has played a significant role in an effective supplier management process. Also the quality level is increased while the cost is reduced. Amid et al. [2009] developed a fuzzy multi-objective model considering not only varying importance of criteria, but also the size of the order quantities. The model makes an improvement in supplier selection assessment and provides managers to allocate orders to vendors efficiently and to manage suppliers effectively. Amid et al. [2011] proposed a weighted max-min fuzzy model as a single objective model to decrease the complexity of the problem. The proposed model provides a simple way to evaluate suppliers and to allocate orders to them.

Pattnaik [2011] created a fuzzy arithmetic model to achieve a desirable level of supplier performance in quality, flexibility, delivery and service. In the proposed model, a number of vendors in a set of criteria are evaluated by decision makers and the importance of these criteria are assigned a particular ratio within linguistic expressions as ‘low’, ‘middle’ and ‘high’. In order to obtain the performance of suppliers, linguistic assessments are converted into fuzzy numbers to build up a fuzzy model. The model helps decision makers to select a convenient supplier for each product.

In the study proposed by Ordoobadi [2009], decision makers examined the importance of each criterion to evaluate vendors and the performance rating of suppliers as the crucial attributes in the decision making process. Fuzzy sets are used to deal with the uncertainties and scoring of the suppliers based on the given criteria. Suppliers are then ranked using their scores. This work was extended, investigating uncertainty on the shapes of membership functions using different modified fuzzy sets [Turk et al., 2014].

A summary overview of the methods used in supplier selection problems is now given. MADM methods can handle large and complex problems in the presence of conflicting criteria and uncertainty. They have a number of advantages. For

instance, AHP is easy to use due to its hierarchical structure and pairwise comparison allow users to give different weight for each criterion. Also MAUT takes uncertainty in real-world applications into account. Moreover, TOPSIS which has a simple structure including a certain number of steps is capable of solving various problems regardless of their attributes. However, due to uncertainty in subjective judgements of decision makers, the lack of information and conflicts among the criteria, MADM methods are unable to produce a precise solution [Velasquez and Hester, 2013]. In order to overcome these difficulties, combining different methods has become increasingly popular such as an integrated AHP and linear programming [Ghodsypour and O'Brien, 1998], the combination of DEA and neural network [Çelebi and Bayraktar, 2008], a combination of fuzzy, AHP and TOPSIS [Wang et al., 2009]. In addition, artificial intelligence methods such as cluster analysis and CBR have been used to cope with complexity and ambiguity in real-world problems. A major advantage of these methods is that they can adapt to changes in environment easily. Nevertheless, they are sensitive to imprecise data [Velasquez and Hester, 2013]. Because of the uncertain nature of supplier selection problems, fuzzy set theory has taken a crucial place in these problems. In the literature, it is applied either as an individual approach or as a hybrid method which comprises of fuzzy and at least one of other methods such as fuzzy and AHP [Galankashi et al., 2016, Kahraman et al., 2003], fuzzy and TOPSIS [Chen, 2000, Wang et al., 2009]. Therefore, in the study, we will use a fuzzy approach to assess the performance of suppliers (detailed in Chapter 4).

2.2.2 Inventory Planning

Inventory management is an integrated approach to plan and control inventory while considering the whole network from suppliers to end users. For an effective inventory management, there are only two essential decisions that need to be made: i) the amount of each order as one objective and ii) the right time to give these orders as another objective [Ziukov, 2015]. These objectives allow the amount of late delivery to be reduced, while minimising costs associated with holding stock and stock outs where the demand exceeds the available stocks in the inventory. Miller et al. [2011], Miller and John [2010] pointed out that a supply chain with a well managed inventory attempts to satisfy demand, preventing stock outs and reducing holding costs.

In order to enhance inventory policies, Hillier and Lieberman [1986] drew attention to the necessity of a scientific inventory management. A scientific inventory management builds a computerised mathematical model with respect to the behaviour of the inventory. It allows companies to achieve an optimum inventory policy recording current information about the inventory and then to assess this information to make an improvement on the proposed model. Since inventory policies have an essential impact on profitability, components of an inventory can be defined as different costs which determine the profitability of a company [Hillier and Lieberman, 1986]. A brief explanation of each cost is now given:

Order Cost is the total price paid to purchase any amount of component. Let x represents an amount of purchasing components, $c(x)$ denotes a function of order cost. The simplest calculation of $c(x)$ is $c \times x$, where c denotes the unit price paid. Another calculation of order cost is:

$$c(x) = \begin{cases} 0 & \text{if } x = 0 \\ m + (c \times x) & \text{if } x > 0 \end{cases} \quad (2.16)$$

where m is the setup cost [Hillier and Lieberman, 1986].

Setup Cost, also known as Batch Cost, is the cost of setting up an order. This cost involves the administrative cost of ordering and setting up any machine to start a production run [Miller et al., 2009].

Holding cost is the cost paid to keep and maintain the inventory that remains in the storage for more than one period. This cost includes “the cost of capital tied up, space, insurance, protection, and taxes attributed to storage” [Hillier and Lieberman, 1986].

Stock-out cost, also known as shortage cost, is charged when the demand exceeds the available stock. Two policies are used to calculate this cost: backlogging and non-backlogging. Based on backlogging, demand is held until it can be provided by the next normal delivery. However, in the non-backlogging policy, the demand exceeds the available stock is either provided by a priority shipment or cancelled by the manufacturer rather than waiting for the next normal delivery [Hillier and Lieberman, 1986].

Transportation cost is the expense to carry an item between customers and production plants or suppliers and production plants. Let x represents transportation cost, $t(x)$ denotes a function of transportation cost; $t(x)$ is calculated by $x \times d$ where d denotes the distance between two nodes [Miller et al., 2009].

In today's competitive environment, most commercial organisations try to manage the flow of materials effectively whilst considering the relationships between vendors, manufacturers, distribution centres, customers and other services in order to maintain their success [Turk et al., 2015]. In addition, recent research indicates that there has been a growing interest in the field of supplier selection along with effective inventory planning [Parhizkari et al., 2013]. Rezaei and Davoodi [2008] addressed a multi-item inventory problem considering the selection of vendors with respect to the quality of products. Their research explores the problem of deciding which materials to order from which supplier and when.

Ghodsypour and O'Brien [2001] described a multiple sourcing problem that considers elements of inventory management defined as, logistics cost, storage costs and subjective aspects of vendors. Mohammaditabar and Ghodsypour [2014a] extended the work of Ghodsypour and O'Brien [2001] by assigning orders to suppliers chosen. They also built a joint order process to reduce the overall cost. The components of inventory are divided into a number of groups to achieve desirable costs of storage and inventory. It shows that this method decreases the overall cost of logistics along with inventory holding cost and ordering cost, considerably.

Supplier selection combined with effective inventory planning has been studied by a number of researchers [Ghodsypour and O'Brien, 2001, Mohammaditabar and Ghodsypour, 2014b, Parhizkari et al., 2013]. The majority of the previous work treated SCM problems as a single objective problem. Although some studies considered multiple objectives, the problem was dealt with using an aggregating approach designed for optimising a single objective which was obtained by scalarising multiple objectives into one via some scalarisation method, such as, weighted sum (detailed in Chapter 3.7).

2.3 Fuzzy Logic in SCM

In the literature, there are many works on supply chain management problems solved by fuzzy approaches. Selected works of fuzzy logic in SCM are given as followings. Petrovic et al. [1998] dealt with an inventory management problem in an uncertain environment by generating fuzzy models and simulations. Uncertainty is associated with customer demand and external supply of raw materials. To achieve a desirable delivery performance with an acceptable total cost for a SC system, stock levels and order quantities for each inventory are examined by considering a finite time horizon in these uncertainties. The fuzzy model provides the optimum ordering levels of each inventory. The supply chain performance is evaluated by simulations. Their work indicates that type-1 fuzzy model and simulations are able to cope with a supply chain problem in the presence of uncertainties. Petrovic et al. [1999] extended the model regarding vagueness in customer demand, supply deliveries along the SC and external supply by focusing on ‘decentralised control of each inventory’ and ‘partial coordination in the inventories control’. The fuzzy model provides an improvement on determination of order quantities and the simulation is used to measure the performance of the supply chain within a finite time horizon under consideration of different inventory controls. Based on results, uncertain supply delivery for the whole supply chain has a significant influence on the behaviour of a supply chain. In the other work of Petrovic [2001], a simulation tool named as SCSIM was developed to investigate the supply chain performance and the behaviour of a supply chain and to achieve a better understanding of a SC interpreting these information.

In order to improve the overall SC performance, Xie et al. [2006] divided a SC problem into several independent sub-problems and optimised them individually. It is assumed that customer demand is uncertain and a fuzzy model is used for each sub-problem. The optimisation process is repeated iteratively to obtain a desirable result for the performance of the supply chain. The study provided a comparison among performances of supply chain under a decentralised control. In addition, Peidro et al. [2009] addressed a supply chain planning problem related to uncertainties in supply, demand and process using a fuzzy-mixed integer linear programming model which aims to minimise the total cost with maximum demands. In order to assess the influence of the fuzzy model on the effectiveness of the SC planning, its behaviour is compared with a deterministic model formed

by the same data. It is found that the capability of the fuzzy model is good enough in order to cope with real situations. Lu et al. [2011] also proposed to use a fuzzy mixed-integer linear programming model in an uncertain environment for SC planning problems concerned with organisation excellence, geographical location and capacity of each SC components. With regard to cost reduction (about 1.19 millions dollars in a year), an important improvement on the cost is achieved by the fuzzy model.

Miller et al. [2008] investigated a resource planning process using type-2 fuzzy sets and a GA instead of traditional approaches such as simulations. The objective of the work is to obtain the best decisions in SC operations within a fuzzy model and the model reliability is controlled by the GA. In terms of the overall cost, the GA finds acceptable solutions as £49,100.00 where an actual solution was £44,635.00. Miller et al. [2009] improved the proposed model using interval type-2 fuzzy logic (IT2FL) to overcome uncertainties in the resource planning process and a GA is applied to the fuzzy model in order to find near-optimal solutions. Based on results, the near-optimal solution is only 1.38% different from the actual solution. A cost effective plan is reached with a sensible distribution network. Miller and John [2010] developed a methodology to measure customer satisfaction by adding a determined service level as a constraint into the fuzzy model. Miller et al. [2011] applied the methodology in real-world complex systems as a large-scale inventory management problem. In this case, IT2FL model is used to evaluate the resource plan calculating costs rather than to obtain the optimum result as previous works of authors [Miller and John, 2010, Miller et al., 2008, 2009]. Also Simulated Annealing is applied to achieve a desirable performance of the fuzzy model. With regard to the customer satisfaction level (found 99.9 percent) and acceptable results for the total cost, it is clear that SA is successfully applied to the optimisation problem modelled by the IT2FL.

Moreover, as mentioned in Section 2.2.1, fuzzy set theory has been applied to supplier selection problems using different approaches such as an evolutionary fuzzy-based approach [Jain et al., 2004], a fuzzy classification of suppliers model [Sarkar and Mohapatra, 2006], a fuzzy linguistic assessments approach [Chen et al., 2006], a fuzzy multi-objective linear model [Amid et al., 2006, 2009], a weighted max-min fuzzy model [Amid et al., 2011], a fuzzy arithmetic approach [Ordoobadi, 2009, Pattnaik, 2011], a fuzzy AHP method [Galankashi et al., 2016, Kahraman et al.,

2003], a fuzzy TOPSIS model [Chen, 2000], a fuzzy hierarchical TOPSIS [Wang et al., 2009], a comparison between a fuzzy AHP and a fuzzy TOPSIS model [Junior et al., 2014], a fuzzy group multiple attribute decision making approach [Chen and Lee, 2010a,b, Chen et al., 2012, Gong, 2013, Wu and Chen, 2007].

In conclusion, through the analysis of different selected papers, it is obvious that fuzzy logic is applied into various types of supply chain problems. Due to the uncertain nature inherent to the supply chain, fuzzy logic is a method which is capable of coping with different types of uncertainties in a supply chain. Also, in order to carry out a good assessment of vendors, a fuzzy approach either as an individual approach or as a hybrid method is chosen for supplier selection problems in most cases (detailed in Section 2.2.1). A major advantage of fuzzy logic is that it provides a more flexible model integrating both qualitative and quantitative aspects of the supply chain. Moreover, it is capable of dealing with imprecise inputs comprised of subjective judgements of decision makers and lack of information. Hence, in this research, interval type-2 fuzzy sets will be used to evaluate suppliers.

2.4 Summary

This Chapter presents concepts and explanations of fuzzy logic including interval type-2 fuzzy sets. In addition, a review of supply chain management (the application for this research) has been supplied.

Based on literature, it could be said that type-2 fuzzy sets are capable of handling uncertainty better than type-1 sets in SCM. In addition, because of the complexity in type-reduction and defuzzification processes in type-2 fuzzy sets, the application fields of IT2FS have been increasing day by day and due to the uncertain nature of SCM there has been a rapid growth in fuzzy applications in SCM. This paves the way for studying a fuzzy approach to deal with supplier selection in the first stage of integrated supply chain management problem proposed in this thesis.

The next chapter provides an overview of multi-objective optimisation including multi-objective evolutionary algorithms. Also, related definitions and literature review of metaheuristics are given.

Chapter 3

Multi-objective Optimisation

This chapter provides an overview of the fundamental concepts in multi-objective optimisation. Metaheuristics used for multi-objective optimisation are reviewed. Finally, a related work covering multi-objective optimisation for SCM is given with regards to supplier selection and inventory management.

3.1 Basic Concepts and Terminology

In single objective optimisation, the objective function is clearly defined with several hard constraints to achieve a single optimal solution while in multi-objective optimization, multiple objective functions are defined to obtain a set of solutions considering a set of constraints [Zitzler, 1999]. Generally, multi-objective problems include a decision variables, b objective functions and co constraints. Objective functions and constraints are functions of the decision variables indicated as follows;

$$\begin{aligned} \text{minimise} \quad & \mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(x), f_2(x), \dots, f_b(x)) \\ \text{subject to:} \quad & \mathbf{e}(\mathbf{x}) = (e_1(x), e_2(x), \dots, e_{co}(x)) \leq 0 \\ \text{where} \quad & \mathbf{x} = (x_1, x_2, \dots, x_a) \in X, \\ & \mathbf{y} = (y_1, y_2, \dots, y_b) \in Y \end{aligned} \tag{3.1}$$

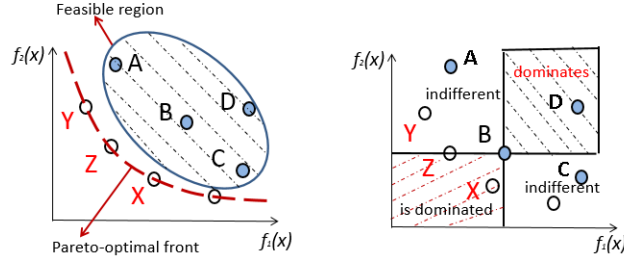


FIGURE 3.1: Pareto optimality in the objective space (left) and the possible relations of solutions in the objective space (right) [Zitzler, 1999].

where $\mathbf{x}$ represents the decision vector while $\mathbf{y}$ is the objective vector, X is denoted as the decision space and Y is the objective space. The constraints $\mathbf{e}(\mathbf{x}) \leq 0$ find out the set of feasible solutions [Zitzler, 1999].

Definition 3.1.1. Feasible Set [Zitzler, 1999]; The set of decision vectors $\mathbf{x}$ which satisfies constraints $\mathbf{e}(\mathbf{x})$ is named as the feasible set $\mathbf{FS}(\mathbf{x})$:

$$\mathbf{FS}(\mathbf{x}) = \{\mathbf{x} \in X | \mathbf{e}(\mathbf{x}) \leq 0\} \quad (3.2)$$

In a single objective optimisation problem, the feasible set is totally ordered according to the objective function $\mathbf{f}(\mathbf{x})$: for two solutions $x_1, x_2 \in \mathbf{FS}(\mathbf{x})$ either $f(x_1) \geq f(x_2)$ or $f(x_1) \leq f(x_2)$. The aim is to find $\mathbf{f}(\mathbf{x})$ with the minimum value only for minimisation problems. Nevertheless, when a number of objectives is considered, the feasible set is partially ordered [Zitzler, 1999].

As an example, in Figure 3.1, $f_1(x)$ represents risk, $f_2(x)$ denotes cost. In this example, two objectives generally conflict with each other: low risk increases the cost while high risk decreases the cost. In Figure 3.1, on the left, there are several points as solutions: A, B, C. Solution Z is better than solution B with lower cost while both solutions have the same risk value. On the other hand, solution Y is better than solution A with lower risk and lower cost. In order to express this situation mathematically, the term *domination* is used [Zitzler, 1999].

Definition 3.1.2. Pareto Dominance [Zitzler, 1999]; For any two decision vectors x_1 and x_2 ;

$$\begin{aligned} x_1 &\succ x_2 \text{ if } f(x_1) > f(x_2) \\ x_1 &\succeq x_2 \text{ if } f(x_1) \geq f(x_2) \\ x_1 &\sim x_2 \text{ if } f(x_1) \not\geq f(x_2) \wedge f(x_2) \not\geq f(x_1) \end{aligned} \quad (3.3)$$

where $x_1 \succ x_2$, $x_1 \succeq x_2$ and $x_1 \sim x_2$ represent ‘ x_1 dominates x_2 ’, ‘ x_1 weakly dominates x_2 ’ and ‘ x_1 is indifferent to x_2 ’ in a sequence. As seen the Equation 3.3, $x_1 \succ x_2$ where x_1 is better than or equal to x_2 in all objectives and $x_1 \succeq x_2$ where is x_1 strictly better than x_2 in at least one objective. In Figure 3.1 on the right, the black dashed line rectangle area shows the region in objective space dominated by solution B while the red dashed line rectangle represents the area which contains the solution vectors dominating solution B. All solutions in the remaining region of the objective space are indifferent to solution B as expressed in the last line of Equation 3.3.

Definition 3.1.3. Pareto Optimality [Zitzler, 1999]; A decision vector $\mathbf{x} \in \mathbf{FS}(\mathbf{x})$ is pareto optimal if there does not exist another solution that dominates it. It is also called as a non-dominated solution. Let consider Solution Z in Figure 3.1 and its corresponding decision vector $\mathbf{x}$ is not dominated by any other decision vector. It is shown as:

$$\mathbf{x} \in \mathbf{PS}(\mathbf{x}) : \mathbf{x} \succ x_1 \quad (3.4)$$

where x_1 denotes another decision vector, $\mathbf{PS}(\mathbf{x})$ is a pareto-optimal set which is a set $\mathbf{PS}(\mathbf{x}) \subseteq \mathbf{FS}(\mathbf{x})$. All pareto-optimal solutions are referred to as the pareto-optimal set $\mathbf{PS}(\mathbf{x})$ where the corresponding objective vectors form the pareto-optimal front or surface. In Figure 3.1, white points demonstrate pareto-optimal solutions: there is no single optimal solution, but a set of optimal trade-off solutions.

Definition 3.1.4. Non-dominated Set [Deb, 2011]; Among a set of solutions $\mathbf{S}(\mathbf{x})$, the non-dominated set of solutions $\mathbf{NS}(\mathbf{x})$ are those that are not dominated by any solution in the set $\mathbf{S}(\mathbf{x})$.

Definition 3.1.5. *Pareto Optimal Front* [Deb, 2001]; The boundary determined by all solution in $PS(x)$ is called the pareto optimal front.

Definition 3.1.6. *Global Pareto Front* [Deb et al., 2000]; The boundary defined by global optimal solutions in a number of mutually exclusive equivalent non-dominated sets is called the global pareto front PF_{global} .

3.2 Performance Metrics

The performance of multi-objective algorithms is assessed using various metrics including number of the non-dominated solutions found, distance of the final pareto set to the global pareto-optimal front (accuracy), distribution of the final pareto set with respect to the pareto-optimal front, and spread of the pareto set (diversity) [Narukawa and Rodemann, 2012, Zitzler, 1999]. When dealing with multi-objective optimisation problems, the purpose is to achieve a desirable non-dominated set. However, for a number of reasons, the assessment of results becomes difficult; i) several solutions are generated rather than one like in a single objective optimisation problem, ii) a number of runs need to be performed to assess the performance of EAs due to their stochastic nature, iii) different entities, such as, coverage, diversity of a set of solutions, could be measured and used as a guidance during the search process [Sarker and Coello Coello, 2002].

In order to handle difficulties, there are several performance metrics proposed in the literature. In Helbig and Engelbrecht [2013], performance metrics are divided into three categories; i) accuracy performance measures: generational distance, inverted generational distance and error ratio, ii) diversity performance measures: number of solutions, spacing metrics, maximum spread, C-metric (coverage metric) and the U-measure, iii) combined performance measures: hypervolume, hypervolume ratio and ϵ -metric. In this study, generational distance, inverted generational distance and hypervolume are used and a brief explanation of each metric is given in the followings.

3.2.1 Generational Distance

Generational Distance (GD) is a performance evaluation method to estimate how far each individual solution in a solution set obtained are from PF_{global} set and GD is defined as [Veldhuizen, 1999];

$$GD = \frac{1}{n} \left(\sum_{i=1}^n d_i^p \right)^{\frac{1}{p}} \quad (3.5)$$

where n is the number of solutions, for $p = 2$, d_i is the Euclidean distance between each of solutions and the nearest member of PF_{global} . The value of $GD = 0$ shows that all individuals generated are in PF_{global} [Coello et al., 2006], hence the lower the GD the better the performance of an algorithm is.

3.2.2 Inverted Generational Distance

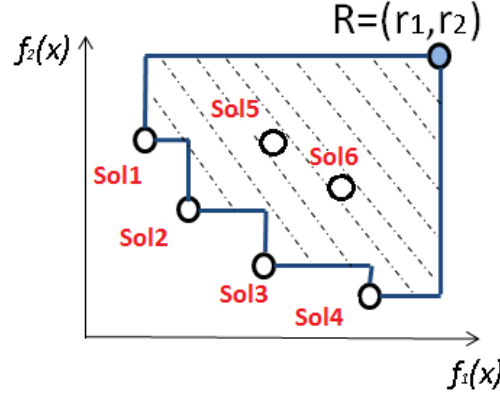
Inverted Generational Distance (IGD) metric was first proposed by Czyzak and Jaskiewicz [1998], calculating the distance between an objective vector and a reference point. However, the term itself, “inverted generational distance” was introduced in [Coello Coello and Reyes Sierra, 2004, Sierra and Coello, 2004].

$$IGD = \frac{1}{m} \left(\sum_{j=1}^m d_j^p \right)^{\frac{1}{p}} \quad (3.6)$$

where m is the number of vectors in PF_{global} , for $p = 2$, d_j is the Euclidean distance between each member of PF_{global} and the nearest solution obtained. The lower the IGD the better the performance of an algorithm is.

3.2.3 Hypervolume

Zitzler et al. [2007] proposed a hypervolume (HV) indicator (in the literature, it is found named as ‘Size of the Space Covered’ or ‘Size of Dominated Space’ by Zitzler and Thiele [1999]). The size of a pareto-front set is computed in the objective space by the non-dominated vectors and generally, the definition of hypervolume

FIGURE 3.2: Hypervolume of a solution set A

indicator is [Brockhoff et al., 2008];

$$I_H(A) = \lambda \left(\bigcup_{l \in A} [f_1(l), r_1] \times \dots \times [f_b(l), r_b] \right) \quad (3.7)$$

where $I_H(A)$ denotes the hypervolume indicator of a solution set $A \subseteq X$ and $I_H(A)$ is bounded by a reference point $r = (r_1, \dots, r_b) \in R_b$. It is assumed that b objective functions $f = (f_1(l), f_2(l), \dots, f_b(l))$ that map solutions $l \in X$ from the decision space X to maximize the hypervolume indicator $I_H(A)$. The Lebesgue measure of a hypervolume set is depicted as $\lambda(HV) = \lambda([f_1(l), r_1] \times [f_2(l), r_2] \times \dots \times [f_b(l), r_b])$ is the b -dimensional hypercuboid consisting of all points in the pareto front set of the solution set A . To provide a better understanding, Figure 3.2 shows a solution set A which consists of six solutions, Sol1, Sol2, Sol3, Sol4, Sol5, Sol6, for a two objective problem. Two reference points are used to bound each objective and they are shown as $R = (r_1, r_2)$. The two-dimensional hypercuboid is calculated consisting of four individuals, Sol1, Sol2, Sol3, Sol4 as solutions of the pareto front set. Unlike the other metrics used in this study, the higher the HV, the better the performance of an algorithm.

3.3 Scalarisation Techniques in Multi-objective Optimisation

In this section, I review some techniques that have been proposed in the literature to solve a scalar optimization problem. The main characteristics of these techniques are that they aggregate the multiple objectives into a single one. The well-known scalarisation techniques are weighted-sum method, the ϵ -constraint method, Benson's method, the augmented weighted Tchebycheff (or Chebychev) method, Tchebycheff method and the weighted max-ordering method [Ehrgott, 2006]. In this study, weighted-sum and Tchebycheff scalarisation methods are used. In the following subsections of this chapter, both methods are introduced.

3.3.1 The Weighted-sum Scalarisation

The weighted-sum approach suggested by Saaty and Gass [1954] is the most commonly used scalarisation technique for multi-objective optimisation problems.

$$\text{minimise } f(x) = \sum_{i=1}^b w_i f_i(x) \text{ where } x \in X \quad (3.8)$$

while w_i are non-zero fractional numbers known as weights and $\sum w_i = 1$.

The main advantage of the approach is that it is simple and easy to use. Also, it attempts to find a desirable pareto optimal solution for convex trade-off surfaces [Caramia and Dell'Olmo, 2008]. However, the weakness of the method could be sensitivity towards weights and it cannot obtain multiple pareto-optimum solutions simultaneously [Kim and de Weck, 2006]. Moreover, it is not capable of searching a solution in non-convex trade-off surfaces [Deb, 2001].

3.3.2 The Tchebycheff Scalarisation

This scalarisation method is proposed by Steuer and Choo [1983] that aims to minimize a function value assuming that the global ideal objective vector known

as the utopian vector is known. In this approach, a multi-objective optimisation problem is aggregated into a single objective problem and defined as:

$$\begin{aligned} &\text{minimise: } \max[w_i(f_i(x) - z_i^*)] \text{ for } i = 1, \dots, b \\ &\text{subject to: } x \in X \end{aligned} \tag{3.9}$$

where $w \in W = \{w \in R^b | 0 < w_i < 1, \sum_{i=1}^b w_i = 1\}$ and z^* is the utopian objective vector. Although, the Tchebycheff approach allows us to achieve every pareto-optimal solution because of the usage of an utopian objective vector, in order to achieve a desirable solution, objective functions need to be normalised. Information about ideal solution z^* is also a requirement for a Tchebycheff method to work [Miettinen, 1999]. In this study, we normalise each objective and determine z^* values to compute the model before hand.

The major advantage of the method could be that it is capable of searching a solution in non-convex trade-off surfaces [Voß et al., 2010]. However, one of the potential dangers of Tchebycheff approach is its dependence upon the utopian points [Hazewinkel, 1987]. In order to obtain reasonable solutions, objectives need to be normalised.

3.4 Metaheuristics

Many real-world problems are too complex and exact methods often fail in providing ‘high quality’ solutions in a ‘reasonable’ amount of time. Metaheuristic is a well-known approach in problem solving to find an approximate (near-optimal) solution when classic techniques fail to find any feasible solution due to the immense search space size. In 1986 Glover [1986] introduced the term of ‘metaheuristic’. It is recently defined by [Sörensen and Glover, 2013, p.960] as: “A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms.”

Metaheuristic approaches can be classified based on their characteristics such as the nature of the search process and use of memory. Sörensen and Glover [2013] divided metaheuristics into three classes; i) constructive metaheuristics,

such as GRASP (greedy randomized adaptive search procedure) [Feo and Resende, 1995], ii) local search metaheuristics known as single point search methods (trajectory methods), such as Simulated Annealing [Kirkpatrick et al., 1983], Tabu Search [Glover, 1986] and iii) population-based metaheuristics, such as Genetic Algorithms [Goldberg, 1989].

Constructive metaheuristics build solutions from processing partial solutions [Feo and Resende, 1995], while local search metaheuristics provide an iterative improvement through the neighbourhood search where a specific solution known as a candidate solution is explored within its neighbourhood with a set of moves iteratively. In contrast, population-based metaheuristics use multiple solutions improving them iteratively to provide exploration (diversification) in the search space simultaneously and exploitation (intensification) in a limited region to generate new solutions which have better performances [Blum and Roli, 2003]. In this work, a single point search method is used to solve the integrated supply chain management problem that has multi-objective nature. Four population based multi-objective algorithms are applied to the same problem with an attempt to detect the best performing approach.

3.4.1 Simulated Annealing

Simulated Annealing (SA) is a well-known local search algorithm (metaheuristic), inspired by the annealing process in the metallurgical industry, for solving computationally difficult optimisation problems [Metropolis et al., 1953]. A typical SA algorithm moves through the search space from a current solution to a new solution generated in the neighbourhood of this current solution, where a neighbour can be attained through some modification. In the algorithm, an initial solution is chosen and the ‘objective value’ of this solution is calculated. A move operator generates a new solution and the ‘objective value’ of this solution is evaluated comparing to the current solution’s ‘objective value’. If the new solution has an improvement on the ‘objective value’, it is accepted and becomes a current solution. If any improvement on the ‘objective value’ of the new solution is not achieved, it may be accepted with some probability. This acceptance probability allows the algorithm to converge to a desirable solution [Kirkpatrick et al., 1983]. In addition, the performance of SA depends on problem type and size [Arostegui et al., 2006,

Hussin and Stutzle, 2014]. For this reason, Simulated Annealing is preferred for the integrated supply chain management problem because of the characteristics of the problem. In the following, further information about Simulated Annealing is provided along with the pseudo-code of the approach and a related literature review.

Algorithm 1 presents pseudo-code of the approach [Turk et al., 2015]. This approach starts with a randomly generated initial solution (S_0). A new solution (S_{New}) is generated using a predefined neighbourhood/move operator (*CreateNeighboringSolution*) within the vicinity of the solution in hand and is compared to the original solution ($S_{Current}$). Based on the solution's 'objective value', measuring its quality (fitness) ($f(.)$) and a probability based on the current 'temperature' ($Temp$), a new solution is either accepted or rejected (line 8 in Algorithm 1). An improving solution is always accepted. SA avoids becoming trapped at a local optimum by accepting worsening solutions with a probability based on temperature, change in the quality of the solution using the Metropolis criterion [Metropolis et al., 1953] during the search process where a uniform random number is generated in between $[0,1]$ by *rand()*. The temperature is initially set to a high value ($Temp_0$) that results in a high probability of inferior solutions being accepted. Then the temperature is slowly decreased (line 16 in Algorithm 1) over the course of a run, reducing the probability of inferior solutions being selected. This iterative process of creating new solutions, evaluating them and accepting/rejecting them is repeated until a termination criteria, i.e., the maximum number of iterations or a final temperature (T_{min}) is reached. In order to maintain a balance between intensification and diversification during the search, we keep track of the best solution (S_{Best}) visited by the SA approach and return that solution at termination. In addition, *CoolingRate* is computed based on the current temperature, termination temperature named as *AbsoluteTemperature* and the number of maximum iterations, *MaxIts* decided by users. It is shown as in Equation 3.10:

$$CoolingRate = e^{\log((AbsoluteTemperature/Temp)/MaxIts)} \quad (3.10)$$

Since its introduction in 1983, SA has been successfully applied to many different real-world optimisation problems. For example, Miller et al. [2012] proposed a method that combines interval type-2 fuzzy logic and SA to investigate inventory

Algorithm 1 Pseudo-code of a generic SA algorithm [Turk et al., 2015]

```

1: Input:  $S_0, Temp_0, Temp_{min}, CoolingRate$ 
2: Output:  $S_{Best}$ 
3:  $\Delta \leftarrow 0; S_{Current}, S_{Best} \leftarrow S_0; Temp \leftarrow Temp_0$ 
4:  $qualityOfCurrent, qualityOfBest \leftarrow f(S_0)$ 
5: while ( $Temp > Temp_{min}$ ) do
6:    $S_{New} \leftarrow CreateNeighboringSolution(S_{Current})$ 
7:    $\Delta \leftarrow obj(S_{New}) - qualityOfCurrent$ 
8:   if ( $(\Delta < 0) \text{ OR } (e^{-\Delta/Temp} > rand(0, 1))$ ) then
9:      $S_{Current} \leftarrow S_{New}$ 
10:     $qualityOfCurrent \leftarrow qualityOfCurrent + \Delta$ 
11:   end if
12:   if ( $qualityOfCurrent < qualityOfBest$ ) then
13:      $S_{Best} \leftarrow S_{Current}$ 
14:      $qualityOfBest \leftarrow qualityOfCurrent$ 
15:   end if
16:    $Temp \leftarrow Temp \times CoolingRate$ 
17: end while

```

management problems in real-world systems. Kalender et al. [2013] studied SA-based hyper-heuristics that combine different methods for mixing multiple neighbourhood operators and Simulated Annealing as a move acceptance method. They tested their approach across a variety of problem domains ranging from timetabling to bin packing.

In addition, over the past three decades, the Simulated Annealing approaches have been extended for multi-objective optimisation. In 1994, Serafini [1994] presented a multi-objective Simulated Annealing technique in a two dimensional objective space to cope with both objectives for a travelling salesman problem. Czyżak and Jaskiewicz [1997] proposed a pareto Simulated Annealing approach combining two objectives with different weight settings to achieve an efficient solution set for a multi-objective knapsack problem. In Ulungu et al. [1998], a nuclear fuel manufacturing problem as an industrial problem is solved using a multi-objective Simulated Annealing approach and Ulungu et al. [1999] developed the proposed approach for a bi-objective knapsack problem problem using different aggregation principles.

Although some studies considered multiple objectives, most of problems were dealt with using an approach designed for optimising a single objective which was obtained by aggregate multiple objectives into one via some scalarisation method,

TABLE 3.1: Related work on Simulated Annealing algorithms range from single objective to multi-objective optimisation problems

References	Problems	Methods	Objective	Data
Miller et al. [2012]	Inventory management problem	Combining IT2FSs and SA	Single	Real
Kalender et al. [2013]	Educational timetabling problem, Boolean satisfiability, One dimensional bin packing, Permutation flow shop.	SA-based hyper heuristics	Single	Real
			Single	Synthetic
			Single	Synthetic
			Single	Synthetic
Serafini [1994]	Travelling salesman problem	Multi-objective Simulated Annealing (MOSA)	Two	Real
Czyżak and Jaskiewicz [1997]	Multi-objective knapsack problem	Pareto SA	Two	Real
Ulungu et al. [1998]	Nuclear fuel manufacturing problem	Interactive MOSA	Two	Real
Ulungu et al. [1999]	Bi-objective knapsack problem	MOSA	Two	Real
Suppaitnarm et al. [2000]	A ranking system problem, A deployment system for a spacecraft, A ten-bar cantilever truss.	MOSA using a multiplicative function for each objective	Two	Real
			Three Three	
Jaskiewicz [2001]	Multi-objective knapsack problem	Local-search based metaheuristics, MOSA	Two	Real
Suman [2003]	System reliability problem	Weight and pareto domination based multi-objective Simulated Annealing (WMOSA), (PDMOSA)	Two	Real
			Two	Real
Suman [2005]	A refinery problem, Two convex pareto front problems, A non-convex pareto front problems Cost optimisation problem, A multi-stage mixed system problem.	Pareto domination based multi-objective Simulated Annealing	Two	Real
			Two	Synthetic
			Two	Synthetic
			Two	Synthetic
			Two	Synthetic
Bandyopadhyay et al. [2008]	Six test problems, DTLZ1-6, <i>DTLZ</i> ₁₅ , <i>DTLZ</i> ₁₀ , <i>DTLZ</i> ₁₅ .	Archived multi-objective Simulated Annealing (AMOSA)	Three	Synthetic
			Five	
			Ten	
			Fifteen	

such as, weighted sum [Demirtas and Ustun, 2008, Serafini, 1994]. However, the weakness of such approaches arises mainly due to their performance sensitivity to varying weights and their inability to obtain multiple trade-off solutions simultaneously after a run [Kim and de Weck, 2006]. Also, Das and Dennis [1997] proved that fixed weights cannot reach some part of the front in an objective space for multi-criteria optimisation problems.

To avoid uncertainty about the procedure of determining objective weights, Suppaitnarm et al. [2000] investigated another multi-objective Simulated Annealing method which uses a multiplicative function for each objective instead of a composite objective function on three real-world problems. Jaskiewicz [2001] proposed an adapting weighted aggregation approach to avoid uncertainty about choosing weights of objectives during the annealing process on the multi-objective knapsack problem. Moreover, Suman [2003] generated two Simulated Annealing

based multi-objective algorithms: weight based multi-objective Simulated Annealing (WMOSA) and pareto domination based multi-objective Simulated Annealing (PDMOSA). The WMOSA algorithm uses a weight vector in the acceptance criterion to support exploration along feasible solutions. The PDMOSA algorithm is formulated using pareto-domination based fitness of a solution in the acceptance criterion. A real-world system reliability problem is solved using these two algorithms. Suman [2005] developed PDMOSA algorithms using a pareto-domination based acceptance criterion using a self-stopping process. These algorithm is tested on 7 different problems varying from real world to mixed integer programming problems.

Bandyopadhyay et al. [2008] proposed a new multi-objective SA named as archived multi-objective Simulated Annealing (AMOSA). An AMOSA stores the pareto front solutions in an archive to determine the acceptance of a new solution. The performance of proposed algorithm is compared to non-dominated sorting genetic algorithm-II (NSGA-II) on nine test problems of three, five, ten and fifteen objectives.

Based on the literature, there have been a few attempts in multi-objective Simulated Annealing methods as shown in Table 3.1. In most of earlier works, a single objective function is formed combining all objectives into one weighted objective and the later works try out to find alternative approaches using a pareto-domination based acceptance criterion. There is a slight conflict between MOSA having fixed weights and a pareto-domination based acceptance criterion. The confusion arises from the ambiguity in determining initial weights. However, a random weight setting could be another way to obtain a set of trade-off solutions through their weights randomly generated.

3.4.2 Evolutionary Algorithm

Evolutionary algorithm (EA) is a generic population based meta-heuristic optimization algorithm inspired by the natural process of evolution where the environmental conditions cause natural selection on a population. Genetic algorithms are the most well-known evolutionary algorithms. Algorithm 2 depicts a simple genetic algorithm where a population of solutions are updated iteratively using

mutation (modifying a solution) and crossover (recombination of solutions) operators [Goldberg, 1989].

Algorithm 2 Pseudo-code of a simple Genetic algorithm

```

1: Input:  $N, f(x)$ 
2: Initialise population  $P$  - size  $N$ 
3: Evaluate objective values  $f(x)$  for each individual in population  $P$ 
4: while (Termination criteria is met) do
5:   Select the best fit individuals for reproduction
6:   Apply Crossover and mutation operations to breed
7:   Replace least fit individuals on population  $P$  with new individuals
8: end while

```

3.4.3 Parameter Tuning

Evolutionary algorithms are formed by setting particular parameters which could have an essential impact on the performance of an evolutionary algorithm. To make an effective search and to reach solutions with better quality, an evolutionary algorithm needs to be configured selecting the best combination of these parameters. Identifying parameters for evolutionary algorithms is one of the challenging tasks in a successful optimization algorithm providing a balance between exploitation of available resources and exploration of a search space. Therefore, parameters have to be chosen and tuned properly since different algorithms require some or all of these parameters [Deb, 2007]. In the literature, two ways are used to distinguish the best combination of parameters; either *parameter tuning* where values of parameters are set in the initialisation process and do not change while the algorithm is running or *parameter control* where parameter values are changing automatically during the run of algorithm. Although over the last decade parameter control has been applied in a number of areas successfully [Eiben and Smit, 2011], a parameter tuning method is used to overcome parameter setting issue in this study.

Evolutionary algorithms are specified by qualitative parameters such as representation, recombination, mutation, parent selection and quantitative parameters such as population size, mutation rate, crossover rate, offspring size [Eiben and Smit, 2011]. In this work, only quantitative parameters are tuned. In the literature, there are various approaches to tune quantitative parameters such as Latin square [Myers and Hancock, 2001], Taguchi orthogonal arrays [Taguchi and Yokoyama, 1993],

REVAC [Nannen and Eiben, 2006], meta-EA [Grefenstette, 1986] and its variations, meta-GA (meta plan) [Mercer and Sampson, 1978], meta-ES (evolution strategy) [Grefenstette, 1986]. The idea behind these methods is to find the best settings using a small set of vectors for quantitative parameters. For this reason, in this study, Taguchi orthogonal arrays is used to tune parameters of each algorithm for population based evolutionary algorithms.

The Taguchi orthogonal arrays [Taguchi and Yokoyama, 1993] is one of the well-known sampling methods to tune parameters with respect to a factorial design. Fisher exhibited factorial designs as a design of experiments to examine impact of several factors (here the parameters) on the mean of a response (here the solution) [Roy, 2010]. In order to decrease the number of experiments required in the full factorial design, in the Taguchi method, fractional factorial experiments (FFEs) were developed [Roy, 2010]. The Taguchi method divides factors into two groups: noise factors N and controllable factors S . This method provides a way for N to decrease the variation around the goal considering orthogonal arrays. The robust design is also selected the design with less noise factors found [Taguchi et al., 2005]. Moreover, there are two common approaches to evaluate the results obtained in the Taguchi method: i) analysis of variance (ANOVA) where experiments repeat once, ii) signal to noise (S/N) for multiple experimental runs [Sadeghi et al., 2014, Taguchi et al., 2005].

Generally, parameter settings of MOEAs are determined according to either existing experience or tuning parameters. For example, in the work of Deb et al. [2002], identical parameter settings for SPEA and PAES were used while a reasonable set was chosen for NSGA-II with regard to the literature. In this study, the best combination of parameters are selected using the Taguchi method for each MOEA and ANOVA is performed to decide the significant parameters as factors and their contribution to the solution as a response.

3.5 Multi-objective Evolutionary Algorithms

The characteristics of evolutionary algorithms (EAs) are different from classical optimisation approaches. First of all, they use a direct search procedure which is able to be applied for a variety of optimisation problems to search directly

for “optimum” values. Secondly, EAs use a population to investigate a solution space. This provides a quick computational overall search and an ability to manage decision variables, objectives and constraints evolving the population based on the best minimum and maximum values. Finally, stochastic operators are used in EAs. It allows an evolutionary optimisation algorithm to handle multiple optima and other complexities considering a global perspective in the search space [Deb, 2011]. Therefore, over the past 30 years, there has been a growing interest in investigating multi-criteria optimisation problems using evolutionary algorithms.

In addition, there have been various Multi-objective Evolutionary Algorithms (MOEAs) presented in the literature and the most preferred MOEAs are: Vector Evaluated Genetic Algorithm (VEGA) [Schaffer, 1985], Multi-Objective Genetic Algorithm (MOGA) [Fonseca and Fleming, 1993], Niched-Pareto Genetic Algorithm (NPGA) [Horn et al., 1994], Non-dominated Sorting Genetic Algorithm (NSGA) [Srinivas and Deb, 1994], Pareto-Archived Evolution Strategy (PAES) [Knowles and Corne, 1999], Strength Pareto Evolutionary Algorithm (SPEA) [Zitzler, 1999], Indicator-based Evolutionary Algorithm (IBEA) [Zitzler and Künzli, 2004], Multi-Objective Messy Genetic Algorithm (MOMGA) [Van Veldhuizen and Lamont, 2000], the microGA [Goldberg, 1989], MOEA/D [Zhang and Li, 2007]. According to Fonseca and Fleming [1995], these algorithms can be classified into three groups considering the way that individuals are evaluated and selected; aggregating approaches, population-based non-pareto approaches and pareto-based approaches.

Aggregation approaches combine all objective functions into a single objective function aggregating them. And then a single comprised solution is produced by applying an evolutionary algorithm [Fonseca and Fleming, 1995]. Weighted sum, the ϵ -constraint and goal programming are well-known aggregation approaches in the literature [Nedjah and Mourelle, 2015]. We explained two aggregation approaches used in the thesis in Section 3.3.

Population-based non-pareto methods divide the population into several sub-populations, each of them evolves toward a single objective and each objective is dealt with one at a time. Vector Evaluated Genetic Algorithm is one example of these approaches [Fonseca and Fleming, 1995].

Pareto-based approaches monitor the population for non-dominated populations evolving them with respect to the dominance relationship between the individuals of the population [Fonseca and Fleming, 1995]. The most popular pareto-based approaches are MOGA, NPGA, NSGA, SPEA and microGA [Nedjah and Mourelle, 2015].

A variety of MOEAs with differing algorithmic components, such as diversity maintenance, replacement, have been previously proposed and more can be found in Konak et al. [2006], Zitzler et al. [2000], Zitzler and Thiele [1999]. Although this thesis considers four MOEAs: NSGA-II, SPEA2, IBEA and NSGA-III, in the following subsections, we will review seven of the most prominent algorithms [Coello Coello, 2002] to investigate advantages and disadvantages of each algorithm.

3.5.1 Vector Evaluated Genetic Algorithm

The Vector Evaluated Genetic Algorithm presented by Schaffer [1985] is a simple variation of genetic algorithms with a difference in the selection process. In VEGA, a number of sub-populations is generated by performing a proportional selection with respect to each objective separately.

Assume that m objectives have to be coped with. First the population size N is divided equally to create sub-populations of size N/m as $P_1, P_1 \dots P_m$ randomly. Then, each solution in the m sub-populations is assigned a fitness value with respect to its corresponding objective function. A new population in size N is formed using proportional selection of these sub-populations [Zitzler and Thiele, 1999]. In VEGA, the crossover and mutation operators are applied in the usual way within each sub-population [Coello Coello, 2002].

In general, the solutions generated by VEGA are locally pareto optimal and not necessarily globally non-dominated [Coello Coello, 2002]. The main advantage of VEGA is its simplicity. It is easy to apply and understand. In addition, it does not need to merge multiple objectives into one fitness function and there is no need to calculate the dominance level of each individual in the entire population. Thus, the time complexity of VEGA is less than other methods. However, one of the potential weaknesses of VEGA is selection bias. This problem arises from

this approach which selects individuals based on one objective without considering other objectives. Schaffer [1985] called this ‘middling’ performance that is acceptable but not outstanding for any of the objective functions. Moreover, another weakness of VEGA is that VEGA is not capable of producing desirable solutions in presence of non-convex search spaces [Zhang and Fujimura, 2010].

3.5.2 Multi-Objective Genetic Algorithm

In 1989, Goldberg [1989] proposed the pareto ranking method where the whole population is evaluated and all non-dominated individuals is assigned as rank ‘1’. Then, these individuals are excluded from the population. After that, all non-dominated values from the rest of the population are assigned as rank ‘2’. Until all solutions have been assigned, these steps are proceeded repeatedly. However, in 1993, Fonseca and Fleming [1993] used Golbers’s suggestion in a different way and called the new method, a Multi-Objective Genetic Algorithm.

In MOGA, the entire population is sorted according to their ranks. All non-dominated individuals are assigned from the best (rank 1) to the worst (rank $n \leq N$) using a linear or an exponential interpolation in the same manner as proposed by Goldberg. And then the average of individual fitnesses in the same rank is calculated to make them samples at the same rank [Ghosh and Dehuri, 2004]. MOGA is easy to use and can be applied in many kinds of multi-objective optimisation problems. The algorithm also converges towards the pareto front [Konak et al., 2006]. Nevertheless, the weakness of this approach is about maintaining diversity. In order to deal with a premature convergence, population diversity is preserved using the niche formation techniques [Fonseca and Fleming, 1993].

3.5.3 Niche-Pareto Genetic Algorithm

Horn et al. [1994] proposed the Niche-Pareto Genetic Algorithm which differs from the other pareto-based approaches in the selection operation. NPGA performs a tournament selection procedure based on the pareto dominance. In general, tournament selection involves the process of random selection of a set of individuals from the population. As an example, two individuals randomly selected are compared against a subset from the entire population (typically 10%

of the population size). If one of them is dominated and the other is not, then the non-dominated individual wins. When both individuals are either dominated or non-dominated, the result of the tournament is chosen with respect to fitness sharing [Coello et al., 2006].

The advantage of this method is that it is faster than MOGA and NSGA due to the fact that the pareto ranking is applied to some subsets of the population at each run rather than the entire population [Coello Coello, 2002]. Also, it achieves desirable non-dominated fronts [Coello Coello, 1996]. However, the convergence of the method depends on the tournament size [Horn et al., 1994]. In addition, the algorithm needs an extra parameter for tournament selection [Konak et al., 2006].

3.5.4 Non-dominated Sorting Genetic Algorithm

Srinivas and Deb [1994] developed Goldberg's suggestion along with a niche method and a selection approach and called a new approach as Non-dominated Sorting Genetic Algorithm (NSGA). NSGA is a variation of simple genetic algorithm ranking solutions with respect to their non-dominance within the population. A niche technique is used to preserve diversity of the population. The crossover and mutation operators remain as usual. The pseudo code for NSGA is shown in Algorithm 3 where population P with size N is evolved n generations to find values of objective functions $f_i(x)$ where $i = 1, \dots, b$ [Coello et al., 2006].

Before the selection is performed, the population is ranked to classify all non-dominated individuals into one category (between line 6 and line 12 in Algorithm 3). Each of them is assigned a large dummy fitness value to provide these individuals with having an equal reproduction potential. To maintain diversity of the population, these non-dominated individuals are shared using their dummy fitness values. Then they are ignored and another layer of non-dominated values are taken into consideration. The process continues until all individuals in the population is classified [Coello et al., 2006, Srinivas and Deb, 1994]. This algorithm eliminates the bias in VEGA and can be extended to higher dimensional and more complicated multi-objective problems. However, the weakness of NSGA is the sharing parameter which causes undesirable diversity in population. Moreover, there is lack of elitism which reduces the performance of the NSGA

Algorithm 3 Pseudo-code of NSGA

```

1: Input:  $N, n, f_i(x), k$ 
2: Initialise Population  $P$  in size  $N$ 
3: Evaluate values of objective functions  $f_i(x)$ 
4: for  $t = 1$  to  $n$  do
5:    $k = 1$ 
6:   while Population  $P$  classified do
7:     Rank Population  $P$  identifying non-dominated individuals
8:     Define them as rank ' $k$ '
9:     Assign a dummy fitness value to them
10:    Remove them from consideration temporarily
11:     $k = k + 1$ 
12:   end while
13:   Select an individual via Stochastic Universal Sampling according to shared dummy fitness values
14:   Apply Crossover (Single Point Based) and Mutation
15: end for

```

Algorithm 4 Pseudo-code of NSGA-II

```

1: Input:  $N, n, f_i(x)$ 
2: Initialize Population  $P_t$  in size  $N$ 
3: Evaluate values of objective functions  $f_i(x)$ 
4: Rank Population  $P_t$  identifying non-dominated individuals
5: Generate Off-spring Population  $Q_t$ 
6: Apply Binary Tournament Selection
7: Apply Recombination and Mutation
8: for  $t = 1$  to  $n$  do
9:   Combine  $P_t$  and  $Q_t$ 
10:  for  $j = 1$  to  $R_t = P_t \cup Q_t$  do
11:    Assign Rank (level) sorting based on pareto dominance and crowding distance
12:    Generate sets of non-dominated vectors along  $R_t$ 
13:    Determine crowding distance
14:  end for
15:  Select  $N$  individuals for next generation of population  $P_{t+1}$  adding solutions from the first front until  $N$  individuals based on non-dominated sorting and crowding distance selection
16:  Generate  $Q_{t+1}$ 
17:  Apply Binary Tournament Selection
18:  Apply Recombination and Mutation
19: end for

```

significantly [Deb et al., 2002]. Deb et al. [2000] proposed a new version of NSGA namely NSGA-II where these limitations are dealt with.

NSGA-II is an elitist MOEA based on a non-dominated sorting method. In order

to provide a better spread of solutions and convergence to pareto-optimal solutions, NSGA-II uses a crowding distance approach to sort individuals [Deb et al., 2002]. Initially, a population P_1 of size N is randomly generated where $t = 1$ and then those N individuals are sorted into different non-domination levels. Then, an offspring population Q_1 of size N is created applying crossover and mutation operators with associated probabilities (rates). P_1 and Q_1 are merged to form R_1 of size $2N$ which includes elite members of both parent and offspring populations. All individuals in R_1 are sorted into a number of non-domination levels such as F_1, F_2 and so on. Starting from F_1 , the next population P_2 is formed until the size of P_2 achieves N . The crowding distance approach is used to obtain N member in P_2 accepting the last level F_n partially. This process is repeated until a termination criterion is reached. It is a given number of generations n for objective functions $f_i(x)$ where $i = 1, \dots, b$ in Algorithm 4 [Deb and Jain, 2014, Deb et al., 2002, Sadeghi et al., 2014].

Furthermore, recently, there has been a growing interest in many-objective (four or more objectives) optimisation problems. Most MOEAs have faced difficulties in solving many-objective optimisation. Difficulties in handling many objectives can be listed as: i) a large fraction of population becomes non-dominated solutions within consideration of the number of objectives, ii) in a large dimensional space, diversity measurement becomes difficult and computationally expensive, iii) recombination operators may be insufficient to assess offspring solutions [Deb and Jain, 2014]. Deb and Jain [2014] developed a NSGA-II procedure with significant changes in the selection operator to overcome these difficulties and called it NSGA-III.

NSGA-III remains similar to NSGA-II apart from replacing the crowding distance in the selection operator with a systematic evaluation of individuals in the population with respect to the reference points [Yuan et al., 2014]. Algorithm 5 presents the pseudo-code of NSGA-III. Initially, a population P_t of size N is randomly generated and then those N individuals are sorted into different non-domination levels. Then, an offspring population Q_t of size N is created applying crossover and mutation operators with associated probabilities (rates). P_t and Q_t are merged to form R_t of size $2N$ which includes elite members of both parent and offspring populations. All individuals in R_1 are sorted into a number of non-domination levels

Algorithm 5 Pseudo-code of NSGA-III

```

1: Input:  $N, n, f_i(x), t, It_{max}$ 
2: Initialize Population  $P_t$  in size  $N$ 
3:  $t = 0$ 
4: while  $t < It_{max}$  do
5:   Generate off-spring population  $Q_t$  using crossover and mutation operators
6:   Combine  $P_t$  and  $Q_t$  to set  $R_t$ 
7:   Generate sets of non-dominated vectors  $F_1, F_2, \dots$  along  $R_t$ 
8:   Set  $S_t = \{\}, j = 1$ ;
9:   while  $|S_t| \leq N$  do
10:     $S_t = S_t \cup F_j$ 
11:     $j = j + 1$ 
12:   end while
13:    $l = j$ 
14:   if  $(|S_t| = N)$  then
15:      $P_{t+1} = S_t$ ; break
16:   else
17:      $P_{t+1} = \cup_{k=1}^{l-1} F_k$ 
18:     Individuals to be chosen from  $F_l$ ,  $K = N - |P_{t+1}|$ 
19:     Normalize objectives  $f_i(x)$  using min and intercept points of each objective
20:     Associate each individual of  $S_t$  to a reference point
21:     Select  $K$  individuals from  $F_l$  to construct  $P_{t+1}$  by the niche preserving operator
22:   end if
23:    $t = t + 1$ 
24: end while

```

such as F_1, F_2 and so on. The rest of the NSGA-III algorithm works quite different from NSGA-II. After sorting R_t let us obtain a new population S_t with size N examining individuals in S_t corresponding to a set of reference points either pre-defined or supplied. All objective values and reference points are first normalised to keep them in an identical range. Then, in order to associate each individual in the population S_t with a reference point, a reference line is determined joining the reference point with the origin of the normalized space. Next, the perpendicular distances between each individual in S_t and their corresponding lines are calculated and each individual is associated to the closest reference line. After that, in order to selectively choose which points will be in the next population P_{t+1} , a niche preserving operator explained in Deb and Jain [2014] is used.

3.5.5 Pareto-Archived Evolution Strategy

Pareto-Archived Evolution Strategy was proposed by Knowles and Corne [1999] using an evolution strategy that a single parent generates a single offspring. Based on this strategy, binary strings are used instead of real parameters and off-springs are generated using bitwise mutations [Deb et al., 2002].

In PAES, an offspring is evaluated comparing with its parent. If the offspring dominates the parent, the offspring is kept as the next parent and the iteration continues. Additionally, if the parent dominates the offspring, a new offspring is generated using a mutation operator. However, if both the offspring and the parent do not dominate each other, they are compared with all non-dominated solutions recorded in an archive. On the one hand, if the offspring is dominated by a number of solutions in the archive, the offspring is discarded and the parent mutates again to find a new offspring for further processing [Deb et al., 2002]. On the other hand, if the offspring dominates any member in the archive, it is recorded as the parent of the next generation and the dominated members are removed from the archive. In addition, if the offspring is not dominated by any member of the archive, in this case, the offspring is joined to the archive as a new non-dominated solution. To choose a new parent, the density of an individual in its neighbourhood is examined and a crowding strategy is used to select a new solution as the parent [Ghosh and Dehuri, 2004].

After a brief explanation of the PAES procedure, the main advantage is that the computational complexity is reduced by not using a niching procedure and another advantage is using a crowding strategy to maintain diversity effectively [Coello Coello, 2002]. However, it is not a population based method and its performance depends on the number of individuals evaluated [Konak et al., 2006].

3.5.6 Strength Pareto Evolutionary Algorithm

Strength Pareto Evolutionary Algorithm introduced by Zitzler [1999], combines different features of several MOEA techniques. In SPEA, non-dominated solutions found are recorded into an external archive called as the external non-dominated set. A strength value of each member of the external set is calculated (similar to the ranking value of MOGA) [Coello Coello, 2002]. Based on these strength

values, the fitness of each individual in the current population is calculated taking into account all external non-dominated solutions which dominate the individual with respect to proximity to the true pareto front and the spreading of solutions. Additionally, the population diversity is maintained using the pareto dominance which provides a proper distribution of solutions along with the pareto front. The size of the external non-dominated set grows continuously and is pruned to keep its size below a certain threshold. An average linkage method is used as a clustering approach in SPEA [Coello et al., 2006]. Algorithm 6 depicts the pseudo-code of SPEA for i objective functions $f_i(x)$ where $i = 1, \dots, b$. In Algorithm 6, population P_t in size N is evolved n generations considering values of objective functions. Hence, one disadvantage of SPEA is using a complex clustering approach and another one is the size of achieve which cannot be controlled automatically.

Algorithm 6 Pseudo-code of SPEA

```

1: Input:  $N, n, f_i(x)$ 
2: Initialize Population  $P_t$  in size  $N$ 
3: Generate an empty external pareto front set  $E_t$ 
4: Evaluate values of objective functions  $f_i(x)$  in Population  $P_t$ 
5: for  $t = 1$  to  $n$  do
6:   Copy non-dominated individuals in  $P_t$  to  $E_t$ 
7:   Eliminate those individuals covered by other individuals within  $E_t$ 
8:   Update non-dominated individuals in  $E_t$ 
9:   if ( $|E_t| > N$ ) then
10:    Prune  $E_t$  using clustering
11:   else
12:    Calculate objective functions of individuals in both population  $P_t$  and  $E_t$ 
13:     $MP \leftarrow T(P_t \cup E_t)$  (Use binary tournament selection with replacement to select
      individuals from  $P_t + E_t$  (multiset union) until the mating pool is full)
14:    Apply crossover and mutation on  $MP$  and add individuals to generate the new
      population  $P_t$ 
15:   end if
16:   Evaluate values of objective functions  $f_i(x)$  in population  $P_t$ 
17: end for

```

In addition, Zitzler et al. [2002] proposed a new version of SPEA named as SPEA2 which tries to eliminate the potential weaknesses in processes of SPEA. SPEA2 is an elitist evolutionary algorithm and works similarly [Coello et al., 2006]. The pseudo-code of SPEA2 is presented in Algorithm 7 where population (P_t) with size (N) is evolved n generations considering fitness value of each individual ($f_i(x)$) where $i = 1, \dots, m$. One of the main differences is that SPEA2 uses an external

archive that consists of the previously found non-dominated solutions (line 6 in Algorithm). In addition, SPEA2 uses an advanced fitness assignment strategy which considers both dominated and dominating individuals. Moreover, the nearest neighbour density measure is used in order to maintain the diversity [Zitzler et al., 2002].

Algorithm 7 Pseudo-code of SPEA2

```

1: Input:  $N, n, f_i(x)$ 
2: Initialize Population  $P_t$  in size  $N$ 
3: Generate an empty external pareto front set  $E_t$ 
4: Evaluate values of objective functions  $f_i(x)$  in Population  $P_t$ 
5: for  $t = 1$  to  $n$  do
6:   Copy non-dominated individuals in  $P_t$  to  $E_t$ 
7:   Eliminate those individuals covered by other individuals within  $E_t$ 
8:   Update non-dominated individuals in  $E_t$ 
9:   if ( $|E_t| > N$ ) then
10:    Use the truncation operator to discard members from  $E_t$ 
11:   else
12:    Calculate objective functions of individuals in both population  $P_t$  and  $E_t$ 
13:     $MP \leftarrow T(P_t \cup E_t)$  (Use binary tournament selection with replacement to select
      individuals from  $P_t + E_t$  (multiset union) until the mating pool is full)
14:    Apply crossover and mutation on  $MP$  and add individuals to generate the new
      population  $P_t$ 
15:   end if
16:   Evaluate values of objective functions  $f_i(x)$  in population  $P_t$ 
17: end for

```

3.5.7 Indicator-based Evolutionary Algorithm

Indicator-based Evolutionary Algorithm [Zitzler and Künzli, 2004] is an integration of different approaches proposed by Fonseca and Fleming [1998], Knowles [2002] to minimise the distance to the non-dominated set and maximise the diversity within a pareto optimal set. The main idea is to calculate the fitness of each individual using an indicator value predetermined by users. In addition, only pairs of individuals are compared instead of considering the entire pareto front sets and diversity preservation mechanism is not required [Zitzler and Künzli, 2004]. However, the disadvantage of IBEA is that it has a high time complexity for the calculation of exact indicator contributions of each solution.

Algorithm 8 presents pseudo-code of IBEA that uses hypervolume indicator I_{HV} . This approach starts with a randomly generated population P_t of size N where

Algorithm 8 Pseudo-code of IBEA

```

1: Input:  $N, n, I_{HV}$ 
2: Initialize Population  $P_t$  of size  $N$ 
3: for  $t = 1$  to  $n$  do
4:   Calculate  $I_{HV}$  values and assign them as fitness values to solutions in  $P_t$ 
5:   Chose an individual  $x \in P$  with the smallest fitness value and remove it from the
     population  $P_t$ 
6:   Perform binary tournament selection with replacement to fill the mating pool
7:   Apply crossover
8:   Apply mutation to the mating pool to generate the new population  $P_t$ 
9: end for

```

the maximum number of generations is defined as n . Fitness of each individual in the population P_t is calculated (line 4 in Algorithm 8). Individual with the worst value of fitness is removed from the population. And then, a binary tournament selection is performed to fill the mating pool. Mutations and crossover are applied to the mating pool and each individual obtained is added to population P_t . When the number of generations is greater than the maximum number of generations, the execution is finished and non-dominated individuals of population P_t are defined as final solutions [Matnei Filho and Vergilio, 2016].

3.5.8 Summary of Multi-objective Optimisation Algorithms

Multi-objective optimisation algorithms can be categorised into two groups: either solving the multi-objective problem as a single objective problem aggregating all objectives into one or using population based approaches which comprise non-pareto and pareto based methods. Although there are many variations of evolutionary multi-objective algorithms in the literature, the cited algorithms in this study are well-known and effective algorithms used in many applications. Based on several survey papers [Coello Coello, 1999, Konak et al., 2006, Xiujuan and Zhongke, 2004, Zitzler et al., 2000, Zitzler and Thiele, 1999], algorithms differ from each other in terms of their fitness assignment, elitism and diversification mechanisms.

Table 3.2 and Table 3.3 show how algorithms differ from each other based on their structural aspects. In Table 3.2, highlights of each algorithm are given and the check mark is used to indicate that the algorithm uses this component. It can be clearly seen that there are two methods, VEGA and NPGA used sub-populations

in order to reduce time complexity during the selection process while convergence insufficiency occurs. Although the rest of algorithms in Table 3.2 use a population to provide better solution sets, only PAES uses a (1+1) GA strategy that a single parent generates one offspring. In addition, there are three algorithms, PAES, SPEA and SPEA2, which record non-dominated solutions into an external archive. Also, Table 3.2 demonstrates which algorithm has a diversity mechanism to avoid premature convergence. In Table 3.2, the elitism mechanism is used in five algorithms; NSGA-II, NSGA-III, PAES, SPEA and SPEA2. Based on the literature, the elitism can improve the performance of algorithms significantly [Zitzler et al., 2000]. Furthermore, Table 3.3 depicts the fitness assignment and diversity procedures for these algorithms. For instance, VEGA uses a proportional selection process considering each objective function in turn and there is no specific mechanism to maintain population diversity. On the other hand, let us look at SPEA2 in Table 3.3. In order to provide more precise search, the fitness assignment strategy of SPEA is improved incorporating density of both individuals dominated and non-dominated and the nearest neighbour density measurement is used to provide good diversity performance sorting the distances between the individuals and its neighbours. To provide a better understanding of the importance of these components on the performance of algorithms, a number of works that investigate their effects comparing these algorithms is given in the following.

TABLE 3.2: Components of well-known multi-objective optimisation algorithms

Methods	Variations on Population				Diversity Mechanism	Elitism
	Sub	Entire	Offspring	External		
<i>VEGA</i>	✓	×	×	×	×	×
<i>MOGA</i>	×	✓	×	×	✓	×
<i>NPGA</i>	✓	×	×	×	✓	×
<i>NSGA</i>	×	✓	✓	×	✓	×
<i>NSGA – II</i>	×	✓	✓	×	✓	✓
<i>NSGA – III</i>	×	✓	✓	×	✓	✓
<i>PAES</i>	×	✓	✓	✓	✓	✓
<i>SPEA</i>	×	✓	✓	✓	✓	✓
<i>SPEA2</i>	×	✓	✓	✓	✓	✓
<i>IBEA</i>	×	✓	×	×	×	×

Zitzler et al. [2000] compared eight algorithms (RAND, FFGA, NPGA, HLGA, VEGA, NSGA, SOEA and SPEA) on six test functions. They involve fronts which have convex, non-convex, discrete, multi-modal, deceptive and non-uniform and they found that SPEA and NSGA achieved better solutions than others with

TABLE 3.3: Fitness assignment and diversity mechanism of well-known multi-objective algorithms

Methods	Fitness Assignment	Diversity Mechanism
<i>VEGA</i>	proportionate selection	×
<i>MOGA</i>	pareto ranking	fitness sharing by niching
<i>NPGA</i>	no fitness assignment, tournament selection	the niche count method
<i>NSGA</i>	ranking based on non- dominated sorting	fitness sharing by niching
<i>NSGA – II</i>	ranking based on non- dominated sorting	crowding distance approach
<i>NSGA – III</i>	perpendicular distance to the reference lines	reference points
<i>PAES</i>	pareto dominance	cell based density
<i>SPEA</i>	ranking based on non-dominated solutions in external archive	clustering to prune external population
<i>SPEA2</i>	strength of non-dominated solutions	nearest neighbourhood density measurement
<i>IBEA</i>	mating selection based on an indicator, such as hyper-volume	×

regard to the distance to the pareto front. SPEA, an elitist method, is also found to be superior over four evolutionary multi-objective algorithms (VEGA, NPGA, NSGA and HLGA) while solving an extended 0/1 knapsack problem. In addition, NSGA with elitism provided almost equivalent performance to the performance of SPEA [Zitzler and Thiele, 1999]. Thus, it is clearly seen that the influence of elitism cannot be underestimated to improve the performance of algorithms. In 2002, Deb et al. [2002] compared the performance of NSGA-II with SPEA and PAES as two elitist algorithms on nine test problems chosen from a number of significant past works. NSGA-II achieved better spread of solutions than SPEA and PAES. Moreover, in the work of Zitzler et al. [2002], the comparisons SPEA2 with other three elitist algorithms, SPEA, NSGA-II and PESA were carried out on a number of continues and combinatorial test problems. Based on the results found, the performance of SPEA2 is better than SPEA. Also, NSGA-II and SPEA2 behaved similarly and achieved the best performance overall.

However, pareto dominance-based algorithms are not capable of generating strong selections among the pareto fronts on many-objective problems since almost all

solutions in the population become non-dominated solutions [Deb and Jain, 2014]. To cope with many objective problems effectively, there are three types of methods: i) aggregation function-based EMO algorithms such as Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D), ii) indicator based algorithms such as IBEA and iii) reference set based algorithms such as NSGA-III [Ishibuchi et al., 2010]. In the work of Deb and Jain [2014], the proposed NSGA-III algorithm is applied to several test problems (considering from 3 to 15 objectives) and compared with MOEA/D and NSGA-II algorithms. The test problems include convex, concave, disjointed, scaled, multi-modal fronts and NSGA-III solved all problems well but while comparing with MOEA/D algorithms, there is no single version which produces efficient solutions for all test problems. However, it is clear that NSGA-III performs much better than NSGA-II on all test problems chosen.

In addition, IBEA was tested on a number of benchmark problems that involve the two-dimensional knapsack problem instance, continuous test functions and problem instances which cover network processor applications. To assess the performance of IBEA with the hypervolume indicator, the solutions of IBEA are compared with SPEA2 and NSGA-II and it is found that IBEA performs significantly better than other two algorithms in the most cases. Only for the KUR test problem, SPEA2 and NSGA-II provide better performance [Zitzler and Künzli, 2004]. Moreover, on a portfolio optimisation problem, IBEA outperforms NSGA-II and SPEA2 achieving a minimum distance from the pareto front [Bhagavatula et al., 2014]. Furthermore, performances of three algorithms are compared on the job-shop scheduling problems using two performance indexes, hypervolume and R2. Quality indexes prove that SPEA2 and IBEA are better than NSGA-II in most cases and there is no statistically difference between SPEA2 and IBEA. Also, it is found that the efficiency of NSGA-II decreases on any case with more than two objectives [Frutos et al., 2014].

In conclusion, it is clear that elitism has a significant influence on the performance of an algorithm in evolutionary multi-objective optimisation preserving the diversity of the population. Also, SPEA2 and NSGA-II are the most prominent as being both popular and effective for multi-objective optimisation [Coello Coello, 2002]. However, they are ineffective in many objective problems. NSGA-III can deal with difficulties in many objective problems and the work of Deb and Jain [2014] indicates that NSGA-III is applied in many areas successfully. Moreover,

IBEA is another MOEA designed without any necessity of diversity preservation methods using indicators such as the hypervolume and performs well in a number of problems. Based on the literature, each algorithm was confronted by different challenges based on problem features and there is no superior method among all MOEAs. Therefore, in this work, we will carry out four algorithms: NSGA-II as an elitist method within the population, SPEA2 as an elitist method using an external population, IBEA as an indicator based algorithm and NSGA-III as a many-objective optimisation algorithm.

3.6 Existing Tools for Multi-objective Optimisation

When constructing and comparing multi-objective optimisation algorithms, it is needed to have an available software package which contains state of the art algorithms, a number of test problems and techniques for performance assessment. However, it is certain that building any software package which satisfies these aspects is difficult and complicated. In the literature, few attempts have been made to address the issue. For example, the implementation of NSGA-II in the C programming language was publicly available. However, it is difficult to implement because of the lack of an object-oriented design [Durillo and Nebro, 2011a]. There is another software framework, namely PISA known as the official implementation of SPEA2 tool [Bleuler et al., 2003]. The disadvantage of PISA is using a shared-file mechanism to communicate via a text-based interface to exchange the selection technique. Thus, a long execution time of an application becomes unacceptable [Durillo et al., 2006].

Moreover, Walking Fish Group (WFG), a C++ based framework was designed for multi-objective optimisation [Huband et al., 2006]. The implementation includes nine test problems, WFG1-WFG9. Nevertheless, WFG does not consist of most of the multi-objective test suites such as ZDT and DTLZ. Hence, a new software framework in java programming language called jMetal was launched. The main idea behind jMetal is to allow users to use different MOEAs in the same software framework reusing the existing code. Also, a fair comparison between a new algorithm and any existing MOEA is provided. jMetal includes various test problems: Zitzler-Deb-Thiele (ZDT), Deb-Thiele-Laumanns-Zitzler (DTLZ), WFG test problems, CEC2009 as unconstrained problems, the Li-Zhang

benchmark, Kursawe, Fonseca and Flemming, Schaffer and constrained problems: Srinivas, Tanaka, Osyczka2, Constr_Ex, Golinski and Water (detailed in the work of Durillo and Nebro [2011a]). In addition, jMetal has an implementation of a number of well-known MOEAs listed as NSGA-II, NSGA-III, SPEA2, PAES, IBEA, PESa-II, OMOPSA, MOCcell, AbYSS, MOEA/D, Denssea, CellDE, GDE3, FastPGA, IBEA, SMPSO, MOCHC, SMS-EMOA. To assess the performance of different algorithms, several quality indicators are provided: Hypervolume, Spread, Generational Distance, Inverted Generational Distance, Epsilon. Moreover, different variable representations are carried out in the jMetal framework: binary, real, binary-coded real, integer and permutation. The tool includes utilities for statistical analyses. The source code, a guidebook and extra informations can be obtained from a web site (<http://jmetal.sourceforge.net/>) [Durillo and Nebro, 2011a]. In addition, there is a MOEA framework which combines PISA and jMetal and provides an user friendly interface (<http://moeaframework.org/>). The disadvantage of the MOEA framework is that it is not including many-objective algorithms. For all these reasons, jMetal is used to run experiments of MOEAs in this research study.

3.7 Multi-objective optimisation in SCM

Due to the complexity of the supply chain with multiple conflicting objectives requiring a search for a set of trade-off solutions, there has been a range of studies applying multi-objective methods. Mastrocinque et al. [2013] addressed the multi-objective supply chain problem which minimises the total cost and the total lead-time by using a swarm-based optimisation method. This proposed method reportedly performed better than an ant colony optimisation method across a test example of a bulldozer supply chain problem formed by different weight setting scenario. In order to optimise the flow of all materials in a supply chain, Zandhessami et al. [2012] presented a fuzzy multi-objective non-linear mixed integer programming (MIP) model considering total transportation costs and capacities of all echelons with three objectives, minimising cost, the quantity of vehicles, and maximising the capacity utilization of factories. A real world data from an automotive component manufacturer also was used. In this work, a genetic algorithm

was also applied to the problem which achieved a better performance than a fuzzy MIP model obtained.

Demirtas and Ustun [2008] described a method which integrates the analytic network process and multi-objective mixed integer linear programming considering both quantitative and qualitative factors in supplier evaluation. The method aimed to allocate optimum quantities for the materials to the selected suppliers with respect to three objectives, minimising the total cost, defective rate of materials from suppliers and maximising the total capacity of purchasing from suppliers. A real world problem tackled by a refrigerator producer was solved using the ϵ -constraint method which takes only one objective and restrict the rest two objectives within user-specified values and Tchebycheff scalarisation which performed the best. In Demirtas and Ustun [2009], the same problem was studied using a weighted sum scalarisation approach for a multi-period planning horizon. Archimedean Goal Programming (AGP) was applied to the problem. Different weights for each goal were used to investigate the behaviour of the approach under different scenarios. Ustun and Demirtas [2008b] addressed the same problem related to the performance of suppliers to determine the optimum quantities for the suppliers based on their performance. An achievement scalarisation function was proposed. Minmax Goal Programming was used to solve the multi-objective problem and the model was improved using AGP. This model provided better results than the model built using only one type of achievement function. Ustun and Demirtas [2008a] extended their previous work and presented an alternative approach based on Tchebycheff for the same problem. It has been shown that Tchebycheff procedure had a significant advantage over goal programming producing pareto-optimal solutions. Results showed that the Tchebycheff procedure obtained the most preferred non-dominated solutions.

Altıparmak et al. [2006] proposed a genetic algorithm approach for multi-objective optimisation of supply chain networks in order to provide a number of alternative solutions to decision makers. In their work, three objectives were considered: minimisation of the total cost, maximisation of customer services and maximisation of capacity. Two different weight settings were used to solve the problem using real world data from a manufacturer of plastic products. Based on pareto-optimum

solutions found, the proposed algorithm was compared to a multi-objective simulated annealing approach. This comparison indicated that GA methods outperformed simulated annealing approach. Che and Chiang [2010] also worked on a GA method for a multi-objective optimisation mathematical model of build-to-order supply chain network problems, considering three objectives: maximisation of quality, minimisation of costs and delivery time. Based on results, desirable trade-offs among the objectives are achieved with an improvement on the solution quality in terms of the average number of pareto-optimal solutions. Mahnam et al. [2009] developed a supply chain inventory model using fuzzy logic and integration of multi-objective particle swarm optimisation and simulation optimisation. The performance of the model was assessed in terms of two aspects: the total cost and fill rate (the percentage of demand satisfied). Based on results found using a synthetic supply chain network problem consists of eight vendors, six production plants and one distribution center, the proposed model increased the flexibility in decision making considering the total cost and fill rate simultaneously in a bi-objective approach.

Liao et al. [2011] proposed an approach to solve a multi-objective location-inventory problem (MOLIP) considering three objectives, minimising the total cost, maximising customer service in two aspects: volume fill rate and responsiveness level (the percentage of satisfaction of the customers within specified coverage distance). The MOLIP model was solved to examine the effects of location, distribution and inventory applying a Non-dominated Sorting Genetic Algorithm-II (NSGA-II). The proposed approach was compared to Strength Pareto Evolutionary Algorithm 2 (SPEA2) and outperformed SPEA2 in terms of the diversity quality of pareto-optimal solutions. Zhang et al. [2016] addressed a supply chain network design model to minimise the cost, to increase customer satisfactory and to reduce the strength of environmental influences using a modified multi-objective artificial bee colony (MOABC) algorithm. A number of problem instances were generated in different scales, ‘small’, ‘medium’ and ‘large’ cases to assess the performance of the proposed algorithm. In this work, first the problem was treated as a single objective optimisation problem and it was solved using the MOABC algorithm. And then, the proposed algorithm and MOGA were applied to the problem as a multi-objective problem with three objectives. The performance of

the MOABC algorithm was found similar to the performance of MOGA. Furthermore, in the work of Shankar et al. [2013], three nodes in SCM: suppliers, production plant and distribution centers were considered to maximise the fill rate and to minimise the total cost using a particle swarm optimisation algorithm. The bi-objective model is carried out to a live case study to show the use of algorithm in practice using data from a pump manufacturer. Based on trade-off solutions, the total cost could be decreased by 3.8% while 100% demand satisfaction was achieved.

The majority of the previous work treated SCM problems as a single objective problem and attempted to solve them using different scalarisation approaches. However, the weakness of such approaches arises mainly due to their performance sensitivity to varying weights and their inability to obtain multiple trade-off solutions simultaneously after a run [Kim and de Weck, 2006]. Therefore, in our study, three MOEAs, NSGA-II, SPEA2 and IBEA and a many-objective evolutionary algorithm namely NSGA-III are used. In addition, previous approaches using swarm based optimisation methods could also suffer from premature convergence [Esmin et al., 2015]. However, in this study, three algorithms, namely NSGA-II, NSGA-III and SPEA2 containing a diversity mechanism are preferred. The third MOEA used in our study, that is IBEA, does not necessitate a diversity preservation method as the indicators such as hypervolume reportedly perform well across a wide range of problems [Zitzler and Künzli, 2004]. Moreover, in none of the previous work covered above, the parameter settings of search methods were tuned. Nevertheless, evolutionary algorithms like any other stochastic local search method required setting of algorithmic parameters and selected setting could have an essential impact on their performance. To make an effective search and to reach solutions with better quality, an evolutionary algorithm needs to be configured selecting the best combination of these parameters. Identifying the best parameter settings for a search algorithm is one of the challenging tasks in designing an effective solution to the problem in hand [Deb, 2007]. However, in our study, parameters of evolutionary algorithms are tuned using the Taguchi approach [Taguchi et al., 2005].

In conclusion, only a few previous studies explored multi-objective supplier selection informed inventory planning. In this study, we investigate a two-stage

integrated solution approach to solve the integrated supply chain problem of supplier selection and inventory planning. The first component involves solving the supplier selection problem using an IT2FS. Following that an inventory model is developed to investigate how supplier risk affects cost. The parameters of four MOEAs are first calibrated using the Taguchi approach [Taguchi and Yokoyama, 1993]. After tuning parameters of these algorithms, each algorithm with the best setting is applied to the problem in order to examine the trade-off between operational cost of a supply chain and risk of suppliers. The performances of NSGA-II, NSGA-III, SPEA2 and IBEA are evaluated using well known metrics on twenty four different problem instances with different characteristics and sizes, where four of them are real world problem instances and 20 of them are randomly generated based on those instances.

3.8 Summary

This chapter presents a brief explanation of metaheuristics. Also, basic concepts of multi-objective optimisation along with a related literature review are given. Based on literature, it could be said that few studies have investigated supplier selection informed inventory planning as a multi-objective optimisation problem. Thus, selection of appropriate suppliers along with managing inventory system efficiently has become an area of open research in imprecise environment. In this work, we will investigate an integrated fuzzy embedded approach which aims to deal with both supplier selection and inventory planning using different multi-objective optimisation methods.

The next chapter provides the problem description of a two-stage fuzzy based optimisation approach to deal with the multi-objective integrated supply chain management problem.

Chapter 4

Integrated Supply Chain Management Problem

In this chapter, a two-stage fuzzy based optimisation approach is provided to deal with a supply chain management problem. The first stage of ranking suppliers is solved using an IT2FS method and in the second stage, a MOSA and different MOEAs with tuned parameters are studied to solve the supplier selection and inventory planning problem considering the information provided from the first stage.

4.1 Problem Description

4.1.1 Stage One: Ranking of Suppliers

The aim of this stage is to achieve an appropriate approach to rank vendors. Firstly, the criteria which have played a critical role in supplier selection are identified by decision makers. Then, the performance of each vendor is assessed based on these criteria [Turk et al., 2014].

TABLE 4.1: Linguistic weights of the attributes represented by type-1 fuzzy sets and its corresponding interval type-2 fuzzy sets (LT: Linguistic terms).

LT	Type-1 fuzzy sets	Interval type-2 fuzzy sets
L	(0.0,0.0,0.2,0.4)	$((0.0,0.0,0.2,0.3),(0.0,0.0,0.2,0.5))$
M	(0.2,0.4,0.4,0.6)	$((0.3,0.4,0.4,0.5),(0.1,0.4,0.4,0.7))$
H	(0.4,0.6,0.6,0.8)	$((0.5,0.6,0.6,0.7),(0.3,0.6,0.6,0.9))$
VH	(0.6,0.8,1.0,1.0)	$((0.7,0.8,1.0,1.0),(0.5,0.8,1.0,1.0))$

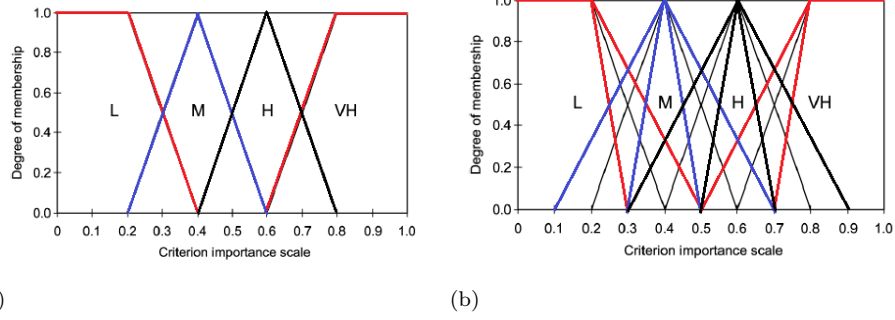


FIGURE 4.1: The Membership functions of the linguistic importance weight, (a) The type-1 MFs [Ordoobadi, 2009] and (b) The interval type-2 MFs.

4.1.1.1 Membership Functions

In the study of Ordoobadi [2009], decision makers have examined two attributes: importance of each criterion to evaluate vendors, and performance rating of suppliers. We extend this work, investigating uncertainty in the supplier selection problem using IT2FS in order to provide a guideline to choose an appropriate vendor.

The importance of each criterion is rated using linguistic weights: ‘low importance (L)’, ‘moderate importance (M)’, ‘high importance (H)’ and ‘very high importance (VH)’ [Ordoobadi, 2009]. The numeric scale defined between 0 and 1 corresponds to the fuzzy numbers shown in Table 4.1. 4.1 indicates the parameters of type-1 and interval type-2 membership functions used to depict each of the linguistic weights. Each type-1 membership function is taken from the original type-1 fuzzy sets of Ordoobadi [2009]. Interval type-2 membership functions are obtained by blurring the original type-1 sets from left to right in the same amount shifting for each set. Figure 4.1 demonstrates their corresponding trapezoidal-shaped membership functions.

TABLE 4.2: Linguistic performance rates represented by type-1 fuzzy sets and its corresponding interval type-2 fuzzy sets (LT: Linguistic terms).

LT	Type-1 fuzzy sets	Interval type-2 fuzzy sets
P	(0,0,2,4)	$((0,0,2,3),(0,0,2,5))$
G	(2,4,4,6)	$((3,4,4,5),(1,4,4,7))$
VG	(4,6,6,8)	$((5,6,6,7),(3,6,6,9))$
EX	(6,8,10,10)	$((7,8,10,10),(5,8,10,10))$

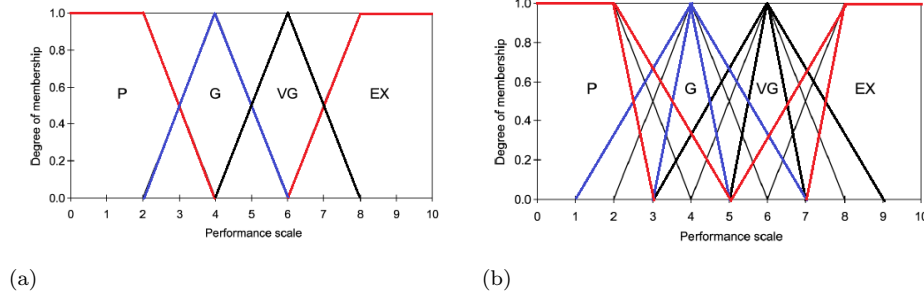


FIGURE 4.2: The membership functions of the linguistic performance rate, (a) The type-1 MFs [Ordoobadi, 2009] and (b) The interval type-2 MFs.

Decision makers assigned each supplier performance using linguistic weights: ‘excellent (EX)’, ‘very good (VG)’, ‘good (G)’ and ‘poor (P)’. The numeric scale determined between 0 and 10 corresponded to the fuzzy numbers of each criterion value [Ordoobadi, 2009, Turk et al., 2014]. Figure 4.2 demonstrates membership functions of both type-1 and interval type-2 in the same manner as denoted previously for modelling the importance weights, and their values are demonstrated in Table 4.2 [Turk et al., 2014].

4.1.1.2 Proposed Method for Ranking Suppliers

The decision makers reviewed a list of criteria and defined four main criteria and several sub-criteria as shown in Figure 4.3. The importance of each criterion is determined using linguistics terms and the linguistic importance weight of each node is converted into trapezoidal interval type-2 fuzzy sets as demonstrated in Table 4.1. Then, the weighted trapezoidal sets of end nodes of each branch are calculated by multiplying all nodes on the same branch illustrated in Figure 4.3 to obtain their membership values. For instance, for type-1 fuzzy sets, w_8 shown

in Figure 4.3 is achieved by multiplying the importance weight of the delivery by the importance weight of the delivery lead time as:

$$w_8 = (0.4, 0.6, 0.6, 0.8)(0.4, 0.6, 0.6, 0.8) = ((0.16, 0.36, 0.36, 0.64)) \quad (4.1)$$

And then, the linguistic terms of suppliers' performance determined by decision makers are converted into fuzzy performance ratings. After identifying the selection criteria and generating appropriate fuzzy membership functions, to measure the performance of suppliers and elicit their ranks, the aggregate fuzzy set for each vendor is calculated by multiplying the fuzzy performance rates matrix by the fuzzy importance weights. Next, fuzzy values are converted through centroid type-reduction and defuzzification methods to crisp values. Finally, based on crisp values that we obtained, the suppliers are ranked. The supplier with the highest crisp value is assigned as rank '1' and all suppliers are assigned from the best (the highest crisp value) to the worst (the lowest crisp value) as shown in Table 5.11.

4.1.2 Stage Two: Supplier Selection Aware Inventory Planning

In the first stage, the performance of each supplier is assessed and all suppliers are ranked based on their performance. In the second stage, an inventory model that incorporates information from the first stage is developed to capture the influence of supplier risk on the total cost of supply chain operation. The integrated SCM problem addressed in this study captures features of multi-product production while considering the different components. It consists of multiple suppliers, manufacturing plants and potential customers varying from experiment to experiment. The time for planning is broken down to 'chunks' of time. The first time period depends on what the original stock levels are [Turk et al., 2015]. Other assumptions are as follows:

1. Every supplier can supply all plants with all components.
2. Every supplier and plant has limited capacity for each component and product.
3. The cost of the whole operation is the cost of: product, order, transport, holding of inventory and stock out.

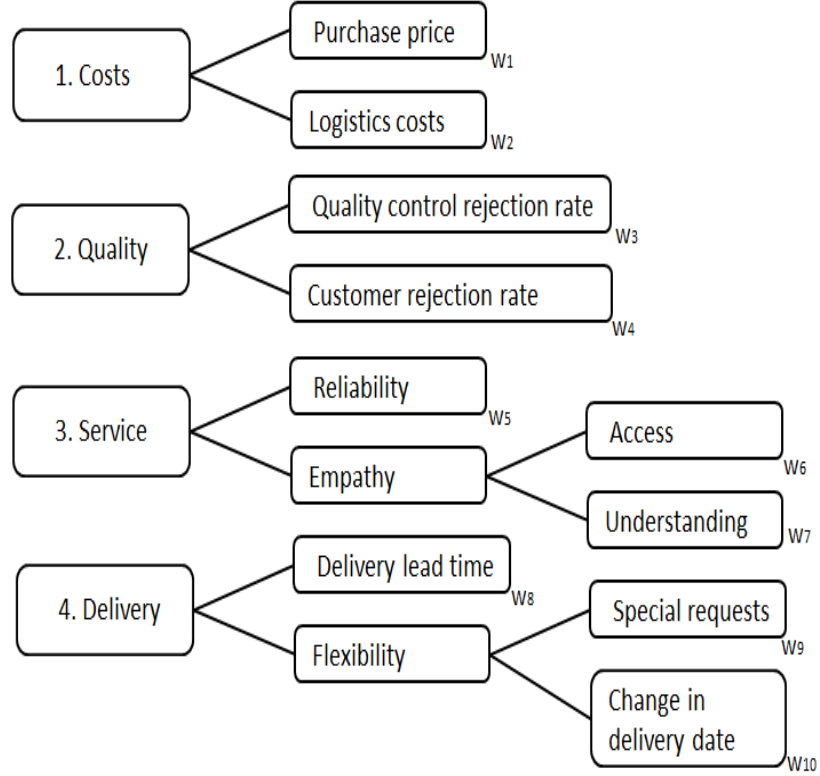


FIGURE 4.3: The criteria and sub-criteria used for selection of suppliers [Ordoobadi, 2009].

4. Distance between nodes are fixed and known.

Moreover, if an order is not in stock, stock-out cost is computed and when we cannot satisfy demand, the products are bought at the full price from competitors [Turk et al., 2015].

4.1.3 Problem Formulation

Presented below is the formulation of the supply chain problem where i , j and k represent a supplier, manufacturing plant and customer, respectively. In addition, a product, indicated by p , is formed using c components in discrete time period denoted by t . Tables 4.3 and 4.4 provide the notation to build the model.

TABLE 4.3: Notation for Decision Variables [Turk et al., 2015]

Variable	Meaning
$P_A(p, j, k, t)$	Amount of product p from plant j to customer k in period t
$C_A(c, i, j, t)$	Amount of component c from supplier i to plant j in period t

TABLE 4.4: Notation used for the formulation of the problem [Turk et al., 2015]

Notation	Meaning
$P_I(p, j, t)$	Inventory of product p at plant j in period t
$C_I(c, j, t)$	Inventory of component c at plant j in period t
$C_S(c, j)$	Initial stock for component c at plant j
$P_S(p, j)$	Initial stock for product p at plant j
$X_C(c, i)$	Component c 's capacity of supplier i
$X_P(p, j)$	Product p 's capacity at plant j
$Y_C(c, i)$	Component c 's cost of supplier i
$Y_P(p, k)$	Product p 's selling price for customer k
$T_C(c, i, j)$	Carrying cost for component c between supplier i and plant j
$T_P(p, j, k)$	Carrying cost for product p between plant j and customer k
$I_C(c, j)$	Component c 's inventory cost at plant j
$I_P(p, j)$	Product p 's inventory cost at plant j
$S_C(c, j)$	Shortage cost at plant j for component c
$S_P(p, j)$	Shortage cost at plant j for product p
$O_C(c, i)$	Ordering cost of supplier i for component c
$M_P(p, j)$	Manufacturing cost for product p at plant j
$S(p, j)$	Setup cost in plant j for product p
$H(p, j)$	Holding cost percentage for product p at plant j
$D_S(i, j)$	Distance between supplier i and plant j
$D_P(j, k)$	Distance between plant j and customer k
$Rank(i)$	Rank of vendor i
$Risk(i)$	Risk of vendor i
$D(p, k, t)$	Customer k demand for product p in each period t
$P_M(p, k, t)$	Non-fulfilment amount of product p for customer k in period t

In this research, the integrated SCM problem is considered as i) a single objective problem aggregating two objectives, ii) a two objective problem as a multi-objective problem and iii) a six objective problem solved by a many-objective optimisation method. The aim of this study is to minimise: i) potential risk endured TR (Equation 4.8) as a result of the supplier selection and ii) the total cost of the supply chain TC (Equation 4.7).

In Equation 4.2, the total cost of inventory is shown for the components and products successively. Equation 4.3 accumulates the transportation cost considering the products in the first row and components in the second row, respectively. In

Equation 4.4, the component order and setup costs as batch costs are added. The manufacturing and shortage costs for each product and each component are included in Equation 4.5 as a total production cost. The stock-out cost is depicted in Equation 4.6 as a penalty cost incurred when the quantity of production does not satisfy the customer demands. Equation 4.7 computes the total cost summing up the above entities for each time period/step.

$$\begin{aligned} \text{minimise } THC = \sum_t \left(\sum_p \sum_j I_P(p, j) \times P_I(p, j, t) \right. \\ \left. + \sum_c \sum_j I_C(c, j) \times C_I(c, j, t) \right) \end{aligned} \quad (4.2)$$

$$\begin{aligned} \text{minimise } TTC = \sum_t \left(\sum_p \sum_j \sum_k \left(P_A(p, j, k, t) \times D_P(j, k) \times T_P(p, j, k) \right) \right. \\ \left. + \sum_c \sum_i \sum_j \left(C_A(c, i, j, t) \times D_S(i, j) \times T_C(c, i, j) \right) \right) \end{aligned} \quad (4.3)$$

$$\begin{aligned} \text{minimise } TBC = \sum_t \left(\sum_c \sum_i \sum_j O_C(c, i) \times C_A(c, i, j, t) \right. \\ \left. + \sum_p \sum_j \sum_k S(p, j) \times P_A(p, j, k, t) \right) \end{aligned} \quad (4.4)$$

$$\begin{aligned} \text{minimise } TPC = \sum_t \left(\sum_p \sum_j \sum_k M_P(p, j) \times P_A(p, j, k, t) \right. \\ \left. + \sum_p \sum_j S_P(p, j) \times P_I(p, j, t) + \sum_c \sum_j S_C(c, j) \times C_I(c, j, t) \right) \end{aligned} \quad (4.5)$$

$$\text{minimise } TSC = \sum_t \left(\sum_p \sum_k P_M(p, k, t) Y_P(p, k) \right) \quad (4.6)$$

$$\text{minimise } TC = THC + TTC + TBC + TPC + TSC \quad (4.7)$$

$$\text{minimise } TR = \sum_t \sum_c \sum_i \sum_j C_A(c, i, j, t) \times Risk(i) \quad (4.8)$$

Equation 4.8 demonstrates the total risk of suppliers with respect to Equation 4.9 which shows the calculation of a coefficient for the risk of each supplier by normalising the supplier rank indicated in Table 5.3.

$$Risk(i) = \frac{\sum_i Rank(i)}{Rank(i)} \quad (4.9)$$

Equation 4.10 depicts the supplier capacity for each period and Equation 4.11 shows the capacity of the plant for each period.

$$\sum_j C_A(c, i, j, t) \leq X_C(c, i) \quad \text{for } \forall c, i, t \quad (4.10)$$

$$\sum_k P_A(p, j, k, t) \leq X_P(p, j) \quad \text{for } \forall p, j, t \quad (4.11)$$

Equation 4.12 shows that demand is satisfied by the production units, and if the production units are not less than the order amount of customer, it is provided from retailers explained as non-fulfilment amount of a product in Table 4.4. Equation 4.13 guarantees that the production units are not more than the order amount of customer. In this model, it is assumed that the first product composes of the first and second components and the second one is produced using the third and

fourth components. Equation 4.14 describes the inventory-control constraints for these components and Equation 4.15 represents inventory-control constraints for each product.

$$P_M(p, k, t) = D(p, k, t) - \sum_j P_A(p, j, k, t) \quad \text{for } \forall p, k, t. \quad (4.12)$$

$$\sum_j P_A(p, j, k, t) \leq D(p, k, t) \quad \text{for } \forall p, k, t. \quad (4.13)$$

$$\sum_i C_A(c, i, j, t) + C_S(c, j) = \sum_k P_A(p, j, k, t) + P_I(p, j, t) + C_I(c, j, t) \quad (4.14)$$

for $\forall j, t$ where $p = 1$ for $c = \{1, 2\}$ and where $p = 2$ for $c = \{3, 4\}$.

$$\begin{aligned} \sum_j \sum_k P_A(p, j, k, t) + \sum_j P_S(p, j) &= \sum_k D(p, k, t) + \sum_j P_I(p, j, t) \\ &+ \sum_k P_M(p, k, t) \end{aligned} \quad \begin{matrix} (4.15) \\ \text{for } \forall p, t. \end{matrix}$$

4.1.4 Problem Instances

In this study, four groups of six problem instances are used, totalling up to twenty four instances (Table 4.5). For each instance, five suppliers denoted from S1 to S5 are considered for simplicity. Each group is formed of a fixed number of suppliers, manufacturing plants and customers, namely $2 \times 2 \times 2$, $3 \times 2 \times 3$, $3 \times 3 \times 3$ and $5 \times 5 \times 5$ as illustrated in Figure 4.4. The flow of goods is also shown in Figure 4.4 using arrows. The planning horizon contains three discrete time periods. The production cost, capacity, minimum order quantity, order quantity and initial stock are all parametrised for the instances. Two different settings are used for the problem instances as summarised in Table 4.6 and Table 4.7.

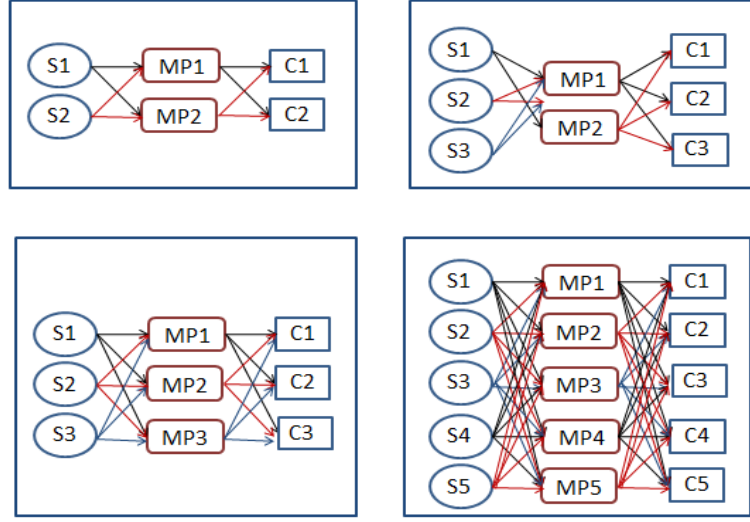


FIGURE 4.4: Representation of systems consisting of a fixed number of suppliers, manufacturing plants and customers, namely $2 \times 2 \times 2$, $3 \times 2 \times 3$, $3 \times 3 \times 3$ and $5 \times 5 \times 5$, from which 24 instances are derived.

TABLE 4.5: Characteristics of the problem instances (S: the number of suppliers, P: the number of manufacturing plants, C: the number of customers, CO: configuration)

Inst	S	P	C	CO	Inst	S	P	C	CO	Inst	S	P	C	CO	Inst	S	P	C	CO
Inst1	2	2	2	C1	Inst7	3	2	3	C1	Inst13	3	3	3	C1	Inst19	5	5	5	C1
Inst2-7	2	2	2	C2	Inst8-13	3	2	3	C2	Inst14-19	3	3	3	C2	Inst20-24	5	5	5	C2

TABLE 4.6: Configuration of problem instance parameters-I

Node Configuration	Label	Production Cost	Capacity		Minimum Order Quantities		Order Quantities		Initial Stock	
		Products	Components	Products	Components	Products	Components	Products	Components	Products
Fixed	C1	(0.5,0.2)	1000	1000	100	100	100	100	250	250
Random	C2	[0.20,0.80]	[500,1000]	[500,1000]	[50,200]	[50,200]	[10,50]	[10,50]	[0.500]	[0.500]

TABLE 4.7: Configuration of problem instance parameters-II

Node Configuration	Label	Suppliers	Cost of Components
Fixed	C1	S1	(0.9,0.15,0.3,0.5)
		S2	(1.2,0.25,0.45,0.7)
		S3	(1.0,0.2,0.35,0.6)
		S4	(1.3,0.17,0.38,0.55)
		S5	(0.95,0.25,0.4,0.75)
Random	C2	All	[0.20,0.80]

The first instances of each group, namely Inst1, Inst7, Inst13 and Inst19 are real-world problem instances [Miller et al., 2012] and use a fixed parameter setting. The remaining instances are randomly generated using a certain range for each parameter setting as shown in Table 4.5.

4.2 Summary

This chapter provides a description of an integrated two-stage fuzzy embedded approach to deal with a SCM problem that composes of both supplier selection and inventory planning of a supply chain problem.

The next chapter presents a single objective algorithm where the two-objective SCM problem is converted into a scalar problem and solved using Multi-objective Simulated Annealing. Firstly, the influence of the membership function shape on the results has been investigated on the problem instance. Secondly, suppliers are ranked using different type-1 and interval type-2 sets. Finally, the results obtained from the IT2FS approach are fed into the next stage and the integrated SCM problem is solved using Multi-objective Simulated Annealing.

Chapter 5

Single Objective Algorithm for the Integrated Problem

This chapter provides implementations of the two-objective problem converted into a scalar problem using two scalarisation approaches. First, in order to rank suppliers, the supplier selection problem is dealt with using IT2FS. In addition, the influence of the membership function shape on the results is investigated on the problem instance. Second, an inventory planning approach that incorporates information from the first stage is developed to capture the influence of supplier risk on the total cost of supply chain operation. We apply a simulated annealing approach to the problem using two scalarisation approaches, weighted sum and Tchbycheff. Additionally, this chapter provides a comparison between the performance of weighted sum and Tchbycheff approaches under different weight settings. Moreover, parameters of the SA algorithm are tuned and comparison between performances of default setting SA and tuned parameter SA algorithms is provided.

5.1 Stage One: Ranking of Suppliers

The proposed approach by Ordoobadi [2009], was applied in a case study comprised three suppliers and 16 selected criteria and sub-criteria as illustrated in Figure 4.3 (as detailed in Section 4.1.1). The importance of criteria is determined by decision makers as shown in Table 5.1. The performance of each supplier is also decided in

TABLE 5.1: The importance of criteria, L: Low importance, M: Moderate importance, H: High importance, VH: Very high importance

Criterion	Its importance	Criterion	Its importance
Costs	H	Empathy	M
Purchase Costs	VH	Access	L
Logistic Costs	H	Understanding	M
Quality	VH	Delivery	H
Quality control rejection rate	H	Delivery lead time	H
Customer rejection rate	M	Flexibility	H
Service	M	Change in delivery time	M
Reliability	M	Special requests	H

TABLE 5.2: The performance of each supplier in terms of selected criteria for three suppliers, EX: Excellent, VG: Very good, G: Good, P: Poor

Supplier	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10
S1	P	VG	VG	G	P	VG	G	P	EX	G
S2	G	G	VG	EX	G	EX	P	G	P	P
S3	G	P	G	VG	EX	P	G	P	VG	VG

TABLE 5.3: Crisp scores and supplier rankings Ordoobadi [2009]

Suppliers	Crisp Scores	Rank
S1	13.50	3
S2	15.78	1
S3	14.44	2

terms of selected criteria as shown in Table 5.2. To evaluate suppliers, the approach explained in Section 4.1.1 is followed and crisp scores are obtained as illustrated in Table 5.3. In this section, firstly, shapes of membership functions of Ordoobadi's work are investigated and the influence of the membership functions' shapes on the result is evaluated by using three different modified fuzzy sets. Secondly, their corresponding interval type-2 fuzzy sets are proposed to measure the performance of suppliers. Finally, the work is carried out using real world data in order to evaluate suppliers under an interval type-2 fuzzy environment.

5.1.1 Effect of the fuzzy membership functions shape on the result

Garibaldi and John [2003] pointed out that choosing membership functions of linguistic terms has played an essential role in fuzzy systems to capture the opportunities to best represent human knowledge. For this reason, this work is carried out on investigation of shapes of membership functions in order to see whether there is any change on the ranking of suppliers. This task is done in a two-step process:

1. In order to demonstrate the effect of uncertainty on fuzzy sets in linguistic terms, ranking is done for three different membership functions. Each value of membership functions is determined based on original type-1 fuzzy sets of Ordoobadi [2009]. In the first approach (Approach 1), membership functions are created by shifting the centre point of the original type-1 membership functions from 0.4 to 0.5 for the criteria importance and from 4 to 5 for performance of suppliers and values of each set are assigned as in Table 5.4 and Table 5.5. In the second and third approaches (Approach 2 and Approach 3), membership functions are created by varying the parameters of membership functions determined in the first approach and changes in the widths of membership functions are increased gradually.
2. For these three approaches, Approach 1, Approach 2 and Approach 3, the suppliers are ranked by using the same way explained in Ordoobadi [2009].

Results of experiments for analysing the effect of uncertainty in membership functions on variation in ranking of suppliers are illustrated in Table 5.6. In comparison to crisp scores of suppliers using original type-1 sets shown in Table 5.3 and results of these three approaches, there are significant raises in fuzzy scores of each supplier because of shifting the centre points from 0.4 to 0.5.

Nevertheless, in the first and second approaches, the rank of suppliers is found the same as the result in Ordoobadi [2009]. In the third approach, S3 became the best among three suppliers. In Table 5.6, it is also clearly seen that the difference between the crisp score of S2 and S3 becomes smaller when the widths of fuzzy sets are increased regardless of the fact that there is no important change on S1. Based on results, it is observed that shapes of membership functions have a significant

TABLE 5.4: Linguistic weights of the attributes represented by type-1 fuzzy sets

Linguistic terms	Ordoobadi's work	Approach 1
Low importance	(0.0,0.0,0.2,0.4)	(0.0,0.0,0.25,0.5)
Moderate Importance	(0.2,0.4,0.4,0.6)	(0.25,0.5,0.5,0.75)
High Importance	(0.4,0.6,0.6,0.8)	(0.5,0.75,0.75,1.0)
Very High Importance	(0.6,0.8,1.0,1.0)	(0.75,1.0,1.0,1.0)
Linguistic terms	Approach 2	Approach 3
Low importance	(0.0,0.0,0.2,0.5)	(0.0,0.0,0.15,0.5)
Moderate Importance	(0.2,0.5,0.5,0.8)	(0.15,0.5,0.5,0.85)
High Importance	(0.5,0.8,0.8,1.0)	(0.5,0.85,0.85,1.0)
Very High Importance	(0.8,0.1,1.0,1.0)	(0.85,1.0,1.0,1.0)

TABLE 5.5: Linguistic performance rates represented by type-1 fuzzy sets

Linguistic terms	Ordoobadi's work	Approach 1
Poor	(0,0,2,4)	(0,0,2.5,5)
Good	(2,4,4,6)	(2.5,5,5,7.5)
Very Good	(4,6,6,8)	(5,7.5,7.5,10)
Excellent	(6.8,10,10)	(7.5,10,10,10)
Linguistic terms	Approach 2	Approach 3
Poor	(0,0,2,5)	(0,0,1.5,5)
Good	(2,5,5,8)	(1.5,5,5,8.5)
Very Good	(5,8,8,10)	(5,8.5,8.5,10)
Excellent	(8,10,10,10)	(8.5,10,10,10)

TABLE 5.6: Ranking of suppliers results for three approaches

Suppliers	Approach 1	Rank	Approach 2	Rank	Approach 3	Rank
S1	25.41	3	27.20	3	29.11	3
S2	27.61	1	29.37	1	31.18	2
S3	27.27	2	29.21	2	31.36	1

role in the rank of suppliers. Thus, these three approaches are carried out on the next stage in order to develop them using interval type-2 fuzzy sets.

TABLE 5.7: Their corresponding interval type-2 fuzzy sets for linguistic weights of attributes, LT: Linguistic terms, L: Low importance, M: Moderate importance, H: High importance, VH: Very high importance

LT	Ordoobadi's work	Approach 1
L	$((0.0,0.0,0.2,0.3),(0.0,0.0,0.2,0.5))$	$((0.0,0.0,0.25,0.4),(0.0,0.0,0.25,0.6))$
M	$((0.3,0.4,0.4,0.5),(0.1,0.4,0.4,0.7))$	$((0.35,0.5,0.5,0.65),(0.15,0.5,0.5,0.85))$
H	$((0.5,0.6,0.6,0.7),(0.3,0.6,0.6,0.9))$	$((0.6,0.75,0.75,0.9),(0.4,0.75,0.75,1.0))$
VH	$((0.7,0.8,1.0,1.0),(0.5,0.8,1.0,1.0))$	$((0.85,1.0,1.0,1.0),(0.65,1.0,1.0,1.0))$
LT	Approach 2	Approach 3
L	$((0.0,0.0,0.2,0.4),(0.0,0.0,0.2,0.6))$	$((0.0,0.0,0.15,0.4),(0.0,0.0,0.15,0.6))$
M	$((0.3,0.5,0.5,0.7),(0.1,0.5,0.5,0.9))$	$((0.25,0.5,0.5,0.75),(0.05,0.5,0.5,0.95))$
H	$((0.6,0.8,0.8,1.0),(0.4,0.8,0.8,1.0))$	$((0.6,0.85,0.85,1.0),(0.4,0.85,0.85,1.0))$
VH	$((0.9,1.0,1.0,1.0),(0.7,1.0,1.0,1.0))$	$((0.95,1.0,1.0,1.0),(0.75,1.0,1.0,1.0))$

5.1.2 Developing approaches with IT2FSs

The type-1 model is extended by using interval type-2 sets. Their corresponding interval type-2 fuzzy sets are indicated in Table 5.7 and Table 5.8. The process for each approach is performed by going through steps explained in Section 4.1.1. Firstly, trapezoidal IT2FSs are generated using information about the importance of the criteria selected by experts. These criteria are expressed in linguistic terms with respect to experts' perceptions. For example, if a criterion's importance weight is 'high' then it is assigned as $((0.5, 0.6, 0.6, 0.7), (0.3, 0.6, 0.6, 0.9))$ as seen in Table 4.1. After all criteria are converted into fuzzy numbers, all nodes on the same branch are multiplied by the previous node as indicated in Figure 4.3. Let w_i indicates the fuzzy importance weight of criterion i where $i = 1, 2, \dots, 10$. For instance, w_8 is achieved by multiplying the importance weight of the delivery by the importance weight of the delivery lead time as:

$$\begin{aligned}
 w_8 &= ((0.5, 0.6, 0.6, 0.7), (0.3, 0.6, 0.6, 0.9))((0.5, 0.6, 0.6, 0.7), (0.3, 0.6, 0.6, 0.9)) \\
 &= ((0.25, 0.36, 0.36, 0.49), (0.09, 0.36, 0.36, 0.81)) \quad (5.1)
 \end{aligned}$$

After all weights are computed in the same manner, values of membership functions for each criterion are achieved as shown in Table 5.9. Then, trapezoidal

TABLE 5.8: Their corresponding interval type-2 fuzzy sets for linguistic performance rates, LT: Linguistic terms, EX: Excellent, VG: Very good, G: Good, P: Poor

LT	Ordoobadi's work	Approach 1
P	$((0,0,2,3),(0,0,2,5))$	$((0,0,2.5,4),(0,0,2.5,6))$
G	$((3,4,4,5),(1,4,4,7))$	$((3.5,5,5,6.5),(1.5,5,5,8.5))$
VG	$((5,6,6,7),(3,6,6,9))$	$((6,7.5,7.5,9),(4,7.5,7.5,10))$
E	$((7,8,10,10),(5,8,10,10))$	$((8.5,10,10,10),(6.5,10,10,10))$
LT	Approach 2	Approach 3
P	$((0,0,2,4),(0,0,2,6))$	$((0,0,1.5,4),(0,0,1.5,6))$
G	$((3,5,5,7),(1,5,5,9))$	$((2.5,5,5,7.5),(0.5,5,5,9.5))$
VG	$((6,8,8,10),(4,8,8,10))$	$((6,8.5,8.5,10),(4,8.5,8.5,10))$
E	$((9,10,10,10),(7,10,10,10))$	$((9.5,10,10,10),(7.5,10,10,10))$

TABLE 5.9: The weighted trapezoidal interval type-2 sets for criteria

Criterion	Weight	Interval type-2 fuzzy sets
w1		$((0.35,0.48,0.60,0.70),(0.15,0.48,0.60,0.90))$
w2		$((0.25,0.36,0.36,0.49),(0.09,0.36,0.36,0.81))$
w3		$((0.35,0.48,0.60,0.70),(0.15,0.48,0.60,0.90))$
w4		$((0.21,0.32,0.40,0.50),(0.05,0.32,0.40,0.70))$
w5		$((0.09,0.16,0.16,0.25),(0.01,0.16,0.16,0.49))$
w6		$((0.00,0.00,0.03,0.08),(0.00,0.00,0.03,0.24))$
w7		$((0.03,0.06,0.06,0.13),(0.00,0.06,0.06,0.34))$
w8		$((0.25,0.36,0.36,0.49),(0.09,0.36,0.36,0.81))$
w9		$((0.08,0.14,0.14,0.24),(0.01,0.14,0.14,0.57))$
w10		$((0.13,0.22,0.22,0.34),(0.03,0.22,0.22,0.73))$

IT2FSs for the performance of vendors are generated in the same way as criteria importance as depicted in Table 5.10. And then the aggregate fuzzy set for each vendor is computed by multiplying the fuzzy performance rates matrix by the fuzzy importance weights. Finally, fuzzy values are converted through Centroid type-reduction and defuzzification methods to crisp values in order to rank suppliers. Table 5.11 and Table 5.12 show the rank of suppliers obtained as results.

In Table 5.11, the crisp score of Ordoobadi's work using type-1 fuzzy sets and the output of our study using interval type-2 fuzzy sets without shifting the centre

points are illustrated. In comparison to the effect of using type-1 and interval type-2 fuzzy sets, the rank of suppliers are found the same, despite that there is an increase in the score of each supplier. In comparison to interval type-2 membership functions created with shifting the center points and changing the widths of these membership functions, the difference in scores of S2 and S3 becomes small and in the third approach, the crisp value of S3 becomes greater than the output of S2. During the experiments, we have observed that there is a relationship between the shape of membership functions reflecting the uncertainty in linguistic terms. As this uncertainty is increased by enlarging the widths of the fuzzy sets, there is a rise in the fuzzy scores of suppliers.

Another important observed feature of this work is in the third approach, the crisp score of S3 is greater than crisp score of S2 in the type-1 fuzzy logic system, however, in the interval type-2 fuzzy logic system, the output of S2 is almost the same as the result of S3. Thus, it could show that interval type-2 FLSs are able to deal with linguistic uncertainty better than type-1 FLSs. Simply because, type-2 fuzzy sets are more effective in capturing the linguistic assessment and considering them as an input of the main problem [Karnik and Mendel, 1999]. Hence, we will use real world data to evaluate suppliers in an interval type-2 fuzzy environment.

We took real world data for supplier evaluation from a manufacturing company in Turkey. We evaluate five suppliers based on the same way as explained above. Table 5.13 depicts the linguistic terms of their performance while the linguistic importance of criteria is kept the same as in the proposed approach. Their crisp scores are obtained as shown in Table 5.14. Depending on the number of suppliers in the associated problem instance, the results obtained from the IT2FS approach for the relevant suppliers are used in the next stage. For example, considering Inst1, where $S = 2$, associated risk values for S1 and S2 are fed into the stage two approach.

5.2 Stage Two: Supplier Selection Aware Inventory Planning

In this part of the work, there are two objectives to optimize: (i) minimise exposure to risk due to the supplier selection and (ii) minimize the total cost of

TABLE 5.10: Suppliers' performance ratings with respect to the selection criteria

Suppliers	w1	w2
S1	((0,0,2,3),(0,0,2,5))	((7,8,10,10),(5,8,10,10))
S2	((3,4,4,5),(1,4,4,7))	((3,4,4,5),(1,4,4,7))
S3	((3,4,4,5),(1,4,4,7))	((0,0,2,3),(0,0,2,5))
Suppliers	w3	w4
S1	((5,6,6,7),(3,6,6,9))	((3,4,4,5),(1,4,4,7))
S2	((5,6,6,7),(3,6,6,9))	((7,8,10,10),(5,8,10,10))
S3	((3,4,4,5),(1,4,4,7))	((5,6,6,7),(3,6,6,9))
Suppliers	w5	w6
S1	((0,0,2,3),(0,0,2,5))	((5,6,6,7),(3,6,6,9))
S2	((3,4,4,5),(1,4,4,7))	((7,8,10,10),(5,8,10,10))
S3	((7,8,10,10),(5,8,10,10))	((0,0,2,3),(0,0,2,5))
Suppliers	w7	w8
S1	((3,4,4,5),(1,4,4,7))	((0,0,2,3),(0,0,2,5))
S2	((0,0,2,3),(0,0,2,5))	((3,4,4,5),(1,4,4,7))
S3	((3,4,4,5),(1,4,4,7))	((0,0,2,3),(0,0,2,5))
Suppliers	w9	w10
S1	((7,8,10,10),(5,8,10,10))	((3,4,4,5),(1,4,4,7))
S2	((0,0,2,3),(0,0,2,5))	((0,0,2,3),(0,0,2,5))
S3	((5,6,6,7),(3,6,6,9))	((5,6,6,7),(3,6,6,9))

TABLE 5.11: Found Scores for Suppliers by Ordoobado's work

Suppliers	Score of T1FS		Score of IT2FS	
	Crisp Scores	Rank	Crisp Scores	Rank
S1	13.50	3	16.72	3
S2	15.78	1	18.56	1
S3	14.44	2	17.79	2

the supply chain, subject to a set of constraints. The weighted sum and Tchebycheff scalarisation are used to convert the two-objective problem of minimizing the vector *TotalFitness* (TF) into a scalar problem as seen in Equation 5.2 and Equation 5.3.

TABLE 5.12: Ranking of suppliers results for three approaches

Suppliers	Approach 1	Rank	Approach 2	Rank	Approach 3	Rank
S1	27.41	3	30.03	3	32.03	3
S2	29.74	1	32.12	1	34.00	2
S3	29.32	2	32.00	2	34.06	1

TABLE 5.13: The performance of each supplier in terms of selected criteria for five suppliers

Supplier	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10
S1	P	G	P	P	P	P	P	P	P	P
S2	VG	G	VG	EX	EX	EX	EX	EX	VG	VG
S3	G	G	G	G	G	G	G	G	VG	G
S4	VG	G	-	-	-	-	-	-	-	-
S5	P	G	G	P	P	P	P	G	G	P

TABLE 5.14: Rank and Risk Values of Suppliers for each Instance

Suppliers	Crisp Score	Inst1-Inst6		Inst7-Inst18		Inst19-Inst24	
		Rank	Risk	Rank	Risk	Rank	Risk
S1	10.88	2	3.28	3	4.90	4	6.60
S2	24.80	1	1.44	1	2.15	1	2.90
S3	17.59	-	-	2	3.03	2	4.10
S4	5.96	-	-	-	-	5	12.05
S5	12.56	-	-	-	-	3	5.72

The form of weighted sum is depicted as:

$$\text{minimise } TF = w_1 TR + w_2 TC \quad (5.2)$$

where TR is the total risk associated with a supply chain (Equation 4.8), TC is the total cost (Equation 4.7). Their respective weights are w_1 and w_2 and $w_1 + w_2 = 1$. Additionally, the form of Tchebycheff problem is shown as:

$$\begin{aligned} \text{minimise } TF = \max \{ & w_1 (TR - \mu_{TR}) / (max_{TR} - \mu_{TR}), \\ & w_2 (TC - \mu_{TC}) / (max_{TC} - \mu_{TC}) \} \end{aligned} \quad (5.3)$$

where μ_{TR} is the idealistic total risk affiliated with a supply chain, μ_{TC} is the desirable total cost, max_{TR} , max_{TC} are the maximum values in 100 experiments

for TR and TC , their respective weights are w_1 and w_2 and $w_1 + w_2 = 1$ [Miettinen, 1999].

5.2.1 Multi-objective Simulated Annealing

Multi-objective simulated annealing approach requires design of several domain specific components, such as, solution representation, neighbourhood/move operator and algorithmic settings, maximum number of iterations and cooling schedule. In this section, we provide the details of the MOSA approach used as a generic optimisation algorithm for solving the supply chain management problem described in the previous chapter. A real-valued *solution representation* is used in MOSA. A solution consists of a 4 dimensional array that represents an inventory plan. The dimensions correspond to the source node, destination node, component/product and time period, respectively. Each element of the array contains a value in $[0,1]$ that denotes the amount of stock added to the inventory of a source node, for a destination node, of a product in a particular time period. For example, if $currentPlan[1,2,3,4]$ contains the value 0.5, this indicates that source node 1 receives/produces 50% of its capacity of product 3 for destination node 2 in period 4. This method of representing inventory ensures that a node can never supply/produce more than its capacity in one time period. Inventory values are also restricted by a minimum order quantity, and order quantities. For example, we might set a minimum order quantity of 100 for a particular product at a particular node, and then restrict orders to units of 100 (e.g. 500, 600, 700, 800 etc.).

TF (Equation 5.2) is used to evaluate the quality (*fitness*) of a candidate solution during the search process. Although it is a challenge to decide on how much weight to apply to each of the objectives [Kim and de Weck, 2006], these settings can be considered to reflect the attitude of a “decision maker” towards cost and risk in our study. More importantly and additionally, by running the algorithm with different weight settings, we can obtain a set of solutions indicating the trade-off between those objectives and the “decision maker” can choose one of the solutions as appropriate.

The algorithm generates an *initial solution* randomly, containing values in $[0,1]$ that represent the proportion of capacity added to each supply node for each destination node of each product in each time period. Then, MOSA attempts to

improve this initial solution iteratively by applying a *neighbourhood operator* and then deciding whether to accept or reject the resultant solution.

A parametrised neighbourhood operator is implemented for perturbing a solution. At each step, a new solution is generated by this operator modifying the settings of a number of randomly selected elements with random values. The number of elements being a discrete parameter value in between 1 and 10. This operator is an adaptive operator using reinforcement learning for parameter control [Özcan et al., 2010]. The number of elements altered is chosen using a reinforcement learning method in which a set of counters is used to record the most successful number of changes. A utility value/score is maintained using a counter for each parameter setting. Initially, a random number of elements between 1 and 10 is chosen and the solution is altered accordingly. At any time, if the changes to the selected elements result in a non-worsening solution, the counter for that parameter value is incremented, otherwise it is decremented. When the next perturbation needs to be made, the algorithm looks for the number of elements with the maximum score, i.e., the most successful. If there are multiple numbers with the maximum value, one is chosen at random. This method encourages the MOSA algorithm to select the number of changes that has resulted in the largest ratio of improvements to changes detrimental to solutions.

5.2.2 MOSA with certain weight settings

The aim of this section is to investigate a trade-off between risk and cost through varying the weights of objectives in MOSA. All weight vectors remain unchanged during the search.

5.2.2.1 Experimental Setup

The supply chain problem instance used in this study contains two suppliers (denoted as S1 and S2 in Table 5.14, two manufacturing plants and two potential customers shown in Figure 4.4. The planning horizon contains only one discrete time period.

We tested MOSA on the problem instance using the information from the previous stage considering 6 different weight settings for w_1 , while $w_2 = 1.0 - w_1$

yielding 6 different scenarios for optimizing TF with weighted sum and Tchebycheff scalarisations of cost and risk objectives as provided in Equations 5.2 and 5.3, respectively. Each scenario Scn-1, Scn-2, Scn-3, Scn-4 and Scn-5 uses 0.0, 0.2, 0.5, 0.8 and 1.0 as a setting for w_1 , respectively. The last scenario (Scn-6) generates 100 random settings for w_1 in $(0,1]$ with $w_2 = 1.0 - w_1$. A single SA run is performed with each setting generating 100 solutions. This experiment is performed to observe the trade-off solutions for the given instance. The cooling schedule for the temperature within MOSA is geometric, resulting in a slow and consistent reduction of temperature over the course of a run. The initial temperature is fixed as 10,000. To allow 100,000 iterations, the cooling rate is computed using Equation 3.10 and then SA terminates. Each SA experiment is repeated 30 times, except for Scn-6.

5.2.2.2 Experimental Results

The stage one fuzzy approach described in Section 4.1.1 yields the output for two suppliers as shown in Table 5.14. S2 has a better ranking than S1 due to low risk. This information is fed into the SA solver described in Section 5.2.1 for inventory planning.

MOSA indeed discovers high quality inventory plans based on the described model for those scenarios as shown in Table 5.15. Clearly, MOSA produces lower mean TR values (higher TC values) as the weight of total supplier risk increases (total cost of the supply chain operation decreases) when weighted sum is used. MOSA provides flexibility for the decision makers enabling them to try out different risk weightings (representing risk-avoiding or risk-taking strategies), and observe how this choice influences inventory planning in terms of the total supplier risk and cost of the supply chain operation. It seems that the MOSA using Tchebycheff scalarisation spends more effort on the improvement of TR as compared to TC .

As for the behaviour of the MOSA during the search process, Figure 5.1 provides the progress plot of the Tchebycheff based objective value (TF) of the current solution in a single run for each scenario as an example. The objective value oscillates radically during the initial phases of the search process. This indicates that large moves are accepted initially as expected, and this behaviour settles down in time and the algorithm seems to reach a plateau for all scenarios. This is

TABLE 5.15: Best and average results in terms of TR , TC obtained from 30 runs for each scenario for Weighted sum and Tchebycheff scalarisation

Scenario	Weighted sum				Tchebycheff			
	Average		Best		Average		Best	
	TR	TC	TR	TC	TR	TC	TR	TC
Scn-1	14867	2769	15504	2630	14867	2769	15504	2630
Scn-2	13004	2995	14605	2743	14557	3144	13706	3063
Scn-3	12259	3151	12628	2908	14881	2756	15863	2640
Scn-4	12179	3350	12628	2888	14875	2752	15853	2630
Scn-5	11839	8145	12269	5600	11839	8145	12269	5600

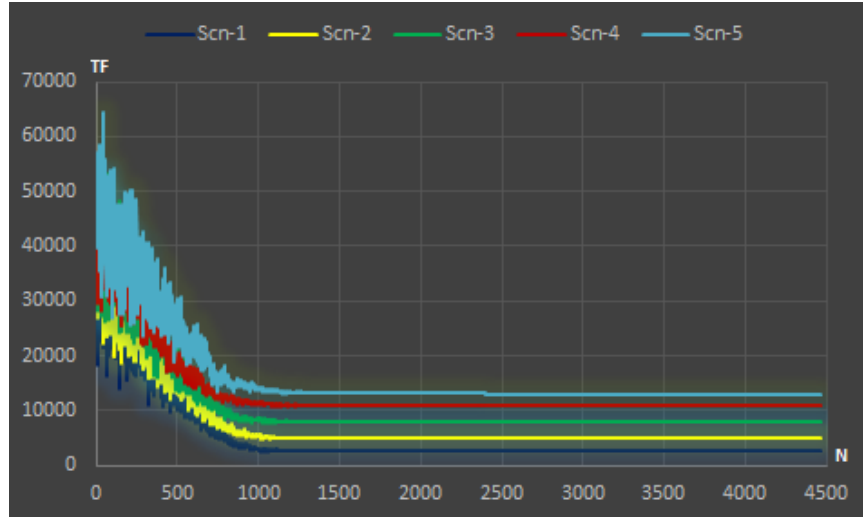


FIGURE 5.1: Behaviour of the SA algorithm in a single run for each scenario using a Tchebycheff scalarisation.

likely due to the chosen cooling rate and initial temperature. Although we tested a set of different values for parameter tuning purposes, we did not observe much performance improvement over different scenario.

Figure 5.2 provides the scatter plot of TR versus TC , obtained after applying the MOSA using weighted sum and Tchebycheff scalarisation approaches under Scn-6. The former approach produces a slightly better pareto front than the latter one with a hypervolume [Zitzler and Thiele, 1999] of 0.75 versus 0.74. Moreover, Figure 5.3 demonstrates another scatter plot of TR versus TC which contains

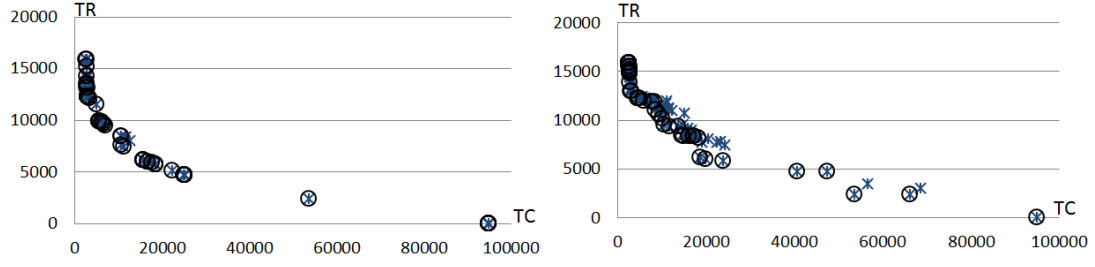


FIGURE 5.2: TR vs TC from 100 runs (each with a random weight setting) of the SA algorithm based on weighted sum (on the left) and Tchebycheff (on the right) scalarisations. Circles indicate the pareto front.

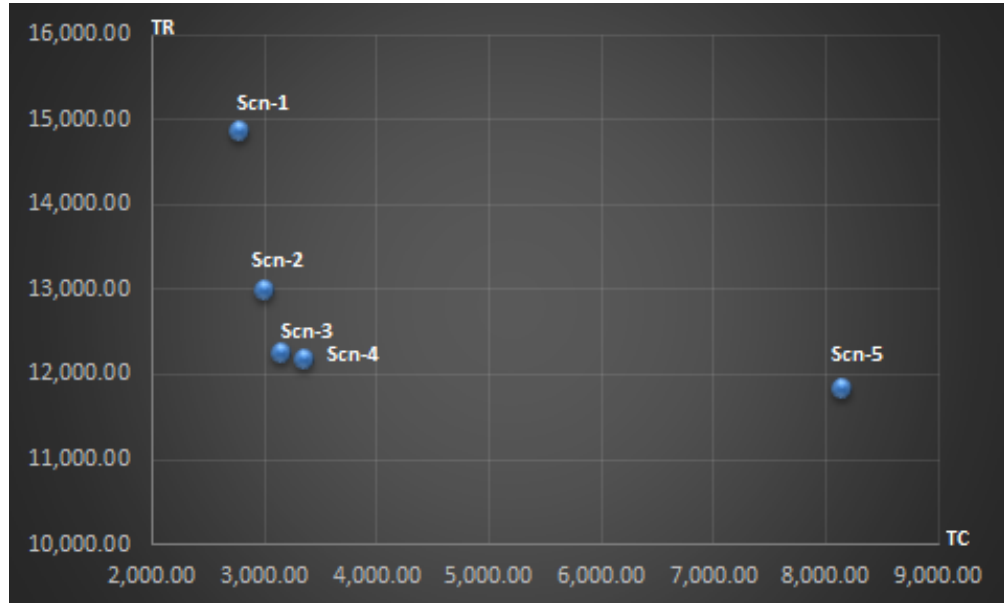


FIGURE 5.3: TR vs TC for five runs (a run for each scenario except Scn-6) of the SA algorithm based on the weighted sum approach.

five solutions. One solution for each scenario is obtained using the weighted sum scalarisation approach. It is clearly seen that it is possible to locate different regions through varying the weights.

In summary, the results indicate that there is certainly a trade-off between risk and cost. The trade-off solutions can be obtained using a simple weighted sum approach via a generic SA optimisation algorithm. Hence, it is possible for decision makers to choose an appropriate strategy based on their aversion to risk and cost. The proposed system could be used in What-If scenarios to see how varying

attitudes to risk affect inventory planning. A decision maker can alternatively choose a solution from the set of trade-off solutions.

5.2.3 MOSA with random weight settings

The previous section provided a MOSA approach to minimize supplier risk and operational costs for inventory planning, simultaneously. MOSA performed well using a learning move operator with an adaptive parameter control. The proposed approach is capable of capturing the trade-off between risk and cost via scalarisation of both objectives, giving flexibility to the decision makers to choose from a set of trade-off solutions for supply chain management. Weighted sum and Tchebycheff scalarisation approaches are tested using six different weight settings for each scenario. However, maintaining the diversity of search direction becomes difficult under certain weight settings as seen in Figure 5.3. Therefore, we develop this approach using random weight settings. The weight vector of each run is generated randomly.

In addition, MOSA approach depends on several parameters (e.g., initial solution, cooling schedule) as explained in Section 5.2.1. The choice of parameters can have an impact on the algorithm performance. Thus, selected parameters of MOSA are calibrated and the effect of tuned parameters on the results is investigated. A comparison between MOSA with tuned parameters and MOSA with default setting is provided.

5.2.3.1 Experimental Setup

The supply chain problem instances used in this work consist of four groups of six problem instances, totalling up to twenty four instances (Table 4.5) as detailed in Section 4.1.4. The planning horizon contains three discrete time periods. We perform MOSA on these problem instances with respect to the information from the previous stage considering random weight settings for w_1 in $(0,1]$, while $w_2 = 1.0 - w_1$ in order to obtain a desirable TF with weighted sum and Tchebycheff scalarisations of cost and risk objectives as given in Equations 5.2 and 5.3, respectively.

TABLE 5.16: Parameter setting options used during the parameter tuning of MOSA

Parameters	MOSA
Initial Temperature	(10,000, 30,000, 50,000)
Absolute Temperature	$(1 \times 10^{-5}, 5 \times 10^{-6}, 1 \times 10^{-6})$

TABLE 5.17: Average rank for MOSA algorithm, with a particular parameter configuration, Init. temp.:Initial temperature, Abs. temp.: Absolute temperature.

Exp.	Init. temp.	Abs. temp.	Average Rank for WS				Average Rank for TCH			
			Inst1	Inst13	Inst19	Result	Inst1	Inst13	Inst19	Result
1	10,000	1×10^{-5}	5.57	5.40	5.07	5.3	4.83	4.53	5.03	4.8
2	10,000	5×10^{-6}	4.93	4.73	5.53	5.1	4.73	5.57	3.93	4.7
3	10,000	1×10^{-6}	4.63	4.77	3.70	4.4	5.27	4.50	5.30	5.0
4	30,000	1×10^{-5}	4.77	4.30	5.63	4.9	5.43	4.40	5.60	5.1
5	30,000	5×10^{-6}	5.73	5.47	5.10	5.4	4.40	5.10	5.03	4.8
6	30,000	1×10^{-6}	4.10	5.17	5.80	5.0	4.57	5.20	5.50	5.1
7	50,000	1×10^{-5}	4.93	5.20	4.33	4.8	5.37	5.47	5.03	5.3
8	50,000	1×10^{-6}	4.77	5.00	4.87	4.9	5.33	4.87	4.67	5.0
9	50,000	1×10^{-6}	5.50	4.97	4.93	5.1	5.07	5.37	4.90	5.1

A single MOSA run is performed with each setting generating 30 solutions in 30 different random settings. The trade-off solutions for the given instances are achieved running these experiments. Each MOSA experiment is repeated 30 times with both a default setting and tuned parameters using weighted sum and Tchebycheff scalarisation approaches.

5.2.3.2 Parameter Tuning

We investigated two control parameters for SA: *initial temperature* and *absolute temperature* shown in Table 5.16. We choose three instances considering their size (Inst1, Inst13 and Inst19) and 9 parameter settings 3^2 to try out all the alternatives. Each trial is repeated 30 times. In order to assess algorithms, the average rank of hypervolume values is computed. To achieve these ranks, first, each run is ranked from the highest to the lowest hypervolume values for 9 parameter settings. For example the highest hypervolume value is assigned as 1. And then the average rank of 30 runs is calculated. In the same manner, for three instances we rank them and get the average rank of three instances as a result shown in Table 5.17.

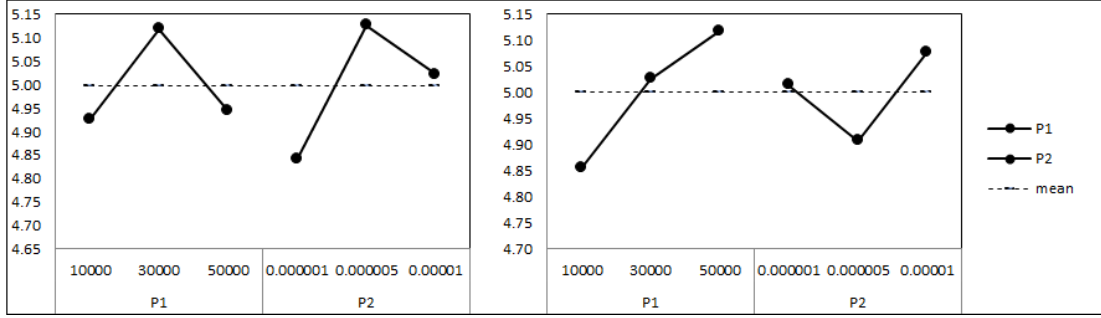


FIGURE 5.4: Main effects plot with mean rank values based on 3 instances in MOSA based on weighted sum (on the left) and Tchebycheff (on the right) scalarisations, P1: Initial temperature, P2: Absolute temperature.

TABLE 5.18: Parameter setting options used during the parameter tuning of three MOEAs, Init. temp.: Initial temperature, Abs. temp.: Absolute temperature, Max. evaluation: Maximum evaluation, N_s : Number of suppliers.

Settings	Init. temp.	Abs. temp	Max. evaluation
<i>DefaultSetting</i>	10,000	1×10^{-5}	$N_s \times 100,000$
<i>TunedParameters_{WS}</i>	10,000	1×10^{-6}	500,000
<i>TunedParameters_{TCH}</i>	10,000	5×10^{-6}	500,000

In order to find the average effect of each parameter setting on the result (the average rank of three instances), the result of each parameter setting is averaged by the number of combinations with the same parameter setting. For instance, let us consider the average effect of an initial temperature 10,000 for the weighted sum scalarisation approach is computed as $(5.3 + 5.1 + 5.4)/3 = 4.9$ where the number of combinations is 3 out of 9. Figure 5.4 demonstrates the main effect plot that consists of the average effect of each parameter setting for MOSA based on weighted sum and Tchebycheff scalarisations. The parameter setting that has the lowest value would be accepted as the best setting for each parameter as summarised in the main effects plot. Therefore, the best configuration for MOSA based on the weighted sum scalarisation is attained as 10,000 for $P1$ and 1×10^{-6} for $P2$ while it is found as 10,000 for $P1$ and 5×10^{-6} for $P2$ for Tchebycheff scalarisation approach (shown in Table 5.18).

5.2.3.3 Experimental Results

To evaluate results of two scalarisation methods for each problem instance in different parameter settings, while solving two objectives supply chain problem we used three metrics, hypervolume, generational distance and inverted generational distance. The averages and standard deviations of these metrics are given in Table 5.19 and Table 5.20 for default parameter settings of MOSA ($MOSA_d$) and MOSA with tuned parameter settings ($MOSA_t$), respectively. Also, both tables summarise results of weighted sum and Tchebycheff approaches obtained during the 30 runs of the MOSA algorithm, for each instance in terms of three performance metrics. We analyse the behaviour of the methods with the aim of highlighting which one works better using one tailed Wilcoxon signed-ranks test.

$$H_0 : P_{Ws} = P_{Tch}$$

$$H_A : P_{Ws} < P_{Tch}$$

Where H_0 represents null hypothesis which asserts that the difference between median values of the two methods is zero and H_A represents alternative hypothesis, i.e, the median value of the weighted sum approach is less than the median of Tchebycheff approach. The confidence level (significance level) used for non-parametric test is 95% (p-value under 0.05).

The following notations are used in Table 5.19 and Table 5.20: Given two algorithms; Ws versus Tch, $>$ ($<$) indicates that Ws(Tch) is better than Tch(Ws) and this performance difference is statistically significant within a confidence interval of 95% and $Ws \geq Tch$ ($Ws \leq Tch$) denotes that Ws(Tch) performs better on average than Tch(Ws) but no statistical significance.

In order to assess results, we start with comparing hypervolume values, higher value represents better quality of the obtained results. Among instances generated using $MOSA_d$, mostly the best values are obtained by the weighted sum approach for 14 out of 24 instances (seen in Table 5.19). An one tailed Wilcoxon signed-ranks test showed that there are clear statistical significant differences between results of both approaches for hypervolume values. As for the remaining instances, Tchebycheff is slightly better than the weighted sum. On the other hand, based on

TABLE 5.19: Performance metrics of two objectives supply chain problem with default settings

Inst.		Hypervolume		Generational Distance		Inverted Generational Distance	
		Ws	Tch	Ws	Tch	Ws	Tch
Inst1	Mean	0.7668	> 0.7613	85,537.3	> 89,387.7	17,492,133.3	> 20,506,033.3
	Std.	0.0068	0.0069	25,146.9	25,152.7	1,446,890.2	1,511,602.3
Inst2	Mean	0.7560	> 0.7548	75,075.6	> 78,235.5	3,312,090.0	≤ 3,173,056.7
	Std.	0.0066	0.0059	31,988.9	24,558.5	506,377.4	452,285.6
Inst3	Mean	0.6853	≤ 0.6875	72,213.0	> 84,563.3	8,358,750.0	≤ 8,243,740.0
	Std.	0.0058	0.0055	23,598.3	33,230.4	410,118.8	345,820.3
Inst4	Mean	0.7138	> 0.7132	66,254.5	≤ 63,889.1	5,086,180.0	> 5,227,973.3
	Std.	0.0074	0.0054	18,551.0	24,458.7	898,987.1	800,542.2
Inst5	Mean	0.7462	> 0.7451	79,374.0	> 80,003.2	2,167,056.7	≤ 2,139,006.7
	Std.	0.0087	0.0059	31,168.2	30,864.7	277,384.2	205,026.2
Inst6	Mean	0.7261	> 0.7253	71,474.3	> 75,652.8	4,044,210.0	< 3,651,440.0
	Std.	0.0079	0.0080	29,508.9	23,345.1	852,454.4	608,008.6
Inst7	Mean	0.7966	> 0.7920	187,879.7	≤ 186,390.0	1,917,516.7	> 2,001,410.0
	Std.	0.0054	0.0050	50,705.2	51,851.3	73,992.2	97,295.2
Inst8	Mean	0.7968	≤ 0.7974	166,509.0	≤ 163,587.0	2,315,020.0	> 2,495,550.0
	Std.	0.0036	0.0030	62,663.2	43,294.1	153,788.2	132,238.1
Inst9	Mean	0.8189	≤ 0.8201	150,651.5	≤ 149,852.1	833,311.3	> 861,560.0
	Std.	0.0043	0.0041	42,818.7	50,542.0	23,332.0	32,212.5
Inst10	Mean	0.7912	≤ 0.7925	174,243.0	> 176,765.4	1,590,490.0	> 1,651,276.7
	Std.	0.0049	0.0041	39,414.0	50,212.4	68,521.5	75,829.3
Inst11	Mean	0.7880	≤ 0.7886	167,373.4	> 176,626.0	2,515,596.7	> 2,724,940.0
	Std.	0.0053	0.0034	47,018.2	49,522.0	139,686.6	152,664.7
Inst12	Mean	0.8055	≤ 0.8057	166,607.3	≤ 157,459.0	1,288,066.7	≤ 1,279,460.0
	Std.	0.0051	0.0042	38,357.7	51,360.4	79,419.7	67,277.8
Inst13	Mean	0.7025	> 0.7005	293,122.7	> 333,814.7	10,816,333.3	< 10,367,240.0
	Std.	0.0081	0.0073	102,169.2	129,162.3	591,374.9	516,510.0
Inst14	Mean	0.7013	≤ 0.7015	287,654.0	≤ 239,689.0	2,469,223.3	> 2,475,790.0
	Std.	0.0037	0.0051	99,436.8	76,087.5	47,035.1	87,083.7
Inst15	Mean	0.7205	> 0.7184	344,926.7	≤ 327,313.7	2,665,466.7	> 2,692,600.0
	Std.	0.0049	0.0036	85,622.0	79,940.6	55,798.4	54,415.3
Inst16	Mean	0.7016	≤ 0.7019	299,391.5	≤ 263,950.0	2,460,990.0	> 2,472,370.0
	Std.	0.0042	0.0059	103,958.0	90,978.0	36,417.0	71,703.9
Inst17	Mean	0.7011	> 0.6977	269,903.9	≤ 204,211.6	4,962,620.0	≤ 4,960,490.0
	Std.	0.0064	0.0049	119,272.2	82,350.2	216,021.7	202,435.6
Inst18	Mean	0.7306	> 0.7277	259,170.0	> 285,349.3	2,026,743.3	> 2,070,156.7
	Std.	0.0066	0.0054	109,002.2	90,659.3	73,994.5	46,294.1
Inst19	Mean	0.7003	≤ 0.7013	2,869,753.3	≤ 2,659,692.7	58,057,000.0	> 58,326,933.3
	Std.	0.0047	0.0051	1,038,701.6	816,361.9	957,290.7	1,241,284.4
Inst20	Mean	0.7321	> 0.7307	2,184,081.7	≤ 1,993,753.3	41,224,233.3	> 41,543,800.0
	Std.	0.0044	0.0041	823,315.2	630,548.9	680,859.9	640,389.8
Inst21	Mean	0.7088	≤ 0.7092	2,763,496.7	≤ 2,503,279.3	56,562,533.3	> 56,811,900.0
	Std.	0.0038	0.0044	791,756.2	864,824.6	756,483.2	726,190.0
Inst22	Mean	0.7028	> 0.7027	2,402,095.7	> 2,521,456.7	56,934,500.0	> 57,888,333.3
	Std.	0.0049	0.0035	797,772.4	742,186.1	1,061,677.1	871,288.5
Inst23	Mean	0.7293	> 0.7271	2,484,562.7	> 2,514,333.7	50,413,033.3	> 51,326,366.7
	Std.	0.0048	0.0037	664,988.3	641,442.9	1,218,816.3	1,036,558.0
Inst24	Mean	0.7337	> 0.7333	2,160,750.3	≤ 1,850,501.7	48,664,666.7	> 48,854,866.7
	Std.	0.0040	0.0037	772,402.7	667,719.2	729,311.9	1,112,939.1

results of $MOSA_t$, the weighted sum approach is superior to Tchebycheff approach on 16 over 24 instances (seen in Table 5.20) and for 7 instances, Tchebycheff is slightly better than weighted sum. For Inst7, Tchebycheff provides significantly better hypervolume value than the weighted sum approach achieved. In addition,

TABLE 5.20: Performance metrics of two objectives supply chain problem with tuned parameters

Inst.		Hypervolume		Generational Distance		Inverted Generational Distance	
		Ws	Tch	Ws	Tch	Ws	Tch
Inst1	Mean	0.7733	> 0.7690	85,462.5	> 95,397.3	16,841,900.0	> 19,371,466.7
	Std.	0.0091	0.0074	20,213.5	30,013.7	927,549.7	1,661,681.2
Inst2	Mean	0.7603	≤ 0.7607	77,280.3	> 75,963.7	3,163,576.7	< 2,930,050.0
	Std.	0.0068	0.0054	22,843.7	28,167.2	493,160.3	375,300.6
Inst3	Mean	0.6903	> 0.6902	67,479.6	> 74,146.4	7,979,166.7	> 7,904,353.3
	Std.	0.0065	0.0051	3,1026.3	25,959.9	445,211.7	329,259.6
Inst4	Mean	0.7204	≤ 0.7217	68,977.7	≤ 63,372.9	5,068,016.7	< 4,480,433.3
	Std.	0.0057	0.0069	25,732.0	22,542.4	819,808.5	620,144.4
Inst5	Mean	0.7543	> 0.7526	74,447.8	> 82,905.9	2,133,026.7	≤ 2,085,860.0
	Std.	0.0082	0.0064	30875.4	27,358.2	253,877.1	228,446.6
Inst6	Mean	0.7323	≤ 0.7336	81,334.6	≤ 75,548.0	3,812,376.7	≤ 3,502,436.7
	Std.	0.0070	0.0089	24,595.9	26,558.6	926,819.3	649,275.1
Inst7	Mean	0.7968	< 0.7983	190,459.0	≤ 189,294.6	1,817,530.0	> 1,948,980.0
	Std.	0.0037	0.0044	54,377.8	59,029.3	99,156.2	129,038.8
Inst8	Mean	0.8001	≤ 0.8010	171,902.3	≤ 165,251.0	2,298,043.3	> 2,524,910.0
	Std.	0.0050	0.0039	49,505.9	42,076.9	106,083.4	165,970.1
Inst9	Mean	0.8209	≤ 0.8228	142,766.3	≤ 135,094.5	814,594.3	> 836,195.0
	Std.	0.0051	0.0037	40,775.3	37,000.8	33,259.6	28,532.8
Inst10	Mean	0.7950	> 0.7940	175,808.3	< 153,149.8	1,553,430.0	> 1,582,030.0
	Std.	0.0051	0.0035	54,708.5	43,056.8	58,100.7	76,013.6
Inst11	Mean	0.7913	> 0.7912	174,361.8	< 153,146.7	2,491,006.7	> 2,639,446.7
	Std.	0.0053	0.0042	48,645.0	51,132.1	97,890.4	165,888.7
Inst12	Mean	0.8090	> 0.8075	168,454.9	< 147,448.6	1,251,120.0	> 1,277,396.7
	Std.	0.0049	0.0035	41,201.9	48,036.1	70,026.2	63,649.7
Inst13	Mean	0.7076	> 0.7045	310,709.0	> 359,591.0	10,727,580.0	< 10,161,913.3
	Std.	0.0074	0.0078	96,764.7	122,292.3	653,156.9	521,745.8
Inst14	Mean	0.7048	> 0.7044	284,985.3	≤ 264,627.0	2,429,616.7	> 2,448,966.7
	Std.	0.0050	0.0052	65,562.0	82,978.8	54,092.0	71,773.2
Inst15	Mean	0.7242	> 0.7231	313,221.0	≤ 285,670.8	2,643,120.0	≤ 2,642,233.1
	Std.	0.0059	0.0056	90,350.0	100,594.1	78,777.9	68,419.5
Inst16	Mean	0.7061	> 0.7045	254,250.8	> 278,954.7	2,396,606.7	> 2,427,793.3
	Std.	0.0048	0.0071	86,671.6	101,791.7	59,931.2	77,949.6
Inst17	Mean	0.7021	≤ 0.7036	256,914.0	< 202,726.9	4,932,020.0	≤ 4,899,156.7
	Std.	0.0057	0.0064	94,232.6	70,751.7	222,404.3	241,587.7
Inst18	Mean	0.7325	> 0.7315	276,170.1	> 271,162.3	2,009,436.7	> 2,027,386.7
	Std.	0.0054	0.0049	85,834.1	75,648.1	52,376.6	54,102.5
Inst19	Mean	0.7022	> 0.7017	2,136,531.7	> 2,466,676.7	57,306,266.7	> 58,258,666.7
	Std.	0.0042	0.0050	607,978.7	698,724.2	1,286,836.7	1,292,007.1
Inst20	Mean	0.7316	> 0.7305	2,064,702.3	≤ 1,956,880.0	41,158,766.7	> 41,367,166.7
	Std.	0.0045	0.0039	678,215.2	508,273.8	812,124.0	881,406.8
Inst21	Mean	0.7090	> 0.7085	2,481,134.0	> 2,657,690.3	56,084,733.3	> 56,737,533.3
	Std.	0.0044	0.0038	893,922.1	842,453.2	960,658.3	1,008,732.4
Inst22	Mean	0.7036	> 0.7019	2,546,636.7	≤ 2,411,970.3	56,947,966.7	> 57,651,666.7
	Std.	0.0036	0.0040	786,735.9	872,907.2	668,585.4	692,134.7
Inst23	Mean	0.7279	≤ 0.7284	2,599,646.7	≤ 2,283,313.3	50,817,900.0	> 51,138,500.0
	Std.	0.0039	0.0044	704,422.5	588,781.1	919,166.4	853,284.7
Inst24	Mean	0.7342	> 0.7334	2,052,283.0	> 2,097,496.7	48,668,333.3	> 49,035,500.0
	Std.	0.0039	0.0036	612,538.5	689,576.2	977,624.8	973,408.6

when the size of problem gets larger, weighted sum achieve the better hypervolume values than Tchebycheff.

Apart from these, another important finding is that the performance difference between $MOSA_d$ and $MOSA_t$ is less than 1% for both approaches on 24 instances as

TABLE 5.21: The average percentage improvement of metrics on 24 instances when tuned parameters are used

	Hypervolume		Generational Distance		Inverted Generational Distance	
	Ws	Tch	Ws	Tch	Ws	Tch
Average	0.4%	0.5%	2.4%	2.5%	1.9%	2.7%

shown in Table 5.21. However, even with this small change in hypervolume value has an influence on the performance of approaches. For instance, for $MOSA_t$, weighted sum achieved better hypervolume values on 8 instances (Inst3, 10, 11, 12, 14, 16, 19, 21) while Tchebycheff provides slightly better results in $MOSA_d$ algorithm. Moreover, the performance of Tchebycheff is improved achieving significantly better result on Inst7 since weighted sum provides better result for $MOSA_d$.

Secondly, we consider generational distance where lower value denotes better quality of results. Although on the one hand, for $MOSA_d$, in some cases, Tchebycheff provides lower values, the one tailed Wilcoxon signed-ranks test shows that there is a significant statistical difference between both approaches and weighted sum is significantly better than Tchebycheff on 11 instances. Also Tchebycheff is slightly better than weighted sum on 13 instances. On the other hand, for $MOSA_t$, weighted sum performs significantly better than Tchebycheff over 10 instances while GD values of Tchebycheff are significantly better than results of weighted sum for four instances (Inst10, 11, 12 and 17). In addition, with respect to generational distance, Tchebycheff is slightly better than weighted sum for 10 out of 24 instances. Based on the size of problem, there is no significant difference on the results of both approaches when the size of problem gets larger for both cases. Moreover, as seen in Table 5.21, the average performance improvements are provided as 2.4% for weighted sum and 2.5% for Tchebycheff in $MOSA_t$. The statistically improvement on the performance of Tchebycheff is clearly seen on six instances (Inst10, 11, 12, 17, 22 and 23) whereas weighted sum provides an improvement on four instances (Inst16, 19, 21 and 24). For instance, for Inst10 and Inst11, Tchebycheff is improved from “significantly worse” to “significantly better”.

Finally, considering inverted generational distance, the weighted sum approach outperforms Tchebycheff for 17 instances in $MOSA_d$ and 17 instances in $MOSA_t$

TABLE 5.22: The summary of SA results

Metrics	Approaches	The number of instances			
		$Ws > Tch$	$Ws \geq Tch$	$Ws < Tch$	$Ws \leq Tch$
HV	$MOSA_d$	14	0	0	10
	$MOSA_t$	16	0	1	7
GD	$MOSA_d$	11	0	0	13
	$MOSA_t$	10	0	4	10
IGD	$MOSA_d$	17	0	2	5
	$MOSA_t$	17	0	3	4

out of 24 instances. This performance variation is statistically significant. Tchebycheff is slightly better than weighted sum on five instances for $MOSA_d$ and four instances for $MOSA_t$. Moreover, Tchebycheff provides better results over Inst6 for $MOSA_d$ algorithm and three instances, Inst2, 4 and 13 for $MOSA_t$ algorithm. To provide better understanding, Table 5.22 demonstrates the summary of results. It can be concluded that mostly the quality of solutions obtained by weighted sum is better than results of Tchebycheff. In addition, the use of default settings is acceptable for MOSA algorithm provided reasonable results, whereas parameter tuning is a fundamental process to achieve better results. Moreover, based on the standard deviation values, more data is located closer to the center of the mean in both approaches for three metrics. But for HV and GD, in most instances, weighed sum achieves smaller standard deviations than Tchebycheff. In addition, for larger sizes, there is an increase on the standard deviation of both approaches. However, in $MOSA_t$, changes in the standard deviation are less for the weighted sum approach.

As a sample, we plotted the pareto front achieved by each scalarisation approach on arbitrarily chosen instances of Inst1, Inst7, Inst13 and Inst19. As it can be observed from Figure 5.5, Tchebycheff produce wider spread of solutions on the pareto front when compared to weighted sum. In addition, these approaches provide a set of trade-off solutions. A common way of (automatically) reducing all trade-off solutions into a ‘preferable’ reasonable single solution is detecting the solution at the *knee point* on the pareto-front. We have used the method presented in [Bechikh et al., 2010] to obtain the a single solution called a knee solution based on the knee point for all problem instances. However, when knee solutions are considered, weighted sum is better than Tchebycheff.

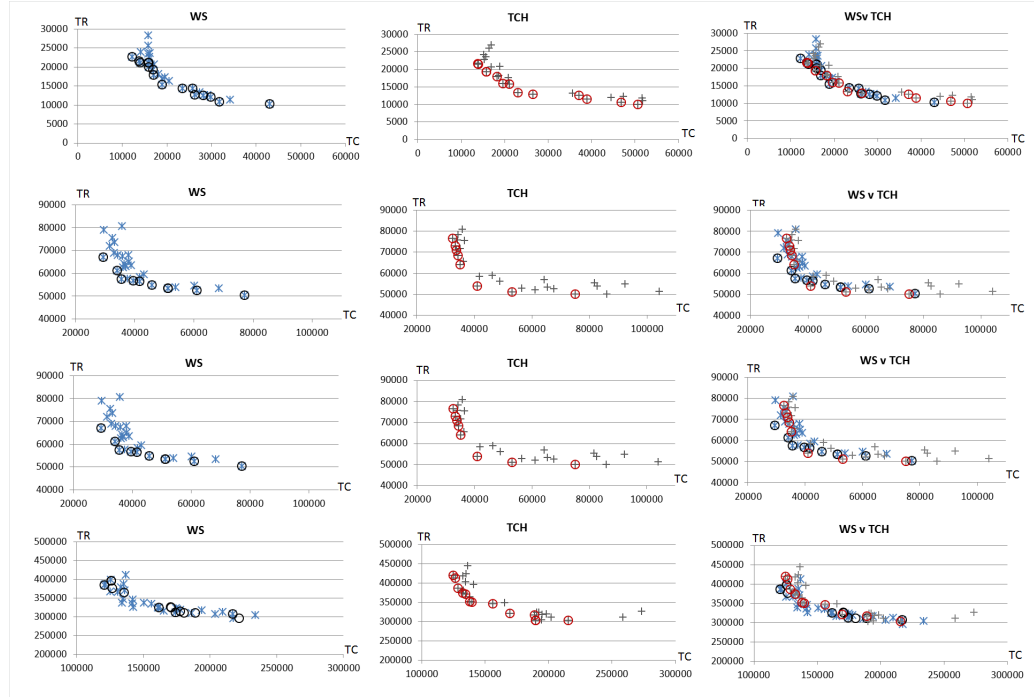


FIGURE 5.5: TR vs TC from 30 runs (each with a random weight setting) of the SA algorithm based on weighted sum (on the left), Tchbycheff (in the middle) and weighted sum versus Tchbycheff (on the right) scalarisations. Circles show the pareto front. The first row represents a $2 \times 2 \times 2$ (supplier, manufacturing plant, customer in a sequence) system, in the second row, a $3 \times 2 \times 3$ system is indicated, the third row shows a $3 \times 3 \times 3$ system and the last row depicts a $5 \times 5 \times 5$ system. For all graphs, instances generated by certain parameters are used.

We have observed that both approaches in the $MOSA_t$ algorithm achieved knee solutions for the majority of the instances satisfying at least 97% of the customer demand as seen service levels in Table 5.23. In our model, we have merged all cost related entities into a single quantity. The results indicate that there could be trade-off in between different constituent costs. For example, Table 5.23 shows that the weighted sum solution is better than the Tchbycheff solution in terms of the total cost. However, service levels are the same for all instances except Inst4 and more importantly, the weighted sum solution provides a better stock out cost while the Tchbycheff solution provides a relatively better production cost in return worsening the cost for the remaining items.

TABLE 5.23: Comparison of two MOSA approaches for two-objective supply chain problem in terms of different cost results which compose the total cost

Inst.		Total Risk	Total Cost	Service Level	Batch Cost	Production Cost	Transportation Cost	Stockout Costs	Holding Cost
Inst1	Ws	10,668	9,087	98.2%	390	6,035	240	1,640	782
	Tch	12,284	9,797	98.2%	410	6,490	272	1,800	825
Inst2	Ws	13,754	10,835.04	99.3%	390	7,019.7	204.94	791.1	2,429.3
	Tch	13,066	10,812.23	98.6%	400	6,321.6	178.73	1,472.8	2,439.1
Inst3	Ws	13,969	11,215.76	99.2%	430	7,433.4	213.36	886	2,253
	Tch	13,252	11,430.09	98.6%	390	7,033.1	208.29	1,595.8	2,202.9
Inst4	Ws	16,232	10,109.67	99.3%	400	7,436.8	210.37	689.3	1,373.2
	Tch	14,011	11,658.52	97.9%	360	7,082.7	186.62	2,475.5	1,553.7
Inst5	Ws	13,057	12,260.45	98.7%	360	7,923	196.25	1,606.5	2,174.7
	Tch	13,039	11,618.54	98.4%	370	7,165.7	218.84	1,871.3	1,992.7
Inst6	Ws	18,163	8,932.72	99.3%	390	5,493.8	192.02	667.5	2,189.4
	Tch	14,801	10,186.69	99.0%	320	6,515.2	180.89	1,047.5	2,123.1
Inst7	Ws	32,120	19,286.5	99.7%	690	15,765	414	575	1,842.5
	Tch	27,243	22,564.5	99.0%	630	17,205	394.5	2,255	2,080
Inst8	Ws	23,790	21,718.28	98.2%	700	15,075	287.88	3,817	1,838.4
	Tch	23,951	22,144.14	98.6%	740	16,039	283.04	3,112	1,970.1
Inst9	Ws	21,987	18,098.05	99.1%	740	13,057	460.75	1,691.5	2,148.8
	Tch	21,987	18,098.05	99.1%	740	13,057	460.75	1,691.5	2,148.8
Inst10	Ws	26,629	21,728.19	98.7%	660	16,242	303.09	2,727.6	1,795.5
	Tch	26,282	22,273.9	98.8%	710	16,500	337.5	2,693.6	2,032.8
Inst11	Ws	31,889	20,622.23	99.1%	770	16,143	382.33	1,933	1,393.9
	Tch	33,116	19,886.01	99.7%	830	17,004	398.31	513.8	1,139.9
Inst12	Ws	26,278	23,352.42	99.5%	800	19,343	322.02	1,258.9	1,628.5
	Tch	29,232	20,891.91	99.2%	730	16,446	329.11	1,691.9	1,694.9
Inst13	Ws	52,260	24,274.5	99.5%	910	18,315	512	1,185	3,352.5
	Tch	48,490	27,202	99.6%	880	21,575	507	1,075	3,165
Inst14	Ws	47,786	22,899.85	99.4%	910	16,969	453.35	1,488.4	3,079.1
	Tch	43,552	25,903.87	99.0%	960	17,930	575.97	2,632.7	3,805.2
Inst15	Ws	45,246	22,182.91	99.5%	990	16,213	524.11	1,182.3	3,273.5
	Tch	45,723	23,094.42	99.5%	960	16,744	535.62	1,220.3	3,634.5
Inst16	Ws	48,455	23,780.6	99.6%	910	18,414	445.7	935.5	3,075.4
	Tch	47,629	24,652.82	99.7%	930	19,651	445.12	695.5	2,931.2
Inst17	Ws	49,542	24,018.36	99.6%	970	18,654	512.36	1,060	2,822
	Tch	47,396	25,406.16	99.5%	1,010	20,052	500.36	1,387.5	2,456.3
Inst18	Ws	44,182	24,638.64	99.2%	920	18,234	507.44	2,074	2,903.2
	Tch	42,338	25,295.37	99.2%	940	18,443	510.17	2,078.5	3,323.7
Inst19	Ws	298,900	89,864	99.2%	2,080	61,492	2,484	7,075	16,733
	Tch	298,720	91,126	99.2%	2,090	61,322	2,729	7,325	17,660
Inst20	Ws	285,550	70,478.4	99.5%	2,320	49,744	1,966.9	3,397.5	13,050
	Tch	269,290	71,256.5	99.4%	2,280	49,546	1,896	4,000.5	13,534
Inst21	Ws	286,890	78,345.4	99.0%	2,390	50,948	2,421.9	8,142.5	14,443
	Tch	291,930	85,922	99.2%	2,400	59,295	2,375	6,973	14,879
Inst22	Ws	299,800	82,563.4	99.1%	2,410	54,838	2,400.4	7,333	15,582
	Tch	294,290	79,864.5	98.8%	2,300	50,319	2,619	9,311.5	15,315
Inst23	Ws	278,520	85,940.2	99.3%	2,290	60,830	2,156.7	5,786.5	14,877
	Tch	273,170	89,862.7	99.4%	2,310	64,723	2,257.2	5,102.5	15,470
Inst24	Ws	281,200	76,284.1	99.6%	2,290	54,542	2,116.1	2,920	14,416
	Tch	271,460	79,007.3	99.5%	2,380	56,149	2,215.3	3,935	14,328

5.3 Summary

In this chapter, we studied interval type-2 fuzzy sets and two generic single point based heuristic optimisation algorithms, each using a different scalarisation method to solve a supply chain management problem considering both supplier selection and inventory planning.

In the first stage, it is observed that there is a relationship between the shape of membership functions reflecting the uncertainty in linguistic terms. Also based on results, interval type-2 fuzzy sets are capable to cope with linguistic uncertainty better than type-1 sets. Thus, a supply chain management problem is solved using interval type-2 fuzzy sets and the results are carried out in the next stage. In the second stage, this study illustrated the multi-objective nature of

the problem where tested on a simple single problem instance by the proposed approaches. We provided an approach which is capable of capturing the trade-off between risk and cost via scalarisation of both objectives. This gives flexibility to the decision makers to choose from a set of trade-off solutions for supply chain management. Also, MOSA performed well using a learning move operator with an adaptive parameter control. Moreover, parameter settings of MOSA does influence its performance. After parameter tuning, we used each approach at its best performance and tested them on 24 problem instances considering three performance metrics (hypervolume, generational distance and inverted generational distance) using the two-objective model on a set of real world and randomly generated problem instances. The statistical analysis of the results indicate that weighted sum scalarisation performs better than Tchebycheff in most instances. In addition, the comparison of $MOSA_d$ and $MOSA_t$ shows that parameter settings have an essential influence on the performance of the SA algorithm.

Although MOSA performed reasonably well on the instances used, in the next chapter, we will extend this work and investigate three well-known MOEAs to solve the problem with an attempt to detect the best performing approach.

Chapter 6

Multi-objective Evolutionary Algorithms for the Integrated Problem

In the previous chapter, a generic local search meta-heuristic is used to solve the problem aggregating two objectives subject to several constraints as described in Chapter 4. In this chapter, four population based meta-heuristic algorithms are used to tackle the problem with an attempt to detect the best performing approach. Firstly, the problem is formulated as a multi-objective problem and we investigate the performance of three MOEAs, NSGA-II, SPEA2 and IBEA. Secondly, the total cost is the sum of 5 different costs, production cost, holding cost, batch cost, transportation cost and stock-out cost, related to production processes. We treat each cost component as a separate objective and solve the integrated supply chain management problem as a many objective problem using NSGA-III. Finally, performance comparison of these algorithms is provided.

6.1 Multi-objective Evolutionary Algorithms

As the goal in multi-objective optimisation based on a metaheuristic is to obtain a set of trade-off solutions at the end of the search process for the decision makers, population based search techniques, in particular multi-objective evolutionary

algorithms are naturally preferred (see Section 3.5). A variety of MOEAs with differing algorithmic components, such as diversity maintenance, replacement, have been previously proposed and more can be found in Konak et al. [2006], Zitzler et al. [2000], Zitzler and Thiele [1999]. This work considers three particular MOEAs: NSGA-II, SPEA2, IBEA (detailed in Section 3.5).

6.1.1 Experimental Setup

We implemented a fuzzy model and run the stage one for a particular supplier to rank suppliers as explained in Chapter 5. Then this information is fed into the multi-objective evolutionary algorithms to solve the integrated problem of supplier selection and inventory planning.

The Jmetal suite [Durillo et al., 2010, Durillo and Nebro, 2011b] is used to run all experiments with the MOEAs. Each trial is repeated for 30 times during the experiments, where each run yields a set of trade-off solutions. A run *terminates* whenever 5000 iterations/generations are exceeded.

A real-valued *chromosome representation* is used to represent a potential inventory plan. This plan is encoded into a 4 dimensional array. Each dimension represents the source node, destination node, component/product and time period, successively. Each array entry contains a value $\in [0,1]$, representing the ratio of raw material or goods added to the inventory with respect to the full capacity of the chosen product at a given source and destination node within a specific time period. For instance, if the value of *currentPlan*[3,1,4,2] is 0.5, this would demonstrate that in period 2, source node 3 is holding 50% of its capacity of product 4 for destination node 1. The holding can never exceed 100% within any time period with the proposed encoding. In addition to this, order amount is decided in certain increments starting from a specified minimum value. For example, assuming an increment of 100 units and a minimum order of 300 units for a particular product at a particular node, the orders are restricted to the increments of 100 starting from 300 (e.g. 300, 400, 500 etc.).

The *initial population* is generated randomly. A binary tournament selection is employed to create a offspring population. Simulated Binary *Crossover* (SBX) and Polynomial *Mutation* operator are used by all MOEAs. The common parameters

of SPEA2 and IBEA include *population size* (P), *crossover probability* (P_c), *distribution index for crossover* (D_c), *distribution index for mutation* (D_m) and *archive size* (A). NSGA-II has the same algorithmic control parameters, excluding the archive size. Crossover and mutation operators involve crossover and mutation probabilities (rates), and crossover and mutation distribution index. Crossover and mutation probabilities are utilised to maintain the frequency of operations. Crossover and mutation distribution index define a probability distribution of offspring solutions to control the spread of offspring solutions based on some measure of distance among the parent solutions. Larger values support to produce “near parent” solutions while smaller values provide a more diverse search [Zeng et al., 2010]. All the algorithmic control parameters are tuned for each particular algorithm.

6.1.2 Results from Ranking of Suppliers

Although the overall two-stage approach operates in an integrated manner, we report the results from stage one separately for the ease of empirical analysis. The stage one fuzzy method for ranking of relevant suppliers for each instance is executed as explained in Section 5.1, yielding an output of score, ranking and risk for each supplier as shown in Table 5.14.

6.1.3 Parameter Tuning of NSGA-II, SPEA2 and IBEA

The Taguchi orthogonal arrays [Taguchi and Yokoyama, 1993] as a design of experiments method is used for parameter tuning of each MOEA for improved performance. We investigated four control parameters for NSGA-II with the following settings: $P \in \{25, 50, 100, 200\}$, $P_c \in \{0.6, 0.7, 0.8, 0.9\}$, D_c and $D_m \in \{5, 10, 15, 20\}$, and five control parameters for SPEA2 and IBEA with the addition of $A \in \{25, 50, 100, 200\}$. The best parameter configuration is determined based on the L^{16} Taguchi orthogonal arrays design as shown in Table 6.1 where the design consists of up to 5 parameters at 4 levels each. There are 16 rows [Taguchi et al., 2005].

We have arbitrarily chosen three instances of different sizes: {Inst1 as a small size of instance, Inst13 as a medium size of instance, Inst19 as a large size of instance} and each of the sixteen parameter settings as provided in Table 6.2 is

TABLE 6.1: L^{16} Taguchi orthogonal arrays design

Experiment number	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5
1	1	1	1	1	1
2	1	2	2	2	2
3	1	3	3	3	3
4	1	4	4	4	4
5	2	1	2	3	4
6	2	2	1	4	3
7	2	3	4	1	2
8	2	4	3	2	1
9	3	1	3	4	2
10	3	2	4	3	1
11	3	3	1	2	4
12	3	4	2	1	3
13	4	1	4	2	3
14	4	2	3	1	4
15	4	3	2	4	1
16	4	4	1	3	2

tested using each MOEA applying to the selected problem instances as required by the L^{16} Taguchi orthogonal array design. In order to assess the performance of each setting for an MOEA, mean rank (per run), which is obtained by ranking each setting with respect to the hypervolume of the pareto set for each run and then averaging the ranks of a setting over all runs and problem instances. A lower value indicates a better performance. As an example in Table 6.2 the parameter settings and average rank of three MOEAs are shown.

A mean rank for a particular parameter value setting to indicate its main effect is computed by taking the average of the rank of all runs with that setting on all instances. For example, let us consider the mean effect of a population size 25 for NSGA-II computed as $(14.87 + 12.14 + 12.27 + 11.00)/4 = 12.60$. This parameter setting corresponds to the first 4 experiments in Table 6.2. Figure 6.1 provides the main effects plot indicating the performance of each parameter value setting. The best configuration for NSGA-II is attained as 200 for P , 0.9 for P_c , 20 for D_c and 10 for D_m . Moreover, ANOVA is utilised to analyse the contribution of each parameter setting on the performance of MOEAs. Table 6.3 summarises the results. The population size as well as distribution index for crossover, distribution index for mutation, and population size parameter settings have significant

TABLE 6.2: Average rank for three multi-objective algorithms, with a particular parameter configuration based on the L^{16} Taguchi orthogonal array

Experiment number	P	P_c	D_c	D_m	A	Average Rank		
						NSGA-II	SPEA2	IBEA
1	25	0.6	5	5	25	14.87	15.8	15.0
2	25	0.7	10	10	50	12.14	11.3	2.6
3	25	0.8	15	15	100	12.27	6.0	10.5
4	25	0.9	20	20	200	11.00	4.8	5.0
5	50	0.6	10	15	200	10.80	13.6	6.4
6	50	0.7	5	20	100	11.84	10.2	8.5
7	50	0.8	20	5	50	8.59	6.3	5.0
8	50	0.9	15	10	25	7.86	5.8	11.6
9	100	0.6	15	20	50	6.71	11.1	7.6
10	100	0.7	20	15	25	4.74	11.4	13.7
11	100	0.8	5	10	200	9.52	6.5	4.8
12	100	0.9	10	5	100	6.09	2.8	11.8
13	200	0.6	20	10	100	3.79	10.1	9.2
14	200	0.7	15	5	200	4.43	9.9	8.0
15	200	0.8	10	20	25	4.53	7.1	12.8
16	200	0.9	5	15	50	6.80	3.3	3.4

TABLE 6.3: ANOVA test results for dismissing the contribution of each parameter for MOEAs in terms of percent contribution

MOEAs	P	P_c	D_c	D_m	A	Residual	Total
NSGA-II	79.28	1.64	17.93	0.12	-	1.03	100%
SPEA2	8.07	4.46	0.17	86.46	0.84	0	100%
IBEA	84.41	2.65	7.82	2.85	2.27	0	100%

TABLE 6.4: Tuned Parameters of three MOEAs

MOEAs	P	P_c	D_c	D_m	A
NSGA-II	200	0.9	20	10	-
SPEA2	50	0.8	15	20	200
IBEA	200	0.9	20	20	50

contribution within a confidence level of 95% on the performance of NSGA-II, SPEA2 and IBEA, respectively.

In the same manner, the parameters of the other two MOEAs are tuned using the Taguchi method and the results achieved are depicted in Table 6.4. Based on the hypervolume values, the tuned three MOEAs outperform all the other settings with the highest hypervolume values for each instance. This parameter setting is accepted as the best parameter setting for each algorithm.

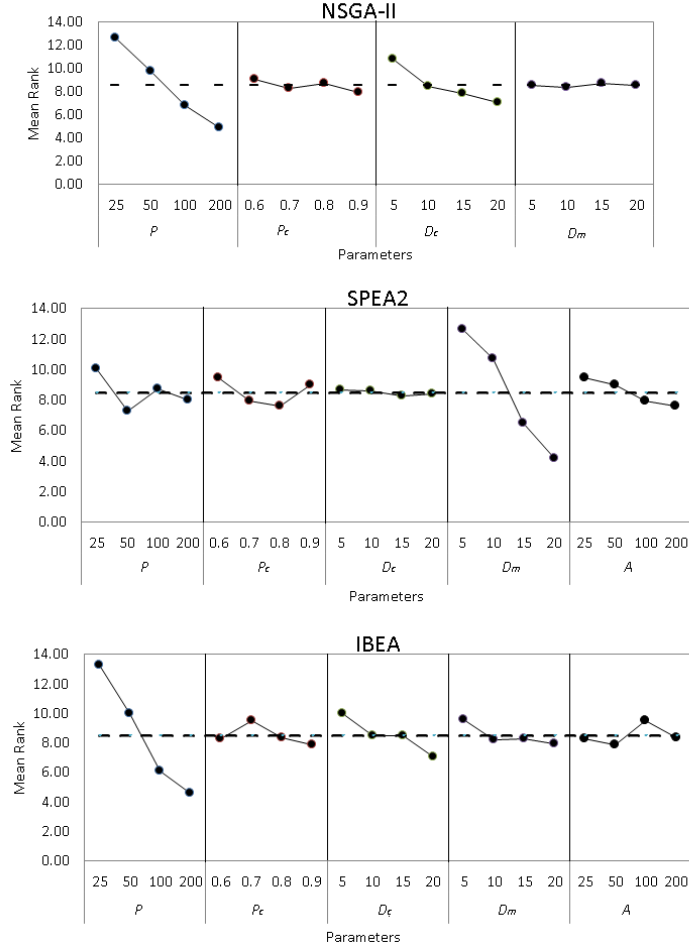


FIGURE 6.1: Main effects plot with mean rank values in three multi-objective algorithms

6.1.4 Experimental Results

NSGA-II, SPEA2 and IBEA are applied to the 24 problem instances of the stage two to discover high quality inventory plans based on the described multi-objective model (See Chapter 4). These MOEAs provide flexibility for the decision makers by enabling them to choose a solution from a set of ‘equal’ quality solutions reflecting different levels of trade-off between the total supplier risk and cost of the supply chain operation.

Table 6.5 provides the performance comparison of all MOEAs across all problem instances based on hypervolume, generational distance and inverted generational distance. We have further performed statistical analysis of results based on one

tailed Wilcoxon signed-rank test. In summary, NSGA-II is the best performing multi-objective algorithm on average across all instances in terms of all metrics as shown in Table 6.5 and we use the following notation in the table.

$$\begin{aligned} H_0 : & \quad P_{NSGA-II} = P_O \\ H_A : & \quad P_{NSGA-II} < P_O \end{aligned}$$

where H_0 represents null hypothesis which asserts that the difference between median values of the two methods is zero and H_A represents alternative hypothesis that the median value of NSGA-II is less than the median of other multi-objective algorithm O . The confidence level (significance level) for the non-parametric test of Wilcoxon signed-rank is set to 95% (p-value under 0.05). The following notations are used in Table 5.19. Let us consider two algorithms; NSGA-II versus O , $>$ ($<$) indicates that NSGA-II (O) is better than O (NSGA-II) and this performance difference is statistically significant within a confidence interval of 95% and $NSGA-II \geq S(NSGA-II) \leq O$ denotes that NSGA-II (O) performs slightly better on average than O (NSGA-II) with no statistical significance.

NSGA-II performs significantly better than SPEA2 and IBEA based on hypervolume on 24 and 15 out of 24 instances, respectively. As for the remaining instances, IBEA is slightly better than NSGA-II for nine instances. When the size of problem is enlarged, NSGA-II still provides significantly better hypervolume values than the other two algorithms (shown as Figure 6.2).

Secondly, with respect to the generational distance, NSGA-II performs significantly better than SPEA2 and IBEA on 13 and 6 out of 24 instances, respectively. For seven instances, SPEA2 is slightly better than NSGA-II. IBEA is slightly better than NSGA-II over 15 instances achieving lower values of generational distance. In addition, SPEA2 provides significantly better generational distance on four instances whereas IBEA performs significantly better than NSGA-II over three instances. NSGA-II provides significantly better results as the size of the problem grows.

Finally, considering inverted generational distance, NSGA-II outperforms SPEA2 and IBEA for 223 and 20 out of 24 instances, respectively. This performance

TABLE 6.5: Performance comparison of NSGA-II, SPEA2 and IBEA for the two-objective supply chain management problem based on three metrics

Inst.		HyperVolume			Generational Distance			Inverted Generational Distance		
		NSGA-II	SPEA2	IBEA	NSGA-II	SPEA2	IBEA	NSGA-II	SPEA2	IBEA
Inst1	Mean	0.7640	> 0.7495	≤ 0.7646	153,991.3	≤ 139,840.9	≤ 152,547.2	18,434,666.7	> 20,393,033.3	> 18,464,566.7
	Std.	0.0068	0.0065	0.0071	42,737.6	31,936.8	42,269.8	824,768.8	952,529.4	994,784.4
Inst2	Mean	0.7527	> 0.7433	> 0.7526	118,211.6	< 102,516.7	≤ 106,338.1	3,099,630.0	> 3,423,510.0	> 3,197,753.3
	Std.	0.0067	0.0062	0.0056	37,823.6	42,539.4	41,053.5	327,324.2	457,428.2	437,830.9
Inst3	Mean	0.6795	> 0.6688	≤ 0.6823	101,073.1	> 107,795.2	> 105,283.5	8,553,806.7	> 9,142,340.0	≤ 8,466,946.7
	Std.	0.0097	0.0086	0.0066	37,281.4	50,138.6	40,734.2	449,454.3	436,251.7	360,211.3
Inst4	Mean	0.7119	> 0.6985	> 0.7107	104,970.8	≤ 90,953.0	≤ 92,559.0	4,662,183.3	> 5,543,083.3	> 4,838,830.0
	Std.	0.0062	0.0067	0.0059	37,193.0	35,798.6	30,644.7	658,605.2	830,206.9	895,956.0
Inst5	Mean	0.7386	> 0.7259	≤ 0.7425	133,082.9	≤ 124,107.6	≤ 128,042.2	2,149,720.0	> 2,345,703.3	≤ 2,115,850.0
	Std.	0.0057	0.0056	0.0051	53,555.0	37,093.7	44,638.5	257,496.4	200,865.4	208,958.1
Inst6	Mean	0.7229	> 0.7141	≤ 0.7256	116,663.4	> 129,015.5	> 120,733.0	3,902,323.3	> 3,980,433.3	≤ 3,604,680.0
	Std.	0.0072	0.0075	0.0083	38,985.2	41,910.7	42,407.7	799,385.5	774,240.2	611,537.1
Inst7	Mean	0.7946	> 0.7807	> 0.7938	320,002.7	> 316,970.7	≤ 277,630.4	1,912,743.3	> 2,144,826.7	> 1,932,760.0
	Std.	0.0058	0.0072	0.0044	77,033.0	74,612.7	85,617.1	95,724.9	91,234.2	77,638.6
Inst8	Mean	0.7964	> 0.7891	≤ 0.7973	301,241.0	≤ 300,575.0	≤ 271,020.7	2,391,006.7	> 2,563,963.3	> 2,412,020.0
	Std.	0.0026	0.0035	0.0029	95,836.2	72,718.8	66,977.2	98,423.9	106,250.1	93,791.7
Inst9	Mean	0.8170	> 0.8102	≤ 0.8174	268,958.7	≤ 252,043.0	≤ 253,180.0	847,869.3	> 903,499.0	> 848,681.7
	Std.	0.0045	0.0044	0.0041	65,784.5	60,263.5	65,029.1	26,366.3	28,412.0	21,764.5
Inst10	Mean	0.7892	> 0.7820	≤ 0.7900	301,086.7	< 257,948.3	< 252,085.3	1,608,503.3	> 1,740,453.3	> 1,611,513.3
	Std.	0.0037	0.0038	0.0042	78,765.2	84,204.9	51,779.5	46,969.5	55,348.0	40,359.0
Inst11	Mean	0.7867	> 0.7808	> 0.7861	227,870.0	> 302,879.7	> 278,308.0	2,595,276.7	> 2,871,066.7	> 2,644,343.3
	Std.	0.0063	0.0051	0.0037	71,414.3	83,599.4	82,282.2	108,674.2	97,127.0	104,287.0
Inst12	Mean	0.8044	> 0.7967	> 0.8020	283,313.3	< 246,669.3	< 256,049.7	1,284,280.0	> 1,370,263.3	> 1,311,563.3
	Std.	0.0035	0.0034	0.0035	70,058.8	78,198.1	83,193.9	51,080.9	49,539.8	65,895.0
Inst13	Mean	0.6999	> 0.6841	> 0.6970	521,598.3	> 533,727.7	≤ 472,593.7	10,467,286.7	> 11,059,466.7	> 10,631,306.7
	Std.	0.0059	0.0074	0.0092	184,974.7	216,917.7	155,807.4	533,005.9	410,021.3	506,606.6
Inst14	Mean	0.6987	> 0.6891	≤ 0.6991	389,654.3	> 432,197.3	≤ 375,909.3	2,481,806.7	> 2,632,243.3	> 2,491,380.0
	Std.	0.0047	0.0054	0.0058	148,033.4	103,388.7	115,054.6	67,294.9	63,263.0	69,367.2
Inst15	Mean	0.7187	> 0.7071	> 0.7162	508,622.0	≤ 467,514.0	≤ 471,610.0	2,676,630.0	> 2,832,610.0	> 2,717,810.0
	Std.	0.0036	0.0054	0.0058	180,539.9	170,014.1	170,862.9	59,602.4	62,423.3	43,087.8
Inst16	Mean	0.6991	> 0.6862	> 0.6987	395,404.3	> 423,801.3	≤ 367,221.3	2,505,200.0	> 2,626,360.0	> 2,513,333.3
	Std.	0.0065	0.0050	0.0063	131,243.3	159,288.7	113,157.4	67,682.4	48,323.5	74,552.0
Inst17	Mean	0.6936	> 0.6820	≤ 0.6941	310,075.0	> 310,731.3	> 354,250.7	5,022,010.0	> 5,231,876.7	> 5,046,300.0
	Std.	0.0059	0.0046	0.0041	128,397.4	137,561.9	132,238.3	238,738.9	172,161.9	157,058.3
Inst18	Mean	0.7267	> 0.7155	> 0.7246	398,581.7	> 448,228.0	≤ 395,568.3	2,059,116.7	< 448,228.0	< 395,568.3
	Std.	0.0056	0.0057	0.0051	135,965.9	122,630.9	114,389.3	42,911.6	122,630.9	114,389.3
Inst19	Mean	0.7006	> 0.6897	> 0.6993	4,053,156.7	≤ 3,871,516.7	> 4,180,700.0	57,760,266.7	> 60,028,733.3	> 58,630,166.7
	Std.	0.0052	0.0047	0.0049	1,100,606.1	1,100,606.1	1,218,248.3	1,092,811.7	1,191,522.5	969,366.1
Inst20	Mean	0.7308	> 0.7213	> 0.7269	2,906,855.7	> 3,267,640.0	> 3,465,779.3	41,542,666.7	> 43,010,466.7	> 42,054,033.3
	Std.	0.0044	0.0047	0.0050	972,084.7	1,338,053.6	1,367,927.4	569,996.9	796,878.4	697,617.6
Inst21	Mean	0.7066	> 0.6983	> 0.7047	3,827,270.0	> 4,300,943.3	≤ 3,777,916.7	56,721,333.3	> 58,427,233.3	> 57,092,600.0
	Std.	0.0045	0.0049	0.0037	988,830.8	1,365,882.9	1,276,234.0	957,691.2	662,402.5	619,656.8
Inst22	Mean	0.7021	> 0.6921	> 0.7000	3,684,286.7	> 3,884,603.3	≤ 3,244,260.0	57,495,800.0	> 59,183,833.3	> 57,571,333.3
	Std.	0.0055	0.0049	0.0044	1,199,193.4	1,234,025.5	912,513.5	783,634.1	995,375.3	1,023,030.5
Inst23	Mean	0.7274	> 0.7184	> 0.7260	4,289,596.7	< 3,499,073.3	< 3,270,576.7	51,158,666.7	> 52,838,233.3	> 51,678,100.0
	Std.	0.0036	0.0039	0.0041	1,422,592.3	1,039,882.9	998,372.4	1,181,358.4	938,460.8	788,069.5
Inst24	Mean	0.7317	> 0.7222	> 0.7299	3,695,396.7	> 3,854,173.3	≤ 3,479,921.7	49,376,333.3	> 51,166,433.3	> 49,669,633.3
	Std.	0.0036	0.0034	0.0034	1,077,348.8	1,178,044.1	1,284,694.8	841,295.5	743,742.8	1,030,337.1

variation is statistically significant. IBEA is slightly better than NSGA-II on three instances. For Inst18, SPEA2 and IBEA are significantly better than NSGA-II. NSGA-II produces better results when compared to SPEA2 and IBEA across the last relatively large six instances formed by the $5 \times 5 \times 5$ model.

As a sample, we plotted the pareto-front achieved by each MOEA on one small and one relatively large arbitrarily chosen instances of Inst1 (Figure 6.2(a)) and Inst19 (Figure 6.2(b)). As it can be observed from Figure 6.2(a), NSGA-II and SPEA2 produce a wider spread of solutions on the pareto-front when compared to IBEA. However, when the size of problem gets larger, the spread of pareto-front achieved by NSGA-II is better than the others as illustrated in Figure 6.2(b).

The multi-objective approaches provide means to the decision makers to select a

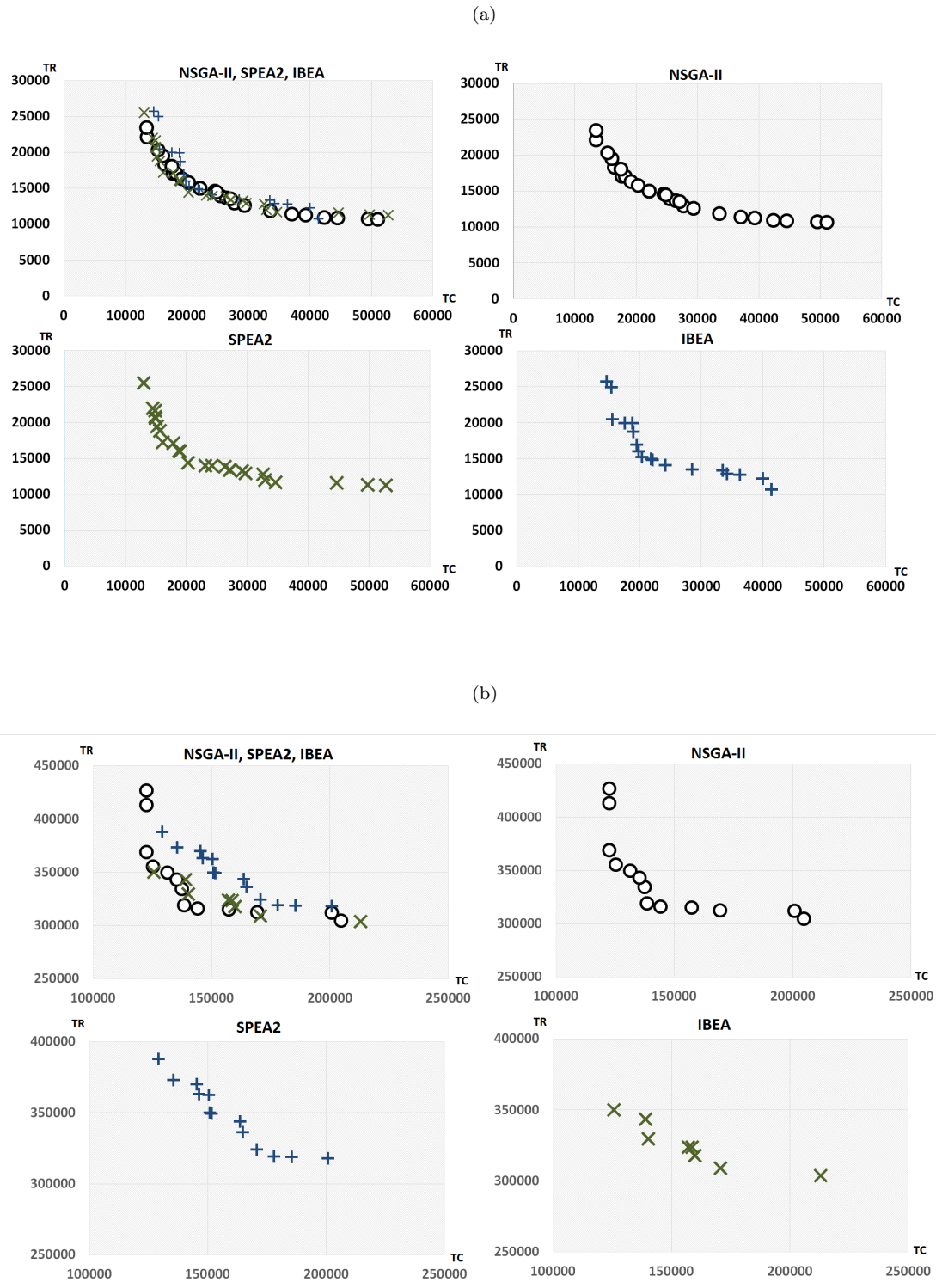


FIGURE 6.2: Final pareto-front plots for NSGA-II, SPEA2 and IBEA based on TR vs TC from a sample run on (a) Inst1 and (b) Inst19.

solution among multiple trade-off solutions. A common way of (automatically) reducing all solutions into a ‘preferable’ reasonable single solution is detecting the solution at the *knee point* on the pareto-front. We have used the method presented in [Bechikh et al., 2010] to obtain the a single solution based on the knee point for all problem instances. The total cost (TC) and risk (TR) objectives computed for each such solution to each instance is summarised in Table 6.6 which also provides all constituent costs considered.

We have observed that all three MOEAs achieved knee solutions for the majority of the instances satisfying at least 91% of the customer demand as seen in Table 6.6. In our model, we have merged all cost related entities into a single quantity. The results indicate that there could be a trade-off in between different constituent costs. For example, Table 6.6 shows that the SPEA2 solution is better than the IBEA solution in terms of both total cost and risk. However, service levels and holding costs are the same and more importantly, the SPEA2 solution provides a better batch cost, transportation cost and stock-out cost while the IBEA solution provides a relatively better production cost in return worsening the cost for the remaining items. Considering Inst23, although in the overall, the IBEA solution is better than (dominates) the SPEA2 solution, however again the SPEA2 solution is better in terms of the production cost.

In summary, in this study, we investigated the performance of the overall multi-objective approach using three well known MOEAs, namely NSGA-II, SPEA2 and IBEA for solving the integrated problem. Although, there are several studies on multi-objective supply chain management [Liao et al., 2011, Shankar et al., 2013, Zhang et al., 2016], to the authors’ knowledge, this is one of the first studies in which the integrated problem of supplier selection and inventory planning has been investigated as a multi-objective problem.

Firstly, parameter setting of an MOEA does influence its performance [Liao et al., 2011, Shankar et al., 2013, Zhang et al., 2016]. After parameter tuning, we used each MOEA at its best performance and applied to 24 problem instances. The empirical results indicate the overall success of NSGA-II interacting well with IT2FSs for SCM. All MOEAs achieved high quality trade-off solutions satisfying the customer demand almost fully in the majority of cases.

TABLE 6.6: Comparison of three MOEAs for two-objective supply chain management problem in terms of different cost results which compose the total cost

Inst.		Total Risk	Total Cost	Service Level	Batch Cost	Production Cost	Transportation Cost	Stockout Costs	Holding Cost
Inst1	NSGA-II	18,153.1	10,964.0	97.26%	520.0	9375.0	251.0	518.0	300.0
	SPEA-II	19,528.4	10,106.0	95.84%	470.0	8305.0	276.0	635.0	420.0
	IBEA	16,777.9	9,327.0	93.57%	490.0	7485.0	251.0	501.0	600.0
Inst2	NSGA-II	17,220.8	9,648.2	98.24%	450.0	6,571.8	221.3	2,235.1	170.0
	SPEA2	15,023.0	10,887.0	98.44%	400.0	7,616.1	195.0	2,505.9	170.0
	IBEA	15,282.9	10,907.1	98.44%	420.0	7,544.9	197.2	2,575.0	170.0
Inst3	NSGA-II	14,424.5	12,888.2	93.37%	430.0	9,173.9	192.0	2,237.3	855.0
	SPEA-II	15,670.2	13,078.5	92.96%	400.0	9,348.9	230.1	2,178.5	921.0
	IBEA	14,761.9	12,614.7	91.83%	390.0	8,851.7	186.3	2,155.8	1,031.0
Inst4	NSGA-II	19,595.1	9,844.7	96.37%	470.0	7,490.6	212.1	1,314.5	357.5
	SPEA-II	18,836.0	10,179.7	94.91%	460.0	7,684.0	215.4	1,302.3	518.1
	IBEA	19,738.8	9,995.67	96.42%	480.0	7,547.7	214.5	1,396.0	375.5
Inst5	NSGA-II	18,930.4	10,049.2	99.68%	430.0	7,602.9	214.3	1,769.5	32.5
	SPEA-II	18,117.0	12,011.4	99.23%	410.0	9,415.7	220.4	1,872.7	92.5
	IBEA	19,611.2	10,917.1	100.00%	460.0	8,242.7	231.9	1,982.4	0
Inst6	NSGA-II	17,608.1	9,392.2	95.66%	420.0	6,329.5	201.3	2,034.0	407.5
	SPEA-II	20,121.5	9,579.1	95.75%	390.0	6,387.9	211.2	2,182.5	407.5
	IBEA	19,342.0	9,258.2	95.60%	420.0	6,093.9	218.6	2,118.2	407.5
Inst7	NSGA-II	30,116.3	21,677.5	95.50%	700.0	17,525.0	443.0	2,034.5	975.0
	SPEA2	34,691.0	24,153.5	96.89%	720.0	19,930.0	454.0	2,299.5	750.0
	IBEA	37,171.0	22,166.5	98.42%	690.0	18,850.0	405.0	1,871.5	350.0
Inst8	NSGA-II	31,009.8	19,507.8	95.29%	790.0	15,975.3	329.0	1,495.4	918.2
	SPEA-II	31,204.9	21,199.4	95.40%	780.0	17,296.1	355.1	1,792.2	975.9
	IBEA	30,823.7	21,021.6	95.78%	790.0	17,161.9	371.1	1,812.5	886.1
Inst9	NSGA-II	29,509.5	18,147.2	97.64%	830.0	14,855.8	368.2	1,664.3	428.9
	SPEA-II	32,241.4	18,565.0	97.97%	780.0	15,209.3	415.3	1,784.4	376.0
	IBEA	31,321.4	18,396.0	98.28%	810.0	15,164.4	393.8	1,711.9	316.0
Inst10	NSGA-II	34,382.4	21,466.9	96.73%	800.0	17,938.8	365.2	1,659.9	703.0
	SPEA-II	34,118.5	20,858.6	97.52%	750.0	17,912.1	385.2	1,293.8	517.6
	IBEA	33,740.0	20,543.1	96.68%	760.0	17,077.0	352.1	1,672.5	681.5
Inst11	NSGA-II	36,838.4	19,783.2	97.90%	820.0	17,128.3	390.3	1,028.7	415.9
	SPEA-II	39,090.7	21,015.5	98.10%	860.0	18,138.3	396.9	1,221.7	398.5
	IBEA	34,277.4	20,688.0	96.71%	790.0	17,694.6	405.0	1,117.3	681.0
Inst12	NSGA-II	34,269.2	23,335.5	99.92%	820.0	20,699.3	362.2	1,435.5	18.5
	SPEA-II	36,621.2	20,809.7	98.93%	820.0	17,842.3	395.5	1,529.4	222.5
	IBEA	33,172.0	20,707.6	98.66%	780.0	17,627.3	414.1	1,608.2	278.0
Inst13	NSGA-II	57,172.0	26,937.5	98.61%	900.0	21,570.0	552.0	3,540.5	375.0
	SPEA2	57,558.4	25,355.5	98.03%	910.0	20,365.0	571.0	3,009.5	500.0
	IBEA	58,166.6	24,311.0	98.15%	850.0	19,400.0	493.0	3,118.0	450.0
Inst14	NSGA-II	51,343.6	23,830.7	95.28%	980.0	18,106.0	482.8	3,136.7	1,125.2
	SPEA-II	55,065.7	23,112.7	94.53%	1,030.0	17,109.6	532.1	3,176.7	1,264.4
	IBEA	54,897.9	23,589.4	96.45%	1,020.0	18,341.4	467.6	2,922.7	837.7
Inst15	NSGA-II	54,569.8	22,645.6	96.57%	1,050.0	17,216.8	547.8	3,054.1	776.9
	SPEA-II	52,777.0	23,810.8	95.11%	990.0	17,628.6	600.7	3,426.5	1,165.0
	IBEA	52,767.3	23,627.9	97.58%	1,020.0	18,254.1	527.3	3,255.4	571.0
Inst16	NSGA-II	55,254.2	23,285.6	96.41%	1,000.0	17,869.4	465.7	3,114.6	836.0
	SPEA-II	53,487.8	24,744.6	95.51%	980.0	19,064.6	489.4	3,099.2	1,111.5
	IBEA	53,513.1	26,253.4	97.93%	990.0	21,260.3	489.0	2,971.7	542.5
Inst17	NSGA-II	54,208.6	24,172.8	97.72%	1,050.0	19,730.3	484.7	2,357.4	550.4
	SPEA-II	60,551.0	25,190.3	98.67%	1,040.0	20,451.3	532.4	2,831.1	335.5
	IBEA	53,314.8	25,557.0	98.19%	1,030.0	20,848.4	511.1	2,704.0	463.5
Inst18	NSGA-II	45,759.5	24,728.6	97.38%	970.0	19,997.8	410.2	2,703.7	647.0
	SPEA-II	46,601.5	24,764.7	96.20%	1,020.0	19,483.9	480.4	2,839.3	941.1
	IBEA	49,917.0	26,563.6	98.55%	1,000.0	21,930.4	439.3	2,809.9	384.0
Inst19	NSGA-II	311,927.9	94,565.5	95.64%	2,130.0	70,468.0	2,249.0	15,593.5	4,125.0
	SPEA-II	295,378.6	93,315.5	93.38%	2,170.0	65,486.0	2,529.0	16,955.5	6,175.0
	IBEA	320,600.3	91,3753.0	95.02%	2,210.0	66,182.0	2,390.0	16,043.0	4,550.0
Inst20	NSGA-II	324,124.7	82,541.2	96.89%	2,450.0	59,478.2	2,411.9	14,750.0	3,451.0
	SPEA-II	320,970.5	82,915.1	94.73%	2,420.0	58,462.3	2,403.4	14,861.4	4,768.0
	IBEA	330,580.6	76,970.3	95.22%	2,440.0	54,655.7	2,319.4	13,934.2	3,621.0
Inst21	NSGA-II	324,124.7	82,541.2	95.82%	2,450.0	59,478.2	2,411.9	14,750.0	3,451.0
	SPEA-II	320,970.5	82,915.1	94.25%	2,440.0	56,262.9	2,347.1	14,646.9	4,662.5
	IBEA	330,580.6	76,970.3	95.06%	2,440.0	55,970.4	2,279.3	14,383.4	3,897.5
Inst22	NSGA-II	329,405.7	81,790.2	93.77%	2,540.0	57,297.1	2,336.9	14,524.2	5,092.0
	SPEA-II	331,145.3	80,359.4	94.20%	2,440.0	56,262.9	2,347.1	14,646.9	4,662.5
	IBEA	327,571.3	78,970.5	95.06%	2,440.0	55,970.4	2,279.3	14,383.4	3,897.5
Inst23	NSGA-II	293,495.9	86,074.0	96.92%	2,300.0	64,535.2	2,336.9	14,540.7	2,651.0
	SPEA2	301,443.0	88,673.1	96.12%	2,310.0	65,358.0	2,210.0	15,355.5	3,439.0
	IBEA	299,561.5	88,115.6	97.88%	2,370.0	66,971.1	2,024.9	14,878.6	1,871.0
Inst24	NSGA-II	293,673.9	73,994.2	96.29%	2380.0	53,026.2	2107.7	13,732.3	2,748.0
	SPEA-II	297,539.6	76,601.4	95.38%	2,360.0	53,991.2	2,196.5	14,515.2	3,538.5
	IBEA	294,919.5	72,489.1	96.75%	2,460.0	52,192.9	2,096.9	13,381.8	2,357.5

The empirical results indicate that there could be even more conflicting objectives which can be considered in the solution model and then simultaneously optimised. Although MOEAs performed reasonably well in this study for the two-objective problem, this might not be the case when the number of objectives is increased. Recently, there has been a growing interest into many-objective (four or more

objectives) optimisation considering that existing MOEAs could struggle in solving such problems [Deb and Jain, 2014] requiring algorithmic improvement. For example, Deb and Jain [2014] developed NSGA-III as an extension to NSGA-II with significant changes in the selection operator to overcome these difficulties. In the next section, we study the trade-off between all contributing factors within the total cost and risk treating each factor as a separate objective and investigate the performance of a many-objective approach to the problem.

6.2 Many-objective Evolutionary Algorithm

Over the last decade, there has been a growing interest in many-objective optimisation algorithms. To solve many-objective optimisation problems, NSGA-III was proposed by Deb and Jain [2014] using an external guidance mechanism which provides diversity maintenance among solutions obtained. Throughout the optimization process, a systematic evaluation of individuals in the population is attempted with respect to the reference points defined either uniformly or preferentially based on information provided by decision makers [Seada and Deb, 2015].

From the previous section, we can see that there could be even more conflicting objectives in the two-objective supply chain management problem. In this section, we explore the possibility of considering conflicting objectives ignored in the two-objective problem and applying NSGA-III to solve this problem. The problem takes into account six objectives, i.e, total risk (Equation 4.8), production cost (Equation 4.5), holding cost (Equation 4.2), batch cost (Equation 4.4), transportation cost (Equation 4.3) and stock-out cost (Equation 4.6).

6.2.1 Experimental Setup

The results found in stage one is carried to the second stage to solve the integrated problem using NSGA-III. The Jmetal suite [Durillo et al., 2010, Durillo and Nebro, 2011b] is used to run all experiments with NSGA-III as a many-objective evolutionary algorithm. Each trial is repeated for 30 times during the experiments, where each run yields a set of trade-off solutions. A run *terminates* whenever 5000 iterations/generations are exceeded.

The same *chromosome representation* explained in the previous section is used to depict a potential inventory plan. The *initial population* is generated randomly. A binary tournament selection is employed to create a offspring population. Simulated Binary *Crossover* (SBX) and Polynomial *Mutation* operators are used. The parameters of NSGA-III include the *population size* (P), *crossover probability* (P_c), *distribution index for crossover* (D_m), *distribution index for mutation* (D_c) and *number of divisions* (N_d). NSGA-III has the same algorithmic control parameters as NSGA-II, including *number of divisions* (N_d). The number of divisions is utilised to determine how many reference points will be used in a reference line to maintain diversity in obtained solutions. All the algorithmic control parameters are tuned.

6.2.2 Parameter Tuning of NSGA-III

Parameters of NSGA-III are tuned using Taguchi orthogonal arrays [Taguchi and Yokoyama, 1993] as a design of experiments method. We investigated five control parameters for NSGA-III with the following potential settings: $P \in \{25, 50, 100, 200\}$, $P_c \in \{0.6, 0.7, 0.8, 0.9\}$, $D_c, D_m \in \{5, 10, 15, 20\}$ and $N_d \in \{3, 4, 5, 6\}$. The L^{16} Taguchi orthogonal arrays design is used to achieve the best parameter configuration. We followed the same way explained in Section 6.1.3 and obtained results as shown in Table 6.7.

The mean effect of each parameter is calculated in the same manner as explained in Section 6.1.3. Figure 6.3 provides the main effects plot indicating the performance of each parameter value setting. The best configuration for NSGA-III is attained as 200 for P , 0.7 for P_c , 10 for D_c , 5 for D_m and 6 for N_d . In addition, the contribution of each parameter setting on the performance of NSGA-III is analysed using ANOVA. Table 6.8 summarises the results. The number of divisions has a significant contribution within a confidence level of 95% on the performance of NSGA-III. In the same manner, hypervolume of three selected instances is used to assess the performance of each parameter setting configured based on the Taguchi method.

TABLE 6.7: Average rank of NSGA-III, with a particular parameter configuration based on the L^{16} Taguchi orthogonal array

Experiment number	P	P_c	D_c	D_m	N_d	Average Rank NSGA-III
1	25	0.6	5	5	3	13.4
2	25	0.7	10	10	4	8.9
3	25	0.8	15	15	5	6.4
4	25	0.9	20	20	6	5.2
5	50	0.6	10	15	6	4.2
6	50	0.7	5	20	5	6.8
7	50	0.8	20	5	4	9.4
8	50	0.9	15	10	3	13.5
9	100	0.6	15	20	4	11.4
10	100	0.7	20	15	3	14.0
11	100	0.8	5	10	6	4.0
12	100	0.9	10	5	5	5.9
13	200	0.6	20	10	5	5.6
14	200	0.7	15	5	6	3.0
15	200	0.8	10	20	3	14.2
16	200	0.9	5	15	4	10.2

TABLE 6.8: ANOVA test results for dismissing the contribution of each parameter for NSGA-III in terms of percent contribution

MOEAs	P	P_c	D_c	D_m	N_d	Error	Total
NSGA-III	0.28	0.28	0.11	2.51	96.81	-	100%

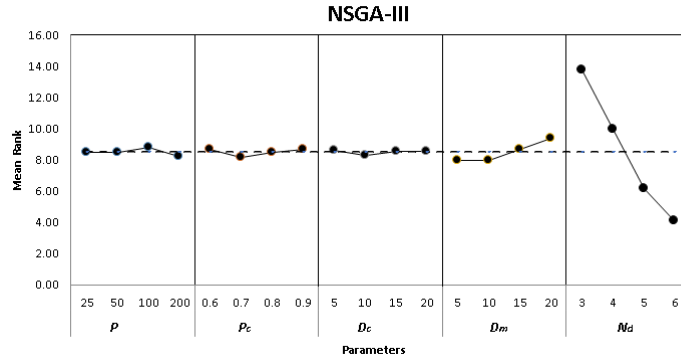


FIGURE 6.3: Main effects plot with mean rank values in NSGA-III

6.2.3 Experimental Results

Given six objective functions, the experimentation is conducted in exactly the same manner as when generating a Pareto set of solutions with two objectives as

explained in Section 6.1. NSGA-III is applied to 24 problem instances. The high-dimensional trade-off front in a many-objective evolutionary algorithm is analysed. For 24 instances, NSGA-III is superior than other methods considering values of hypervolume, generational distance and inverted generational distance achieved as shown in Table 6.10 (further comparison given in the following section).

Six objectives, total risk, total production cost, total holding cost, total batch cost, total transportation cost and total stock-out cost computed for each knee solution to each instance are summarised in Table 6.9. We have observed that NSGA-III achieved knee solutions for the majority of the instances with a low customer satisfaction rate. To find reasons for poor levels of customer service and to visualise relationship among objectives, Figure 6.4 depicts the pareto optimal set of Inst19 as an example in ‘parallel coordinates’ [Inselberg, 2009] generated by using NSGA-III. Each green line represents a pareto optimal solution and indicates change through objectives from one to another. The black line shows the knee solution for Inst19. Figure 6.5 displays the same data set but this time, there are two lines and each line represents a single solution in the pareto set of Inst19. In Figure 6.5, the solution 1 with a high risk has low production and stock-out costs while the solution 2 represents low risk scenario with high production and stock-out costs. Also, the high risk solution consists of relatively low holding cost.

Moreover, from the visualisation, we can observe that there is obvious inverse-correlation between the transportation cost and holding cost as seen in Figure 6.4. In this sense, decreasing the holding cost will increase transportation cost. However, the correlation between other objectives is not quite as obvious. Therefore, the relationship between risk and each cost is investigated in Figure 6.6. There is no specific relationship between risk and production cost, transportation cost and batch cost. It is obviously seen that there is negative relationship between risk and holding cost, and between risk and stock-out cost.

In summary, we have explored the performance of NSGA-III, in the six objective integrated SCM problem. Based on performance metrics, NSGA-III performed reasonably well in this study. However, the empirical results indicate that NSGA-III did not achieve high quality trade-off solutions satisfying at least 90% of the customer demand while each cost is considered as an objective. In addition, to investigate conflicting objectives, the parallel coordinates figures are used visualising the relationship among objectives. It is observed that there is an obvious

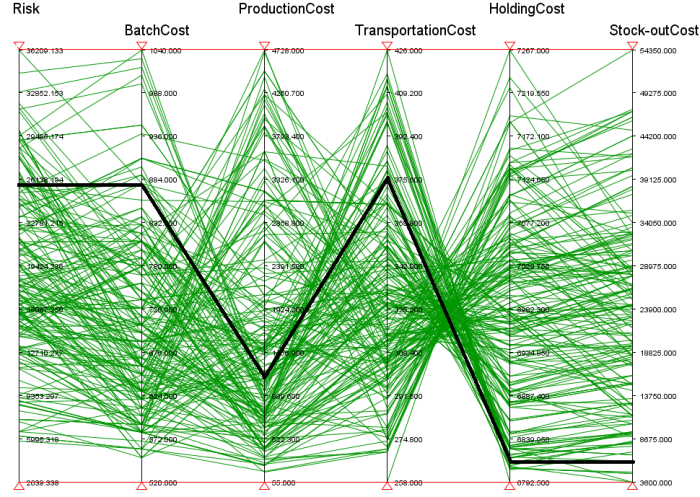


FIGURE 6.4: Results of NSGA-III in parallel coordinate for Inst19; green lines show trade-off solutions among objectives and black line indicates the knee solution.

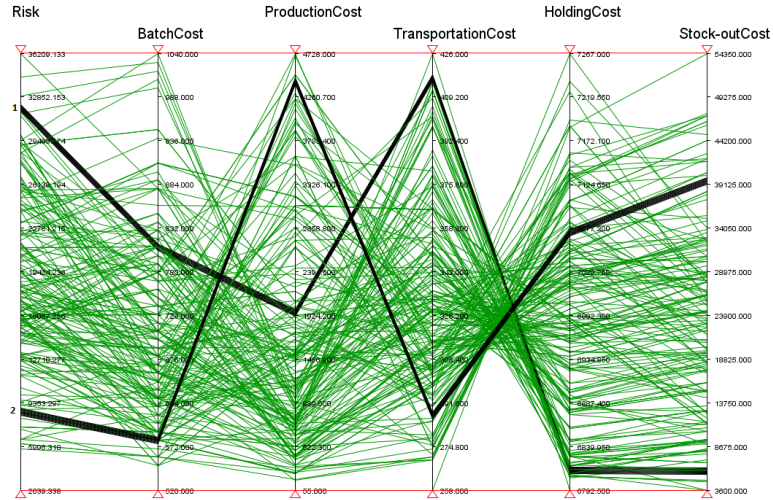


FIGURE 6.5: Two solutions of NSGA-III highlighted in parallel coordinate; Trade-off among objectives for Inst19

relationship between some objectives such as risk and stock-out cost. Objective reduction can be alternative way removing the redundant objectives in the original objective set. To improve performance and to achieve acceptable cost results, some objectives might be excluded and the problem will be solved again. Also the model can be improved to reduce the stock-out cost.

TABLE 6.9: Objective wise results of NSGA-III. TR: Total Risk, TC: Total Cost, TBC: Total Batch Cost, TPC: Total Production Cost, TTC: Total Transportation Cost, TSC: Total Stock-out Cost, THC: Total Holding Cost.

Inst.	TR	TC	Service Level	TBC	TPC	TTC	TSC	THC
Inst1	10,875.0	3,120.0	83.97%	330.0	2,100.0	190.0	0.0	500.0
Inst2	7,554.3	3,812.3	85.68%	300.0	879.0	176.2	1,911.1	546.0
Inst3	7,394.3	4,189.4	78.02%	270.0	1,426.6	166.8	1,405.0	921.0
Inst4	10,398.8	4,667.7	69.45%	270.0	1,643.5	192.2	1,136.1	1,425.9
Inst5	1,814.0	5,768.6	46.81%	200.0	703.0	122.9	1,674.5	3,068.2
Inst6	9,052.5	5,133.9	83.91%	230.0	2,111.6	182.7	1,783.7	826.0
Inst7	15,108.8	6,332.5	61.78%	490.0	2,220.0	269.0	933.5	2,420.0
Inst8	21,172.1	5,480.9	80.29%	700.0	2,524.2	307.9	868.6	1,080.3
Inst9	15,617.4	4,741.2	80.25%	680.0	1,963.0	302.5	859.5	936.3
Inst10	16,217.0	5,872.8	57.83%	600.0	1,695.1	267.1	834.1	2,476.4
Inst11	19,834.3	6,005.7	76.31%	710.0	3,089.6	322.9	460.4	1,422.8
Inst12	11,343.0	8,720.8	67.07%	480.0	4,023.6	243.7	1,101.3	2,872.1
Inst13	11,051.8	7,043.5	55.21%	390.0	1,460.0	281.0	1,757.5	3,155.0
Inst14	17,517.9	7,601.5	66.56%	540.0	2,679.9	285.5	1,554.2	2,541.9
Inst15	8,401.4	6,595.7	59.07%	480.0	1,404.1	238.1	1,774.0	2,699.5
Inst16	11,969.5	6,430.6	64.05%	460.0	1,748.5	300.7	1,609.3	2,312.1
Inst17	12,135.5	9,952.9	60.74%	480.0	3,554.5	290.0	1,721.3	3,907.1
Inst18	6,130.3	7,093.1	51.47%	410.0	1,250.1	247.0	1,743.8	3,442.2
Inst19	16,688.4	10,312.5	83.42%	680.0	690.0	348.0	6,884.5	1,710.0
Inst20	24,902.3	11,994.0	79.19%	970.0	1,657.1	442.6	6,427.7	2,496.5
Inst21	16,885.4	11,549.7	83.39%	830.0	1,380.8	365.9	7,054.8	1,918.2
Inst22	22,838.1	10,990.0	93.93%	1,090.0	1,715.7	365.4	7,152.3	666.6
Inst23	15,405.6	9,925.7	99.53%	890.0	1,121.2	358.4	7,509.5	46.5
Inst24	23,212.4	9,713.2	91.88%	1,020.0	829.9	417.1	6,657.8	788.4

6.3 A Comparative Analysis of Approaches

In this section, a comparison between three different algorithms represent single, multi and many objective algorithms is provided on the integrated problem. The objective version of problem formulation is varied. Single and multi-objective algorithms consider two objective problem (the aggregation of the total risk and total cost Equation 5.2) for single objective algorithm, total cost (Equation 4.7) and total risk (Equation 4.8) separately for multi-objective algorithm) while the six objective problem (considering the total risk (Equation 4.8), production cost (Equation 4.5), holding cost (Equation 4.2), batch cost (Equation 4.4), transportation cost (Equation 4.3) and holding cost (Equation 4.2)) is taken into account by a many-objective algorithm. Three different performance metrics and their cost results are considered to investigate their performance. In order to tackle difficulties of the high-dimensional trade-off front for the six objective problem, we aggregate all cost values together and evaluated them same as the two objective

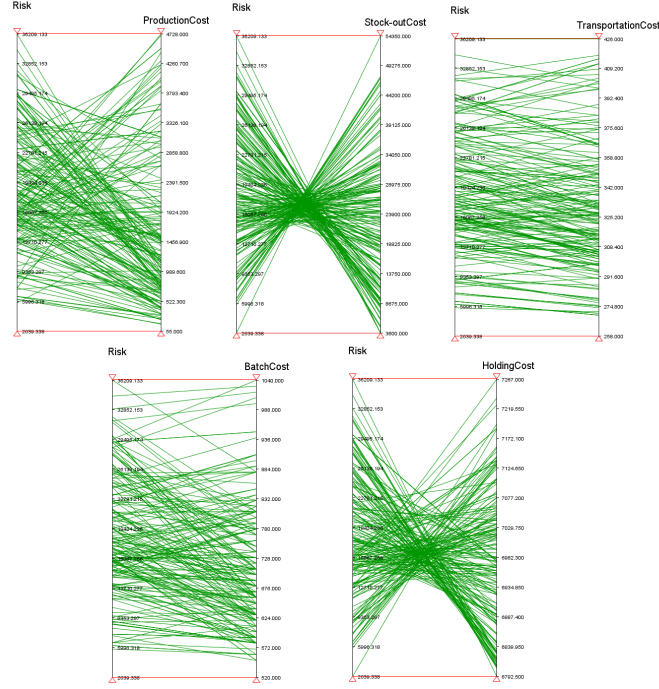


FIGURE 6.6: Risk vs Cost objectives' results of NSGA-III in parallel coordinate for Inst19

problem (to find pareto-front sets and its performance). Each algorithm is chosen based on its performance as explained in Chapter 5 and the previous section. For instance, in the single objective optimisation, a simulated annealing approach with tuned parameters using weighted sum $MOSA_{Ws}$ which performs better than the Tchebycheff approach in most instances is selected. In addition, global pareto front used to measure generational distance and inverted generational distance is defined from all results achieved by all algorithms.

First, $MOSA_{Ws}$, NSGA-II and NSGA-III are compared with each other to investigate their performances. $MOSA_{Ws}$ performs well providing significantly better hypervolume and generational distance results for all over twenty four instances while NSGA-II is capable of finding closer solutions to the global pareto front. Then, considering inverted generational distance, $MOSA_{Ws}$ outperforms NSGA-II for twenty one out of twenty four instances. This performance variation is statistically significant. NSGA-II is slightly better than $MOSA_{Ws}$ on an instance, Inst2. For two instances, Inst4 and Inst13, NSGA-II produces significantly better results than $MOSA_{Ws}$. On the other hand, NSGA-III is superior when compared to $MOSA_{Ws}$ and NSGA-II across all instances as shown in tables, Table 6.10

TABLE 6.10: A comparison of single ($MOSA_{W_s}$), multi (NSGA-II) and many objective (NSGA-III) evolutionary algorithms based on three performance metrics

Inst.	HyperVolume				Generational Distance				Inverted Generational Distance							
		$MOSA_{W_s}$	NSGA-II	NSGA-III	$MOSA_{W_s}$	NSGA-II	NSGA-III	$MOSA_{W_s}$	NSGA-II	NSGA-III	$MOSA_{W_s}$	NSGA-II	NSGA-III			
Inst1	Mean	0.7733	>	0.7640	<	0.8617	85,462.5	>	153,991.3	<	14,191.2	16,841,900.0	>	18,434,666.7	<	465,424.7
	Std.	0.0091		0.0068		0.0178	20,213.5		42,737.6		1,248.6	927,549.7		824,768.8		178,732.2
Inst2	Mean	0.7603	>	0.7527	<	0.9026	77,280.3	>	118,211.6	<	20,818.4	3,163,576.7	<	3,099,630.0	<	165,035.9
	Std.	0.0068		0.0067		0.0070	22,843.7		37,823.6		9,786.4	493,160.3		327,324.2		55,906.3
Inst3	Mean	0.6903	>	0.6795	<	0.8263	67,479.6	>	101,073.1	<	46,785.9	7,979,166.7	>	8,553,806.7	<	2,189,103.3
	Std.	0.0065		0.0097		0.0084	31,026.3		37,281.4		19,769.8	445,211.7		449,454.3		1,026,613.6
Inst4	Mean	0.7204	>	0.7119	<	0.8383	68,977.7	>	104,970.8	<	30,252.1	5,068,016.7	<	4,662,183.3	<	457,079.3
	Std.	0.0057		0.0062		0.0102	25,732.0		37,193.0		23,904.7	819,808.5		658,605.2		271,723.8
Inst5	Mean	0.7543	>	0.7386	<	0.8971	74,447.8	>	133,082.9	<	46,179.5	2,133,026.7	>	2,149,720.0	<	298,860.6
	Std.	0.0082		0.0057		0.0091	30,875.4		53,555.0		29,988.4	253,877.1		257,496.4		127,470.0
Inst6	Mean	0.7323	>	0.7229	<	0.8693	81,334.6	>	116,663.4	<	58,723.5	3,812,376.7	>	3,902,323.3	<	396,973.3
	Std.	0.0070		0.0072		0.0045	24,595.9		38,985.2		28,026.6	926,819.3		799,385.5		149,382.0
Inst7	Mean	0.7968	>	0.7946	<	0.8770	190,459.0	>	320,002.7	<	51,260.7	1,817,530.0	>	1,912,743.3	<	168,246.7
	Std.	0.0037		0.0058		0.0078	54,377.8		77,033.0		24,129.5	99,156.2		95,724.9		34,780.2
Inst8	Mean	0.8001	>	0.7964	<	0.8717	171,902.3	>	301,241.0	<	19,298.9	271,020.7	>	2,391,006.7	<	85,388.1
	Std.	0.0050		0.0026		0.0060	49,505.9		95,836.2		11,332.0	66,977.2		98,423.9		19,731.9
Inst9	Mean	0.8209	>	0.8170	<	0.8899	142,766.3	>	268,958.7	<	118,589.7	814,594.3	>	847,869.3	<	199,444.9
	Std.	0.0051		0.0045		0.0153	40,775.3		65,784.5		122,349.9	33,259.6		26,366.3		136,917.6
Inst10	Mean	0.7950	>	0.7892	<	0.8731	175,808.3	>	301,086.7	<	31,074.0	1,553,430.0	>	1,608,503.3	<	141,270.7
	Std.	0.0051		0.0037		0.0078	54,708.5		78,765.2		11,447.5	58,100.7		46,969.5		41,997.4
Inst11	Mean	0.7913	>	0.7867	<	0.8630	174,361.8	>	227,870.0	<	61,447.7	2,491,006.7	>	2,595,276.7	<	162,999.2
	Std.	0.0053		0.0063		0.0077	48,645.0		71,414.3		65,245.3	97,890.4		108,674.2		47,410.7
Inst12	Mean	0.8090	>	0.8044	<	0.8802	168,454.9	>	283,313.3	<	82,517.8	1,251,120.0	>	1,284,280.0	<	233,954.7
	Std.	0.0049		0.0035		0.0082	41,201.9		70,058.8		44,704.8	70,026.2		51,080.9		73,336.6
Inst13	Mean	0.7076	>	0.6999	<	0.9206	310,709.0	>	521,598.3	<	76,152.2	10,727,580.0	<	10,467,286.7	<	713,037.7
	Std.	0.0074		0.0059		0.0088	96,764.7		184,974.7		41,477.9	653,156.9		533,005.9		279,123.7
Inst14	Mean	0.7048	>	0.6987	<	0.8746	284,985.3	>	389,654.3	<	79,809.1	2,429,616.7	>	2,481,806.7	<	212,990.4
	Std.	0.0050		0.0047		0.0100	65,562.0		148,033.4		34,490.0	54,092.0		67,294.9		57,665.8
Inst15	Mean	0.7242	>	0.7187	<	0.9263	313,221.0	>	508,622.0	<	94,034.5	2,643,120.0	>	2,676,630.0	<	204,205.7
	Std.	0.0059		0.0036		0.0074	90,350.0		180,539.9		46,892.7	78,777.9		59,602.4		57,232.2
Inst16	Mean	0.7061	>	0.6991	<	0.8980	254,250.8	>	395,404.3	<	90,109.0	2,396,606.7	>	2,505,200.0	<	182,753.4
	Std.	0.0048		0.0065		0.0070	86,671.6		131,243.3		44,550.3	59,931.2		67,682.4		53,838.6
Inst17	Mean	0.7021	>	0.6936	<	0.8862	256,914.0	>	310,075.0	<	115,584.6	4,932,020.0	>	5,022,010.0	<	534,597.3
	Std.	0.0057		0.0059		0.0120	94,232.6		128,397.4		64,992.4	222,404.3		238,738.9		190,702.7
Inst18	Mean	0.7325	>	0.7267	<	0.9083	276,170.1	>	398,581.7	<	131,657.6	2,009,436.7	>	2,059,116.7	<	209,729.7
	Std.	0.0054		0.0056		0.0110	85,834.1		135,965.9		70,231.8	52,376.6		42,911.6		45,540.7
Inst19	Mean	0.7022	>	0.7006	<	0.9837	2,136,531.7	>	4,053,156.7	<	43,111.0	57,306,266.7	>	57,760,266.7	<	247,124.0
	Std.	0.0042		0.0052		0.0016	607,978.7		994,025.7		19,239.8	1,286,836.7		1,092,811.7		72,672.2
Inst20	Mean	0.7316	>	0.7308	<	0.9792	2,064,702.3	>	2,906,855.7	<	117,175.2	41,158,766.7	>	41,542,666.7	<	301,284.0
	Std.	0.0045		0.0044		0.0026	678,215.2		972,084.7		51,612.7	812,124.0		569,996.9		65,545.4
Inst21	Mean	0.7090	>	0.7066	<	0.9847	2,481,134.0	>	3,827,270.0	<	57,679.7	56,084,733.3	>	56,721,333.3	<	307,158.7
	Std.	0.0044		0.0045		0.0017	893,922.1		988,830.8		17,709.2	960,658.3		957,691.2		92,594.1
Inst22	Mean	0.7036	>	0.7021	<	0.9862	2,546,636.7	>	3,684,286.7	<	67,881.1	56,947,966.7	>	57,495,800.0	<	229,310.3
	Std.	0.0036		0.0055		0.0013	786,735.9		1,199,193.4		29,465.3	668,585.4		783,634.1		58,393.4
Inst23	Mean	0.7279	>	0.7274	<	0.9886	2,599,646.7	>	4,289,596.7	<	47,219.0	50,817,900.0	>	51,158,666.7	<	179,730.0
	Std.	0.0039		0.0036		0.0007	704,422.5		1,422,592.3		13,601.8	919,166.4		1,181,358.4		49,710.5
Inst24	Mean	0.7342	>	0.7317	<	0.9852	2,052,283.0	>	3,695,396.7	<	75,444.5	48,668,333.3	>	49,376,333.3	<	261,063.7
	Std.	0.0039		0.0036		0.0014	612,538.5		1,077,348.8		30,343.3	977,624.8		841,295.5		58,524.4

where a comparison between $MOSA_{W_s}$ and NSGA-III is given and Table 6.11 where a comparison between NSGA-II and NSGA-III is given.

As a sample, we plotted the pareto-front achieved by each algorithm on Inst1 as shown in Figure 6.7. As it can be observed from Figure 6.7, NSGA-II and NSGA-III produce a wider spread of solutions on the pareto-front when compared to $MOSA_{W_s}$.

We search for a single solution by $MOSA_{W_s}$ for each run and execute $MOSA_{W_s}$ several times using random weight settings to obtain multiple solutions by $MOSA_{W_s}$ as detailed in Chapter 5. The solution found each run sometimes also does not belong to the pareto set. Thus, we observed difficulties in comparing several solutions from multiple runs of $MOSA_{W_s}$ with a large number of solutions

TABLE 6.11: Performance comparison between NSGA-II and NSGA-III

Inst.		Hypervolume			Generational Distance			Inverted Generational Distance		
		NSGA-II		NSGA-III	NSGA-II		NSGA-III	NSGA-II		NSGA-III
Inst1	Mean	0.7640	<	0.8617	153,991.3	<	14,191.2	18,434,666.7	<	465,424.7
	Std.	0.0068		0.0178	42,737.6		12,478.6	824,768.8		178,732.2
Inst2	Mean	0.7527	<	0.9026	118,211.6	<	20,818.4	3,099,630.0	<	165,035.9
	Std.	0.0067		0.0070	37,823.6		9,786.4	327,324.2		55,906.3
Inst3	Mean	0.6795	<	0.8263	101,073.1	<	46,785.9	8,553,806.7	<	2,189,103.3
	Std.	0.0097		0.0084	37,281.4		19,769.8	449,454.3		1,026,613.6
Inst4	Mean	0.7119	<	0.8383	104,970.8	<	30,252.1	4,662,183.3	<	457,079.3
	Std.	0.0062		0.0102	37,193.0		23,904.7	658,605.2		271,723.8
Inst5	Mean	0.7386	<	0.8971	133,082.9	<	46,179.5	2,149,720.0	<	298,860.6
	Std.	0.0057		0.0091	53,555.0		29,988.4	257,496.4		127,470.0
Inst6	Mean	0.7229	<	0.8693	116,663.4	<	58,723.5	3,902,323.3	<	396,973.3
	Std.	0.0072		0.0045	38,985.2		28,026.6	799,385.5		149,382.0
Inst7	Mean	0.7946	<	0.8770	320,002.7	<	51,260.7	1,912,743.3	<	168,246.7
	Std.	0.0058		0.0078	77,033.0		24,129.5	95,724.9		34,780.2
Inst8	Mean	0.7964	<	0.8717	301,241.0	<	19,298.9	2,391,006.7	<	85,388.1
	Std.	0.0026		0.0060	95,836.2		11,332.0	98,423.9		19,731.9
Inst9	Mean	0.8170	<	0.8899	268,958.7	<	118,589.7	847,869.3	<	199,444.9
	Std.	0.0045		0.0153	65,784.5		122,349.9	26,366.3		136,917.6
Inst10	Mean	0.7892	<	0.8731	301,086.7	<	31,074.0	1,608,503.3	<	141,270.7
	Std.	0.0037		0.0078	78,765.2		11,447.5	46,969.5		41,997.4
Inst11	Mean	0.7867	<	0.8630	227,870.0	<	61,447.7	2,595,276.7	<	162,999.2
	Std.	0.0063		0.0077	71,414.3		65,245.3	108,674.2		47,410.7
Inst12	Mean	0.8044	<	0.8802	283,313.3	<	82,517.8	1,284,280.0	<	233,954.7
	Std.	0.0035		0.0082	70,058.8		44,704.8	51,080.9		73,336.6
Inst13	Mean	0.6999	<	0.9206	521,598.3	<	76,152.2	10,467,286.7	<	713,037.7
	Std.	0.0059		0.0088	184,974.7		41,477.9	533,005.9		279,123.7
Inst14	Mean	0.6987	<	0.8746	389,654.3	<	79,809.1	2,481,806.7	<	212,990.4
	Std.	0.0047		0.0100	148,033.4		34,490.0	67,294.9		57,665.8
Inst15	Mean	0.7187	<	0.9263	508,622.0	<	94,034.5	2,676,630.0	<	204,205.7
	Std.	0.0036		0.0074	180,539.9		46,892.7	59,602.4		57,232.2
Inst16	Mean	0.6991	<	0.8980	395,404.3	<	90,109.0	2,505,200.0	<	182,753.4
	Std.	0.0065		0.0070	131,243.3		44,550.3	67,682.4		53,838.6
Inst17	Mean	0.6936	<	0.8862	310,075.0	<	115,584.6	5,022,010.0	<	534,597.3
	Std.	0.0059		0.0120	128,397.4		64,992.4	238,738.9		190,702.7
Inst18	Mean	0.7267	<	0.9083	398,581.7	<	131,657.6	2,059,116.7	<	209,729.7
	Std.	0.0056		0.0110	135,965.9		70,231.8	42,911.6		45,540.7
Inst19	Mean	0.7006	<	0.9837	4,053,156.7	<	43,111.0	57,760,266.7	<	247,124.0
	Std.	0.0052		0.0016	994,025.7		19,239.8	1,092,811.7		72,672.2
Inst20	Mean	0.7308	<	0.9792	2,906,855.7	<	117,175.2	41,542,666.7	<	301,284.0
	Std.	0.0044		0.0026	972,084.7		51,612.7	569,996.9		65,545.4
Inst21	Mean	0.7066	<	0.9847	3,827,270.0	<	57,679.7	56,721,333.3	<	307,158.7
	Std.	0.0045		0.0017	988,830.8		17,709.2	957,691.2		92,594.1
Inst22	Mean	0.7021	<	0.9862	3,684,286.7	<	67,881.1	57,495,800.0	<	229,310.3
	Std.	0.0055		0.0013	1,199,193.4		29,465.3	783,634.1		58,393.4
Inst23	Mean	0.7274	<	0.9886	4,289,596.7	<	47,219.0	51,158,666.7	<	179,730.0
	Std.	0.0036		0.0007	1,422,592.3		13,601.8	1,181,358.4		49,710.5
Inst24	Mean	0.7317	<	0.9852	3,695,396.7	<	75,444.5	49,376,333.3	<	261,063.7
	Std.	0.0036		0.0014	1,077,348.8		30,343.3	841,295.5		58,524.4

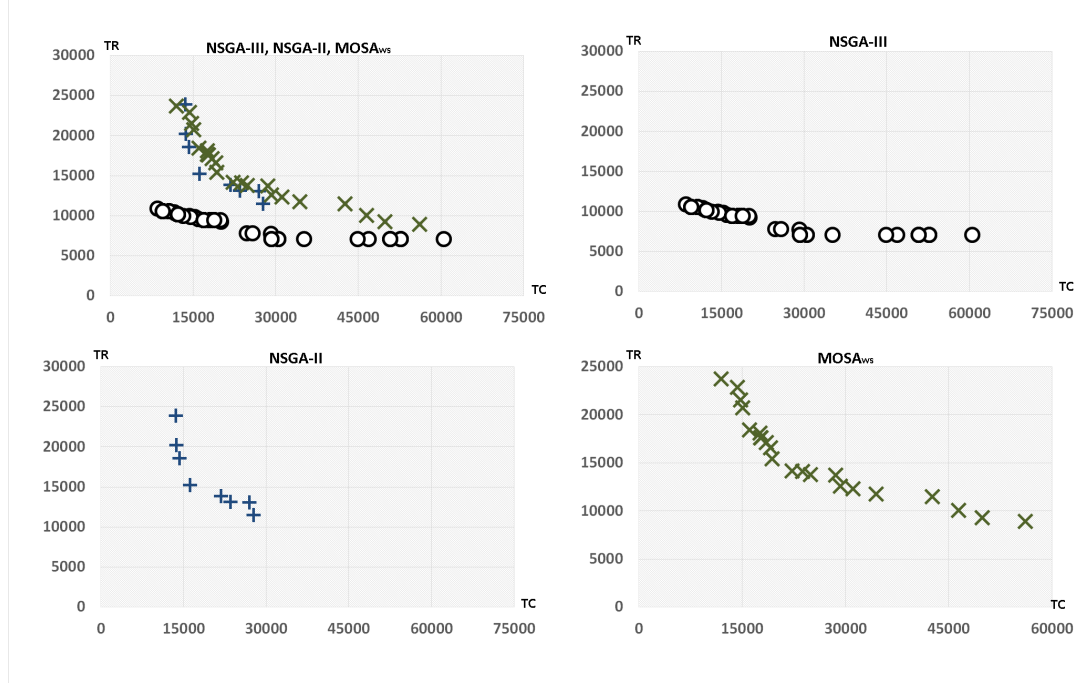


FIGURE 6.7: Pareto-front plots for NSGA-III, NSGA-II and $MOSA_{ws}$ based on TR vs TC from a sample run on Inst1

from a single run of NSGA-II and NSGA-III as seen in Figure 6.7. For this reason, to provide better spread in $MOSA_{ws}$, the number of runs could be increased.

In addition, in NSGA-II and NSGA-III, without any prior assumption (such as assigning weights to each objective in $MOSA_{ws}$), equally good solutions are searched in a run. However, NSGA-III generates high-dimensional pareto fronts that larger iterations and population might be needed to adequately represent the problem. Moreover, with regard to applicability of the problem, cost results show that NSGA-II and $MOSA_{ws}$ achieved at least 90% of ‘satisfied’ customers while NSGA-III satisfied customer expectations poorly as seen in Table 6.9.

6.4 Summary

This chapter provides an investigation of four meta-heuristic algorithms on the integrated SCM problem. Also, this chapter analysed performances of the overall multi-objective approach using three well known MOEAs, namely NSGA-II, SPEA2 and IBEA applied to the problem. First, the optimal parameter setting

is found for each algorithm. After tuning, each algorithm is tested on twenty four problem instances. The results show the overall success of NSGA-II comparing to SPEA2 and IBEA. High quality trade-off solutions which satisfy the customer demand at least 90% are achieved by all MOEAs while minimising objectives simultaneously.

Moreover, the empirical results led us to investigate the trade-off between all contributing costs to the total cost and risk, separately. The many-objective optimisation algorithm, NSGA-III is applied to the six objective formulation of the same problem. NSGA-III performs better than three MOEAs in terms of three performance metrics used in the study. However, it is found that NSGA-III cannot satisfy customer expectations while producing high stock-out cost.

Finally, based on these findings, this chapter presents an investigation on the performance of three algorithms representing single, multi and many objective optimisation namely $MOSA_{Ws}$, NSGA-II and NSGA-III, respectively while single, two and six objective formulations of the same problem are used in sequence. The empirical results over twenty four instances show that $MOSA_{Ws}$ performs better than NSGA-II in majority of the cases. Also, NSGA-III outperforms $MOSA_{Ws}$ and NSGA-II over all instances. In the next chapter, the conclusion and possible directions for future work are given.

Chapter 7

Conclusion

This thesis has explored a two-stage integrated approach which considers both the supplier selection and inventory planning. The combination of interval type-2 fuzzy sets and multi-objective evolutionary algorithms as a solution technique has been proposed to tackle the integrated supply chain management problem. The problem is solved by single, multi and many objective optimisation algorithms, respectively. In order to successfully address this idea, a number of angles have been considered. One of these has been the selection of appropriate methods and algorithms to build and solve the proposed problem. After a general overview of methods used in the literature, in consequence of the uncertain nature of supplier selection problems, interval type-2 fuzzy sets are used to assess the performance of suppliers. To solve the problem as a single objective problem, a single point search method, SA known as MOSA for multi-objective optimisation is chosen. After reviewing the structural aspects of MOEAs and applications, NSGA-II, SPEA2 and IBEA are preferred while the integrated SCM problem is treated as a multi-objective problem. For the many objective formulation of the problem, NSGA-III is used due to its effectiveness as a solution method [Deb and Jain, 2014].

Following on from this, the formulation of the problem is defined based on the number of objectives varied. The thesis has also considered the parameter settings of algorithms. The parameters of each algorithm are tuned to obtain the best combination of parameters for the algorithm. Furthermore, to evaluate the performance of each algorithm and to provide a comparative analysis of algorithms selected in terms of their performances, three metrics, hypervolume, generational

distance and inverted generational distance which measure the algorithm performance in terms of diversification of the population and convergence towards the pareto front are chosen. In the following section, the contributions of each chapter will be analysed in more detail giving summary of work.

7.1 Summary of Work

In this thesis, a two-stage integrated approach is addressed to the integrated supply chain management problem which aims to minimise the conflicting objectives of supply chain operation cost and supplier risk. The problem is treated as a single, multi or many objective optimisation problem.

Firstly, the multi-objective problem is converted into a scalar problem using two different scalarisation methods; the weighted sum and Tchebycheff and a MOSA approach is applied to the problem using different weight settings. We show trade-offs between risk and cost through varying weights of objectives in MOSA. That study illustrated the multi-objective nature of the problem testing the proposed approaches on a single objective problem. Secondly, this work is extended using three well known multi-objective evolutionary algorithms, namely NSGA-II, SPEA2 and IBEA to solve the problem with an attempt to detect the best performing approach. The performances of these algorithms are evaluated using well known metrics, hypervolume, generational distance and inverse generational distance on twenty four different problem instances, where four of them are real world problem instances and 20 of them are randomly generated based on those instances.

The empirical results indicate the overall success of NSGA-II on the integrated SCM problem. Three MOEAs, NSGA-II, SPEA2 and IBEA, achieved high quality trade-off solutions satisfying the customer demand almost 92% in majority of the cases. Also, based on the results, there are even more conflicting objectives which can be considered in the solution model and then simultaneously optimised. Although MOEAs performed reasonably well in this study for the two-objective problem, this might not be the case when the number of objectives are increased. Finally, we investigate each cost component of the total cost as a separate objective

and we obtain a six objective integrated SCM problem. This problem is solved using a many-objective approach namely NSGA-III.

7.2 Research Contributions

The key results of the research presented in the thesis are summarised as follows:

- Based on the literature, in most of the research, supplier selection and inventory management problems are tackled separately. To the authors' knowledge, this is one of the first studies in which the integrated problem of supplier selection and inventory planning has been investigated as a multi-objective problem. In Chapter 4, the integrated SCM problem is explained.
- In Chapter 5, the integrated supply chain management model is built using both information about supplier selection and inventory planning and the multi-objective nature of the integrated supply chain management problem is shown testing on a simple problem instance. The problem is formulated as single, multi and many-objective optimisation problem, respectively. Minimising the total risk of suppliers and minimising the total cost of inventory are defined as objectives. Two objectives are aggregated into a single objective using two scalarisation approaches, weighted sum and Tchbycheff and the single objective problem is solved using a generic local search meta-heuristic, Simulated Annealing. The proposed approach provides a system that decision makers can alternatively choose a solution from the set of trade-off solutions obtained using different risk versus cost scenario. It is observed that based on three performance metrics, the performance of MOSA using the weighted sum scalarisation technique is better than MOSA using the Tchbycheff scalarisation technique in most instances.
- In Chapter 6.1, three MOEAs, NSGA-II, SPEA2 and IBEA are used to deal with the two objective integrated problem. Based on the performance evaluation, NSGA-II performs better than other two algorithms, SPEA2 and IBEA on average across 24 instances. Based on the knee solution of each algorithm, all MOEAs achieved high quality trade-off solutions while the customer satisfaction rate is at least 91% in the majority of instances.

- In Chapter 6.2, in order to tackle conflicting objectives in the total cost, the problem is reformulated as a six objective problem and solved using NSGA-III. In the two objective problem, the total cost and total risk are determined as objectives while in the six objective version of the same problem, total risk, total production cost, total holding cost, total batch cost, total transportation cost and total stock-out cost are considered as objectives. It is found that NSGA-III is superior than other algorithms based on three performance metrics for 24 instances. However, NSGA-III achieve a low customer satisfaction rate for the majority of the instances while generating high stock-out cost.
- In Chapter 6.3, based on the comparative analysis of algorithms, $MOSA_{WS}$ provides better performance than NSGA-II in most instances and NSGA-III outperforms NSGA-II, $MOSA_{WS}$ over 24 instances because of its selection operator developed to overcome many-objective optimisation difficulties using a systematic evaluation of individuals in the population. The empirical results also indicate that NSGA-II, $MOSA_{WS}$ achieved the high quality trade-off solutions which satisfy the customer expectation at least 90% in the majority of instances. However, NSGA-III cannot achieve good customer satisfaction.
- In Chapter 5.2, based on the comparison between performances of MOSA with default parameter settings and MOSA with tuned parameter settings, it is shown that parameters have an essential influence on the performance of algorithm. Parameters of each algorithm used in this study are tuned and an explanation can be found in Chapter 5 and Chapter 6.
- The problem instances used in this thesis are provided as a benchmark and are made publically available from https://www.researchgate.net/publication/311873391_SCMproblem with DOI : 10.13140/RG.2.2.21820.64643.

7.3 Future Work

A number of possible research directions can be pursued in order to extend the work presented in this thesis.

- Chapter 4 introduced an integrated supply chain management model. This model could be improved in three ways;
 - i) More constraints can be added into the proposed model such as the *defective item cost* which is encountered during the production process, *the rejection rate* which is the percentage of order from a supply that are rejected. These types of constraints make the model reflected a real-world problem in a more effective way.
 - ii) A risk value could be assigned to stock-out items. In Chapter 4, the stock-out cost is defined as a penalty cost incurred when the quantity of product is less than the customer demands. In the proposed model, it is assumed that the missing amount of any product is bought from a competitor with a retailers price. When the stock-out cost is in the total cost, the model produces reasonable results. However, in Chapter 6, for many-objective optimisation, stock-out cost increased sharply when it is taken into account as an objective. It is suggested that the stock-out items could be linked to a risk value and affect total risk to produce acceptable results.
 - iii) The number of objectives would be reduced to four for many-objective optimisation based on relationships among objectives found as in the parallel coordinates figures in Chapter 6.2.
- Another future study could be applying the approach to new unseen instances possibly even larger than the ones used in this study and/or changing the decision makers' supplier related preferences creating more instances. We provide the problem instances used in this study as a benchmark along with our implementation of the approach for future research.
- In terms of the solution methods for tackling the integrated supply chain management problem, three well-known multi-objective evolutionary algorithms, NSGA-II, SPEA2 and IBEA are applied to two objective version of the problem. It would be interesting to explore other MOEA optimisers. There is various meta-heuristics choices and therefore great scope for research. For example, it would be worth investigating MOEA/D method [Zhang and Li, 2007] proposed as an advanced method to see whether it is capable of achieving the best trade-off solutions.

- Following the algorithmic experiments of Chapter 5, it is discussed that the number of parameters in an algorithm may affect the ability of algorithm. Despite the considerable research effort that has recently been devoted to meta-heuristics, this would be an interesting area for further investigation on parameter tuning methods to provide better understanding about algorithmic performance during running experiments on problem instances.

References

- Altıparmak, F., Gen, M., Lin, L., Paksoy, T., 2006. A genetic algorithm approach for multi-objective optimization of supply chain networks. *Computers and Industrial Engineering* 51, 196 – 215. Special Issue on Computational Intelligence and Information Technology: Applications to Industrial Engineering 33rd.
- Amid, A., Ghodsypour, S., O’Brien, C., 2006. Fuzzy multiobjective linear model for supplier selection in a supply chain. *International Journal of Production Economics* 104, 394 – 407.
- Amid, A., Ghodsypour, S., O’Brien, C., 2009. A weighted additive fuzzy multiobjective model for the supplier selection problem under price breaks in a supply chain. *International Journal of Production Economics* 121, 323 – 332.
- Amid, A., Ghodsypour, S., O’Brien, C., 2011. A weighted max–min model for fuzzy multi-objective supplier selection in a supply chain. *International Journal of Production Economics* 131, 139 – 145.
- Arostegui, M.A., Kadipasaoglu, S.N., Khumawala, B.M., 2006. An empirical comparison of tabu search, simulated annealing, and genetic algorithms for facilities location problems. *International Journal of Production Economics* 103, 742 – 754.
- Bandyopadhyay, S., Saha, S., Maulik, U., Deb, K., 2008. A simulated annealing-based multiobjective optimization algorithm: Amosa. *Trans. Evol. Comp* 12, 269–283.
- Bechikh, S., Ben Said, L., Ghédira, K., 2010. Searching for knee regions in multi-objective optimization using mobile reference points, in: *Proceedings of the 2010 ACM Symposium on Applied Computing*, ACM, New York, NY, USA. pp. 1118–1125.

- Bhagavatula, S.S., Sanjeevi, S.G., Kumar, D., Yadav, C.K., 2014. Multi-objective indicator based evolutionary algorithm for portfolio optimization, in: Advance Computing Conference (IACC), 2014 IEEE International, pp. 1206–1210.
- Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E., 2003. Pisa - a platform and programming language independent interface for search algorithms, Springer. pp. 494–508.
- Blum, C., Roli, A., 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.* 35, 268–308.
- de Boer, L., Labro, E., Morlacchi, P., 2001. A review of methods supporting supplier selection. *European Journal of Purchasing & Supply Management* 7, 75–89.
- Brockhoff, D., Friedrich, T., Neumann, F., 2008. Analyzing Hypervolume Indicator Based Algorithms, in: Rudolph, G., et al. (Eds.), *Conference on Parallel Problem Solving From Nature (PPSN X)*, Springer. pp. 651–660.
- Cappello, A., Losch, M., Schmitz, C., 2008. *Achieving Top Performance in Supply Chain Management*. Wiley-VCH Verlag GmbH and Co. KGaA. pp. 281–296.
- Caramia, M., Dell’Olmo, P., 2008. *Multi-objective Management in Freight Logistics: Increasing Capacity, Service Level and Safety with Optimization Algorithms*. Springer London. URL: https://books.google.co.uk/books?id=U4NJ_1AdrZ8C.
- Charnes, A., Cooper, W., Rhodes, E., 1978. Measuring the efficiency of decision making units. *European Journal of Operational Research* 2, 429 – 444.
- Che, Z., Chiang, C., 2010. A modified pareto genetic algorithm for multi-objective build-to-order supply chain planning with product assembly. *Advances in Engineering Software* 41, 1011 – 1022. *Advances in Structural Optimization*.
- Che, Z.H., Wang, H.S., 2010. A hybrid approach for supplier cluster analysis. *Comput. Math. Appl.* 59, 745–763.
- Chen, C.T., 2000. Extensions of the topsis for group decision-making under fuzzy environment. *Fuzzy Sets and Systems* 114, 1 – 9.

- Chen, C.T., Lin, C.T., Huang, S.F., 2006. A fuzzy approach for supplier evaluation and selection in supply chain management. *International Journal of Production Economics* 102, 289 – 301.
- Chen, S.M., Lee, L.W., 2010a. Fuzzy multiple attributes group decision-making based on the interval type-2 topsis method. *Expert Syst. Appl.* 37, 2790–2798.
- Chen, S.M., Lee, L.W., 2010b. Fuzzy multiple attributes group decision-making based on the ranking values and the arithmetic operations of interval type-2 fuzzy sets. *Expert Syst. Appl.* 37, 824–833.
- Chen, S.M., Yang, M.W., Lee, L.W., Yang, S.W., 2012. Fuzzy multiple attributes group decision-making based on ranking interval type-2 fuzzy sets. *Expert Syst. Appl.* 39, 5295–5308.
- Choy, K., Fan, K.K., Lo, V., 2003a. Development of an intelligent customer supplier relationship management system: the application of case based reasoning. *Industrial Management and Data Systems* 103, 263–274.
- Choy, K., Lee, W., Lo, V., 2002. Development of a case based intelligent customer–supplier relationship management system. *Expert Systems with Applications* 23, 281 – 297.
- Choy, K., Lee, W., Lo, V., 2003b. Design of an intelligent supplier relationship management system: a hybrid case based neural network approach. *Expert Systems with Applications* 24, 225 – 237. *Intelligent Electronic Commerce*.
- Christopher, M.L., 1992. *Logistics and Supply Chain Management*. Pitman Publishing, London.
- Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V., 2006. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- Coello Coello, C., Reyes Sierra, M., 2004. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm, in: Monroy, R., Arroyo-Figueroa, G., Sucar, L., Sossa, H. (Eds.), *MICAI 2004: Advances in Artificial Intelligence*. Springer Berlin Heidelberg. volume 2972 of *Lecture Notes in Computer Science*, pp. 688–697.

- Coello Coello, C.A., 1996. An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design. Ph.D. thesis. New Orleans, LA, USA. AAI9639654.
- Coello Coello, C.A., 1999. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems* 1, 269–308.
- Coello Coello, C.A., 2002. Evolutionary Multi-Objective Optimization: A Critical Review. Springer US, Boston, MA. pp. 117–146.
- Cooper, M.C., Lambert, D.M., Pagh, J.D., 1997. Supply chain management: More than a new name for logistics. *The International Journal of Logistics Management* 8, 1–14.
- Czyżak, P., Jaskiewicz, A., 1997. Pareto Simulated Annealing. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 297–307.
- Czyzak, P., Jaskiewicz, A., 1998. Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis* 7, 34–47.
- Das, I., Dennis, J.E., 1997. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural optimization* 14, 63–69.
- Deb, K., 2001. Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Inc., New York, NY, USA.
- Deb, K., 2007. Evolutionary Multi-Objective Optimization Without Additional Parameters. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 241–257.
- Deb, K., 2011. Multi-objective optimisation using evolutionary algorithms: An introduction, in: Wang, L., Ng, A.H.C., Deb, K. (Eds.), *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*. Springer London, pp. 3–34. doi:10.1007/978-0-85729-652-8_1.
- Deb, K., Agrawal, S., Pratap, A., Meyarivan, T., 2000. A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 849–858.

- Deb, K., Jain, H., 2014. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation* 18, 577–601.
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multi-objective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on* 6, 182–197. doi:10.1109/4235.996017.
- Demirtas, E.A., Ustun, O., 2008. An integrated multiobjective decision making process for supplier selection and order allocation. *Omega* 36, 76 – 90. Special Issue Section: Papers presented at the {INFORMS} conference, Atlanta, 2003.
- Demirtas, E.A., Ustun, O., 2009. Analytic network process and multi-period goal programming integration in purchasing decisions. *Computers & Industrial Engineering* 56, 677 – 690. *Design and Management of Production Systems*.
- DuBois, D., 1997. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, Inc., Orlando, FL, USA.
- Durillo, J., Nebro, A., Alba, E., 2010. The jmetal framework for multi-objective optimization: Design and architecture, in: *CEC 2010, Barcelona, Spain*. pp. 4138–4325.
- Durillo, J.J., Nebro, A.J., 2011a. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software* 42, 760–771.
- Durillo, J.J., Nebro, A.J., 2011b. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software* 42, 760–771.
- Durillo, J.J., Nebro, A.J., Luna, F., Dorronsoro, B., Alba, E., 2006. jMetal: A Java Framework for Developing Multi-Objective Optimization Metaheuristics. Technical Report ITI-2006-10. E.T.S.I. Informática, Campus de Teatinos.
- Ehrgott, M., 2006. A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research* 147, 343–360.
- Eiben, A., Smit, S., 2011. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation* 1, 19 – 31.

- Çelebi, D., Bayraktar, D., 2008. An integrated neural network and data envelopment analysis for supplier evaluation under incomplete information. *Expert Systems with Applications* 35, 1698 – 1710.
- Esmin, A.A., Coelho, R.A., Matwin, S., 2015. A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data. *Artif. Intell. Rev.* 44, 23–45.
- Feo, T.A., Resende, M.G.C., 1995. Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6, 109–133.
- Fonseca, C.M., Fleming, P.J., 1993. Genetic algorithms for multiobjective optimization: Formulation discussion and generalization, in: *Proceedings of the 5th International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. pp. 416–423.
- Fonseca, C.M., Fleming, P.J., 1995. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation* 3, 1–16.
- Fonseca, C.M., Fleming, P.J., 1998. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. i. a unified formulation. *Trans. Sys. Man Cyber. Part A* 28, 26–37.
- Forrester, J., 1958. Industrial Dynamics: A Major Breakthrough for Decision Makers. *Harvard Business Review* 36, 37–66.
- Forrester, J.W., 1961. Industrial dynamics. M.I.T. Press, Cambridge, Mass.
- Frutos, M., Mendez, M., Tohme, F., Broz, D., 2014. Comparison of multiobjective evolutionary algorithms for operations scheduling under machine availability constraints. *The Scientific World Journal* 2013.
- Galankashi, M.R., Helmi, S.A., Hashemzahi, P., 2016. Supplier selection in automobile industry: A mixed balanced scorecard–fuzzy ahp approach. *Alexandria Engineering Journal* 55, 93 – 100.
- Garibaldi, J.M., John, R.I., 2003. Choosing membership functions of linguistic terms, in: *Proceedings of the IEEE Intl. Conf. on Fuzzy Systems (FUZZ-IEEE 2003)*, St Louis, USA. URL: <http://www.cs.nott.ac.uk/~jmg/papers/fuzzieee-03.pdf>.

- Gencer, C., Gürpınar, D., 2007. Analytic network process in supplier selection: A case study in an electronic firm. *Applied Mathematical Modelling* 31, 2475 – 2486.
- Ghodsypour, S., O’Brien, C., 1998. A decision support system for supplier selection using an integrated analytic hierarchy process and linear programming. *International Journal of Production Economics* 56, 199 – 212.
- Ghodsypour, S., O’Brien, C., 2001. The total cost of logistics in supplier selection, under conditions of multiple sourcing, multiple criteria and capacity constraint. *International Journal of Production Economics* 73, 15 – 27.
- Ghosh, A., Dehuri, S., 2004. Evolutionary algorithms for multi-criterion optimisation: A survey. *International Journal of Computing & Information Science* 2, 38–57.
- Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research* 13, 533 – 549.
- Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st ed., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Gong, Y., 2013. Fuzzy multi-attribute group decision making method based on interval type-2 fuzzy sets and applications to global supplier selection. *International Journal of Fuzzy Systems* 15.
- Greenfield, S., Chiclana, F., John, R., Coupland, S., 2012. The sampling method of defuzzification for type-2 fuzzy sets: Experimental evaluation. *Information Sciences* 189, 77 – 92.
- Grefenstette, J.J., 1986. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics* 16, 122–128.
- Hagras, H., 2007. Type-2 flcs: A new generation of fuzzy controllers. *IEEE Computational Intelligence Magazine* 2, 30–43.
- Hazewinkel, M., 1987. *Encyclopaedia of Mathematics (1)*. *Encyclopaedia of Mathematics: An Updated and Annotated Translation of the Soviet "Mathematical Encyclopaedia"*, Springer. URL: <https://books.google.co.uk/books?id=WzbzxytdAuUC>.

- Helbig, M., Engelbrecht, A.P., 2013. Performance measures for dynamic multi-objective optimisation algorithms. *Information Sciences* 250, 61 – 81.
- Hillier, F.S., Lieberman, G.J., 1986. *Introduction to Operations Research*, 4th Ed. Holden-Day, Inc., San Francisco, CA, USA.
- Ho, W., Xu, X., Dey, P.K., 2010. Multi-criteria decision making approaches for supplier evaluation and selection: A literature review. *Eur. J. Oper. Res.* 202, 16 – 24.
- Horn, J., Nafpliotis, N., Goldberg, D.E., 1994. A niched pareto genetic algorithm for multiobjective optimization, in: *IN PROCEEDINGS OF THE FIRST IEEE CONFERENCE ON EVOLUTIONARY COMPUTATION, IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE*, pp. 82–87.
- Huband, S., Hingston, P., Barone, L., While, L., 2006. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation* 10, 477–506.
- Hussin, M.S., Stutzle, T., 2014. Tabu search vs. simulated annealing as a function of the size of quadratic assignment problem instances. *Computers and Operations Research* 43, 286 – 291.
- Hwang, C., Yoon, K., 1981. *Multiple attribut decision making : Methods and applications*. Springer-Verlag.
- Inselberg, A., 2009. *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- Ishibuchi, H., Tsukamoto, N., Sakane, Y., Nojima, Y., 2010. Indicator-based evolutionary algorithm with hypervolume approximation by achievement scalarizing functions, in: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA*. pp. 527–534.
- Jain, V., Tiwari, M., Chan, F., 2004. Evaluation of the supplier performance using an evolutionary fuzzy based approach. *Manufacturing Technology Management Journal* 15, 735 – 744.
- Jaszkiewicz, A., 2001. Comparison of local search-based metaheuristics on the multiple objective knapsack problem. *Foundations of Computing and Decision Sciences* Vol. 26, No. 1, 99–120.

- Junior, F.R.L., Osiro, L., Carpinetti, L.C.R., 2014. A comparison between fuzzy ahp and fuzzy topsis methods to supplier selection. *Applied Soft Computing* 21, 194 – 209.
- Kahraman, C., 2008. *Multi-Criteria Decision Making Methods and Fuzzy Sets*. Springer US, Boston, MA. pp. 1–18.
- Kahraman, C., Cebeci, U., Ulukan, Z., 2003. Multi-criteria supplier selection using fuzzy ahp. *Logistics Information Management* 16, 382 – 394.
- Kalender, M., Kheiri, A., Özcan, E., Burke, E.K., 2013. A greedy gradient-simulated annealing selection hyper-heuristic. *Soft Computing* 17, 2279–2292.
- Karnik, N., Mendel, J., 1998. Introduction to type-2 fuzzy logic systems, in: *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence.*, pp. 915–920.
- Karnik, N.N., Mendel, J.M., 1999. Type-2 fuzzy logic systems. *Fuzzy Systems, IEEE Transactions on* 7, 643–658.
- Karpak, B., Kumcu, E., Kasuganti, R.R., 2001. Purchasing materials in the supply chain: managing a multi-objective task. *European Journal of Purchasing and Supply Management* 7, 209 – 216.
- Katsikeas, C.S., Paparoidamis, N.G., Katsikea, E., 2004. Supply source selection criteria: The impact of supplier performance on distributor performance. *Industrial Marketing Management* 33, 755 – 764.
- Keah, C.T., Kannan, V.R., Handfield, R.B., 1998. Supply chain management: Supplier performance and firm performance. *International Journal of Purchasing and Materials Management* 34, 2–9.
- Kim, I., de Weck, O., 2006. Adaptive weighted sum method for multiobjective optimization: a new method for pareto front generation. *Structural and Multidisciplinary Optimization* 31, 105–116.
- Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. *SCIENCE* 220, 671–680.
- Knowles, J., Corne, D., 1999. The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation, in: *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, p. 105 Vol. 1.

- Knowles, J.D., 2002. Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization. Ph.D. thesis. University of Reading.
- Konak, A., Coit, D.W., Smith, A.E., 2006. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety* 91, 992 – 1007. Special Issue - Genetic Algorithms and ReliabilitySpecial Issue - Genetic Algorithms and Reliability.
- Liao, S.H., Hsieh, C.L., Lai, P.J., 2011. An evolutionary approach for multi-objective optimization of the integrated location–inventory distribution network problem in vendor-managed inventory. *Expert Systems with Applications* 38, 6768 – 6776.
- Lu, M.M.S., Irajpour, A., Hamidi, N., Mavi, R.K., 2011. Designing excellent supply chain network in uncertainty enviroment with fuzzy logic. *International Journal of Business and Management* 6, 241 – 251.
- Mahnam, M., Yadollahpour, M.R., Famil-Dardashti, V., Hejazi, S.R., 2009. Supply chain modeling in uncertain environment with bi-objective approach. *Computers and Industrial Engineering* 56, 1535 – 1544.
- Mastrocinque, E., Yuce, B., Lambiase, A., Packianather, M., 2013. A multi-objective optimization for supply chain network using the bees algorithm. *International Journal of Engineering Business Management* 5, 1–11. doi:D0I: 10.5772/56754.
- Matnei Filho, R.A., Vergilio, S.R., 2016. A multi-objective test data generation approach for mutation testing of feature models. *Journal of Software Engineering Research and Development* 4, 4.
- Mendel, J., Hagrass, H., Tan, W.W., Melek, W.W., Ying, H., 2014. *Introduction To Type-2 Fuzzy Logic Control: Theory and Applications*. 1st ed., Wiley-IEEE Press.
- Mendel, J.M., 2001. *Uncertain rule-based fuzzy logic system: introduction and new directions*. 1st ed., Prentice–Hall PTR.
- Mendel, J.M., John, R., Liu, F., 2006. Interval type-2 fuzzy logic systems made simple. *IEEE T. Fuzzy Systems* 14, 808–821.

- Mendel, J.M., John, R.B., 2002. Type-2 fuzzy sets made simple. *Fuzzy Systems, IEEE Transactions on* 10, 117–127.
- Mentzer, J.T., DeWitt, W., Keebler, J.S., Min, S., Nix, N.W., Smith, C.D., Zacharia, Z.G., 2001. Defining supply chain management. *Journal of Business Logistics* 22, 1–25.
- Mercer, R., Sampson, J., 1978. Adaptive search using a reproductive meta-plan. *Kybernetes* 7, 215–228.
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. *Journal of Chemical Physics* 21, 1087–1092.
- Miettinen, K., 1999. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, MA.
- Miller, S., Gongora, M., Garibaldi, J., John, R., 2012. Interval type-2 fuzzy modelling and stochastic search for real-world inventory management. *Soft Computing* 16, 1447–1459. URL: <http://dx.doi.org/10.1007/s00500-012-0848-y>, doi:10.1007/s00500-012-0848-y.
- Miller, S., Gongora, M., John, R., 2011. Interval type-2 fuzzy modelling and simulated annealing for real-world inventory management, in: *Hybrid Artificial Intelligence Systems 2011 (HAIS2011)* 23-25 May, 2011 Wroclaw, Poland, pp. 231 – 238.
- Miller, S., John, R., 2010. An interval type-2 fuzzy multiple echelon supply chain model. *Knowledge-Based Systems* 23, 363 – 368.
- Miller, S., Popova, V., John, R., Gongora, M., 2008. Improving resource planning with soft computing techniques, in: *Proceedings of UKCI 2008*, De Montfort University, Leicester, UK.. pp. 37 – 42.
- Miller, S., Popova, V., John, R., Gongora, M., 2009. An interval type-2 fuzzy distribution network, in: *Proceedings of 2009 IFSA World Congress*, Lisbon, Portugal. pp. 697 – 702.
- Mohammaditabar, D., Ghodsypour, S.H., 2014a. A supplier-selection model with classification and joint replenishment of inventory items. *International Journal of Systems Science* 0, 1–10.

- Mohammaditabar, D., Ghodsypour, S.H., 2014b. A supplier-selection model with classification and joint replenishment of inventory items. *International Journal of Systems Science* 0, 1–10.
- Myers, R., Hancock, E.R., 2001. Empirical modelling of genetic algorithms. *Evol. Comput.* 9, 461–493.
- Nannen, V., Eiben, A., 2006. A method for parameter calibration and relevance estimation in evolutionary algorithms, in: *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, ACM, New York, NY, USA. pp. 183–190.
- Narukawa, K., Rodemann, T., 2012. Examining the performance of evolutionary many-objective optimization algorithms on a real-world application, in: *Proceedings of the 2012 Sixth International Conference on Genetic and Evolutionary Computing*, IEEE Computer Society, Washington, DC, USA. pp. 316–319.
- Nedjah, N., Mourelle, L.d.M., 2015. Evolutionary multi-objective optimisation: A survey. *Int. J. Bio-Inspired Comput.* 7, 1–25.
- Ordoobadi, S.M., 2009. Development of a supplier selection model using fuzzy logic. *International Journal of Supplier Chain Management* 14, 314 – 327.
- Özcan, E., Misir, M., Ochoa, G., Burke, E.K., 2010. A reinforcement learning - great-deluge hyper-heuristic for examination timetabling. *Int. J. of Applied Metaheuristic Computing* 1, 39–59.
- Parhizkari, M., Amiri, M., Mousakhani, M., 2013. A multiple criteria decision making technique for supplier selection and inventory management strategy: A case of multi-product and multi-supplier problem. *Decision Science Letters* 2, 185 – 190.
- Pattnaik, M., 2011. Supplier selection strategies on fuzzy decision space. *General Mathematics Notes* 4, 49 – 69.
- Peidro, D., Mula, J., Poler, R., Verdegay, J.L., 2009. Fuzzy optimization for supply chain planning under supply, demand and process uncertainties. *Fuzzy Sets and Systems* 160, 2640 – 2657.

- Petrovic, D., 2001. Simulation of supply chain behaviour and performance in an uncertain environment. *International Journal of Production Economics* 71, 429 – 438. Tenth International Symposium on Inventories.
- Petrovic, D., Roy, R., Petrovic, R., 1998. Modelling and simulation of a supply chain in an uncertain environment. *European Journal of Operational Research* 109, 299–309.
- Petrovic, D., Roy, R., Petrovic, R., 1999. Supply chain modelling using fuzzy sets. *International Journal of Production Economics* 59, 443 – 453.
- Ramanathan, R., 2007. Supplier selection problem: integrating dea with the approaches of total cost of ownership and ahp. *Supply Chain Management: An International Journal* 12, 258–261.
- Rao, R.V., 2007. Introduction to Multiple Attribute Decision-making (MADM) Methods. Springer London, London. pp. 27–41.
- Rezaei, J., Davoodi, M., 2008. A deterministic, multi-item inventory model with supplier selection and imperfect quality. *Applied Mathematical Modelling* 32, 2106 – 2116.
- Ross, T.J., 2004. Fuzzy Logic with Engineering Applications. John Wiley & Sons.
- Roy, R., 2010. A Primer on the Taguchi Method, Second Edition:. Society of Manufacturing Engineers.
- Saaty, R., 1987. The analytic hierarchy process—what it is and how it is used. *Mathematical Modelling* 9, 161 – 176.
- Saaty, T., Gass, S., 1954. Parametric objective function (part 1). *Journal of the Operations Research Society of America* 2, 316–319.
- Sadeghi, J., Sadeghi, S., Niaki, S.T.A., 2014. A hybrid vendor managed inventory and redundancy allocation optimization problem in supply chain management: An nsga-ii with tuned parameters. *Computers and Operations Research* 41, 53 – 64.
- Sanayei, A., Mousavi, S.F., Abdi, M., Mohaghar, A., 2008. An integrated group decision-making process for supplier selection and order allocation using multi-attribute utility theory and linear programming. *Journal of the Franklin Institute* 345, 731 – 747.

- Sarkar, A., Mohapatra, P.K., 2006. Evaluation of supplier capability and performance: A method for supply base reduction. *Journal of Purchasing and Supply Management* 12, 148 – 163.
- Sarker, R., Coello Coello, C., 2002. Assessment methodologies for multiobjective evolutionary algorithms, in: *Evolutionary Optimization*. Springer US. volume 48 of *International Series in Operations Research & Management Science*, pp. 177–195.
- Sarkis, J., Talluri, S., 2002. A model for strategic supplier selection. *Journal of Supply Chain Management* 38, 18–28.
- Saunders, M.J., 1997. *Strategic Purchasing and Supply Chain Management*. Pitman Publishing, London.
- Schaffer, J.D., 1985. Multiple objective optimization with vector evaluated genetic algorithms, in: *Proceedings of the 1st International Conference on Genetic Algorithms*, L. Erlbaum Associates Inc., Hillsdale, NJ, USA. pp. 93–100. URL: <http://dl.acm.org/citation.cfm?id=645511.657079>.
- Seada, H., Deb, K., 2015. Effect of selection operator on nsga-iii in single, multi, and many-objective optimization., in: *CEC, IEEE*. pp. 2915–2922.
- Serafini, P., 1994. *Simulated Annealing for Multi Objective Optimization Problems*. Springer New York, New York, NY. pp. 283–292.
- Setak, M., Sharifi, S., madian, A.A., 2012. Supplier selection and order allocation models in supply chain management: a review. *World Applied Sciences Journal* 18, 55 – 72.
- Shaik, M., Abdul-Kader, W., 2011. Green supplier selection generic framework: a multi-attribute utility theory approach. *International Journal of Sustainable Engineering* 4, 37–56.
- Shankar, B.L., Basavarajappa, S., Kadadevaramath, R.S., Chen, J.C., 2013. A bi-objective optimization of supply chain design and distribution operations using non-dominated sorting algorithm: A case study. *Expert Systems with Applications* 40, 5730 – 5739.

- Sierra, M.R., Coello, C.A.C., 2004. A new multi-objective particle swarm optimizer with improved selection and diversity mechanisms, in: *Proceeding of the 2004 Congress on Evolutionary Computation*, pp. 1–39.
- Sörensen, K., Glover, F.W., 2013. *Metaheuristics*. Springer US, Boston, MA. pp. 960–970.
- Srinivas, N., Deb, K., 1994. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.* 2, 221–248.
- Steuer, R., Choo, E.U., 1983. An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical Programming* 26, 326–344. doi:10.1007/BF02591870.
- Suman, B., 2003. Simulated annealing-based multiobjective algorithms and their application for system reliability. *Engineering Optimization* 35, 391–416.
- Suman, B., 2005. Study of self-stopping pdmosa and performance measure in multiobjective optimization. *Computers and Chemical Engineering* 29, 1131 – 1147.
- Suppapitnarm, A., Seffen, K.A., Parks, G.T., Clarkson, P.J., 2000. A simulated annealing algorithm for multiobjective optimization. *Engineering Optimization* 33, 59–85.
- Taguchi, G., Chowdhury, S., Wu, Y., 2005. *Taguchi's Quality Engineering Handbook*. Wiley.
- Taguchi, G., Yokoyama, Y., 1993. *Taguchi methods: design of experiments*. TAGUCHI METHODS SERIES, ASI Press.
- Tahriri, F., Osman, M.R., Ali, A., Yusuff, R.M., Esfandiary, A., 2008. Ahp approach for supplier evaluation and selection in a steel manufacturing company. *Journal of Industrial Engineering and Management* 1, 54 – 76.
- Thomas, D.J., Griffin, P.M., 1996. Coordinated supply chain management. *Eur. J. Oper. Res.* 94, 1 – 15.
- Turk, S., John, R., Özcan, E., 2014. Interval type-2 fuzzy sets in supplier selection, in: *14th UK Workshop on Computational Intelligence UKCI2014*, pp. 1–7.

- Turk, S., Miller, S., Özcan, E., John, R., 2015. A simulated annealing approach to supplier selection aware inventory planning, in: IEEE Congress on Evolutionary Computation, CEC 2015, Sendai, Japan, May 25-28, 2015, pp. 1799–1806.
- Ulungu, E., Teghem, J., Fortemps, P., Tuyttens, D., 1999. MOSA method: a tool for solving multiobjective combinatorial optimization problems. *Journal of Multi-Criteria Decision Analysis* 8, 221–236.
- Ulungu, E., Teghem, J., Ost, C., 1998. Interactive simulated annealing in a multiobjective framework: application to an industrial problem. *Journal of the Operational Research Society* 49, 1044–1050.
- Ustun, O., Demirtas, E.A., 2008a. An integrated multi-objective decision-making process for multi-period lot-sizing with supplier selection. *Omega* 36, 509 – 521. Special Issue on Logistics: New Perspectives and Challenges.
- Ustun, O., Demirtas, E.A., 2008b. Multi-period lot-sizing with supplier selection using achievement scalarizing functions. *Computers & Industrial Engineering* 54, 918 – 931.
- Van Veldhuizen, D.A., Lamont, G.B., 2000. Multiobjective optimization with messy genetic algorithms, in: *Proceedings of the 2000 ACM Symposium on Applied Computing - Volume 1*, ACM, New York, NY, USA. pp. 470–476.
- Velasquez, M., Hester, P.T., 2013. An analysis of multi-criteria decision making methods. *International Journal of Operations Research* 10, 56 – 66.
- Veldhuizen, D.A.V., 1999. Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. Technical Report. *Evolutionary Computation*.
- Vitasek, K.e., 2010. Supply chain management terms and glossary, *Supply Chain Visions*.
- Voß, T., Hansen, N., Igel, C., 2010. Improved step size adaptation for the mo-cma-es, in: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, ACM, New York, NY, USA. pp. 487–494.
- Wang, J.W., Cheng, C.H., Huang, K.C., 2009. Fuzzy hierarchical topsis for supplier selection. *Applied Soft Computing* 9, 377 – 386.

- Weber, C.A., 1996. A data envelopment analysis approach to measuring vendor performance. *Supply Chain Management: An International Journal* 1, 28–39.
- Weber, C.A., Current, J., Desai, A., 2000. An optimization approach to determining the number of vendors to employ. *Supply Chain Management: An International Journal* 5, 90–98.
- Wu, D., Tan, W.W., 2005. Type-2 fls modeling capability analysis, in: *The 14th IEEE International Conference on Fuzzy Systems*, pp. 242 – 247.
- Wu, Z., Chen, Y., 2007. The maximizing deviation method for group multiple attribute decision making under linguistic environment. *Fuzzy Sets Systems* 158, 1608–1617.
- Xie, Y., Petrovic, D., Burnham, K., 2006. A heuristic procedure for the two-level control of serial supply chains under fuzzy customer demand. *International Journal of Production Economics* 102, 37 – 50.
- Xiujuan, L., Zhongke, S., 2004. Overview of multi-objective optimization methods. *Journal of Systems Engineering and Electronics* 15, 142–146.
- Yuan, Y., Xu, H., Wang, B., 2014. An improved nsga-iii procedure for evolutionary many-objective optimization, in: *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, ACM, New York, NY, USA. pp. 661–668.
- Zadeh, L., 1965. Fuzzy sets. *Information and Control* 8, 338 – 353.
- Zadeh, L., 1975. The concept of a linguistic variable and its applications to approximate reasoning. *Inform Science* 8, 199 – 249.
- Zandhessami, H., Zonozi, M., Afshari, M., 2012. Optimizing a multi-echelon supply chain network flow using nonlinear fuzzy multi-objective integer programming: Genetic algorithm approach. *Management Science Letters* 2, 1871–1884.
- Zarandi, M.H.F., Zarani, M.M.F., Saghiri, S., 2007. Five crips and fuzzy models for supply chain of an automotive manufacturing system. *International Journal of Management Science and Engineering Management* 2, 178 – 196.
- Zeng, F., Low, M.Y.H., Decraene, J., Zhou, S., Cai, W., 2010. Self-adaptive mechanism for multi-objective evolutionary algorithms, in: *International Multi-Conference of Engineers and Scientists*, Hong Kong, March 17-19, 2010, pp. –.

- Zhang, Q., Li, H., 2007. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *Trans. Evol. Comp* 11, 712–731.
- Zhang, S., Lee, C.K.M., Wu, K., Choy, K.L., 2016. Multi-objective optimization for sustainable supply chain network design considering multiple distribution channels. *Expert Systems with Applications* 65, 87 – 99.
- Zhang, W., Fujimura, S., 2010. Improved vector evaluated genetic algorithm with archive for solving multiobjective pps problem, in: *E-Product E-Service and E-Entertainment (ICEEE)*, 2010 International Conference on, pp. 1–4.
- Zhao, J., Bose, B.K., 2002. Evaluation of membership functions for fuzzy logic controlled induction motor drive, in: *IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02*, pp. 229–234 vol.1.
- Zitzler, E., 1999. Evolutionary algorithms for multiobjective optimization: Methods and applications.
- Zitzler, E., Brockhoff, D., Thiele, L., 2007. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration, in: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (Eds.), *Evolutionary Multi-Criterion Optimization*. Springer Berlin Heidelberg. volume 4403 of *Lecture Notes in Computer Science*, pp. 862–876.
- Zitzler, E., Deb, K., Thiele, L., 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.* 8, 173–195.
- Zitzler, E., Künzli, S., 2004. Indicator-Based Selection in Multiobjective Search, in: Yao, X., et al. (Eds.), *Conference on Parallel Problem Solving from Nature (PPSN VIII)*, Springer. pp. 832–842.
- Zitzler, E., Laumanns, M., Thiele, L., 2002. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization, in: *Evolutionary Methods for Design, Optimisation, and Control, CIMNE*, Barcelona, Spain. pp. 95–100.
- Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *Evolutionary Computation, IEEE Transactions on* 3, 257–271.

-
- Ziukov, S., 2015. A literature review on models of inventory management under uncertainty. *Business Systems and Economics* 5, 26 – 35.