DESIGN OF A MULTI-ECHELON SUPPLY CHAIN NETWORK USING MULTI-OBJECTIVE DECISION MAKING APPROACH

SAYEM AHMED

DEPARTMENT OF INDUSTRIAL AND PRODUCTION ENGINEERING (IPE)

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY (BUET)
DHAKA, BANGLADESH

13 August, 2017

DESIGN OF A MULTI-ECHELON SUPPLY CHAIN NETWORK USING MULTI-OBJECTIVE DECISION MAKING APPROACH

By

SAYEM AHMED

A Thesis Submitted to the
Department of Industrial and Production Engineering,
Bangladesh University of Engineering and Technology
in Partial Fulfillment of the requirements for the Degree of

MASTER OF SCIENCE IN INDUSTRIAL and PRODUCTION ENGINEERING

DEPARTMENT OF INDUSTRIAL AND PRODUCTION ENGINEERING (IPE)

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY (BUET)
DHAKA, BANGLADESH

13 August, 2017

CERTIFICATE OF APPROVAL

The thesis entitled as **"Design of a Multi-Echelon Supply Chain Network using Multi-Objective Decision Making Approach"** submitted by Sayem Ahmed, Student No. 1015082019, Session- October 2015, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M.Sc. in Industrial and Production Engineering on August 13, 2017.

BOARD OF EXAMINERS

Dr. M. Ahsan Akhtar Hasin — Chairman (Supervisor)
Professor
Department of IPE, BUET, Dhaka

Dr. Nafis Ahmad — Member (Ex- Officio)
Professor & Head
Department of IPE, BUET, Dhaka

Dr. Syed Mithun Ali — Member
Assistant Professor
Department of IPE, BUET, Dhaka

Dr. A. F. M. Anwarul Haque — Member (External)
Professor,
Department of Mechanical and Production Engineering, AUST, Dhaka

CANDIDATE'S DECLARATION

It is hereby declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Sayem Ahmed

This work is dedicated to

My Loving Parents & My Family

ACKNOWLEDGEMENT

All credits go to Allah, the most benevolent and the Almighty, for his boundless grace in successful completion of this thesis.

At the very beginning the author would like to expresses his sincere gratitude and respect to honorable teacher & thesis supervisor, Dr. M. Ahsan Akhtar Hasin, Professor, Department of Industrial and Production Engineering (IPE), Bangladesh University of Engineering and Technology (BUET), Dhaka, for his thoughtful suggestions, timely advice, constant guidance and encouragement throughout the progress of this research work.

The author also expressed his sincere gratitude to Dr. Nafis Ahmad, Professor and Head, Department of IPE, BUET, Dr. Syed Mithun Ali, Assistant Professor, Department of IPE, BUET and Dr. A. F. M. Anwarul Haque, Professor, Department of Mechanical and Production Engineering, AUST, Dhaka for their constructive remarks and for kindly evaluating this research.

The author is very much grateful to Ms. Ineen Sultana, Mr. Saiful Islam and Mr. Mamunur Rahman for their kind support and guidance from abroad.

The author also thankful to Monowar Anjum, undergraduate student, CSE, BUET for his constant co-operation and assistance in developing the algorithm and computer coding. The author also expresses his heartiest thanks to his students M. M. Mahbubur Rahman Sajid and Md. Nazmus Sakib Arik, undergraduate student, IPE, AUST for their support to complete this research work.

The author is indebted to the officials especially the management, Chief Engineer and other high ranked officers of Walton Hi-tech Industries for providing necessary data and opinions.

The author is grateful to all the writers and publishers of the books and journal papers that have taken as references while conducting this research.

Finally, the author would like to thank all of his colleagues of the Department of Mechanical and Production Engineering, AUST and friends for their cooperation and motivation to complete the thesis timely. And the author would also like to extend his thanks to his parents whose continuous inspiration, sacrifice and support encouraged him to complete the thesis successfully.

ABSTRACT

A supply chain is a set of facilities, suppliers, customers, products and methods of controlling inventory, purchasing, and distribution. In a supply chain, the flow of goods between a supplier and customer passes through several stages, and each stage may consist of many facilities.

The planning is designed to integrate supplier selection, product assembly, distribution center (DC) location selection as well as the logistic distribution system of the supply chain in order to meet market demands. In this research, the supplier selection problem is integrated with production decision and distributor location problems and a new mathematical model is proposed. While the most successful companies are aiming for total customer satisfaction, it is important to quantify service quality of suppliers so Taguchi loss function is considered in decision making process for calculating the loss in monetary terms, in case of poor supply performance.

With multiple suppliers and multiple customer needs, the assembly model can be divided into several sub-assembly steps by applicable sequence. Considering three evaluation criteria, namely costs, delivery time, and quality a multi-objective optimization mathematical model is established in this study. The multi-objective problems usually have no unique optimal solution, and the Pareto genetic algorithm (PaGA) can find good tradeoffs among all the objectives. Therefore, this study proposes a modified Pareto genetic algorithm (mPaGA) to improve the solution quality through revision of crossover and mutation operations. The results show that the proposed algorithm gives high quality solutions as well as better computational times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SCM	:	Supply Chain Management
SC	:	Supply Chain
BOSC	:	Build-to-Order Supply Chain
SCND	:	Supply Chain Network Design
MSCN	:	Multi-Stage Supply Chain Network
DC	:	Distribution Center
MOO	:	Multi-Objective Optimization
MOOP	:	Multi-Objective Optimization Problem
GA	:	Genetic Algorithm
MOGA	:	Multi-Objective Genetic Algorithm
PaGA	:	Pareto Genetic Algorithm
tPaGA	:	Traditional Pareto Genetic Algorithm
mPaGA	:	Modified Pareto Genetic Algorithm
DM	:	Decision Maker
LP	:	Linear Programming
MATLAB	:	Matrix Laboratory
QLF	:	Quality Loss Function
SN	:	Signal to Noise

Chapter-1

INTRODUCTION

1.1 INTRODUCTION

A supply chain is a set of facilities, supplies, customers, products and methods of controlling Inventory, purchasing, and distribution. The chain links suppliers and customers, beginning with the production of raw material by a supplier, and ending with the consumption of a product by the customer. In a supply chain, the flow of goods between a supplier and a customer passes through several echelons, and each echelon may consist of many facilities.

The problem of supply chain network design is very broad and means different things to different enterprises. It generally refers to a strategic activity that will take one or more of the following decisions:

- Where to locate new facilities (production, storage, logistics, etc.)?
- Significant changes to existing facilities, e.g. expansion, and contraction or closure.
- Sourcing decisions (what suppliers and supply base to use for each facility).
- Allocation decisions (e.g., what products should be produced at each production facility)
- Which markets should be served by which warehouses, etc.

Logistics network design is one of the most important strategic decisions in supply chain management. Decisions on sourcing, the number of facilities, their locations and capacities and the quantity of flow between them affect both costs and customer service levels. As such, effective and efficient supply chain design can constitute a sustainable competitive advantage for firms. Since opening and closing a facility is both an expensive and time-consuming process, changing network design is impossible in the short run. Because tactical and operational decisions are conditional upon strategic decisions, the configuration of logistics networks will become a constraint for tactical and operational level decisions. Transportation which is related with logistics management means the flow of products or information in right time, in the right quantity (at the right cost) to the right place. The flow of goods between a supplier and a customer passes through several echelons, and each echelon may consist of many facilities. An optimized Transportation system helps to earn company's profits and growths and reduce the distortion of information in each echelon. This is normally achieved when the company manages to reduce transportation time and transportation costs, to increase service level, to limit risks connected to the supply chain, to gain a vision of the supply chain and tools to control it, or to foresee future changes and to adapt to them. Transportation Network is vital which starts from carrying raw materials, finished or semi-finished goods to manufacturer, finished goods to warehouse, retailers and at the end to customer and ending with customer returns. There are lot more probable parameters out there so this network should be solved by considering uncertain circumstances.

Supply chains are networks of logistic and manufacturing activities starting with raw material sourcing and ending with the distribution of finished goods to markets. The performance of a supply chain for a given product-market critically depends on the structure of its production distribution network (i.e. the number, location, mission, technology and capacity of the facilities of the firms involved), the quality of suppliers, allocation decisions and other sourcing decisions. The exact nature of the logistics network design problems encountered in practice depends very much on the industrial context in which they occur.

1.2 RATIONALE OF THE STUDY

Today, in the competitive markets, supply chain works as an interrelating network of suppliers, manufacturers, distributors, and customers, to satisfy customer demands. In 21st century, supply chain management (SCM) is a key strategy to improve competitive edge. The build-to-order supply chain (BOSC) model is a key operation model for providing services/products at present. This study focuses on performing the supply chain planning for the BOSC network. In BOSC model, production activities are not executed until receiving orders from customers, that can effectively reduce the costs of demand prediction and inventory and credibly reflect market demands.

Today, product delivery to customers in a suitable time with desirable quality and minimum cost is a complicated process that needs several internal and external organizational transactions. Since efficiency and responsiveness are two generic strategies for supply chain network design, coordination of these transactions is an important issue. According to Liao et al. (2011), the aim of efficiency is to optimize costs and quality and responsiveness aims to satisfy customer demands quickly. Coordination in the supply chain is a process that leads to planning, cooperation, and control of material, parts, and final products in all levels of supply chain including strategic, tactical, and operational levels (Stevens 1989). In the strategic layer, supplier selection is one of the most crucial decisions that affect the whole performance of supply chain.

As BOSC is started, supplier selection becomes the priority. After a proper part supplier is selected, product assembly begins. Consequently, we should select suitable assembly planning and a qualified assembly factory, and distribute the assembled products in accordance with customer requirements.

Supplier selection decisions become more important when the other strategic decisions such as locating of distribution centers (DCs) and production planning should be made simultaneously. In the previous studies on supply chain, most of researchers considered supplier selection, product assembly, and DC locating problems separately (e.g., Che 2010, Glickman and White 2008, Gunasekaran and Ngai 2009, Ha and Krishnan 2008, Sawik 2010). Integrating different operations in supply chain results in a decision-making process in which two or more conflicting objectives should be considered. Therefore, multi-criteria decision-making techniques can be used as well-known tools. In terms of supplier selection, Dickson (1966) proposed 23 criteria for supplier selection in his research and identified quality, delivery, and performance history as the three major criteria. Several criteria may affect on the performance of suppliers, hence Weber et al. (1991) reviewed many articles which considered more than one criterion such as cost, delivery time, and quality. Wadhwa and Ravindran (2007) and Che and Chiang (2010) used cost, quality, and time for evaluation criteria. In this research, not only these criteria are considered for supplier selection, but also extend to production planning and DC locating of entire supply chain to form a multi-objective problem. Furthermore, this study considers the problem of assigning the DCs to customers too. On the other hand, a novel multiple objective mathematical model including minimization of cost and delivery time and maximization of quality is presented. The model integrates strategic decisions with tactical ones considering capacity constraints, supply and demand equilibrium, and flow balance. Che and Chiang (2010) demonstrated that a multi-echelon logistics system with assembly problem is a non-deterministic polynomial (NP)-hard problem. It becomes even more complex when the location of DCs is considered. Coordinating all parts of the supply chain is a multilevel process and needs important decisions that affect on the performance of each part and whole supply chain and some conflicting objectives should be considered. Simaria and Vilarinho (2004) argued that the genetic algorithm (GA) can be used to effectively solve the assembly line balancing problem. Sha and Che (2006), Altiparmak et al. (2006), and Xu et al. (2008) had applied GA to deal with supply chain problems. In addition, Poulos et al. (2001) and Hosung et al. (2005) have successfully employed PaGA for dealing with the multi-objective optimization problems. The multi-stage supply chain network considered in this research consists of four stages: supplier, manufacturer, retailer and customer. The problem deals with determining the optimal transportation network in order to satisfy the customer demands of several products by selecting the best suppliers and location of distribution centers with the minimum transportation cost, delivery time and quality.

While the most successful companies are aiming for total customer satisfaction, it is important to quantify service quality of suppliers (Li 2003, Liao and Kao 2010). Taguchi loss function was used to determine the potential losses and measure the service performances (Magdalena 2012, Sharma and Balan 2013). Integration of the Taguchi loss function and TOPSIS method to select an optimal supplier in supply chain management was proposed by Lavanpriya et al. (2013).

Therefore, this research aims to develop a mathematical model to minimize total cost and delivery time along with maximizing quality level and service performance using Taguchi quality loss function, for supplier selection and DC allocation.

So, a modified pareto genetic algorithm (mPaGA) based on pareto genetic algorithm (PaGA) is applied to deal with multi-objective, multi-echelon supply chain optimization model to select the right supplier and to allocate product volumes to right distribution centers. In the proposed algorithm, the weighting method is used to make different importance for evaluation criteria and convert the multi-objective problem to a single objective optimization one.

1.3 OBJECTIVES WITH SPECIFIC AIMS

The specific objectives of this research are as follows:

- To formulate a multi-echelon supply chain model to select the right supplier and to allocate product volumes to right distribution centers for minimum distribution of time and cost
- To incorporate Taguchi loss function in decision making process for calculating the loss in monetary terms, in case of poor supply performance
- To solve the model using a suitable meta-heuristic to achieve Pareto optimal solution

1.4 OUTLINE OF METHODOLOGY

In order to carry out this research work, steps that have been adopted are mentioned below:

- The current distribution network, along with performance measures have been studied
- Current research works have been studied as to know how similar cases were solved, in different demographic patterns, and identify the limitations and scopes for improvements
- A multi-objective model has been developed to design an improved supply chain network in case of Bangladesh market and local consumers
- Taguchi loss function has been incorporated to identify loss in terms of cost for poor service performance
- The model has been solved through programming in JAVA using Genetic Algorithm, to minimize total cost (fixed infrastructure cost, purchase cost, ordering cost, transportation cost, poor quality penalty cost), and delivery time, and maximize quality.
- Data regarding appropriate supply chain parameters and variables, such as suppliers' maximum capacity, ordering cost, procurement cost, quality of each supplier, internal and external transportation cost, etc. have been identified and collected
- A set of solutions have been investigated, each of which satisfies the objectives at an acceptable level without being dominated by any other solution
- An entire Pareto optimal solution set has been determined
- A single solution that satisfies the subjective preferences of a human decision maker has been found

- The combination of suppliers and product volumes have been determined to transport through the supply chain network

1.5 LIMITATIONS OF THE STUDY

Supply chain is a very complicated system. It includes many parameters and critical partnership phenomena. The stake holding matters are there to affect the operation of the supply chain. Collective decision making along with local targets of individual partner affects every day's decisions. The trading partners are located at different socio-economic environment. The trading policies framed at corporate level are sometimes overlooked. Hundreds of conflicts that arise in the network cannot be addressed with mathematical formulations only. The demand distribution of this particular product is greatly affected by seasonality factors. There are peak demand seasons and off-peak demand and moderate demand seasons exist practically. There exists some uncertainty in principal raw material procurement and this produces some capacity limitations for the production unit but it was assumed that the capacity is ample and shortages are not permitted in all echelons of supply chain to support the distribution supply chain. Product assembly plans are pre-known and suppliers do not require to produce all types of parts and may produce some of them.

1.6 ORGANIZATION OF THE THESIS

The thesis is organized into seven chapters with a list of references & appendices.

Chapter 1 entitled as "Introduction" which describes about the general concepts of supply chain management. Some motivational information & rationale behind this thesis work is also discussed on this chapter. The research objectives and outcomes are also outlined here with some guideline of research methodologies.

Chapter 2 discusses the literature review of supply chain, BOSC, Taguchi quality loss function, multi-objective supply chain modeling, genetic algorithm, PaGA. In this chapter, a details review on the literature is delivered which gives a germane idea on the background and gaps in this research area. This provides the main inspiration of this research work.

The theoretical background of supply chain network design, taguchi quality loss function, signal to noise ratio, multi-objective optimization etc. are discussed in Chapter 3 entitled as “Theoretical Background”. Different Genetic algorithm parameters, basic information regarding GA are also focused in this chapter.

Chapter 4 presents a mathematical formulation and detail description of the model for multi-objective supply chain considering supplier selection, locating DCs, allocation, and manufacturing planning. Assumptions, set of indices, parameters, variables, objective functions and constraints are explained in this chapter.

In Chapter 5, how the models are implemented is discussed and introduced the solution techniques that can be applied to solve this novel model. The proposed mPaGA algorithm is explained in this section.

Chapter 6 is the most crucial part which discusses and analyzes the result obtained for a hypothetical example of the model formulated in chapter 4 through the solution technique chosen in chapter 5.

Chapter 7 lists the contributions of this research, summarizes the conclusions, limitations of the proposed model, and suggestions for further research which are followed by a list of references & appendices focused on the programming languages for the proposed model.

Chapter-2

LITERATURE REVIEW

In this chapter, the theoretical background of the literature research is explained.
In order to have a better overview of the foundations, several broader concepts that are important for the research presented. Having these concepts explained, in can be passed towards the core literature review. Thus, the theoretical foundations include:

1. Supply Chain Management (SCM)
2. Facility Location and Supply Chain Network Design
3. Taguchi Quality Loss Function
4. Multi-Objective Optimization Models
5. Genetic Algorithms and Pareto Genetic Algorithms
6. Multi-Echelon Multi-Objective Decision Making Approach

The first point of the theoretical foundations should give a quick overview on what is supply chain management, why it is important, and what are the main aspect encompassing. After having a basic idea on SCM, the idea of sustainability in SCM can be presented. Here it moves towards the scope of the thesis, and by explaining the major concepts of sustainability, the basis for the other components of the foundations is set. Hence, explaining the facility location problem and supply chain network design is the last topic that should be covered in order to have a comprehensive picture, and various multi-stage, multi-objective optimization techniques are discussed before entering the core literature review part. The last point of the theoretical background is a crucial aspect for the research, where a specific approach is followed in order to structure and analyze the literature. Having that, the main research gaps can be identified, and form the research questions that should be answered in the following chapters.

2.1 SUPPLY CHAIN MANAGEMENT (SCM)

A supply chain is an integrated process which includes all activities associated with the flow and transformation of goods from raw material stage through to the end user. It also involves the integration of transformation of information which flows up and down the supply chain (Chopra and Meindle, 2003). It is a set of approaches utilized to efficiently integrate suppliers, manufactures, warehouses, and stores so that merchandise is produced and distributed at the right quantities, to the right locations and at the right time in order to minimize system wide costs while satisfying service level requirements (Simchi-Levi et al., 2002).

The supply chain activities begin with a customer order and end when a satisfied customer has paid for the purchase. Supply chain management (SCM) is the term used to describe the management of the flow of materials, information, and funds across the entire supply chain, from suppliers to component producers to final assemblers to distribution (warehouses and retailers), and ultimately to the consumer. In fact, it often includes after-sales service and returns or recycling (Beamon, 1998).

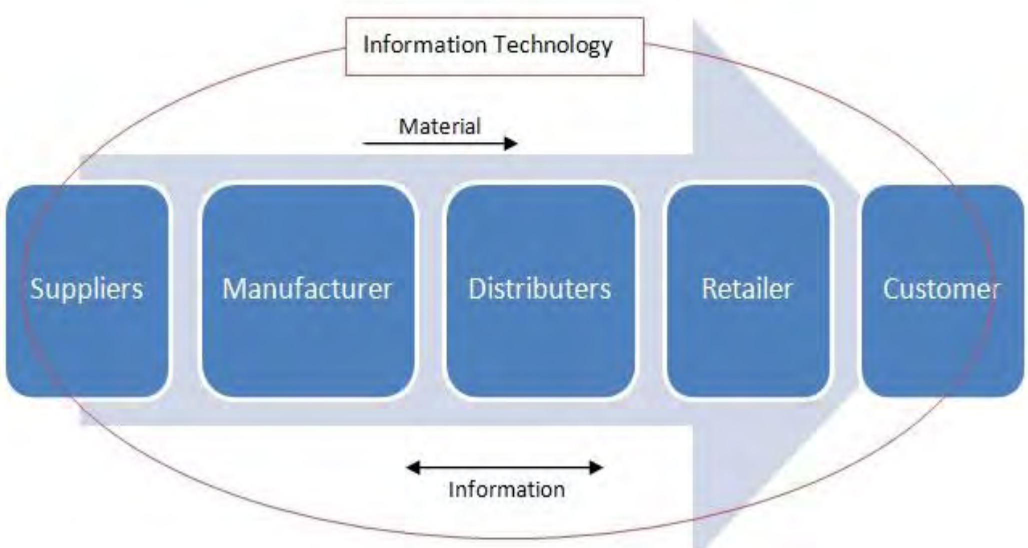


Figure 2.1: Different Stages of Supply Chain (Chopra and Meindle, 2003)

In modern age, supply chains are networks of firms who use their capabilities and resources in order to deliver value to the end consumer. No firms are able to own or control completely their own supply chains without considering other. Globalization, information technology and modern logistics capabilities have created a global market where companies can take advantage of the opportunity to source internationally. Thus, companies have specialized and partnered globally with other companies. These companies have then to increasingly focus on logistics and supply chain coordination. Such coordination is an essential business process nowadays. Modern supply chain management does not concentrate on optimizing own objectives as this self-serving focus often results in poor performance. However, a sequence of local optimum policies does not bring about a globally optimum solution. Munson et al. (2003) summarize it as follows ―When each member of a group tries to maximize his or her own benefit without regard to the impact on other members of the group, the overall effectiveness may suffer. Such inefficiencies often creep in when rational members of supply chains optimize individually instead of coordinating their efforts.

Supply chain professionals play major roles in the design and management of supply chains. In the design of supply chains, they help determine whether a product or service is provided by the firm itself (insourcing) or by another firm elsewhere (outsourcing). In the management of supply chains, supply chain professionals coordinate production among multiple providers, ensuring that production and transport of goods happen with minimal quality control or inventory problems. One goal of a well-designed and maintained supply chain for a product is to successfully build the product at minimal cost and supply the product at minimal transportation time with maximum customer service performance. Such a supply chain could be considered a competitive advantage for a firm.

There are many decisions to be made in supply chains. These include-

- What products to make and what their designs should be;
- How much, when, where and from whom to buy product;
- How much, when and where to produce product;
- How much and when to ship from one facility to another;
- How much, when and where to store product;
- How much, when and where to charge for products; and
- How much, when and where to provide facility capacity.

A supply chain network can be strategically designed in such a way as to reduce the cost and time of the supply chain; it has been suggested by experts that 80% of supply chain costs are determined by location of facilities and the flow of product between the facilities. Supply chain network design is sometimes referred to as 'Network Modeling', due to the fact a mathematical model can be created to optimize the supply chain network.

2.1.1 Build-to-Order Supply Chain (BOSC)

The concept of supply chain management (SCM), first developed by Houlihan (1985), is a key development in logistics. In the initial development, it only dealt with physical distribution and transportation operations with industrial dynamic technology. Jones and Riley (1985) defined SCM as an integrative approach to deal with the planning and control of the materials flow from suppliers to end-users. Gunasekaran et al. (2008) pointed out that SCM is the most important business strategy for increasing organizational competence in the 21st century. Prasad et al. (2005) argued that today's supply chains are increasingly moving towards a build-to-order (BTO) environment from a Made to Stock (MTS) one as a way to gain competitive edge. They examined the differences of SCM in BTO and MTS, and suggested that MTS was transformed into BTO must focus on three dimensions: information complexity, operational independence, and supplier integration. The BTO strategies have pure customization, and in order to meet the particular customer needs that the total quantity needs to be on hand. Gunasekaran (2005) figured that a successful business should not only avoid production activities lacking economic benefits, but also create marketing capabilities that are not imitable. In particular, BOSC is a supply chain model designed for various industries and can be used to address market demands and offer cost-effectiveness. According to Krajewski et al. (2005), the production model has already shifted from mass production to mass customization over a wide spectrum of products, ranging from personal computers, bikes, to automobiles, which are made in the BOSC model, where customers may select parts assembly depending on their preference. Howard et al. (2006) proposed that build-to-order (BTO) has been hailed as a production strategy that fits the demands of the 21st century, where a considerable challenge for the industry is addressed. Based on the findings of these literatures, BOSC is a successful supply chain model that is currently widely in use. Achieving the purpose of SCM in this model becomes a critical issue. Consequently, this study proceeds to manage supplier selection, assembly planning, and logistics distribution in the BOSC model.

2.2 FACILITY LOCATION AND SUPPLY CHAIN NETWORK DESIGN

Distribution network design consist of determining the best way to transfer goods from the supply to the demand points by choosing the structure of the network (layers, different kinds of facilities operating at different layers, their number and their location), while minimizing the overall costs. Distribution network design problems involve strategic decisions which influence tactical and operational decisions. In particular, they involve facility location, transportation and inventory decisions, which affect the cost of the distribution system and the quality of the customer service level. So, they are core problems for each company.

Emerging from the above discussion, distribution network design problems involve a lot of integrated decisions, which are difficult to consider all together. Generally, some simplifying assumptions have been adopted in the literature, and only some aspects related to the complex network decisions have been modeled. For instance, in the past some authors dealt with distribution network design problems as pure location problems, without trying to address and integrate the different types of strategic decisions.

Supply chain networks provide the infrastructure for the production, storage, and distribution of products as varied as pharmaceuticals, vehicles, computers, food products, furniture, and clothing, throughout the globe. Hence, the design of supply chain networks is a topic of engineering importance since it involves the determination of both the sites and the levels of operation of the relevant facilities that enable the manufacture, storage, and delivery of products to the consumers.

Simultaneously, sustainability of supply chains has emerged as a major theme in both research and practice since the impacts of climate change have made both producers and consumers more cognizant of their decision-making and how their decisions affect the environment. Decisions regarding SCN optimization and design are highly concerned with every single part of a chain such as production facilities, distributer centers, suppliers and customers and every type of flow and connection between these nodes of the network. The initial phase of the strategic planning is the definition of the type of network that should be designed. Depending on the business needs, various types of networks can be identified. Thus, within a supply chain, one can distinguish between production network and distribution network. The first includes the operational facilities for production and inventory of products, while the distribution network includes inventory and distribution facilities for serving the customers. Anyway, the current researching on facility locations is not taking a single perspective, but is including models that are mixing the decision factors for both production and distribution network design. The reasons behind stands in the needs of the markets and the strategies of the companies, which want to grasp as much value as possible by developing multifunctional networks globally. As a matter of simplification and unification, the term ―Supply Chain Network Design is used, but the focus is on production network design (even though taking into considerations distribution network design aspects). In a SCND, there are possibilities of different facility locations globally, according to the customers that need to be served, and according to the suppliers that are supposed to serve the facilities. Supply chain networks and strategic planning can be done in both qualitative and quantitative ways. However, the interest of this thesis is the modeling of the supply chain problems from a mathematical point of view. Thus, in order to address the facility location problem and the wider problem of SCND, several mathematical models are presented for explaining the basics. These models are part of Operations Research, and through the time, researches developed both single and multi-objective ones. The traditional models are always cost - based, and taking into considerations the minimization of the costs as one single objective. Another typical single objective is maximization of profits. However, there is a global recognition that facility locations are not a single objective problem, and the models evaluating them should be improved by including different factors regarding SCND Flexibility, lead time, sustainability, customer satisfaction, or quality are some of those factors that can be included in multi-objective models. The multi-objective mathematical models have a goal of finding a set of non-dominating solutions that with higher strategic decisions can lead to a more comprehensive decision-making process.

In the recent years, the multi-stage supply chain network (MSCN) design problem has received a great deal of attention due to the increasing competitiveness generated by the overflowing market globalization. This competitiveness forces the companies to assure their customers a high level of service and, at the same time, to control costs and maximize profits along the overall supply chain in order to hamper their competitors' rise. The network design problem plays a key role in the long-term strategic decision-making that managers need to optimize to keep efficient the whole supply chain. The outcome of a MSCN design problem is usually twofold:

i. identifying the optimal number and the location of the manufacturing plants and distribution centers to be added to each stage of the network;
ii. selecting the optimal logistic transportation trees representing the flow of the optimal quantities of raw materials, semi-finished and final products from suppliers to customers throughout the stages of the supply chain.

The optimization of the MSCN design problem requires the preliminary definition of one or more performance metrics allowing the efficiency and effectiveness of different supply chain configurations to be quantified and compared. A stream of the literature research in this field refers to the single objective approach pursuing the minimization of the total cost of the supply chain such that the product demand from several customers is satisfied without exceeding the capacities of the network facilities, see the recent contributions of Jayaraman and Ross (2003), Yan et al. (2003), Gen and Syarif (2005), Truong and Azadivar (2005), Amiri (2006). This approach usually involves making tradeoffs among the cost components of the system that include: (i) the costs of opening and operating the facilities and (ii) the inbound and outbound transportation costs.

2.3 TAGUCHI QUALITY LOSS FUNCTION

The quality loss function developed by Genichi Taguchi. Taguchi proposed his approach of quality loss in the late 60s. He considered product specification or product species are related to product functions and market, while the product quality is related to loss and market size. Product species affect the market size by offering more feature, and satisfying more customer needs. On quality side, poor quality might damage the willingness of purchase and the reputation of company, which results in a reduction in market size. So, he used the traditional concept of quality as the conformance to specifications, and proposed the loss to be beyond the internal cost as the loss from external, such as loss from customer's inconvenience and dissatisfaction, warranty loss and so on (Taguchi 2005).

In traditional system, if a product measurement falls within a specification limit, the product is accepted: otherwise, product is rejected. The loss function is a means to quantify, on a monetary scale, the loss incurred when a product or its production process deviates from the customer-desired value in terms of one or more key characteristics. This loss includes long-term losses related to poor reliability and the cost of warrantee, excess inventory, customer dissatisfaction, and eventually, loss of market share. Even though researchers attempt to construct many types of quality loss functions, there is a general consensus that the Taguchi loss function may be a better approximation for the measurement of customer dissatisfaction with product quality. Generally, three types of loss functions are used to calculate the Taguchi loss: the first type is a two-sided loss function, where a nominal value is the target and deviation from either side of the target is allowed as long as it remains within the specification limits. Any deviation from the target value will result in a loss and zero loss occurs only when the characteristic measurement is equal to the target value. The second and third types of loss functions are one-sided functions where deviations from the target are allowed only in one direction. These loss functions are referred to as "larger is better" and "smaller is better". In the smaller the better model, the zero point is the assumed best target value. The larger the better case assumes some larger value as the target. The loss functions are provided below, where L_{SB}, L_{BB}, and L_{NB} are the losses incurred in the smaller the better case, bigger the better case and nominal the best case, respectively; k is a loss coefficient; m is a customer target value and the random value y represents the quality characteristic measurement.

Smaller the Better: $L_{SB} = k(y)^2$ (2.1)

Bigger the Better: $L_{BB} = k\ (1/y^2)$ (2.2)

Nominal the Best: $L_{NB} = k\ (y-m)^2$ (2.3)

The mathematical details of the loss function can be found in Cho and Leonard (1997), Phillips and Cho (1998; 2000), Kim and Cho (2000), and Teeravaraprug and Cho (2002). Taguchi loss function is applied in limited manner by researchers in the traditional vendor selection problem (Liao and Kao 2010; Ordoobadi 2009; Magdalena, 2012) and it is used to determine the suitable green vendor for the outsourcing activity by measuring the loss due to the performance level of vendor for predefined selection factors.

2.4 MULTI-OBJECTIVE OPTIMIZATION MODELS

As a final part of SCND design problems, an introduction in multi-objective problems is presented. Since the focus on this research is multi-objective optimization, these problems are described here briefly. MOO can be both deterministic and stochastic. They should be considered as a different type of more complex and complete SCND problems. Multi-objective problems, as already mentioned before, are trying to model and solve SCND problems with more than one objective. These objectives can be both minimizing or maximizing ones, and usually conflicting between each other. A good problem of this type is defined by more than one objective function, set of constrains, and a technique for obtaining solutions. Since there is an existing trade-off between the objectives of the problem, there should not be one unique solution, but on the contrary a set of non-dominated solutions. This set of solutions means that none of the objectives can be improved, without compromising another one. Usually a trade-off between the objectives is present. Therefore, there must be higher decision-making criteria that will choose a preferred solution from the set. There are different types of solving techniques that are able to provide a good set of the so-called Pareto optimal solutions, which are based on some traditional approaches, or on evolutionary algorithms. The problems that can be addressed here, are existing in a significant number, such as minimizing different types of cost, transportation distances, time for traveling, lost sales, investments, or maximizing profit, customer satisfaction, quality assurance. However, current trends are moving towards sustainability not just from a financial point of view, but also from an environmental and social one. Sustainable Supply Chain Network Design recognizes that the long term competitive advantage should be achieved through the alignment of economic, social and environmental goals.

Supply chain optimization has got more attention from many researchers in recent years. In the supply chain literature, there are two general types of optimization problems: local optimization and global optimization. In total supply chain, there is collaboration among members to maximize the chain's profit (Gheidar-Kheljani et al. 2009). Each member attempts to reach the local optimum solutions for its problems, but it does not guarantee the global optimum strategies for the whole supply chain.

Multi-Objective Optimization (MOO) involves simultaneous optimization of problems with at least two objective functions which are conflicting in nature. In MOO, there is no single optimum solution, but a number of them exist that are optimal called Pareto fronts. There are several approaches of finding the Pareto fronts in MOO models. One can aggregate all the objectives into a single objective by scalarization using the weighted sum, distance functions, goal programming and ε−constraint (Konak et al. 2006, Coello 1999). In contrast, evolutionary algorithms are able to generate and maintain multiple solutions in one simulation. Commonly used evolutionary algorithms include genetic algorithms which have been shown to give a good approximation of the Pareto fronts (Deb et al., 2002, Konak et al., 2006).

The supply chain network design problem has been optimized as single objective problem (Amiri 2006, Arntzen et al., 1995). However, the results obtained from such models were not a true wholesomely optimal and to an extent misleading since an optimal solution for a specific scenario is not optimal to the other entities' objectives.

2.5 GENETIC ALGORITHM AND PARETO GENETIC ALGORITHM

2.5.1 Genetic Algorithm

Genetic Algorithms (GAs) are a class of algorithm, which are powerful optimization tools that imitate the natural process of evolution and Darwin's principal of "Survival of the Fittest". Genetic algorithms are optimized in a similar manner, by simulating the Darwinian evolutionary process and naturally occurring genetic operators on chromosomes. Genetic algorithms are used to solve extremely complex search and optimization problems which prove difficult using analytical or simple enumeration methods. GAs do not carry out examinations sequentially but search in parallel mode using a multi individual population, where each individual is being examined at the same time. GAs provide an efficient and robust method of obtaining global optimization in difficult problems. GAs do not require derivative information found in analytical optimization. A GA works well with numerically generated data, experimental data, or analytical functions, and has the ability to jump out of local minimum, i.e. has the ability to find the global optimum.

Genetic algorithms were developed by J. Holland in the 1970s to understand the adaptive processes of natural systems. Then, in the 1980s, he applied genetic algorithm (GA) for optimizing and machine learning problems. GA belongs to a very popular class of evolutionary algorithms that use crossover and mutation operators and a selection procedure to generate new population. In the new population, strong species have greater chance to pass their genes to future generations via reproduction. In recent years, there has been a growing interest in using genetic algorithms to solve many single and multi-objective problems that are mostly NP-hard and combinatorial.

The SCN design problems have been solved by means of evolutionary algorithms. The first attempt to go beyond the traditional mathematical modeling and to exploit the efficiency of the genetic algorithms (GA) to solve the SCN design problems modeled as the balanced allocation of customers to multiple and incapacitated distribution has been proposed by Zhou et al. (2002), Chan et al. (2004) optimized the demand allocation for a two-stage, single product, and partially capacitated, collaborative SCN design problem.

Yeh (2006) presented a memetic algorithm for the minimization the total transportation cost of a given distribution network design problem, involving the choice of facilities (plants and distribution centers) to be opened and the distribution network design to be configured. The proposed algorithm was developed by combining the following methods: the genetic algorithm (GA), the multi-greedy heuristic method, the linear programming technique and three local search methods. Gen et al. (2008) propose a GA based approach to cope with network multiple objective problems.

During the last decade, there has been a growing interest using genetic algorithms (GA) to solve a variety of single and multi-objective problems in production and operations management that are combinatorial and NP hard (Gen and Cheng 2000, Dimopoulos and Zalzala 2000, Aytug et al. 2003).

2.5.2 Pareto Genetic Algorithm

In reality, with the majority of the problems we encounter in real life are multi-objective optimization problems. These objective functions are mutually conflicted. Improving one of the objectives will compromise other objective functions. Namely, multiple performance measures for multi-objective optimization problems are non-commensurable, and we can find a set of solutions that is called Pareto optimal set. The individuals in the solution set are called non-dominated or non-inferior (Zitzler 2004). Fujita et al. (1998) and Zitzler (1999) viewed the GA as a study approach to solve multi-objective optimization problems because it features randomization and large-scale parallel search. Recently, some scholars have introduced Pareto's concept into the GA to improve the solving capability. Traditional GA can only obtain an optimal solution, but PaGA can obtain the Pareto optimal set by adding some mechanisms based on traditional GA. Besides, the Pareto optimal solutions set can also be provided for decision-makers' reference and evaluation. Zhu et al. (2000) indicated that PaGA is a method derived from the improvement of an ordinary GA. During the improvement, three calculation processes, round, niche, and Pareto optimal solutions set filter are added to the ordinary GA, and a case application is also introduced. The conclusion illustrates PaGA can search across the Pareto optimal solution set, and with the participation of decision-makers, the optimum solution can be identified. Fei et al. (2006) argued that all we need to do is to attach Pareto's concept to an ordinary GA and add population sorting, niche technique, and several elitists preserved to obtain PaGA, which is ideal for solving multi-objective optimization problems. Fei et al. (2006) applied it to the lever of the tower hoist hanger for solving a dual-objective optimization problem – identifying the optimum lever's mass and the angle of the hanger – to obtain more options for designers. For other PaGA applications, please refer to relevant research of Poulos et al. (2001), Hosung et al. (2005), Zhou and Xiang (2006), and Chakraborti et al. (2006).

2.6 MULTI-ECHELON MULTI-OBJECTIVE DECISION MAKING APPROACH

A supply chain (SC) is a set of facilities, suppliers, customers, products and methods of controlling inventory, purchasing and distribution (Sabri and Beamon 2000). It is aimed at providing customers with products they want in a timely way and efficiently and as profitable as possible. The main objective is to enhance the operational efficiency, profitability and competitive position of a firm and its supplier chain partners (Min and Zhou 2002). Each independent entity of supply chain has inherent objective function to maximize in business transactions for profit maximization (Pinto, 2004). In a supply chain network (SCN) managers need to make strategic decisions that are viable for the business. The decisions range from what product to produce and their design, how much, when and from where to buy a product, how much, where and when to produce a product, etc. These strategic decisions are made with uncertainty about product demands, costs, prices, lead times and quality. Besides, the environment in which these decisions are made is competitive.

Christopher (2004) defined supply chain management as the management of upstream and downstream relationships with suppliers and customers to deliver superior customer value at less cost to the supply chain as a whole. Harrison and Hoek (2005) described the supply chain management as a plan and control all of the processes that link partners in a supply chain together in order to meet end-customers' requirements. Nowadays, supply chain management (SCM) which covers production planning for entire supply chain from the raw material supplier to the end customer has become the foundation for operations management. Since SCM has become the core of the enterprise management in the 21st century, there is a high interest to exploit the full potential of SCM in enhancing organizational competitiveness.

The supply chain (SC) has been viewed as a network of facilities that performs the procurement of raw material, transformation of raw material to intermediate and end products, and distribution of finished products to retailers or directly to customers. These facilities, which usually belong to different companies, consist of production plants, distribution centers, and end-product stockpiles. They are integrated in such a way that a change in any one of them affects the performance of others. Substantial work has been done in the field of optimal SC control. Various SC strategies and different aspects of SC management have been investigated in the literature.

However, most of the developed models study only isolated parts of the SC. Structuring a global supply chain is a complex decision-making process. The complexity arises from the need to integrate several decisions each of which with a relevant contribution to the performance of the whole system. In such problems, the typical input includes a set of markets, a set of products to be manufactured and/or distributed, demand forecasts for the different markets and some information about future conditions (e.g. production and transportation costs). Making use of the above information, companies must decide where facilities (e.g. plants, distribution centers) should be set operating, how to allocate procurement/production activities to the different facilities, and how to plan the transportation of products through the supply chain network in order to satisfy customer demands. Often, the objective considered is the minimization of the costs for building and operating the network.

The coordination and alignment of decisions among entities are more important when a supply chain is being designed and structured. Supplier selection, locating DCs, manufacturing planning, and allocation problems are the primary decisions that must be made in the beginning of a supply chain design process.

In the real world, the major problem is the existence of multiple conflicting objectives to find a set of Pareto optimal solutions. In the Pareto optimal set, an objective cannot be improved without deteriorating other objectives. There are various methods to find Pareto optimal solutions (Ho 2010). Weighting method is one of the most popular methods used to reach the Pareto optimal set in problems with different weighted objectives.

Salhi and Rand (1989) recognized the error introduced into location problems by ignoring the interdependence between routing and location decisions. Some papers focused on the relationships between facilities and transportation costs, stressing that location of distribution facilities and routing of vehicles from facilities are interdependent decisions.

Network optimization is a major research area in logistics. Goetschalckx et al. (2002) review production-distribution networks, emphasizing global supply chains. Geoffrion and Powers (1995) study strategic distribution system design. Both papers indicate decision variables used in various models: The number, locations, and types of facilities; the quantities of intermediate and finished products manufactured, stored, and shipped. Researchers have addressed in detail various subsets of these issues. Review of facility location is given by Owen et al. (1998). Nozick and Turnquist (1998) include inventory costs in a fixed-charge facility location model, while Hindi and Basta (1994) treat two-stage multi-commodity distribution. Bhatnagar et al. (1993) combine the production plans of several firms, while Dogan and Goetschalckx (1999) examine a network with multiple production stages, integrating suppliers to end customers with seasonal demand.

Sadrzadeh (2012) presented a genetic algorithm-based meta-heuristic to solve the facility layout problem (FLP) in a manufacturing system, where the material flow pattern of the multi-line layout was considered with the multi-products. The matrix encoding technique had been used for the chromosomes under the objective of minimizing the total material handling cost. The proposed algorithm produced a table with the descending order of the data corresponding to the input values of the flow and cost data. The generated table was used to create a schematic representation of the facilities, which in turn was utilized to heuristically generate the initial population of the chromosomes and to handle the heuristic crossover and mutation operators.

The multi-objective unequal area facility layout problem of Ripon et al. (2011) was again solved by Ripon et al. (2013) to obtain pareto optimality by using another evolutionary approach. This research employed the variable neighborhood search (VNS) with an adaptive scheme that presents the final layouts as a set of Pareto optimal solutions.

Aiello et al. (2012) considered four objectives in developing their facility layout design. They took into account material handling costs, aspect ratio, closeness and distance requests among the departments. This paper proposed a new multi objective genetic algorithm (MOGA) for solving unequal area facility layout problems. The genetic algorithm suggested is based upon the slicing structure where the relative locations of the facilities on the floor were represented by a location matrix encoded in two chromosomes.

Historically, researchers have focused relatively early on the design of production/distribution systems (Canel et al. 2001). Typically, discrete facility location models were proposed which possibly included some additional features but that still had a limited scope and were not able to deal with many realistic supply chain requirements. However, in the last decade, much research has been done to progressively develop more comprehensive (but tractable) models that can better capture the essence of many supply chain network design (SCND) problems and become a useful tool in the decision-making process. It also becomes clear that many aspects of practical relevance in supply chain management (SCM) are still far from being fully integrated in the models existing in the literature (Melo et al. 2006, Melo et al. 2009). Ferrio and Wassick (2008) proposed a multi-product chemical supply network include production sites, an arbitrary number of DCs, and customer zones (CZs). This problem was formulated as a MILP model for redesigning and optimizing of network. Their model was analyzed by GAMS/CPLEX mathematical programming solver. Liu and Papageorgiou (2013) proposed a multi-objective production-distribution and capacity planning model by considering costs, response, and service level in a universal SC. Their model is solved by means of the ε-constraints and Lexicographic mini-max methods.

According to Ho et al. (2010) there is only one paper that used GA in the supplier selection problem. Ding et al. (2005) proposed an optimization methodology based on GA for supplier selection problem. Different possible configurations of the selected suppliers are provided in the presented procedure. Each possible configuration was then evaluated with respect to the key performance indicators. Taherkhani and Seifbarghy (2012) proposed a multi-echelon supply chain to minimize the total cost of supply chain consists of purchasing, assembling, and transportation costs between levels. Then, the model is solved by a SA based heuristic. Kadadevaramath et al. (2012) presented the modeling and optimization of a three echelon SCN using the PSO/intelligence algorithms.

The optimization of a supply chain is related to selecting the optimum resource options in order to satisfy the objective function/functions (Roy 1971). The single objective (Jayaraman and Pirkul 2001) based supply chain models are mostly aimed at finding the minimum total cost (Amiri 2006). However, the modeling of a supply chain requires more than single-objective such as lead-time minimization, inventory level minimization, service level maximization, environmental impact maximization and so on (Voudouris 1996). Sometimes these objectives may cause conflicts such as increasing the service level usually causes a growth in costs. Therefore, the aim must be to find trade-off solutions to satisfy the conflicting objectives. In multi-objective optimization problems, there is no single optimum solution, but there is a solution set which creates Pareto optimal solutions. Pareto optimal solutions are a set of trade-offs between different objectives and, are non-dominated solutions (Deb 2001); i.e. there is no other solution which would improve an objective without causing a worsening in at least one of the other objectives.

Most literature models only consider single criterion for the supply chain planning and optimization, such as cost (Georgiadis et al. 2011, Kopanos et al. 2012) and profit (Verderame and Floudas 2009, Amini and Li 2011). One of the earliest papers using multi-objective method for supply chain (Weber and Current 1993) proposed a multi objective approach for vendor selection, considering three objectives including the purchases cost, number of late deliveries, and rejected units. Syarif et al. (2002) designed a multi-echelon SCN in order to select of the plants and DCs to be opened and the distribution network design to satisfy the demand, which was solved using a spanning tree-based Genetic algorithm (GA).

There is no single algorithm which can find the best solution for all types of optimization problems according to the no-free lunch theorem (Wolpert and Macready 1997), in literature, several models have been proposed to solve supply chain design problems to get the Pareto optimal solutions. Most of these models are based on genetic algorithm and its enhanced versions (Altiparmak et al. 2006, Liao et al. 2011).

Multi-objective optimization was used in different aspect of supply chain. One fine example is build-to-order supply chain (BOSC) which is defined as Build to order (BTO) and sometimes referred to as make to order or made to order (MTO), is a production approach where products are not built until a confirmed order for products is received. BTO is the oldest style of order fulfillment and is the most appropriate approach used for highly customized or low volume products. It is a key operation model for providing services/products at present. Study at Che and Chiang (2010) focuses on performing the supply chain planning for the BOSC network. The planning is designed to integrate supplier selection, product assembly, as well as the logistic distribution system of the supply chain in order to meet market demands. With multiple suppliers and multiple customer needs, the assembly model can be divided into several sub-assembly steps by applicable sequence. Considering three evaluation criteria, namely costs, delivery time, and quality, a multi-objective optimization mathematical model was established for the BOSC planning in this study.

Several studies have been undertaken where the SCN problems have been optimized as Multi Objective Optimization (Lee et al. 2002) problems and the results are more than encouraging. Amodeo et al. optimized a supply chain as a multi-objective optimization using genetic algorithms and simulation model (Amodeo 2007). Objectives considered were, minimizing inventory cost and maximizing service level. It was concluded that this approach obtained inventory policies better than the ones used in practice then with reduced costs and improved service level. Another multi-objective model was proposed to assigned suppliers to warehouses and warehouses to customers and used multi-objective optimization modeling framework for minimizing system-wide costs and maximizing customer satisfaction (Erol and Ferrell 2004). Chen and Lee (2004) developed a multi-product, multi-stage, and multi-period scheduling model for a multistage SCN with uncertain demands and prices. They simultaneously optimized the following objectives: fair profit distribution among all participants, safe inventory levels, maximum customer services, and robustness of decision to uncertain demands.

Altiparmak et al. (2006) proposed a model with three objective functions to minimize total cost, to maximize total customer demand satisfied, and to minimize the unused capacity of distribution centers. Here, transportation time was handled as a constraint that determined a set of feasible distribution centers able to deliver the product to the customer before a due date. They proposed a new procedure based on genetic algorithm to find the set of Pareto optimal solutions for a multi-objective supply chain network design problem. They used simple crossover and mutation operators in which both of them are segment based and each segment of offspring is randomly selected with an equal chance among the corresponding segments of parents.

There has been a growing interest of using evolutionary algorithms to solve multi-objective optimization problems recently (Deb 2001, Pinto 2004, Farahani and Elahipanah 2008). Different models have been developed with different objective functions where evolutionary algorithms have been used to find Pareto fronts. Sabri and Beamon (2000) developed an integrated multi-objective supply chain model for strategic and operational under certainties of products-delivery and demands. Similarly, Melachrinoudis et al. (2005) worked on a bi-objective optimization with cost minimization and service level maximization as objectives. Farahani and Elahipanah (2008) set up a bi-objective model for the distribution network of a supply chain which produces one product in a three-echelon supply chain design. Zeng (1998) emphasized the importance of the lead time cost tradeoff, associated to the transportation modes available between pairs of nodes in the network. A mixed integer programming model was proposed to design the supply chain optimizing both objectives. In the model proposed by Graves and Willems (2005) cost and time were combined in the objective function. The supply chain was configured selecting alternatives at each stage of the production and distribution network. A dynamic programming algorithm was used to solve this problem. In recent years multi-objective problems in supply chain design have been treated with more emphasis taking advantage of increased computational resources and new methods. Chan et al. (2006) presented a multi-objective model that optimized a combined objective function with weights. Some of the criteria included cost and time functions, and one of the components of time was transportation time.

The review by Farahani et al. (2010) about multi-criteria models for facility location problems describes some works where metrics of cost and service level are considered. The metaheuristic methods mentioned include multi objective versions of Scatter Search, Tabu Search, Simulated Annealing, Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO).

However, some other metaheuristics that were created for multi-objective applications were also mentioned, like Simple Evolutionary Algorithm for Multi-Objective Optimization (SEAMO), Strength Pareto Evolutionary Algorithm version 2 (SPEA2), Pareto Envelop based Selection Algorithm (PESA), Non-dominated Sorting Genetic Algorithm II (NSGA-II), Vector Evaluated Genetic Algorithm (VEGA), and the Multi-Objective Genetic Algorithm (MOGA).

It is interesting to note the review by Griffis et al. (2012) where they presented the use of metaheuristics in logistics and supply chain management from year 1991 to 2012. Near 15% of the applications were in the area of supply chain design. They highlight the use of Simulated Annealing and Tabu Search among local search metaheuristics, with minor attention in the literature to greedy randomized adaptive search procedure (GRASP), variable neighborhood search (VNS) and others. In terms of population search techniques, the most popular have been Genetic Algorithms and Ant Colony Optimization, with fewer mentions for Scatter Search, Particle Swarm Optimization, and others. However, in this review it is evident the few applications of multi objective metaheuristics, especially for supply chain design problems.

In the literature, several models have been proposed to solve supply chain design problems to get the Pareto optimal solutions. Most of these models are based on genetic algorithms and the fuzzy logic approach. Work has been done on the facility location problem of a four echelons supply chain (suppliers, plants, DCs and customers) (Altiparmak et al. 2006). The objectives of this work are to minimize the total cost, maximize customer services and the capacity utilization balance for DCs using a genetic algorithm-based approach.

Yeh and Chuang (2011) developed an optimum mathematical planning model for green partner selection, which involves four objectives including time, cost, product quality, and green appraisal score; this model employed two types of genetic algorithms to solve multi-objectives and then to find the set of Pareto optimal solutions. The crossover and mutation operators are segment based and the selection mechanism is based on the number of parents and offspring in the current generation. In this study, the weighted sum approach that can generate a greater number of solutions has been proposed.

While the most successful companies are aiming for total customer satisfaction, it is important to quantify service quality of suppliers (Li 2003, Liao and Kao 2010). Taguchi loss function (quality loss function) is a method of measuring loss as a result of the product or service not meeting the required standard (Taguchi et al. 1989). The purpose of calculating loss is to quantitatively evaluate the quality loss caused by the variation (Magdalena 2012). Taguchi loss function is applied to various sector problems such as health care applications (Taner and Antony 2000, 2006); real estate industry (Festervand et al. 2001) and manufacturing sector (Hui and Leung 1994). Taguchi loss function can be effectively applied in many areas due to its distinctive flexibility. However, various applications of Taguchi loss function were hardly exposed in literature (Ordoobadi 2009). The concept behind Taguchi's Quadratic Loss Function (QLF) is to calculate the amount of loss for the company.

Liao (2010) proposed an integrated method to solve the supplier selection problem. The author attributed the method in three phases. Delphi technique was used to identify the criteria such as quality, price, on time delivery, and customer service. Then using Taguchi loss function and AHP, the author prioritized the best supplier and selected the best performing supplier. Use of mixed integer programming has also been made to address the supplier selection problem. Taguchi loss function was used to determine the potential losses and measure the service performances (Magdalena 2012, Sharma and Balan 2013). According to Magdalena (2012), loss in question is the cost of maintenance, the cost of failure, adverse effects to the environment such as pollution or excessive production cost. Integration of the Taguchi loss function and TOPSIS method to select an optimal supplier in supply chain management was proposed by Lavanpriya et al. (2013).

In this study, we proposed a new mPaGA approach based on PaGA for multi-objective optimization of SCNs which is one of the NP hard problems. Three objectives were considered: (1) minimization of total cost comprised of fixed costs of plants and distribution centers (DCs), inbound and outbound distribution costs, (2) minimization of transportation / delivery time, and (3) maximization of service performance of suppliers. The proposed mPaGA was designed to generate Pareto-optimal solutions and to investigate the effectiveness of the proposed mPaGA, an experimental study using actual data from a company, was carried out.

2.7 IDENTIFICATION OF KEY FINDINGS AND RESEARCH GAPS

By performing a critical analysis on the structured literature, it was able to identify the key findings in the current research field. Furthermore, comparing the researching streams between authors led towards identification of the research gaps and definition of the thesis' research direction.

The most recent comprehensive review for facility location and supply chain management demonstrated that most of the literature deals with deterministic models when compared with stochastic ones (approximately 82% against 18%) (Melo et al. 2009). So, majority of papers in the literature on integrated SCND problems are for the deterministic environment, although recently some research in SCND problems under uncertainty is increasing significantly.

Most of the proposed models were developed over a midterm planning horizon while some researchers considered a multi-level planning approach. Minimization of total cost and, to a lesser extent, maximization of the entire chain profit, as a single objective, was often the objective function of most of the proposed models. In case of multi-objective optimization which is very few in number compared to the single objective optimization models, most of them considered service performance level which is often expressed as cost or quality. Few researchers used transportation time, robustness, quality level, lead time, capacity utilization rate, risk etc. as second objective along with cost or profit. Most of these multi-objective models deals with two objectives.

There are very few models which incorporated more than two objectives. Moreover, qualitative performance measures in addressing multi-objective SC problems were greatly ignored.

However, the most important fact is, though the aim of achieving minimum supply chain cost throughout the supply chain network, and ensuring maximum service performance level within minimum delivery time are highly correlated to improve supply chain performance, very few works represent the joint consideration of these three.

And the most widely used modelling approach was genetic algorithm based model. For solution procedure, researchers used several approaches among them particle swarm optimization, different versions of genetic algorithms and biomimetic algorithms was used mostly.

So, a modified pareto genetic algorithm (mPaGA) based on pareto genetic algorithm (PaGA) is applied to deal with multi-objective optimization problem.

Finally, it can be concluded that, a number of studies and researches are conducted on location selection and supplier selection separately, but none of the above-mentioned papers considered the distribution center allocation with supplier selection simultaneously.

This gap of the existing literature inspires the author of this work to come up with a multi-objective, multi-echelon SCND model under deterministic conditions with the consideration of minimizing cost (fixed cost, variable cost and transportation cost), transportation time, maximization of service performance level using Taguchi quality loss function along with the decisions regarding supplier selection and location of distribution centers.

Chapter-3

THEORETICAL BACKGROUND

3.1 INTRODUCTION

Supply chain management has emerged as one of the major arena for companies in recent years. Decisions regarding SCN optimization and design are highly concerned with every single part of a chain such as production facilities, distributer centers, suppliers and customers and every type of flow and connection between these nodes of the network. In this chapter, the details about supply chain, distribution center allocation and supply chain network design, Taguchi quality loss function has been discussed. Multi-objective approach is used to sum up the searching criterions or objectives for cost-time-quality optimization. A heuristic approach, Genetic algorithm is used to solve the problem. So, Multi-objective approach and Genetic algorithm in this research work are briefly discussed step by step to get clear background.

3.2 SUPPLY CHAIN MANAGEMENT

Supply chain management is a cross-functional approach that includes managing the movement of raw materials into an organization, certain aspects of the internal processing of materials into finished goods, and the movement of finished goods out of the organization and toward the end consumer. Supply chain management encompasses different types of decisions, starting from facility locations and capacity planning, but also going downwards hierarchically, such as procurement, production, or distribution decisions.

Supply Chain Management is a relatively new research stream in the past 20 years, and it is an ever-growing research topic. Since the globalization of the companies and the supply chains contributed for more complex operations and network structures, the need of SCM has emerged. In order to introduce the concept of supply chain management, first, what supply chain is, has to be understood. According to Mentzer, supply chain is "a set of three or more entities (organizations or individuals) directly involved in the upstream and downstream flows of products, services, finances, and/or information from a source to a customer". It can be constituted of production and distribution facilities of a focal company, suppliers' facilities and customers. Supply chain is focused on the focal organization activities, without any integrations inside the supply chain. An integration is a step further that organizes the materials, production, and physical distribution activities.

In today's competitive world, the success of an industry is contingent upon the management of its supply chain. Supply chain management is a relatively new term. It crystallizes concepts about integrated business planning that have been espoused by logistics experts, strategists, and operations research practitioners. Today, integrated planning is finally possible due to advances in information technology, but most companies still have much to learn about implementing new analytical tool needed to achieve it. In order to stay competitive and continue to survive they need to be their competitors. World class organizations now realize that non-integrated manufacturing process, nonintegrated distribution process and poor relationship with suppliers and customers are inadequate for their success. Supply chains encompass a series of steps that add value through time, place, and material transformation. A typical supply chain network is given in Figure 3.1.

All organizations have or can purchase the components to build a supply chain network, it is the collection of physical locations, transportation vehicles and supporting systems through which the products and services your firm markets are managed and ultimately delivered. Physical locations included in a supply chain network can be manufacturing plants, storage warehouses, and carrier cross docks, major distribution centers, ports, and intermodal terminals whether owned by the company, the suppliers, the transport carrier, a third-party logistics provider, a retail store or the end customer. Transportation modes that operate within a supply chain network can include the many different types of trucks, trains for boxcar or intermodal unit movement, container ships or cargo planes. The systems which can be utilized to manage and improve a supply chain network include order management systems, warehouse management system, transportation management systems, strategic logistics modeling, inventory management systems, replenishment systems, supply chain visibility, optimization tools and more.

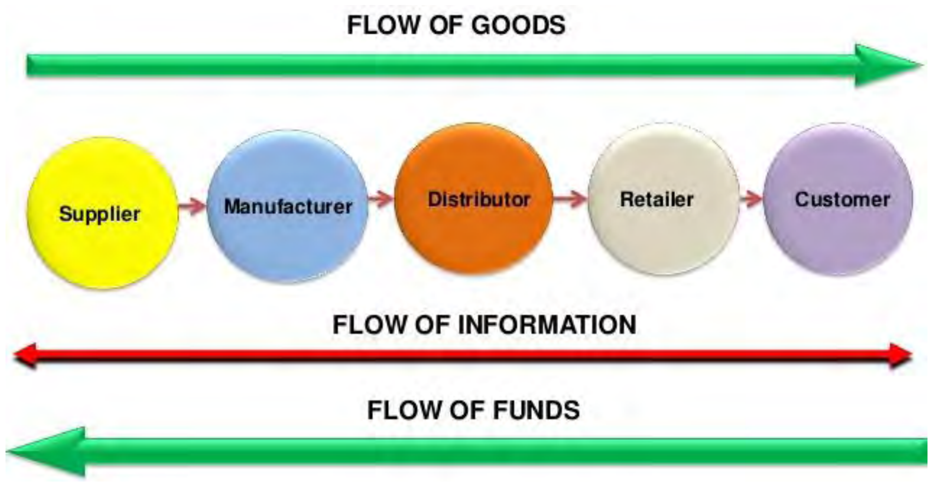


Figure 3.1: Supply Chain Network

SCM goes beyond the logistics processes and the internal integration of a supply chain of a single supplier, organization and customer. It can be defined as “the task of integrating organizational units along a supply chain and coordinating materials, information and financial flows in order to fulfil customer demands with the aim of improving competitiveness of the supply chain as a whole. From the definition, it can be seen that some aspects as coordination, information and financial flows, as well as competitiveness, are complementary with the definition of supply chain (Figure 3.2).

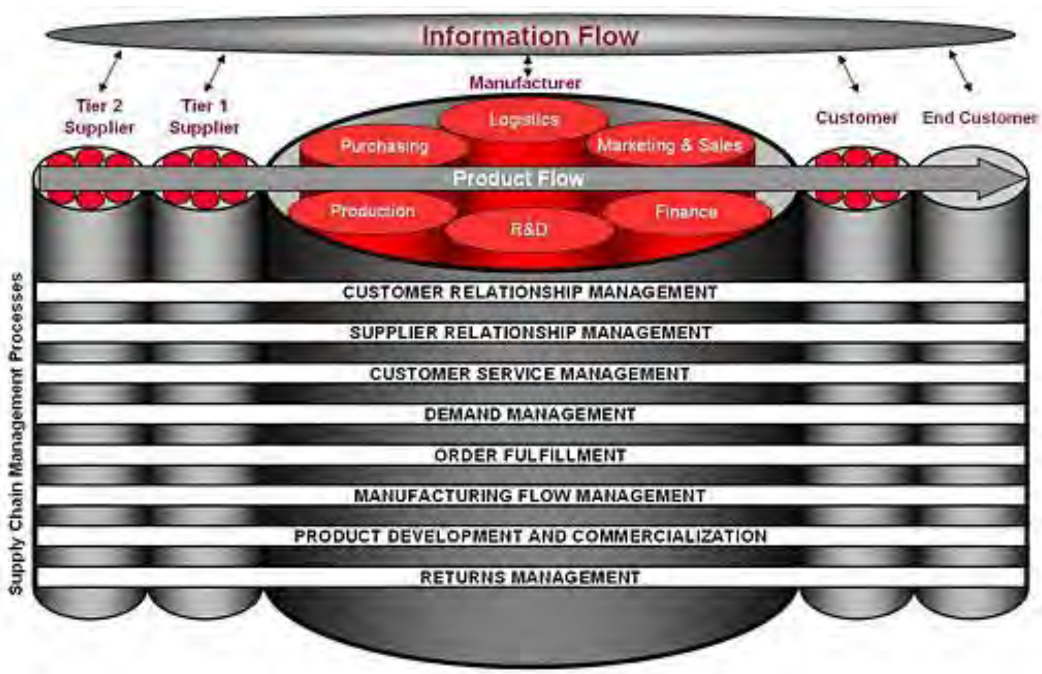


Figure 3.2: Supply Chain Management

The goal of SCM is to integrate and coordinate all the components of the supply chain in an efficient and effective way in order to make the supply chain competitive, with a final goal of being focused on the customer requirements. There are several aspects of the SCM that are critical, which can be presented by the house of SCM (Figure 3.3).

Figure 3.3: House of Supply Chain Management

3.3 SUPPLY CHAIN NETWORK DESIGN

Designing a SCN involves creating a network that incorporates all the facilities, means of production, products, and transportation assets owned by the organization or those not owned by the organization but which immediately support the supply chain operations and product flow. The design should also include details of the number and location of facilities: plants, warehouses, and supplier base. Therefore, it can be said that a SCN design is the combination of nodes with capability and capacity, connected by lanes to help products move between facilities.

An important component in SCN design and analysis is the establishment of appropriate performance measures. A performance measure, or a set of performance measures, is used to determine efficiency and/or effectiveness of an existing system, to compare alternative systems, and to design proposed systems. These measures are categorized as qualitative and quantitative.

Customer satisfaction, flexibility, and effective risk management belong to qualitative performance measures.

Logistics network design is one of the most important strategic decisions in supply chain management. Decisions on the number of facilities, their locations and capacities and the quantity of flow between them affect both costs and customer service levels. As such, effective and efficient supply chain design can constitute a sustainable competitive advantage for firms. Since opening and closing a facility is both an expensive and time-consuming process, changing network design is impossible in the short run. Because tactical and operational decisions are conditional upon strategic decisions, the configuration of logistics networks will become a constraint for tactical and operational level decisions. Over the last two decades, many companies such as Kodak and Xerox have focused on remanufacturing and recovery activities and have achieved significant successes in this area. Logistics network design is based on three objectives – cost minimization, time minimization, and logistics customer service maximization. Not all of these objectives can be achieved simultaneously since they may be in conflict. For example, minimizing costs and simultaneously maximizing service are incompatible.

Logistics is important because it creates value - value for customers and suppliers of the firm, and value for the firm's stakeholders. Value in logistics is expressed in terms of time and place. Products and services have little or no value unless they are in the possession of customers when (time) and where (place) they wish to consume them. To many firms throughout the world, logistics has become an increasingly important value-adding process for a number of reasons.

Thus, the main objective of the supply chain is to maximize the profitability of not just a single entity but rather all the entities taking part in the supply chain.

3.3.1 Demand

Demand planning has become one of the hot buttons of supply chain management. Demand Planning & Forecasting is both an art and a science. It requires informed judgment, business expertise, and technical skills. Done well, it can provide a true competitive edge and increased sales, while managing inventory and maintaining best-in-class customer service. Forecasting depends on both a structured process and modeling. Both are equally important.

Both demand management and demand planning call for managing the demand, that is, what to do if demand is greater than supply or if supply is greater than demand? It's in the best of interest of a company to balance supply with demand. This requires demand sensing (reading and interpreting the signals of the market) and demand shaping (shaping the demand to optimize sales and profit). For effective demand planning, we have to set some business rules, that is, how to deal with different issues that might arise. For example, how to allocate products among different distribution center or customers when there is a shortage. We have to prioritize the distribution centers or customers according to the demand. But we have to consider other factors also. If demand of a distribution center or customer is high, but delivery time is high, transportation cost is high also then we won’t deliver that distribution center first. Because this type of distribution center or customer won’t help the company for making profit. The distribution center or customer which demand is high as well as profit margin is also high because of low cost, delivery time is low, transportation cost is also low, we have to select that distribution center first.

3.3.2 Delivery Time Optimization

The amount of time that elapses between when a process starts and when it is completed is called delivery time. In case of supply chain network, it is actually the time between transport the product or materials from one stage to another stage of SCN. Delivery time is examined closely in manufacturing, supply chain management and project management, as companies want to reduce the amount of time it takes to deliver products to the market. In business, lead time minimization is normally preferred. The longer lead times volatile fuel prices and risks such as unavoidable delays, make estimating the cost and time associated with transportation difficult. As a result, companies incur high expedite and inventory costs. So, for optimizing the distribution network we have to minimize delivery time considering other factors such as: profit margin, demand, transportation cost etc.

Time optimization means how quickly the product reach to the customer. Optimal transportation time from supply chains is a source of competitive advantage that affects the bottom line.

Optimizing transportation time is a non-trivial problem that requires a sophisticated solution. Recently there has been a growing interest in research in supply chain network optimization problems. An important component in supply chain design is determining how an effective supply chain design is achieved, given a performance measure, or a set of performance measures.

3.3.3 Cost Optimization

Cost optimization optimizes cost and one or more responses at the same time to determine the factor settings that are both cost-effective and produce acceptable values for the responses. Often the factor settings that produce the best results are the most expensive to do. Cost optimization determines a compromise between minimizing cost and optimizing the responses.

Supply chain network design is a powerful modeling approach proven to deliver significant reduction in supply chain costs and improvements in service levels by better aligning supply chain strategies. It incorporates end-to-end supply chain cost, including purchase, production, warehousing, inventory and transportation. While this is considered a strategic supply chain planning initiative, organizations can gain competitive advantage by running supply chain network scenarios, evaluating and proactively implementing changes in response to dynamic business scenarios like new product introduction, changes in demand pattern, addition of new supply sources, changes in tax laws and so on.

Transport systems face requirements to increase their capacity and to reduce the costs of movements. All users (e.g. individuals, enterprises, institutions, governments, etc.) have to negotiate or bid for the transfer of goods, people, information and capital because supplies, distribution systems, tariffs, salaries, locations, marketing techniques as well as fuel costs are changing constantly. Transport costs are a monetary measure of what the transport provider must pay to produce transportation services. They come as fixed (infrastructure) and variable (operating) costs, depending on a variety of conditions related to geography, infrastructure, administrative barriers, energy, and on how products are carried. Three major components, related to transactions, shipments and the friction of distance, impact on transport costs. Better transportation management helps companies improve their overall supply chain efficiency.

3.4 TAGUCHI QUALITY LOSS FUNCTION

In the early 1980s, Dr Taguchi proposed the following statement relating to the quality of a product: *"Quality is the financial Loss to society after the article is shipped".*

This is one of the many concepts which were developed by Dr Taguchi. However, the above statement somehow depicts inequality, since a loss to society is not a desirable characteristic. The idea of quality is related to something which is new, beautiful and good which in the engineering sense, must have many features or functionalities. We can rearrange the above definition and still retain the basic concept of Quality to denote a positive attribute as follows:

"Quality is the avoidance of financial loss to society after the product is shipped"

The important point here is the fact that quality is related to monetary loss and not to any other factors or conditions. Even though the actual loss maybe the loss of functionality to the product, or other losses such as pollution, time, noise, etc. The overall effect is a financial loss. It can also be expanded to include the development, and manufacturing phases of a product.

A poorly designed product begins to impart losses to society from the very start of the production stage, and continues to do so, until steps are taken to improve its functionality and performance. There are two major categories of loss to society with respect to the product quality:

The first category relates to the losses incurred as a result of harmful effects to society.

The second category relates to the losses arising because of excessive variation in functional performances. The second category has a dominant impact to the design stages of the product and will be discussed here.

The conventional method of computing the cost of quality is based on the number of parts rejected and reworked. This method of quality evaluation is incapable of distinguishing between two samples, which are both within the specification limits, but with different distributions of targeted properties.

Taguchi loss function (Quality loss function) is a method of measuring loss as a result of the product not meeting the standard specifications (Taguchi, 1989). The purpose of calculating loss is to quantitatively evaluate the quality loss caused by the variation. Loss Function considers the willingness of consumers to obtain a more consistent product and the company's desire to produce products with low cost. Minimization of losses suffered by consumers is a strategy that encourages uniformity of the products and reduces costs of production and consumption. Taguchi loss is useful for the company to identify not only the rejected and reworked scrap but also the possibility of environmental pollution, the use of not long-lasting products, or other negative effects. Loss for the company is the cost due to deviation from the target value.

The concept behind the Taguchi's Quadratic Loss Function (QLF) is to calculate the amount of loss for the company. QLF is a mathematical model that links quality loss to the value of money resulting from the deviation of the quality of the specification from the desired target. Loss in question is the cost of maintenance, the cost of failure, adverse effects to the environment such as pollution or excessive production cost. Based on the loss function approach, the quality characteristics measured by Taguchi can be divided into three categories, namely:

- Nominal the best: It is a quality characteristics value which can be positive or negative. Values are measured by predetermined target value, and this target can be the center or some shift in two-sided specification limit called two-sided equal specification Taguchi loss function (Figure 3.4) and it can be formulated as

  $$L(y) = k\,(y-m)^2 \qquad (3.1)$$

  where $L(y)$ is the loss associated with a particular value of quality character y; m is the nominal value of the specification; k is the average loss coefficient, and its value is constant depending on the cost at the specification limit and the width (e.g., $m \pm \Delta$) of the specification; where Δ is the customer's tolerance.

  The closer it gets to the target value, the better the quality.

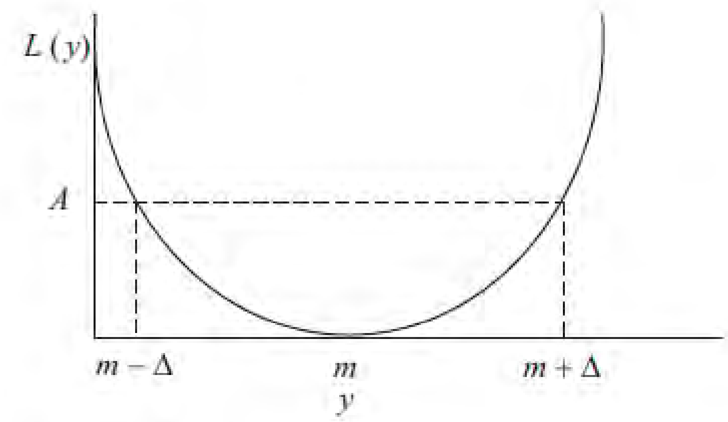


Figure 3.4: Nominal the Best Loss Function

- Lower the better: It is a non-negative measurable characteristic with respect to the ideal value of zero. The nearer it gets to zero, the better the quality and it is called one-sided minimum specification limit (Figure 3.5) and it is formulated as

  $L_{SB} = k(y)^2$, $k = (A/\Delta^2)$ (3.2)

  where A is the average quality loss.

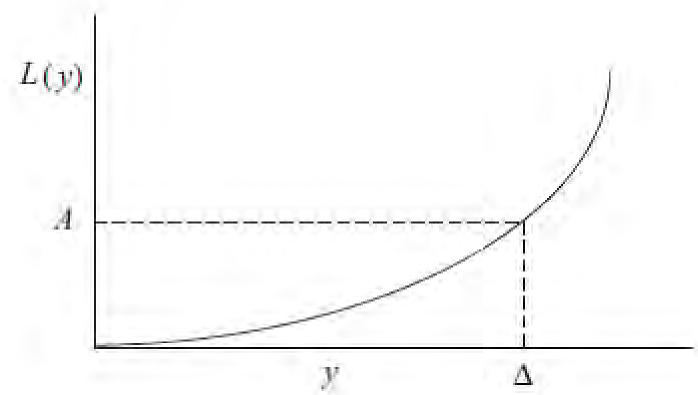


Figure 3.5: Smaller the Better Loss Function

- Larger the better: It is a non-negative measurable characteristics value with respect to the ideal value of infinity. The closer it approaches infinity, the better the quality and it is called one-sided maximum specification limit (Figure 3.6) and it is formulated as

  $L_{BB} = k\,(1/y^2), k = (A\Delta^2)$ (3.3)

  where A is the average quality loss.

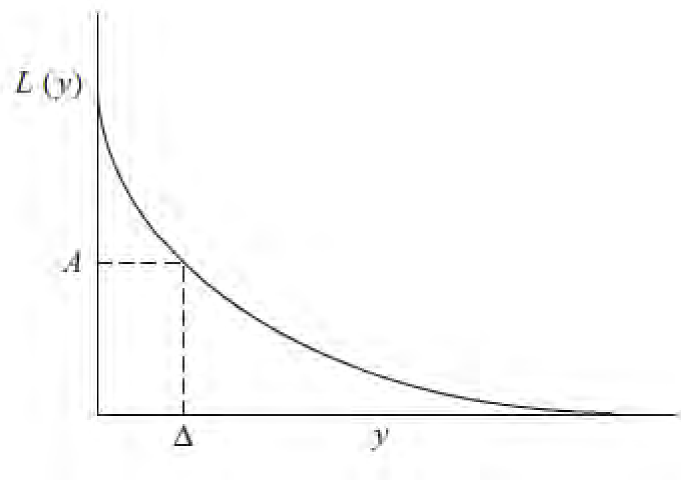


Figure 3.6: Larger the Better Loss Function

The application of Taguchi loss functions can be an excellent tool when faced with determining the utility of competing scheduling policies or practices. A critical insight of Taguchi's theory is that the total expected cost can be reduced by moving the mean closer to the target and reducing variance. Thus, even if the average prosthetic component is aligned properly, cost reductions can be obtained by reducing the variance of the alignment. Variance reduction is a central goal of quality engineering, and Taguchi's expected loss function highlights how reducing variance can reduce cost to society without changing the mean deviation from an ideal value.

3.4.1 Signal-to-Noise (S/N) Ratios

The product of ideal quality should always respond in exactly the same manner to the signals provided by the user. When you turn the key in the ignition of your car you expect that the starter motor turns and the engine starts. In the ideal-quality car, the starting process would always proceed in exactly the same manner - for example, after three turns of the starter motor the engine comes to life. If, in response to the same signal (turning the ignition key) there is random variability in this process, then you have less than ideal quality. For example, due to such uncontrollable factors as extreme cold, humidity, engine wear, etc. the engine may sometimes start only after turning over 20 times and finally not start at all. This example illustrates the key principle in measuring quality according to Taguchi: You want to minimize the variability in the product's performance in response to noise factors while maximizing the variability in response to signal factors.

Noise factors are those that are not under the control of the operator of a product. In the car example, those factors include temperature changes, different qualities of gasoline, engine wear, etc. Signal factors are those factors that are set or controlled by the operator of the product to make use of its intended functions (turning the ignition key to start the car).

Noise factors are primarily response for causing a product's performance to deviate from its target value. Hence, parameter design seeks to identify settings of the control factors which make the product insensitive to variations in the noise factors, i.e., make the product more robust, without actually eliminating the causes of variation. The preferred parameter settings are then determined through analysis of the "signal-to-noise" (S/N) ratio where factor levels that maximize the appropriate S/N ratio are optimal.

Finally, the goal of your quality improvement effort is to find the best settings of factors under your control that are involved in the production process, in order to maximize the S/N ratio; thus, the factors in the experiment represent control factors.

There are three standard types of S/N ratios depending on the desired performance response.

Nominal the best (for reducing variability around a target):

$$S/N_N = 10 * \log_{10} [(\text{Mean}^2/\text{Variance}) - (1/n)] \qquad (3.4)$$

Smaller the better (for making the system response as small as possible):

$$S/N_S = -10 * \log_{10} [(\Sigma y)^2/n] \qquad (3.5)$$

Larger the better (for making the system response as large as possible):

$$S/N_L = -10 * \log_{10} [\Sigma(1/y^2)/n] \qquad (3.6)$$

These S/N ratios are derived from the quadratic loss function and are expressed in a decibel scale. Once all of the SN ratios have been computed for each run of an experiment, Taguchi advocates a graphical approach to analyze the data. In the graphical approach, the S/N ratios and average responses are plotted for each factor against each of its levels. The graphs are then examined to "pick the winner," i.e., pick the factor level which (1) best maximize S/N and (2) bring the mean on target (or maximize or minimize the mean, as the case may be).

3.5 MULTI-OBJECTIVE OPTIMIZATION

Many engineering sectors are challenged by multi-objective optimization problems. Even if the idea behind these problems is simple and well established, the implementation of any procedure to solve them is not a trivial task. Optimizing a problem means finding a set of decision variables which satisfies constraints and optimizes simultaneously a vector function. Solving a multi-objective optimization problem is sometimes understood as approximating or computing all or a representative set of Pareto optimal solutions. When decision making is emphasized, the objective of solving a multi-objective optimization problem is referred to supporting a decision maker in finding the most preferred Pareto optimal solution according to his/her subjective preferences. The underlying assumption is that one solution to the problem must be identified to be implemented in practice. Here, a human decision maker (DM) plays an important role. The DM is expected to be an expert in the problem domain.

A multi-objective optimization problem, as its name suggests, has a number of objectives that need to be optimized. One of the striking differences between single-objective and multi-objective optimization is that in multi-objective optimization one needs to take care of two spaces.

- Decision variable (search) space
- Objective space

Decision variable space is the domain of the candidate solutions. Its axes are the attributes/parameters of the candidate solutions. Objective space is the space where candidate solutions are projected through the objective functions and its axes are the objective functions of the problem. The objective functions map the candidate solutions from the decision variable space to objective space. Position of a candidate solution in the objective space determines how fit it is in the competition with other fellow candidate solutions for selection.

3.5.1 Criteria of Multi-Objective Optimization

In contrast with single objective optimization, multi-objective problems are much more difficult and complex. Firstly, no single unique solution is the best; instead, a set of non-dominated solutions should be found in order to get a good approximation to the true Pareto front. Secondly, even if an algorithm can find solution points on the Pareto front, there is no guarantee that multiple Pareto points will distribute along the front uniformly, often they do not. Thirdly, algorithms which work well for single objective optimization usually cannot directly work for multi-objective problems, unless under special circumstances such as combining multi-objectives into a single objective using some weighted sum method. Substantial modifications are needed to make algorithms for single objective optimization work. In addition to these difficulties, a further challenge is how to generate solutions with enough diversity so that new solutions can sample the search space efficiently.

Real life problems consist of multiple objectives. In earlier attempts of finding solutions to these problems simpler single objective algorithms were used and a weighted average was considered. With this approach proper justification to all the objectives was not possible and hence the outcome in most of the cases remained far away from the actual solutions. To give importance to all the objectives of a problem, multi-objective approach is adopted. In this approach, obtaining a single perfect solution that simultaneously satisfy all the objective functions is impossible, generally a set of solutions known as Pareto-optimal solutions are obtained; and thus to have a proper trade-off among the solutions ensuring diversity became an important issue. So the basic objective is to provide proper justification to all the objectives in a real life problem and to have a good trade-off among the solutions obtained.

3.5.2 Pareto-Optimal Solutions

An Italian economist, Vilfredo Pareto in nineteenth century developed a concept named Pareto optimum, in his studies of economic efficiency and income distribution. This concept is now broadly used in game theory, production engineering and social sciences.

Pareto optimum technique compares two solutions in multi-objective optimization that has no unified criterion with respect to optima. Such solutions (normally referred to as non-dominated or

Pareto optimal solutions) do necessitate improvement in any objective function without sacrificing at least one of the other objective functions. The region defined by Pareto optimal solutions is called the Pareto front, and the objective of multi-objective optimization is to establish the entire Pareto front for the problem instead of a single best solution.

In multi-objective optimizing problems where the objective functions are conflicting in nature, each objective function may have a different individual optimal solution. So, to satisfy all objective functions, a set of optimal solutions is required instead of one optimal solution. The reason for the optimality of many solutions is that no one objective function can be considered to be better than any other. These solutions are non-dominated solutions. Let P be a set of non-dominated solutions. Then,

- Any two solutions of P must be non-dominated with respect to each other.
- Any solution not belonging to P is dominated by at least one member of P.

Actually, a solution x1 is said to dominate another solution x2 if both the following conditions are true:

- The solution x1 is not worse than x2 in all objectives, or fj(x1) is better than fj(x2) for all j ϵ {1, 2, . . . , M}; where fj is the j^{th} objective function and M is the number of objectives.
- The solution x1 is strictly better than x2 in at least one objective, or fj(x1) is better than fj(x2) for at least one j ϵ {1, 2, . . . , M}.

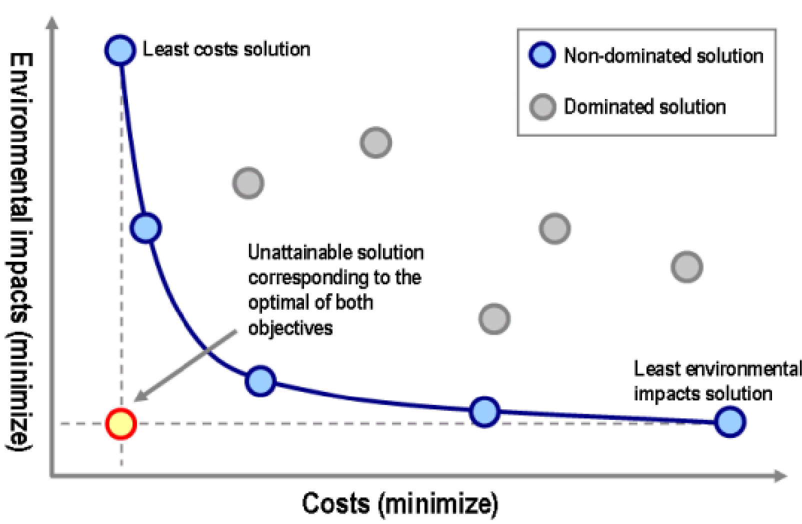


Figure 3.7: Pareto Optimal Solutions

Let us illustrate the Pareto optimality with "cost and environmental impacts" of an algorithm shown in Figure 3.7. In this problem, we have to minimize cost and environmental impacts. In multi-objective optimization, we have two goals:

- Progressing towards the Pareto-optimal front
- Maintaining a diverse set of solutions in the non-dominated front

Pareto-optimal front of a multi-objective optimization problem is the ideal solution set of that problem. In a GA, we start with a random collection of candidate solutions, most likely they are not the best possible solutions and with the help of genetic operators we progress iteratively towards the best possible solution set after a number of generations. While progressing towards the Pareto-optimal set, it is also required to maintain a diverse set of solutions to provide good trade-off alternatives to the end user.

3.6 EVOLUTIONARY ALGORITHMS

As discussed previously, the traditional methods have many drawbacks that are embedded in their solving approaches, where they fail in finding a set of optimal solutions, or they even fail at finding the global Pareto-optimal. Therefore, for optimizing multi-objective problems, new types of algorithms recently have been more and more diffused. The main advantage of these new methods is that they progress simultaneously with a set of possible solutions. By confronting the objective functions and analyzing at the same time various trade-offs, solutions of the overall set of Pareto optimal are generated every single run of the algorithm. This was not a case in the traditional methods, where every new run one solution out of the Pareto-optimal set is generated. The dependency on the weights and parameters is avoided here as well by the generation of populations and setting up operators that are improving the solutions, without the need of the previous procedures. Thus, the solutions here are not sensitive on the weights and constraints definition, as well as the iteration's settings.

Another strong reason for using Evolutionary Algorithms in MOO is that complex problems, with numerous objective functions, variables and constrains can be implemented and solved without problems. Furthermore, the dependency on the decision maker from the traditional methods is no longer existing here, in the priory stages of the optimization, since the setting and population of the model is done by the algorithm itself. Like this, a more robust approach is conducted, where the continuity of the Pareto frontier is not having the issues of the gaps and jumps between solutions, from the traditional methods.

Nevertheless, it can be concluded that searching for a global optimum solution as in the traditional methods is not efficient. The decision-making process has the need of faster ways for generating bigger sets of Pareto-optimal solutions, that will not be focused on a specific decision, but will give a comprehensive picture of the optimization of the overall problem. This has the origins in designing of new products, were the variety of solutions and the conflicting trade-offs between objectives are emphasized strongly. Thus, having a variety of close-enough (to the optimal) solutions are favored in these algorithms in front of the obtaining process for just one global optimal solution.

Evolutionary Algorithms, especially the Multi-Objective Evolutionary Algorithms (MOEA) are aiming at minimizing the distance between the solutions generated and the Pareto optimal set, while maximizing the diversity in creating unique solutions. The approach that is followed lays down in the human genetics, where the mutation of the genes, and the reproduction processes are taken as a guideline for developing these algorithms. In fact, the most representative group of EAs are the Genetic Algorithms, which are fostering the idea of using an initial population and by combination and mutation of the genes of the solutions, iteratively to arrive towards a Pareto-optimal set of solutions. Genetic algorithms, searching from a population of points, seem particularly suited to multi-objective optimization. Their ability to find global optima while being able to cope with discontinuous and noisy functions has motivated an increasing number of applications in engineering and related fields. The GAs are created by John Holland in the middle ‘70s and have a variety of implementation fields, classified mainly in engineering, industrial and scientific. Some of the areas where are mostly applied are: electrical engineering, hydraulics, aeronautics, robotics, controlling, telecommunications, design and manufacturing, management, scheduling, supply chain design, as well as sciences like physics, medicine and computer science.

3.7 GENETIC ALGORITHM

Genetic algorithms are search and optimization tools that enable the fittest candidate among strings to survive and reproduce based on random information search and exchange imitating the natural biological selection. In the middle of the twentieth century some computer scientists worked on evolutionary systems with the notion that this will yield to an optimization mechanism for an array of engineering queries. In the computer science field of artificial intelligence, a Genetic Algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. Professor John Holland in 1975 proposed an attractive class of computational models, called Genetic Algorithms (GA), that mimic the biological evolution process for solving problems in a wide domain. The mechanisms under GA have been analyzed and explained later by Goldberg, De Jong, Davis, Muehlenbein, Chakraborti, Fogel, Vose and many others. Genetic algorithms have three major applications, namely, intelligent search, optimization and machine learning. Holland's theory has been further developed and now Genetic Algorithms (GAs) stand up as a powerful tool for solving search and optimization problems. Genetic algorithms are based on the principle of genetics and evolution.

Charles Darwin stated the theory of natural evolution in the origin of species. Over several generations, biological organisms evolve based on the principle of natural selection "survival of the fittest" to reach certain remarkable tasks. The perfect shapes of the albatross wring the efficiency and the similarity between sharks and dolphins and so on, are best examples of achievement of random evolution over intelligence. Thus, it works so well in nature, as a result it should be interesting to simulate natural evolution and to develop a method, which solves concrete, and search optimization problems. In nature, an individual in population competes with each other for virtual resources like food, shelter and so on. Also in the same species, individuals compete to attract mates for reproduction. Due to this selection, poorly performing individuals have less chance to survive, and the most adapted or "fit" individuals produce a relatively large number of offspring's. It can also be noted that during reproduction, a recombination of the good characteristics of each ancestor can produce "best fit" offspring whose fitness is greater than that of a parent. After a few generations, species evolve spontaneously to become more and more adapted to their environment (Tsoukalas and Uhrig 1997).

GAs procedure begins by generating an initial collection (referred to as population) of random solution that are encoded in the form of strings called chromosomes or the genotype or the genome. The fitness of each individual chromosome is determined by evaluating its performance with respect to an objective function. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are selected (natural survival of the fittest process) and best chromosomes exchange information to produce offspring genes that are evaluated and can replace less fit member in the population. This process is continuing until the terminating condition is met. Thus, the optimum solution is found at the end of the GA process.

3.7.1 Biological Background

The science that deals with the mechanisms responsible for similarities and differences in a species is called Genetics, the science which helps to differentiate between heredity and variations. The concepts of Genetic Algorithms are directly derived from natural evolution or genetics. The main terminologies involved in the biological background of species are as follows:

3.7.1.1 The Cell

Every animal/human cell is a complex of many “small” factories that work together. The center of all this is the cell nucleus. The genetic information is contained in the cell nucleus.

3.7.1.2 Chromosomes

All the genetic information gets stored in the chromosomes. The chromosomes are divided into several parts called genes. Genes code the properties of species i.e., the characteristics of an individual. The possibilities of the genes for one property are called allele and a gene can take different alleles. For example, there is a gene for eye color, and all the different possible alleles are black, brown, blue and green (since no one has red or violet eyes). The set of all possible alleles present in a particular population forms a gene pool. This gene pool can determine all the different possible variations for the future generations. The size of the gene pool helps in determining the diversity of the individuals in the population. The set of all the genes of a specific species is called genome. Each and every gene has a unique position on the genome called locus. In fact, most living organisms store their genome on several chromosomes, but in the Genetic Algorithms (GAs), all the genes are usually stored on the same chromosomes (Goldberg, 1989). Thus, chromosomes and genomes are synonyms with one other in GAs.

3.7.1.3 Genetics

For a particular individual, the entire combination of genes is called genotype. The phenotype describes the physical aspect of decoding a genotype to produce the phenotype. One interesting point of evolution is that selection is always done on the phenotype whereas the reproduction recombines genotype. Thus, morphogenesis plays a key role between selection and reproduction.

3.7.1.4 Reproduction

Reproduction of species via genetic information is carried out by, Mitosis and Meiosis. In Mitosis, the same genetic information is copied to new offspring. There is no exchange of information. This is a normal way of growing of multi cell structures, like organs. When meiotic division takes place, genetic information is shared between the parents in order to create new offspring.

3.7.1.5 Selection

The origin of species is based on “Preservation of favorable variations and rejection of unfavorable variations”. The variation refers to the differences shown by the individual of a species and also by offspring’s of the same parents. There are more individuals born than can survive, so there is a continuous struggle for life. Individuals with an advantage have a greater chance for survive i.e., the survival of the fittest. As a result, natural selection plays a major role in this survival process.

A Genetic Algorithms operates through a simple cycle of stages:

i) Creation of a “population” of strings,
ii) Evaluation of each string,
iii) Selection of best strings and
iv) Genetic manipulation to create new population of strings.

3.7.2 Methodology

In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached. Genetic algorithms find application in bioinformatics, phylogenetics, computational science, engineering, economics, chemistry, manufacturing, mathematics, physics and other fields.

A typical genetic algorithm requires a genetic representation of the solution domain and a fitness function to evaluate the solution domain. A standard representation of the solution is as an array of bits. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case. Tree-like representations are explored in genetic programming and graph-form representations are explored in evolutionary programming; a mix of both linear chromosomes and trees is explored in gene expression programming. The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. For instance, in the knapsack problem one wants to maximize the total value of objects that can be put in a knapsack of some fixed capacity. A representation of a solution might be an array of bits, where each bit represents a different object, and the value of the bit (0 or 1) represents whether or not the object is in the knapsack. Not every such representation is valid, as the size of objects may exceed the capacity of the knapsack. The fitness of the solution is the sum of values of all objects in the knapsack if the representation is valid or 0 otherwise. In some problems, it is hard or even impossible to define the fitness expression; in these cases, interactive genetic algorithms are used.

Once the genetic representation and the fitness function are defined, a GA proceeds to initialize a population of solutions (usually randomly) and then to improve it through repetitive application of the mutation, crossover, inversion and selection operators. Each cycle in Genetic Algorithms produces a new generation of possible solutions for a given problem. In the first phase, an initial population, describing representatives of the potential solution, is created to initiate the search process. The elements of the population are encoded into bit-strings, called chromosomes. The performance of the strings, often called fitness, is then evaluated with the help of some functions, representing the constraints of the problem. Depending on the fitness of the chromosomes, they are selected for a subsequent genetic manipulation process. It should be noted that the selection process is mainly responsible for assuring survival of the best-fit individuals.

3.7.3 GA Parameters

3.7.3.1 Key Elements

The two distinct elements in the GA are individuals and populations. An individual is a single solution while the population is the set of individuals currently involved in the search process.

i. Individual: An individual is a single solution.

ii. Population: A population is a collection of individuals. A population consists of a number of individual being tested, the phenotype parameters defining the individuals.

3.7.3.2 Genes

Genes are the basic “instructions” for building a Generic Algorithms. A chromosome is a sequence of genes. Genes may describe a possible solution to a problem, without actually being the solution. A gene is a bit string of arbitrary lengths.

3.7.3.3 Fitness

The fitness of an individual in a genetic algorithm is the value of an objective function for its phenotype. For calculating fitness, the chromosome has to be first decoded and the objective function has to be evaluated. The fitness not only indicates how good the solution is, but also corresponds to how close the chromosome is to the optimal one.

3.7.3.4 Encoding

Encoding is a process of representing individual genes. The process can be performed using bits, numbers, trees, arrays, lists or any other objects. The encoding depends mainly on solving the problem. For example, one can encode directly real or integer numbers.

a) Binary Encoding

The most common way of encoding is a binary string, which would be represented as

Chromosome 1	1 1 1 1 1 0 0 1 1 0 1 0
Chromosome 2	0 1 1 1 0 0 0 0 1 1 0 0

Each chromosome encodes a binary (bit) string. Each bit in the string can represent some characteristics of the solution. Every bit string therefore is a solution but not necessarily the best solution. Binary encoding gives many possible chromosomes with a smaller number of alleles. On the other hand this encoding is not natural for many problems and sometimes corrections must be made after genetic operation is completed (Sivanandam and Deepa, 2008). Binary coded strings with 1s and 0s are mostly used. The length of the string depends on the accuracy.

b) Octal Encoding

This encoding uses string made up of octal numbers (0–7).

Chromosome 1	03467216
Chromosome 2	15723314

c) Hexadecimal Encoding

This encoding uses string made up of hexadecimal numbers (0–9, A–F).

Chromosome 1	9CE7
Chromosome 2	3DBA

d) Permutation Encoding (Real Number Coding)

Every chromosome is a string of numbers, which represents the number in sequence. In permutation encoding, every chromosome is a string of integer/real values, which represents number in a sequence. Permutation encoding is only useful for ordering problems like traveling sales man etc.

Chromosome A	1	5	3	2	6	7	9	8	3
Chromosome B	8	5	6	3	7	5	6	1	8

e) Value Encoding

Every chromosome is a string of values and the values can be anything connected to the problem. This encoding produces best results for some special problems. Direct value encoding can be used in problems, where some complicated values, such as real numbers, are used.

Use of binary encoding for this type of problems would be very difficult. In value encoding, every chromosome is a string of some values. Values can be anything connected to problem, form numbers, real numbers or chars to some complicated objects.

Chromosome A	1.0	5	3	2	6	7
Chromosome B	A	B	D	E	I	F
Chromosome C	back	right	Back	left	left	Right

f) Tree Encoding

This encoding is mainly used for evolving program expressions for genetic programming. Every chromosome is a tree of some objects such as functions and commands of programming language.

3.7.3.5 Breeding

The breeding process is the heart of the genetic algorithm. It is in this process, the search process creates new and hopefully fitter individuals (Sivanandam and Deepa, 2008). The breeding cycle consists of four steps. They are:

a. Selecting parents.

b. Crossing the parents to create new individuals (offspring or children).

c. Mutation of the offsprings.

d. Replacing old individuals in the population with the new ones.

a) Selection

Selection is the process of choosing two parents from the population for crossing. After deciding on an encoding, the next step is to decide how to perform selection. The purpose of selection is to emphasize fitter individuals in the population in hopes that their off springs have higher fitness. Chromosomes are selected from the initial population to be parents for reproduction. Selection is a method that randomly picks chromosomes out of the population according to their evaluation function. The higher the fitness function, the more chance an individual has to be selected. The various selection methods are discussed as follows:

i. Roulette Wheel Selection

Roulette selection is one of the traditional GA selection techniques. The commonly used reproduction operator is the proportionate reproductive operator where a string is selected from the mating pool with a probability proportional to the fitness. The principle of roulette selection is a linear search through a roulette wheel with the slots in the wheel weighted in proportion to the individual's fitness values. A target value is set, which is a random proportion of the sum of the fit nesses in the population (Sivanandam and Deepa, 2008). The population is stepped through until the target value is reached. This is a moderately strong selection technique.

ii. Random Selection

This technique randomly selects a parent from the population. In terms of disruption of genetic codes, random selection is a little more disruptive, on average, than roulette wheel selection.

iii. Rank Selection

Rank Selection ranks the population and every chromosome receives fitness from the ranking. The worst has fitness 1 and the best has fitness N. It results in slow convergence but prevents too quick convergence. It also keeps up selection pressure when the fitness variance is low.

iv. Tournament Selection

The tournament selection strategy provides selective pressure by holding a tournament competition among Nu individuals. The best individual from the tournament is the one with the highest fitness, which is the winner of Nu. Tournament competitions and the winner are then inserted into the mating pool. The tournament competition is repeated until the mating pool for generating new offspring is filled. The mating pool comprising of the tournament winner has higher average population fitness. The fitness difference provides the selection pressure, which drives GA to improve the fitness of the succeeding genes.

b) Crossover (Recombination)

Crossover is the process of taking two parent solutions and producing from them a child. After the selection (reproduction) process, the population is enriched with better individuals. Crossover operator is applied to the mating pool with the hope that it creates a better offspring.
Crossover is a recombination operator that proceeds in three steps:

i. The reproduction operator selects at random a pair of two individual strings for the mating.
ii. A cross site is selected at random along the string length.
iii. Finally, the position values are swapped between the two strings following the cross site.

i. Single Point Crossover

The traditional genetic algorithm uses single point crossover, where the two mating chromosomes are cut once at corresponding points and the sections after the cuts are exchanged.

Chromosome 1	**1 1 1 1 1**	**0 0 1 1 0 1 0**
Chromosome 2	**0 1 1 1 0**	**0 0 0 1 1 0 0**

Offspring 1	**1 1 1 1 1 0 0 0 1 1 0 0**
Offspring 2	**0 1 1 1 0 0 0 1 1 0 1 0**

ii. Two Point Crossover

In two-point crossover, two crossover points are chosen and the contents between these points are exchanged between two mated parents.

Chromosome 1	**2 3 4 5 6**	**7 8 9 0 1**	**5 7**
Chromosome 2	**A C B E F**	**B R T S E**	**R G**

Offspring 1	**2**	**3**	**4**	**5**	**6**	**B**	**R**	**T**	**S**	**E**	**5**	**7**
Offspring 2	**A**	**C**	**B**	**E**	**F**	**7**	**8**	**9**	**0**	**1**	**R**	**G**

iii. Uniform Crossover

Uniform crossover is quite different from the N-point crossover. Each gene in the offspring is created by copying the corresponding gene from one or the other parent chosen according to a random generated binary crossover mask of the same length as the chromosomes. Where there is a 1 in the crossover mask, the gene is copied from the first parent, and where there is a 0 in the mask the gene is copied from the second parent. The number of effective crossing point is not fixed, but will average L/2 (where L is the chromosome length).

Chromosome 1	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**1**
Chromosome 2	**0**	**0**	**0**	**1**	**1**	**0**	**1**	**0**
Mask	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**0**
Offspring 1	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**
Offspring 2	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**

iv. Three Parent Crossover

In this crossover technique, three parents are randomly chosen. Each bit of the first parent is compared with the bit of the second parent. If both are the same, the bit is taken for the offspring otherwise; the bit from the third parent is taken for the offspring.

Chromosome 1	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**1**
Chromosome 2	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**1**
Chromosome 3	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**0**
Offspring 1	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**1**

c) Mutation

After crossover, the strings are subjected to mutation. Mutation prevents the algorithm to be trapped in a local minimum. Mutation plays the role of recovering the lost genetic materials as well as for randomly disturbing genetic information If crossover is supposed to exploit the current solution to find better ones, mutation is supposed to help for the exploration of the whole search space. It introduces new genetic structures in the population by randomly modifying some of its building blocks. Mutation helps escape from local minima's trap and maintains diversity in the population. There are many different forms of mutation for the different kinds of representation. The important parameter in the mutation technique is the mutation probability (Pm). The mutation probability decides how often parts of chromosome will be mutated. If there is no mutation, offspring are generated immediately after crossover (or directly copied) without any change.

d) Replacement

Replacement is the last stage of any breeding cycle. Basically, there are two kinds of methods for maintaining the population: generational updates and steady state updates. The basic generational update scheme consists in producing N children from a population of size N to form the population at the next time step (generation), and this new population of children completely replaces the parent selection. In a steady state update, new individuals are inserted in the population as soon as they are created, as opposed to the generational update where an entire new generation is produced at each time step.

3.7.4 Search Termination

In short, the various stopping condition are listed as follows:

a. Maximum generations:

The genetic algorithm stops when the specified number of generations has evolved.

b. Elapsed time:

The genetic process will end when a specified time has elapsed.

c. No change in fitness:

The genetic process will end if there is no change to the population's best fitness for a specified number of generations. In this case, the highest-ranking solution's fitness reaches a plateau such that successive iterations no longer produce better results.

3.7.5 Outline of Genetic Algorithm

An algorithm is a series of steps for solving a problem. A genetic algorithm is a problem-solving method that uses genetics as its model of problem solving. It's a search technique to find approximate solutions to optimization and search problems. The basic of genetic algorithm is as follows:

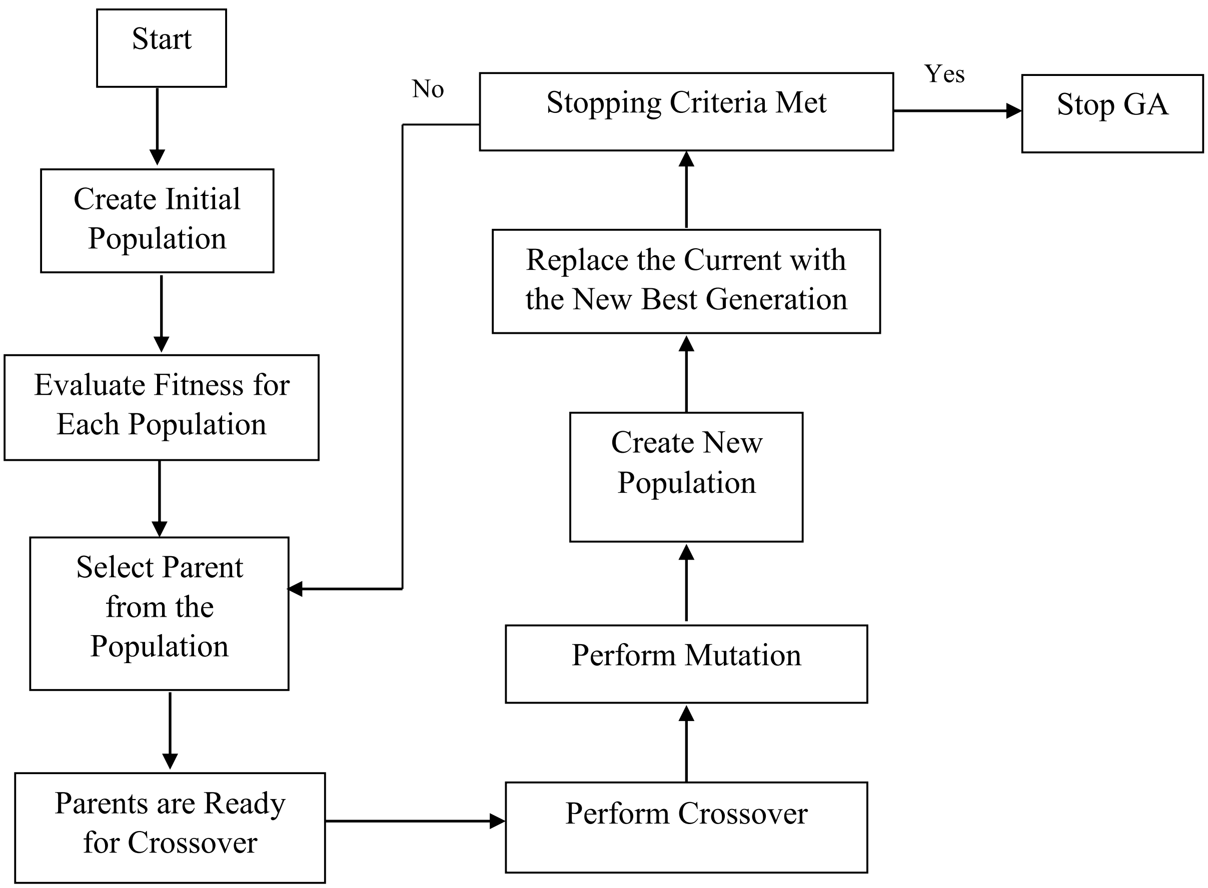


Figure 3.8: Outline of Genetic Algorithm

3.7.6 Advantages and Limitations of Genetic Algorithm

The advantages of genetic algorithm include:

- **i.** Solution space is wider
- **ii.** Robust search algorithm
- **iii.** Require no knowledge or gradient information about the response surface

- **iv.** Discontinuities present on the response surface have little effect on overall optimization performance
- **v.** The fitness landscape is complex
- **vi.** Easy to discover global optimum
- **vii.** The problem has multi objective function
- **viii.** Easily modified for different problems
- **ix.** Handles noisy functions well
- **x.** Handles large search spaces easily
- **xi.** Very robust to difficulties in the evaluation of the objective function
- **xii.** They are resistant to becoming trapped in local optima
- **xiii.** They perform very well for large-scale optimization problems
- **xiv.** Can be employed for a wide variety of optimization problems

The limitations of genetic algorithm include:

- **i.** Random search is time consuming
- **ii.** Cannot tell when or if an optimum solution is obtained
- **iii.** Configuration is not straightforward
- **iv.** The problem of identifying fitness function
- **v.** Premature convergence occurs
- **vi.** The problem of choosing the various parameters like the size of the population, mutation rate, cross over rate, the selection method and its strength
- **vii.** Cannot easily incorporate problem specific information
- **viii.** Not good at identifying local optima
- **ix.** No effective terminator

Currently, Genetic Algorithms is used along with neural networks and fuzzy logic for solving more complex problems. Because of their joint usage in many problems, these together are often referred to by a generic name: “soft-computing”.

Chapter-4

MODEL FORMULATION

4.1 PROBLEM IDENTIFICATION

An important component in SCN design and analysis is the establishment of appropriate performance measures. A performance measure, or a set of performance measures, is used to determine efficiency and/or effectiveness of an existing system, to compare alternative systems, and to design proposed systems. These measures are categorized as qualitative and quantitative. Suppliers service level, Customer satisfaction, robustness and flexibility belong to qualitative performance measures. Quantitative performance measures are also categorized by: (1) objectives that are based directly on cost or profit such as cost minimization, quality maximization, profit maximization, etc. and (2) objectives that are based on some measure of customer responsiveness such as delivery time minimization, customer response time minimization, lead time minimization, etc.

The network design problem is one of the most comprehensive strategic decision problems that need to be optimized for long-term efficient operation of whole supply chain. It determines the number, location, capacity and type of plants, warehouses, and distribution centers to be used. It also establishes distribution channels, and the amount of materials and items to consume, produce, and ship from suppliers to customers. SCN design problems cover wide range of formulations ranged from simple single product type to complex multi-product one, and from linear deterministic models to complex non-linear stochastic ones.

In recent years, the supply chain network (SCN) design problem has been gaining importance due to increasing competitiveness introduced by the market globalization. Firms are obliged to maintain high customer service levels while at the same time they are forced to reduce cost and maintain profit margins. Traditionally, marketing, distribution, planning, manufacturing, and purchasing organizations along the supply chain operated independently. These organizations have their own objectives and these are often conflicting. But, there is a need for a mechanism through which these different functions can be integrated together. Supply chain management (SCM) is a strategy through which such integration can be achieved.

4.2 PROBLEM STATEMENT

The problem considered in this research is a hypothetical one which is generic in nature and can be applied to any situation where a manufacturing plant produces products and uses assembly factories, and distribution centers (DCs) to transfer products to customer zones. Therefore, it is a four-echelon supply chain network which aims to minimize total cost and delivery time along with maximizing quality level and service performance using Taguchi quality loss function, for supplier selection and DC allocation. An illustration of the echelons is given in the Figure 4.1 below.

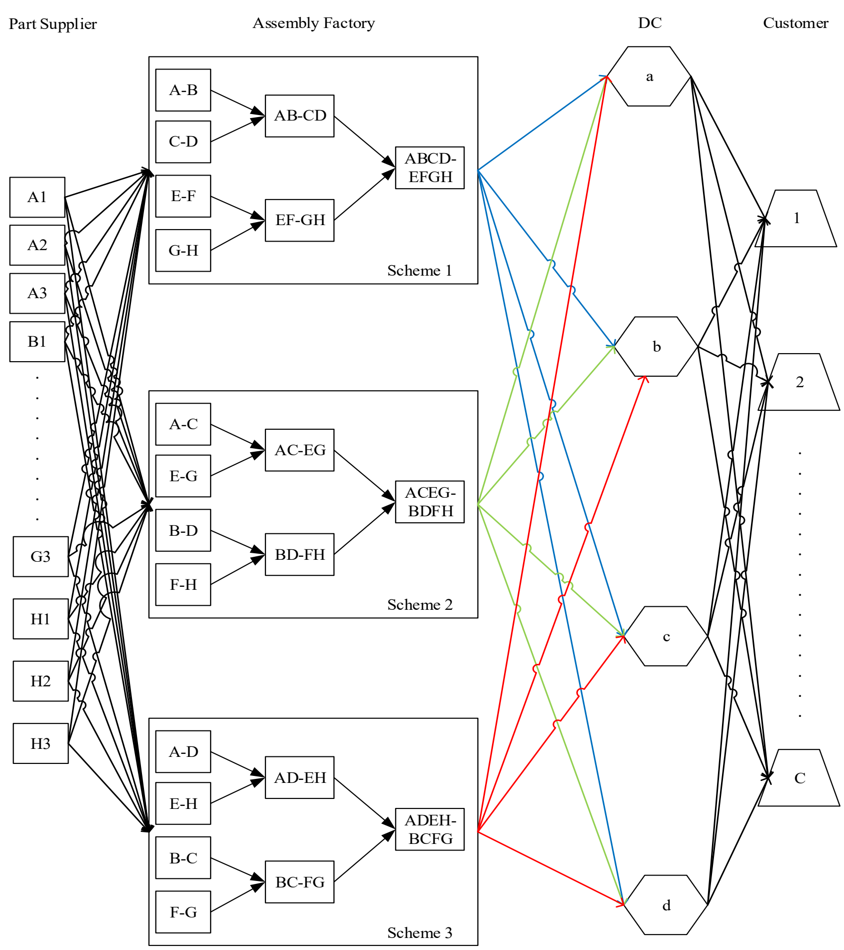


Figure 4.1: Arrangement of Echelons

It has been considered that the company has three assembly factories for converting the raw materials into finished products. In the first echelon, raw materials are purchased from twenty four suppliers depending on the minimum ordering cost, procurement cost, transportation cost and maximum quality grade. In the second echelon, three assembly factories assemble the raw material parts and produce a finished product. In the third echelon, three assembly factories transport the finished products to four different distribution centers and finally in the fourth echelon the DCs transport the products to potential customer zones. Hence, the number and location of assembly factories, DCs and customers, along with capacities of suppliers and demands of customers respectively, are known in advance.

So, the trade-off between costs, transportation time and supplier service performance leads the author of this work to formulate a mixed integer tri-objective SCN optimization model. One criterion tries to minimize the cost of selecting suppliers (ordering cost, procurement cost, transportation cost), assembly factories (transportation cost), distribution centers (transportation cost, fixed infrastructure cost), customer (transportation cost). The other two criteria cover the delivery time and supplier service level respectively.

4.3 ASSUMPTIONS OF THE STUDY

- Parameters are deterministic and known in advance and shortages are not permitted in all echelons of supply chain.
- Planning horizon is infinite.
- Objective functions (i.e., cost, delivery time, and quality) can have different weights.
- Product assembly plans are pre-known.
- Parts assembly is under the assembly schemes.
- Assembly activities begin when all parts are available.
- Assembly and transportation losses are not considered.
- The available candidate locations for locating the DCs are known.
- Locations of suppliers, plant, and customers are fixed and predefined.
- Suppliers do not require to produce all types of parts and may produce some of them.

4.4 NOMENCLATURE

4.4.1 Sets and Indices

i	Index of part, $i=1, 2, \ldots, I$
I	Total number of parts
x	Index of supplier, $x=1, 2, \ldots, Xi$
X_i	Total number of suppliers for i^{th} part
p	Index of assembly scheme, $p=1, 2, \ldots, P$
P	Total number of assembly schemes
d	Index of DC, $d=1, 2, \ldots, D$
D	Total number of DCs
f_d	Setup cost of DCs
c	Index of customer, $c=1, 2, \ldots, C$
C	Total number of Customers
$U_{i,x}$	Maximum production capacity of x^{th} supplier for i^{th} part
n	Index of assembly stage, $n=1, 2, \ldots, N^p$
N^p	Total number of assembly stages for p^{th} assembly scheme
a,b	Assembly factory a/b in assembly stage, $a= 1, 2, \ldots, A^p{}_n$; $b= 1, 2, \ldots, B^p{}_n$
$A^p{}_n$, $B^p{}_n$	Total number of assembly factories for n^{th} assembly stage in p^{th} assembly scheme

4.4.2 Parameters

D_c	Demand of c^{th} customer
$OC_{i,x}$	Order cost of i^{th} part from x^{th} supplier
$PC_{i,x}$	Purchase cost of i^{th} part from x^{th} supplier
TC_{i,x_a}	Transport cost of i^{th} part from x^{th} supplier to factory a in first assembly stage
$TC_{n,a\_n+1,b}$	Transport cost from factory a in assembly stage n to factory b in assembly stage n+1
$TC_{a,d}$	Transport cost from factory a in final assembly stage to DC d
$TC_{d,c}$	Transport cost from d^{th} DC to c^{th} customer
TT_{i,x_a}	Transport time in initial assembly stage, from x^{th} supplier to factory a for part i
$TT_{n,a\_n+1,b}$	Transport time from factory a in assembly stage n to factory b in assembly stage n+1, $n=1, 2, 3, \ldots, N^p-1$

$TT_{n-1,b\_n,a}$	Transport time from factory b in assembly stage n−1 to factory a in assembly stage n, n=2, 3, 4, …, N^p
$TT_{a,d}$	Transport time in final assembly stage from factory a to DC d
$TT_{d,c}$	Transport time in from d^{th} DC to c^{th} customer
$Q_{i,x}$	Quality of part i from supplier x
k	Loss co-efficient
A	Target value
Δ	Specification limit

4.4.3 Variables

W^p_{i,x_a}	Quantity of part i ordered and transported from supplier x to factory a in initial assembly stage in assembly scheme p
$W^p_{n,a\_n+1,b}$	Quantity of semi-finished products transported from factory a in an assembly stage n to factory b in assembly stage n+1 in assembly scheme p, n=1, 2, 3, …, N^p−1
$W^p_{n-1,b\_n,a}$	Quantity of semi-finished products from factory b in assembly stage n+1 to factory a in assembly stage n in assembly scheme p, n=2, 3, ..., N^p
$W^p_{a,d}$	Quantity of finished products in factory a in final assembly stage to DC d in assembly scheme p
$W^p_{d.c}$	Quantity of finished products in DC d to customer c in assembly scheme p
$M^p_{1,a}$	Maximum transport time of parts from x^{th} supplier to factory a for part i in initial assembly stage 1 in assembly scheme p
$M^p_{n,a}$	Maximum transport time of parts or semi-finished products to factory a in assembly stage n in assembly scheme p
M^p_d	Maximum transport time of product to distribution center d in assembly scheme p
M^p_c	Maximum transport time of product to customer c in assembly scheme p

4.4.4 Binary Variables

$S^p_{i,x}$	1: order part i from supplier x in assembly scheme p; 0: otherwise
y_d	1: location of DC in d^{th} center; 0: otherwise

4.5 MODEL FORMULATION

4.5.1 Objective Functions

Minimum Cost =

$$\sum_{i=1}^{I}\sum_{x=1}^{X_i} OC_{i,x}\, f\left(S_{i,x}^{p}\right) + \sum_{d=1}^{D} f_d\, y_d + k\sum_{i=1}^{I}\sum_{x=1}^{X_i}\frac{1}{Q_{i,x}^{2}}$$

$$+\sum_{p=1}^{P}\left\{\begin{array}{l} \sum_{i=1}^{I}\sum_{x=1}^{X_i}\sum_{a=1}^{A_1^p}\left[\left(PC_{i,x}+TC_{i,x}\right)W_{i,x\_a}^{p}\right] \\ +\sum_{n=1}^{N^p-1}\sum_{a=1}^{A_n^p}\sum_{b=1}^{B_n^p}\left(TC_{n,a\_n+1,b}W_{n,a\_n+1,b}^{p}\right) \\ +\sum_{a=1}^{A_{N^p}^{p}}\sum_{d=1}^{D} y_d\left(TC_{a,d}W_{a,d}^{p}\right) \\ +\sum_{d=1}^{D}\sum_{c=1}^{C}\left(TC_{d,c}W_{d,c}^{p}\right) \end{array}\right\}$$

…………………. (4.1)

$$f\left(S_{i,x}^{p}\right)=\begin{cases}1 \text{ if } \sum_{p=1}^{P} S_{i,x}^{p}>0 \quad \forall i,\forall x\\ 0 \quad \text{otherwise}\end{cases}$$

…………………. (4.2)

$$y_d=\begin{cases}1 \text{ if } f\left(S_{i,x}^{p}\right)>0 \ \forall i,\forall x\\ 0 \quad \text{otherwise}\end{cases}$$

…………………. (4.3)

$$k = A\Delta^2$$

…………………. (4.4)

Minimum Time =

$$Max\left\{TT_{d,c}.y_d + M_d^p \mid 1\le p\le P, 1\le c\le C, 1\le d\le D\right\}$$

…………………. (4.5)

$$M_{1,a}^{p} = Max\left\{TT_{i,x\_a}\, f(W_{i,x\_a}^{p}) \mid 1\le i\le I, 1\le x\le X\right\}\forall p;\forall a$$

…………………. (4.6)

$$f(W_{i,x\_a}^{p}) = \begin{cases} 1 \ \text{ if } \ f(W_{i,x\_a}^{p}) > 0 \ \forall \text{p}, \ \forall i, \ \forall x, \forall a \\ 0 \quad \text{otherwise} \end{cases}$$

…………………. (4.7)

$$M_{n,a}^{p} = Max\{M_{n-1,b}^{p} + TT_{n-1,b\_n,a} f(W_{n-1,b\_n,a}^{p}) \mid 1 \le b \le B_{n}^{p} \quad \forall p; \forall a; \quad n = 2,3,4,...,N^{p}\}$$

…………………. (4.8)

$$f(W_{n-1,b\_n,a}^{p}) = \begin{cases} 1 \ \text{ if } \ \text{W}_{n-1,b\_n,a}^{p} > 0 \ \forall \text{p}, \ \forall \text{a}, \ \forall b; \quad n = 2,3,4,...,N^{p} \\ 0 \quad \text{otherwise} \end{cases}$$

…………………. (4.9)

$$M_{d}^{p} = Max\left\{M_{N^{p}.a}^{p} + TT_{a,d} f\left(W_{a\_d}^{p}\right) \mid 1 \le a \le A_{N^{p}}^{p} \ \ \forall \text{p}; \forall d\right\}$$

…………………. (4.10)

$$f\left(W_{a\_d}^{p}\right) = \begin{cases} 1 \ \text{ if } \ \text{W}_{a,d}^{p} > 0 \ \forall \text{p}, \ \forall \text{a}, \ \forall d \\ 0 \quad \text{otherwise} \end{cases}$$

…………………. (4.11)

Maximum Quality =

$$\sum_{p=1}^{P} \sum_{i=1}^{I} \sum_{x=1}^{X_i} \sum_{a=1}^{A_1^p} \left(Q_{i,x} W_{i,x\_a}^{p}\right)$$

…………………. (4.12)

4.5.2 Constraints

$$\sum_{p=1}^{P} \sum_{x=1}^{X_i} \sum_{a=1}^{A_1^p} W_{i,x\_a}^{p} = \sum_{c=1}^{C} D_c \ \ \forall i$$

…………………. (4.13)

$$\sum_{p=1}^{P} \sum_{a=1}^{A_1^p} W_{i,x\_a}^{p} \le U_{i,x} \quad \forall i; \forall x$$

…………………. (4.14)

$$\sum_{i=1}^{I} \sum_{x=1}^{X_i} W_{i,x\_a}^{p} = \sum_{b=1}^{B_2^p} W_{1,a\_2,b}^{p} \quad \forall p; \forall a$$

…………………. (4.15)

$$\sum_{b=1}^{B_{n-1}^{p}} W_{n-1,b\_n,a}^{p} = \sum_{b=1}^{B_{n+1}^{p}} W_{n,a\_n+1,b}^{p} \quad \forall p; \forall a; \quad n = 2,3,4,..., N^{p} - 1$$

………………… (4.16)

$$\sum_{b=1}^{B_{N^{p}-1}^{p}} W_{N^{p}-1,b\_N^{p},a}^{p} = \sum_{d=1}^{D} W_{a,d}^{p} \quad \forall p; \forall a$$

………………… (4.17)

$$\sum_{d=1}^{D} W_{a,d}^{p} = \sum_{c=1}^{C} W_{d,c}^{p} \quad \forall p; \forall a$$

………………… (4.18)

$$\sum_{d=1}^{D} W_{d,c}^{p} = D_{c} \quad \forall p; \forall c$$

………………… (4.19)

4.5.3 Limit of Variables

$S_{i,x}^{p} = 0$ or $1 \quad \forall p; \forall i; \forall x$

………………… (4.20)

$\mathrm{y}_{d} = 0$ or $1 \quad \forall d$

………………… (4.21)

$W_{i,x\_a}^{p} \geq 0$ and $W_{i,x\_a}^{p} \in$ Integer $\quad \forall p; \forall i; \forall x; \forall a$

………………… (4.22)

$W_{n,a\_n+1,b}^{p} \geq 0$ and $W_{n,a\_n+1,b}^{p} \in$ Integer $\quad \forall p; \forall a; \forall b; \quad n = 1,2,3,4,..., N^{p} - 1$

………………… (4.23)

$W_{n-1,b\_n,a}^{p} \geq 0$ and $W_{n-1,b\_n,a}^{p} \in$ Integer $\quad \forall p; \forall a; \forall b; \quad n = 2,3,4,..., N^{p}$

………………… (4.24)

$W_{a\_d}^{p} \geq 0$ and $W_{a\_d}^{p} \in$ Integer $\quad \forall p; \forall a; \forall d$

………………… (4.25)

$W_{d\_c}^{p} \geq 0$ and $W_{d\_c}^{p} \in$ Integer $\quad \forall p; \forall d; \forall c$

………………… (4.26)

$M_{1,a}^{p} \geq 0$ and $M_{1,a}^{p} \in$ Integer $\quad \forall p; \forall a$

………………… (4.27)

$$M_{n,a}^{p} \geq 0 \text{ and } M_{n,a}^{p} \in \text{ Integer } \quad \forall p; \forall n; \forall a$$

………………… (4.28)

$$M_{d}^{p} \geq 0 \text{ and } M_{d}^{p} \in \text{ Integer } \quad \forall p; \forall d$$

………………… (4.29)

$$M_{c}^{p} \geq 0 \text{ and } M_{c}^{p} \in \text{ Integer } \quad \forall p; \forall c$$

………………… (4.30)

4.6 MODEL DESCRIPTION

The first objective in equation (4.1) minimizes the sum of total cost; consists of ordering cost, construction cost of DCs, quality loss cost, purchasing cost and transportation cost from suppliers to assembly factories, transportation cost from one assembly stage to another stage, transportation cost from assembly factories to DCs, and transportation cost from DCs to customers. Ordering cost and construction cost are fixed. Purchasing cost is the price associated with the purchasing of any item. Equation (4.2) and (4.3) represent binary decision variables restriction. Equation (4.4) represents loss co-efficient of quality loss cost.

The second objective in equation (4.5 to 4.11) stand for minimization of delivery time from suppliers to customers through distribution centers is considered.

Equation (4.12) is used for maximizing the quality of suppliers' products as the third objective of proposed model.

In this section, constraints of proposed model are explained. Equation (4.13) stands to ensure the equilibrium of supply and demand between suppliers and customers. Suppliers have their own constraint which is a pre-known production capacity. Equation (4.14) shows that the quantity of each part cannot exceed the supplier's production capacity. The most important constraint is flow equilibrium. Inflow should be equal to out-flow for factories in all assembly lines. Equation (4.15) stands to ensure that total quantity of parts shifted to assembly factories is equal to the total amount of all parts purchased from the suppliers. Similarly, equation (4.16) ensures that the quantity of all semi-finished products must be equal at each stage of assembly factories. Equation (4.17) represents that the quantity of all finished products must be equal to total demand quantity of all

DCs. Equation (4.18) ensures that total demand quantity of all DCs must be equal to the quantity of all finished products. Equation (4.19) represents that total demand quantity shifted from of all DCs to the customers is equal to customer demands.

The binary variables indicate in equation (4.20) and (4.21) whether to purchase items from a supplier or not, and whether to set up DC or not respectively.

Other equations from (4.22 to 4.30) are non-negativity constraints and each variable in these constraints are integer.

Chapter-5

MODEL IMPLEMENTATION

The mathematical model developed in Chapter 4 must be tested for two reasons, firstly for its functionality to be proven, and secondly the outcomes of the conflicting objectives to be analyzed. Therefore, a test problem has to be designed, for which the model is solved with a specific solving technique, and at the end results from the tests can be analyzed and presented in a suitable way. For that reason, this chapter is divided in three sections:

1. Case Description and Data Population
2. Solving Method Characteristics

First of all, a use case example is constructed and populated with all the necessary data. The example has to be suitable for testing the problem, in order to have proper outcomes for assessing the model. Furthermore, the solving method characteristics and settings have to be determined, for having an efficient approach towards the solving process. At the end, the actual tests should be performed, and the solutions are to be analyzed in a proper way for having quality conclusions in the next chapter. Since this is a multi-objective problem, the attention is set on the various trade-off analysis between the conflicting objective functions.

5.1 CASE DESCRIPTION

In designing a testing problem, several aspects were taken into considerations. It began from the availability of data, and the possibility of having real business examples. If real case data is not available, complexity arises in assuming hypothetical data and to make the data close to real business. The transformation of the data for populating the models can be sometimes a highly complex task. Having the reasons above, and the complexity of the problem and innovations that are embedded in the model, as well as the problems that the research faced with finding real business case on which the model can be tested, gave some directions. It is chosen an option for designing a test problem specifically suitable for this research, based on a hypothetical supply chain, where the data is conducted by reasonable assumptions and the author used the data provided at Che and Chiang (2010) as the base of assumption.

In particular, it is chosen a four-level supply chain compiled out of twenty-four suppliers, three assembly factories for converting raw materials to final product, four distribution centers (DCs) among which any one can be chosen to satisfy the objective and six customer zones to be served. The supply chain problem is assuming production of a product consists of eight parts, where two potential assembly factories can be chosen for its production. The reason for these simplifications is partially explained above, but it lays in the complexity of the mathematical model. Since it is an NP-hard linear problem, leads towards the fact that solving this kind of problems can be a highly delicate procedure, where not just using evolutionary algorithms should be a standard way, but even developing new ones, or customizing the existing ones. This is considered as not efficient and reasonable to be done within a frame of this thesis research. However, the testing case can be upgraded in the future, where a more complex supply chain problem will be addressed. For this reason, there must be existing real data, and a specific action dedicated for designing.

5.2 DATA COLLECTION

In order to enable the solving of the model to generate concrete results, it has to be populated with data. For that reason, a collection of data was performed, where data was picked from various sources (Che and Chiang 2010), some reasonable assumptions were taken, and all the necessary calculations were done. For having a fully functional use case, the data was collected for several elements, mainly focused on the costs, transportation time, transportation cost, quality grade, capacity of suppliers, customer demand and capacities of the different facilities of the supply chain. All the data is presented on the basis of the basis of the assumptions (hypothetical data).

Table 5.1: Suppliers Data Sheet and the Cost and Time of Transport from Suppliers to Assembly Factory

Supplier	Capacity	Ordering Cost	Procurement Cost	Quality Grade	Transportation Cost/Time		
					Scheme 1	Scheme 2	Scheme 3
A1	700	4500	90	9	11/14	14/11	13/12
A2	800	5000	85	8	13/11	12/12	11/13
A3	900	4200	95	10	15/10	10/15	12/13
B1	750	4400	97	10	14/10	10/14	11/13
B2	650	4000	93	9	12/15	15/12	13/14
B3	850	4500	88	9	11/14	12/13	14/11
C1	750	4800	87	8	10/14	14/10	12/12
C2	600	4300	92	9	12/14	13/13	14/12
C3	700	4000	96	10	13/11	11/13	12/12
D1	900	4800	99	10	15/10	12/13	10/15
D2	800	4300	94	9	14/11	13/12	11/14
D3	700	4900	89	8	11/15	15/11	13/13
E1	700	4600	91	9	14/11	11/14	12/13
E2	850	4100	95	9	13/15	15/13	14/14
E3	600	4000	98	10	10/14	13/11	14/10
F1	700	4700	86	8	12/14	14/12	13/13
F2	850	4500	89	9	15/11	11/15	13/13
F3	800	4200	94	9	11/14	13/12	14/11
G1	900	4700	92	9	10/15	15/10	12/13
G2	750	4300	97	10	14/10	12/12	10/14
G3	700	5000	90	9	12/12	13/11	11/13
H1	650	4000	98	10	14/12	13/13	12/14
H2	750	4400	93	9	11/14	14/11	13/12
H3	650	4900	89	8	10/15	15/10	13/12

Table 5.2: The Cost and Time of Transport from the Assembly Factory of the Supply Chain to Another Assembly Factory and DCs

Transportation Cost/Time					
Scheme 1		Scheme 2		Scheme 3	
AB→ ABCD	15/20	AC→ ACEG	20/15	AD→ ADEH	19/16
CD→ ABCD	17/16	EG→ ACEG	16/17	EH→ ADEH	16/17
EF→ EFGH	19/17	BD→ BDFH	18/18	BC→ BCFG	17/19
GH→ EFGH	16/17	FH→ BDFH	15/18	FG→ BCFG	18/15
ABCD→ ABCDEFGH	26/30	ACEG→ ACEGBDFH	30/26	ADEH→ ADEHBCFG	28/28
EFGH→ ABCDEFGH	36/40	BDFH→ ACEGBDFH	27/27	BCFG→ ADEHBCFG	25/29
ABCDEFGH→ a	36/40	ACEGBDFH→ a	39/37	ADEHBCFG→ a	40/36
ABCDEFGH→ b	35/38	ACEGBDFH→ b	36/37	ADEHBCFG→ b	38/35
ABCDEFGH→ c	38/35	ACEGBDFH→ c	35/38	ADEHBCFG→ c	36/37
ABCDEFGH→ d	40/37	ACEGBDFH→ d	38/39	ADEHBCFG→ d	37/40

Table 5.3: The Cost and Time of Transport from DCs to Customers

Transportation Cost/Time							
DC \ Customer		Customer					
		1	2	3	4	5	6
DC	a	26/30	25/28	28/25	30/27	32/28	23/25
	b	29/27	26/27	25/28	28/29	30/32	25/27
	c	30/26	28/25	26/27	27/30	32/25	26/25
	d	26/28	27/26	24/28	31/30	25/24	28/32

Table 5.4: Demand Data of DCs

DCs	Demand
a	200
b	150
c	250
d	200

Table 5.5: Demand Data of Customers

Customers	Demand
1	150
2	150
3	100
4	100
5	100
6	200

Data for loss coefficient (k):

The value of the loss coefficient (k) is calculated based on the decision maker's desired specification limits. For example,

A = Target value = 100%

Δ = Specification limit = 80% (lower specification limit set by the decision maker)

$k = A\Delta^2 = 100 \times (0.8)^2 = 64$

$L = 64/(0.8)^2 = 100$ (Taguchi loss at lower specification limit)

5.3 SOLVING METHOD CHARACTERISTICS

For the solving process, as mentioned before in Chapter 4, a fast and elitist Modified Pareto Genetic Algorithm (mPaGA) is used. For that reason, a mPaGA program in Java is used. MATLAB is also used to draw different pareto front. The complete code is presented in Appendix A, while here the procedure of implementing it is presented.

In order to use the code, there are necessary adjustment that should be done. These adjustments are not in the scripts themselves, but in the way of writing the functions in Java programming language. It was necessary to transform the mathematical formulation of the model into a specific programming language. While the writing of the code in the traditional linear solvers is intuitive, in this case there was a need of transforming the formulation model completely in the programming language.

In this research, a modified genetic algorithm is proposed as a population-based method. For this coding, the crossover and mutation operators are conducted for the improvement. Processes of the proposed algorithm are as follows:

Step 1: Chromosome Coding

Each chromosome is a string of bits that represents a pattern of feasible solution. A string consists of P assembly schemes and the chromosome is coded with integer. Each scheme consists of two segments. The 1st segment represents a transportation pattern from suppliers to assembly factory, and its gene values represent the transportation quantities from suppliers to assembly factories. The 2nd segment represents a transportation pattern from assembly factory to customer, and its gene values represent the transportation quantities from assembly factories to customers.

Step 2: Generation of Initial Population

With all constraints (e.g. equilibrium of supply and demand, limit of production capacity, flow equilibrium, and limit of variables) satisfied, the value of each genetic code of the initial population is generated.

Step 3: Calculation of Objective Function Value

Substitute the genetic value of each population in the optimization mathematical model for BOSC problems. We can obtain objective function values Cost, Time and Quality from objective functions.

Step 4: Population Sorting

Based on the objective function value of each chromosome, the population is sorted into non-dominated solution sets by grade. The non-dominated solution individuals are selected from the population and defined as grade 1 fist of all. And then, for the remaining individuals of the population, the non-dominated solution individuals are selected and defined as grade 2. Screen the individuals and increase the grade value until the entire population is scored. The individual assigned the smaller grade value, represents it is the better one.

Step 5: Calculation of Fitness Function Value

Individuals in the same grade have the same fitness function value, which guarantees that solutions in the same grade to have the same reproduction ratio. The fitness function is as shown in equation below, in which F(X) represents the fitness function value for individual X, m indicates the maximum grade, j(X) denotes the grade for individual X, and n_j denotes the number of solutions for grade j.

$$F(X) = \frac{m - j(X) + 1}{\sum_{j=1}^{m}(j \times n_j)}$$

Step 7: Reproduction

The roulette wheel method is a quite common reproduction method in the GA, and that is applied in this study.

Step 8: Crossover Operator

Randomly select two chromosomes from the population and select equal-sized assembly schemes from each selected chromosome. And then the selected schemes attempt to interchange their positions.

For maintaining the equilibrium of supply and demand, the equilibrium mechanism is proposed after crossover operator has been used.

Step 9: Mutation Operator

A two-stage mutation operator is presented for this problem. In the first stage, randomly select an assembly scheme from chromosome, and mutate the value of each gene in this selected scheme with the same ratio. In the second stage, randomly select genes mutated in stage one, and then mutate their values. Based on mutated scheme string, proceed with the elitist preservation to obtain a new population while satisfying all constraints. For maintaining the feasibility of each individual, the feasibility-adjustment mechanism is used after the mutation operator has been used.

Step 10: Elitist Preservation Strategy

Integrate the parental generation with the filial generation into a population. Proceed with sorting based on the objective function value of each individual, preserve the first 50% of individuals in the population and eliminate the other 50%.

Step 11: Generation of New Population

The individuals preserved in the previous step become a new population.

Step 12: Judgment of Stop Condition

The stop condition is set as the generation number; it will stop when the evolution number reaches the generation number.

Step 13: Obtainment of Optimal Strategy

The solution provides the optimal strategy for BOSC planning which involves the following information: the selected suppliers and procurement quantity, and transportation quantity of logistic in the supply chain.

The procedure of mPaGA is shown below in Figure 5.1.

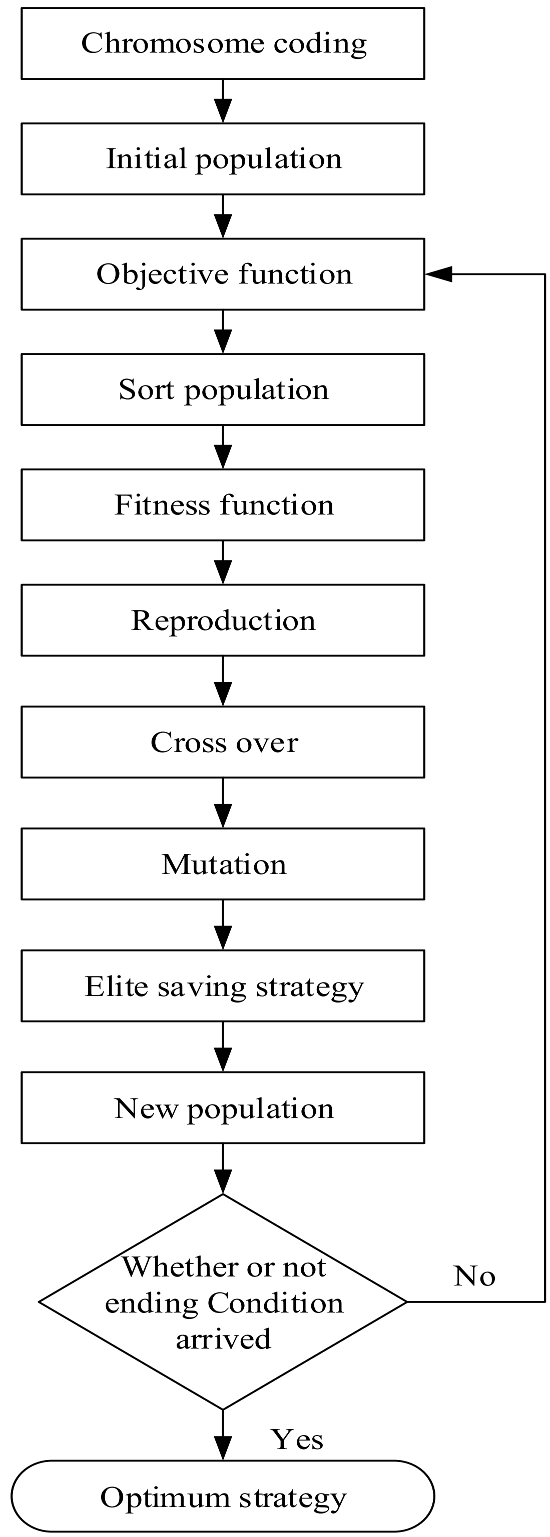


Figure 5.1: Procedure of mPaGA

The procedure of deterministic algorithm from DC to Customer problem can be algorithmically stated as follows.

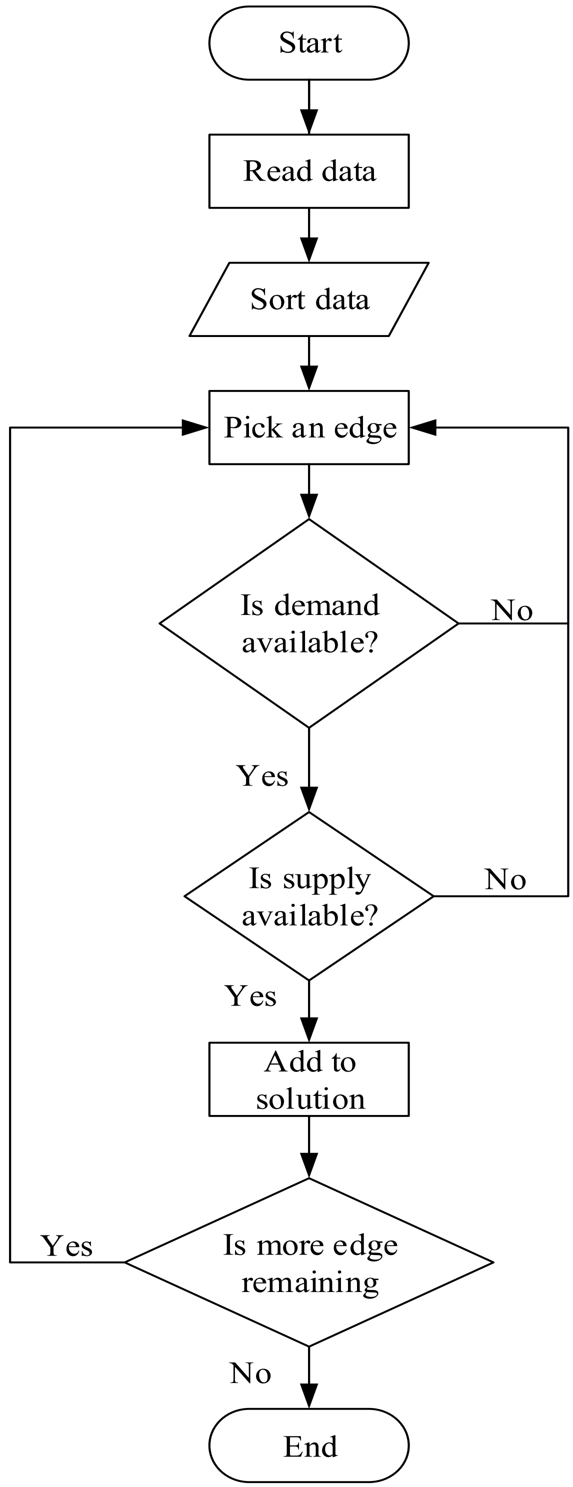


Figure 5.2: Flow Chart of Deterministic Algorithm from DC to Customer Problem

Moreover, in order the behavior of the algorithm to be understood, and test the mathematical model in a coded version, several small problems were run in order to define the best settings. As mentioned in the previous section, parameters of the genetic algorithm are crossover, mutation, population size, and generation number. Different levels of these factors are shown in Table 5.6.

The Taguchi method is employed for design of optimum parameter combination. Taguchi classifies objective functions into three categories: the smaller-the-better type, the larger-the-better type, and the nominal-is-best type. Since almost all objective functions in minimization problems are classified in the smaller-the-better type, the corresponding S/N ratio is
$S/N_S = -10 * \log_{10} [(\Sigma(\text{Fitness Value})^2/n]$.

Table 5.6: Factor Levels of Proposed Algorithm

Factor	Symbol	Level	Type
Initial Population	A	4	A(1)=100, A(2)=150, A(3)=200, A(4)=250
Generation Number	B	4	B(1)=500, B(2)=1000, B(3)=1500, B(4)=2000
Crossover Number	C	4	C(1)=2500, C(2)=4000, C(3)=5500, C(4)=7000
Mutation Number	D	4	D(1)=1000, D(2)=1500, D(3)=2000, D(4)=2500

The signal to noise ratio (SN ratio) has been used to find the largest one in responses as the optimal level of the parameters. L_{16} (4^4) number of run has used, therefore only 16 experiments for setting the parameters of proposed algorithm have been needed. Minitab 17 have been used to obtain the response diagram of the S/N ratio values, as shown in figure 5.3, from which the optimal parameter combination is initial population: 100, generation number: 1500, crossover: 2500, and mutation 2000.

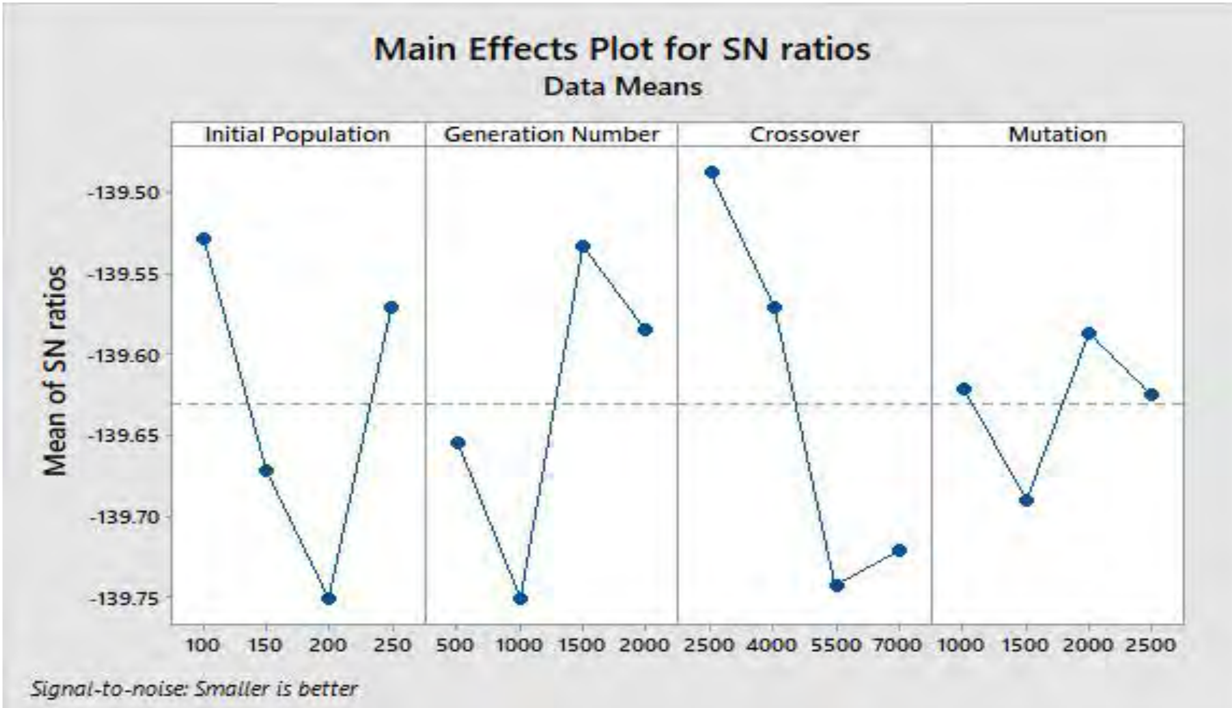


Figure 5.3: Main Effects Plot for S/N Ratios

The algorithms are written by Microsoft Visio Professional 2013, the code is implemented in Java SE 8u141, and are ran in Intel Core i3 CPU 2.00 GHz with 8 GB RAM. Even though the speed was not a problem in the testing of the code, the assessment of the algorithm is out of scope for this research. There are no intentions of performing tests on the particular efficiency of the algorithm and comparing it with other approaches. Rather, it is used as a novelty in the optimization of supply chain network designs, where it is aimed for generation of sufficient quality Pareto-optimal solutions.

Chapter-6

RESULTS & FINDINGS

In this thesis work a multi-echelon supply chain network has been developed. It's a multi-criteria optimization model which minimizes supply chain costs, delivery time and maximizes supplier service quality. To show the effectiveness of the model a numerical problem has been developed and later it is solved using Modified Pareto Genetic Algorithm (mPaGA).

The proposed multi-objective time-cost-quality optimization problem has been solved with modified pareto genetic algorithm technique which is coded on Java SE 8u141 and ran on Intel Core i3 CPU 2.00 GHz with 8 GB RAM.

Since in a multi-objective optimization problem (MOOP), there can never exist a single absolute solution that can satisfy all the objectives to their best. For two or more objectives, each objective corresponds to a different optimal solution, but none of the trade-off solutions is optimal with respect to all objectives. Therefore, multi-objective swarm intelligence techniques like MOGA do not try to find one optimal solution but all the trade-off solutions. Though an optimal solution may have minimum total combined objective function value, it may not be the best solution with respect to all the objectives simultaneously. Because of the nature of MOOP, solution may be optimal with respect to one objective or the total fitness may be minimum by best satisfying all the objectives but may be a poor candidate for a particular objective. The proposed algorithm should determine combinations of suppliers and product volumes to transport through the supply chain network. This is a multi-criteria problem that considers cost, quality, and transportation times. The problems consider different objectives, but the constraint space is the same for them. Hence, it is desirable to generate many optimal solutions considering all the three objectives. Therefore, MOGA is designed to obtain non-dominated Pareto optimal solutions, where no one solution is better than another.

In this section, an illustrative example involving a four-stage supply chain network is presented to demonstrate the effectiveness of the proposed approach (mPaGA). Figure 6.1, Figure 6.2, Figure 6.3, and Figure 6.4 show the typical four-stage supply chain network which includes part suppliers, assembly factories, distribution centers, and customers. A product is composed of eight parts (A, B, . . ., G, H) are assembled at the factories, and each part has three possible suppliers (A1, A2, A3, . . ., H1, H2, H3). Each supplier has different capacity, ordering cost, procurement cost, quality grade, transportation costs, and transportation time. There are three assembly plans based on assembly factories and feasible assembly methods. So, there are 24 individual suppliers to supply 8 types of raw materials for 3 assembly schemes. These three assembly schemes are arranged to assemble the raw materials and produced finished products. These finished products are transferred to four distribution centers (a, b, c, and d) based on the demand of DCs and considering transportation time and cost. After that DCs will send products to six customers (1, 2, 3, 4, 5, and 6) based on the customer demand and considering transportation time and cost. Earlier stated six customers (1, 2, 3, 4, 5 and 6) require 150, 150, 100, 100, 100 and 200 units, respectively.

6.1 RESULT OBTAINED USING MULTI-OBJECTIVE GENETIC ALGORITHM (MOGA)

After linking the database and applying the parameter setting obtained from Taguchi method (S/N ratio), calculate a Pareto optimal solution set. The Pareto optimal solution set includes 100 Pareto optimal solutions, and one of the distribution out of 100 is as shown in Figure 6.1, Figure 6.2, Figure 6.3 and Figure 6.4.

For maintaining total customer demand of 800, three optimum amount of finished goods from each scheme to be transferred to DCs. They are 223 from scheme-1 (Figure 6.1), 576 from scheme-2 (Figure 6.2), and 1 from scheme-3 (Figure 6.3). Considering cost and time, and demand of DCs, the fixed number of finished goods from each scheme are delivered to four DCs.

Distribution planning from DC to customer is shown in Figure 6.4. Stored finished goods are being delivered to customers from DCs based on customer demand, transportation time and transportation cost.

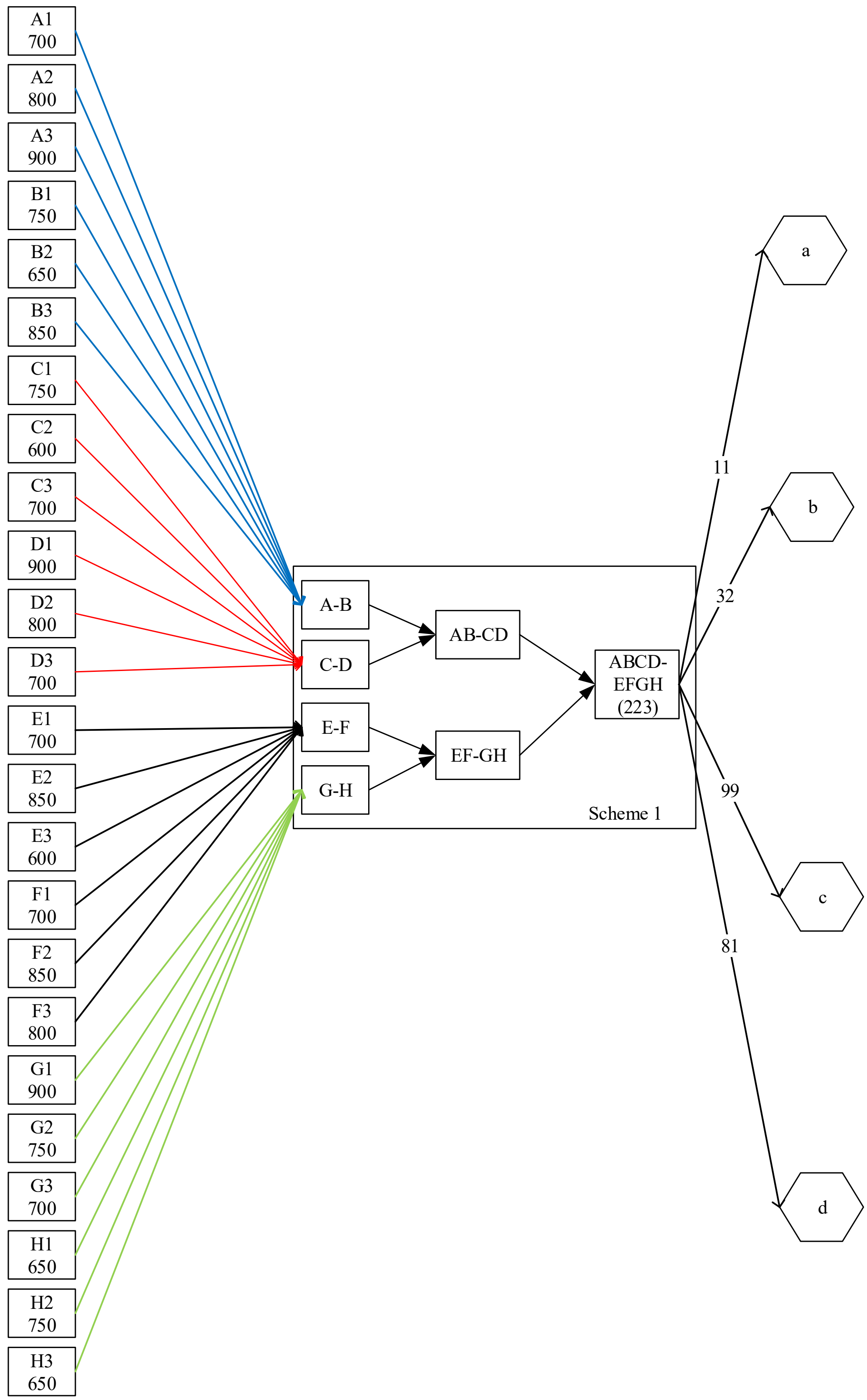


Figure 6.1: Distribution Planning for the Supply Chain (Scheme-1)

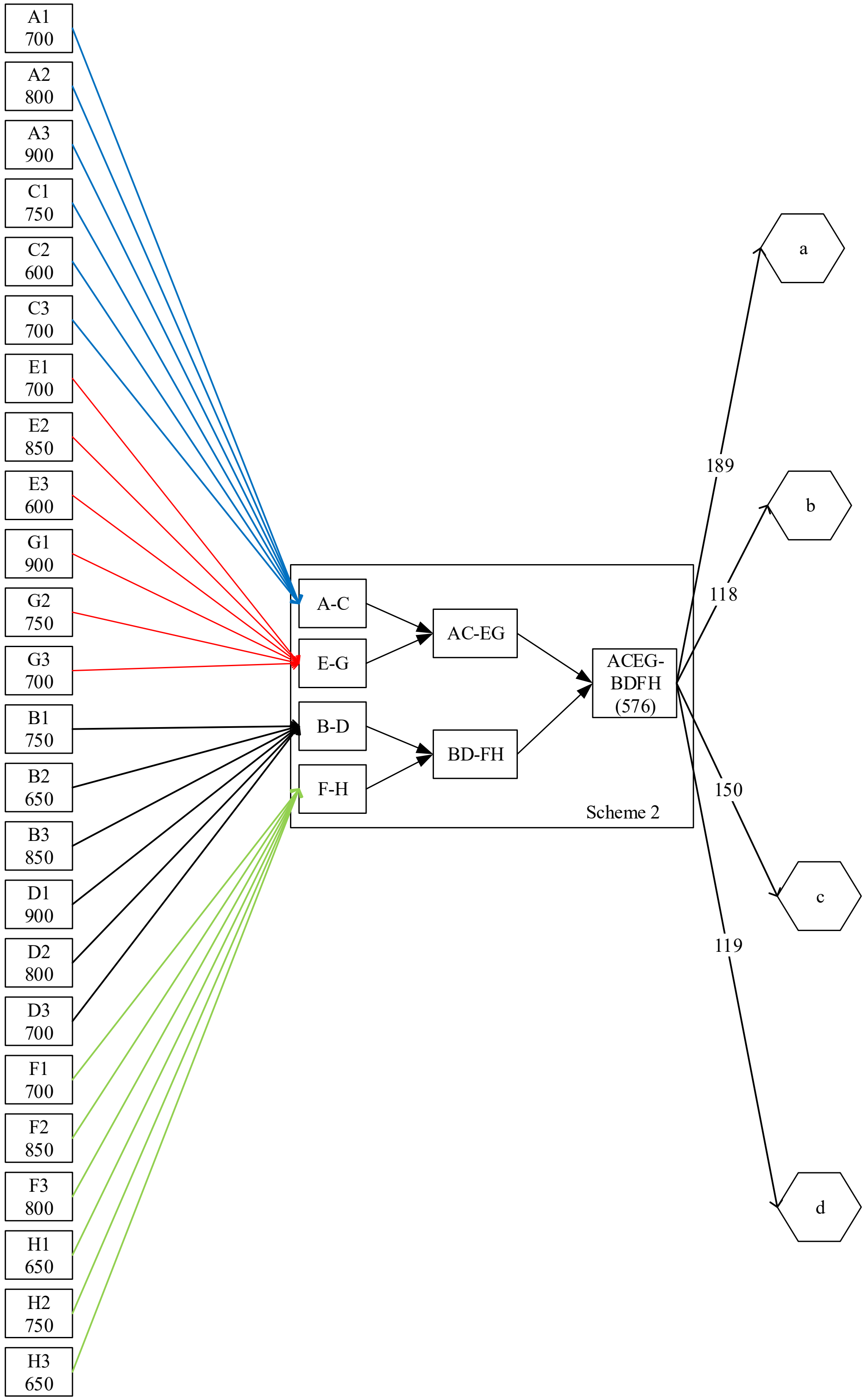


Figure 6.2: Distribution Planning for the Supply Chain (Scheme-2)

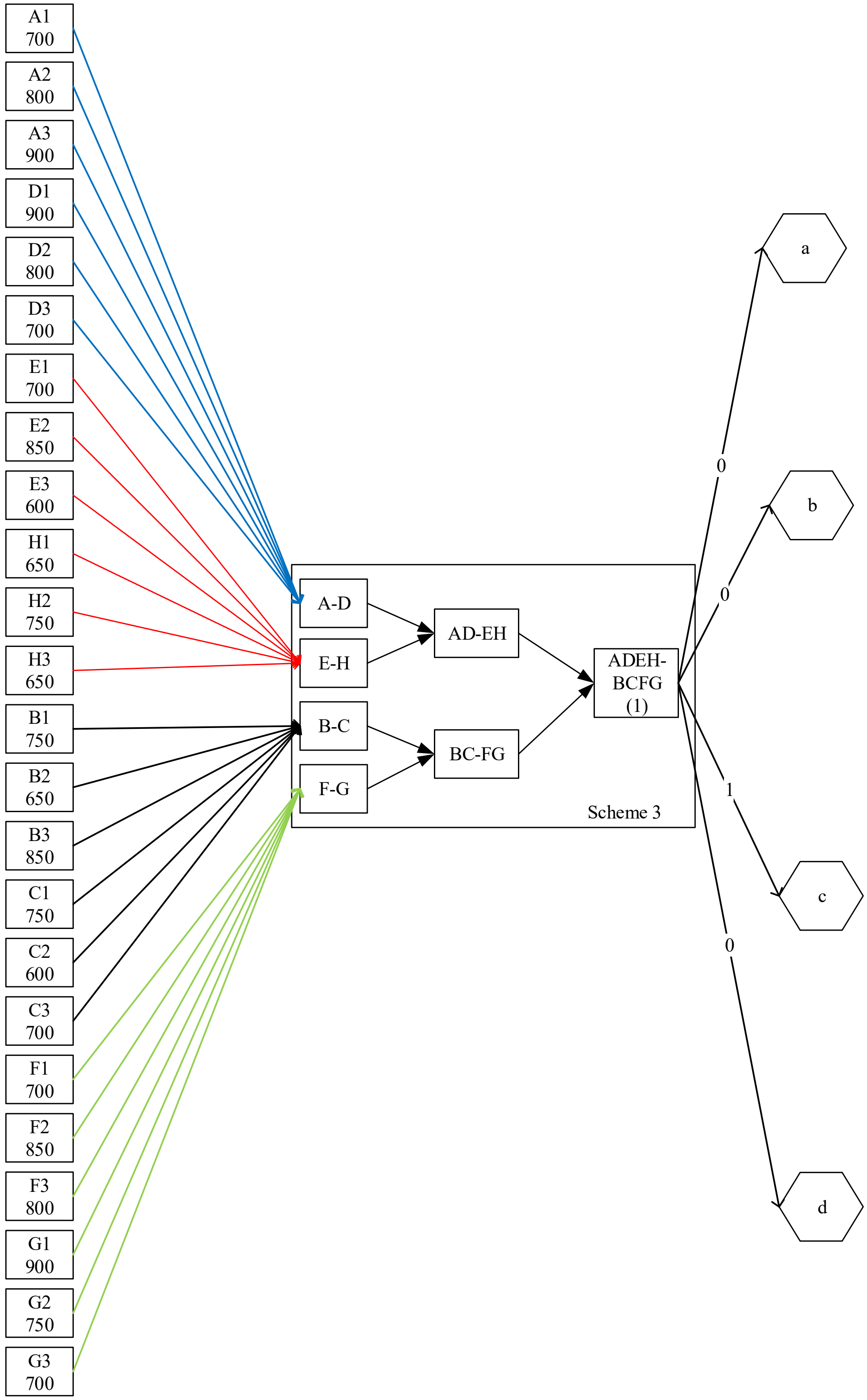


Figure 6.3: Distribution Planning for the Supply Chain (Scheme-3)

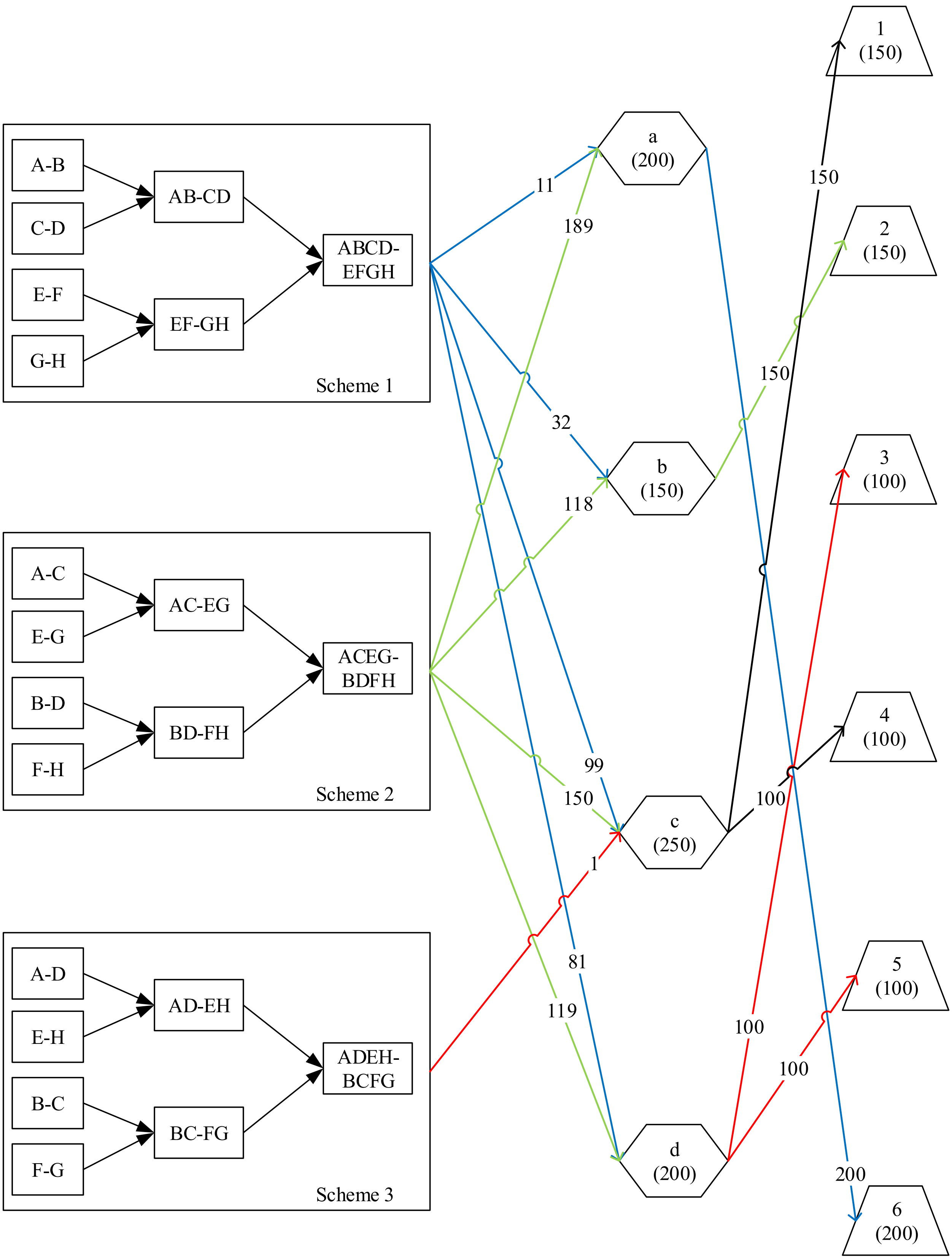


Figure 6.4: Distribution Planning for the Supply Chain (DC to Customer)

Some of the Pareto optimal solutions are shown in the table below.

Table 6.1: Pareto Optimal Solutions Obtained from Proposed Algorithm

Pareto Solution Number	Cost	Time	Quality	Pareto Solution Number	Cost	Time	Quality
1	1088506	115	57587	32	1141761	115	58962
2	1111550	115	57650	33	1137659	115	57576
3	1111655	115	57652	34	1137484	115	57513
4	1116400	115	57670	35	1137516	115	57514
5	1121109	115	57668	36	1140374	115	58466
6	1123468	115	58155	37	1137822	115	57612
7	1124812	115	58411	38	1137858	115	57623
8	1136284	115	57924	39	1137516	115	57504
9	1136284	115	57924	40	1138121	115	57705
10	1136284	115	57924	41	1137741	115	57577
11	1136284	115	57924	42	1137980	115	57652
12	1136284	115	57924	43	1137477	115	57483
13	1136284	115	57924	44	1142019	115	58995
14	1136284	115	57924	45	1137768	115	57574
15	1136284	115	57924	46	1137735	115	57562
16	1135992	115	57821	47	1137520	115	57487
17	1138313	115	58027	48	1137879	115	57605
18	1138313	115	58027	49	1137751	115	57556
19	1138313	115	58027	50	1137872	115	57591
20	1138313	115	58027	51	1137924	115	57607
21	1138313	115	58027	52	1137780	115	57557
22	1138313	115	58027	53	1138034	115	57639
23	1138313	115	58027	54	1137900	115	57594
24	1142002	115	59230	55	1137606	115	57492
25	1141291	115	58901	56	1142464	115	59110
26	1142044	115	59135	57	1137726	115	57530
27	1138643	115	57992	58	1138078	115	57645
28	1141198	115	58841	59	1137771	115	57533
29	1137285	115	57525	60	1138116	115	57648
30	1137298	115	57500	61	1138046	115	57624
31	1137328	115	57495	62	1138472	115	57765

Pareto Solution Number	Cost	Time	Quality	Pareto Solution Number	Cost	Time	Quality
63	1138002	115	57605	82	1138149	115	57616
64	1138104	115	57638	83	1138269	115	57653
65	1137970	115	57590	84	1138188	115	57625
66	1141789	115	58858	85	1138060	115	57582
67	1137838	115	57539	86	1138040	115	57573
68	1138062	115	57606	87	1138043	115	57573
69	1138075	115	57607	88	1143083	115	59250
70	1138283	115	57674	89	1138347	115	57669
71	1138155	115	57631	90	1138213	115	57623
72	1140705	115	58481	91	1138122	115	57592
73	1138344	115	57692	92	1138090	115	57579
74	1138089	115	57606	93	1138255	115	57628
75	1138089	115	57606	94	1138529	115	57719
76	1138089	115	57606	95	1142004	115	58877
77	1138089	115	57606	96	1138167	115	57591
78	1138089	115	57606	97	1138628	115	57743
79	1138223	115	57642	98	1138059	115	57553
80	1137896	115	57533	99	1138387	115	57662
81	1138211	115	57637	100	1138256	115	57618

From the above table, it is evident that, no single solution is better than others with respect to all three objectives. Therefore, these Pareto optimal solutions are non-dominated. Hence this will be useful for the decision makers as they can choose the solution that suits them best from available set of multiple optimum trade off solutions. So, it can be said that the developed model is truly multi-objective in nature and MOGA successfully solve this multi-objective problem.

Correlations among the objective function can be estimated from the Pareto fronts of the corresponding objective functions, which are shown in Figure 6.5, Figure 6.6 and Figure 6.7. The Pareto front between objective 1 and objective 3 shows that if objective 1 increases than objective 3 increases, i.e. the relationship is positive. But Pareto front between objective 1 and objective 2, and objective 3 and objective 2 doesn't show any clear relationship or trend. So, it is clearly evident from the below figures that most of the solutions obtained by this algorithm are Pareto optimal, as we cannot further improve (minimize or maximize) a particular objective without sacrificing others.

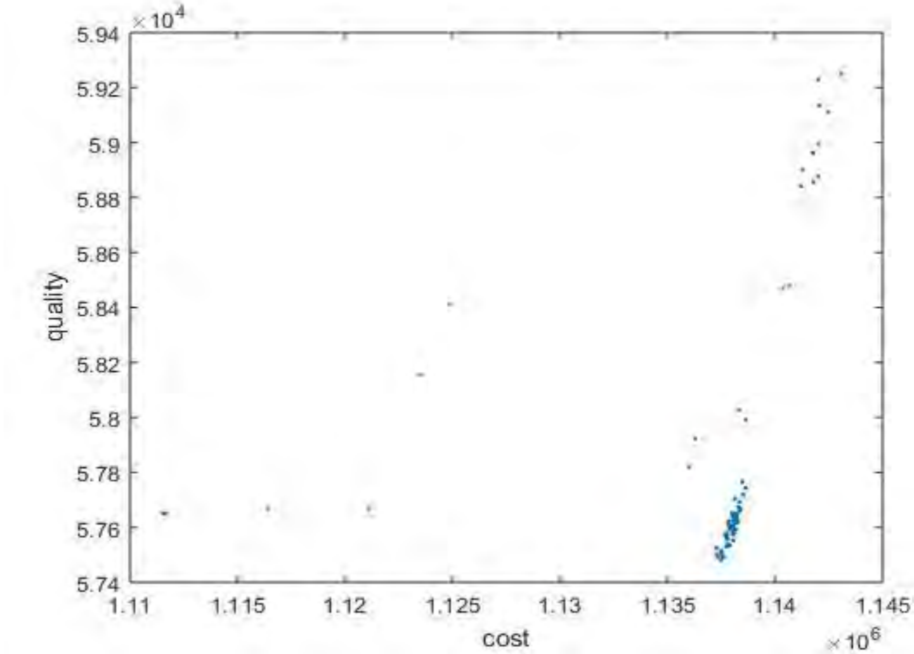


Figure 6.5: Pareto Front between Objective 1 (cost) and Objective 3 (quality)

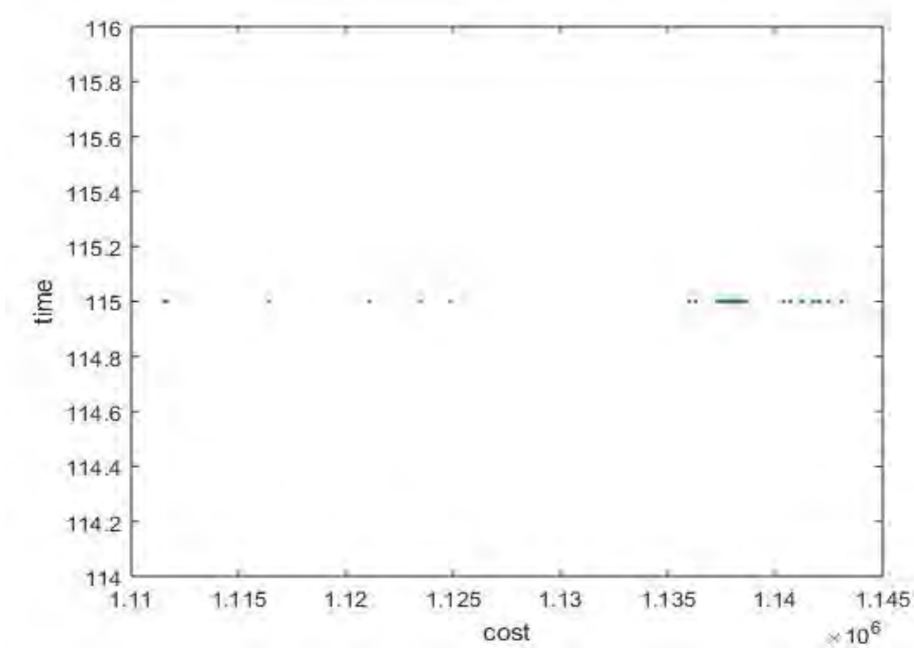


Figure 6.6: Pareto Front between Objective 1 (cost) and Objective 2 (time)

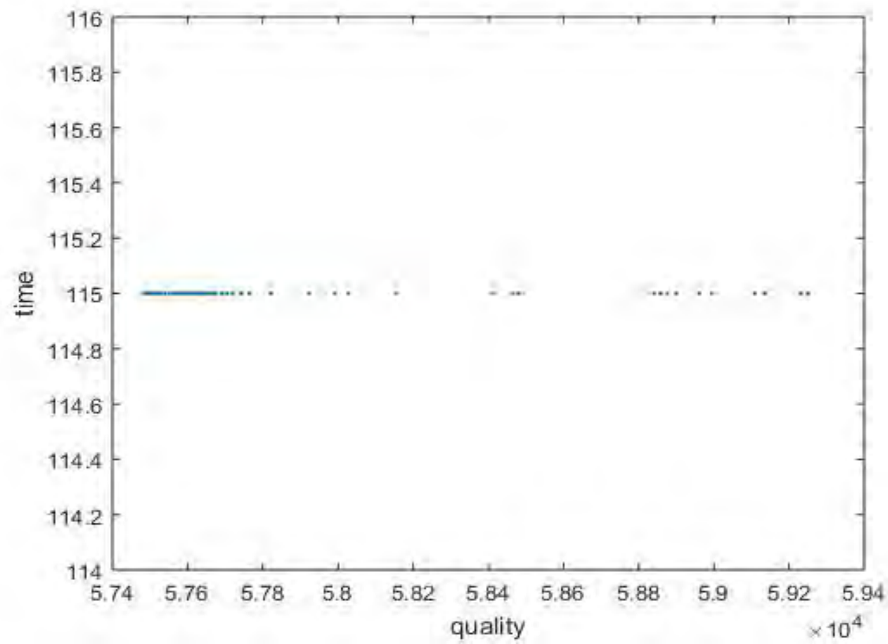


Figure 6.7: Pareto Front between Objective 3 (quality) and Objective 2 (time)

One of the Pareto optimal solutions flow assignments considering all objectives is shown in Figure 6.8. The cost of this solution is 1088506, quality loss cost 79, time is 115, and quality is 57587. From calculation results, assembly plans 1 and 2 are able to choose better suppliers for purchasing parts within the constraint of production capacity, with all customer needs satisfied.

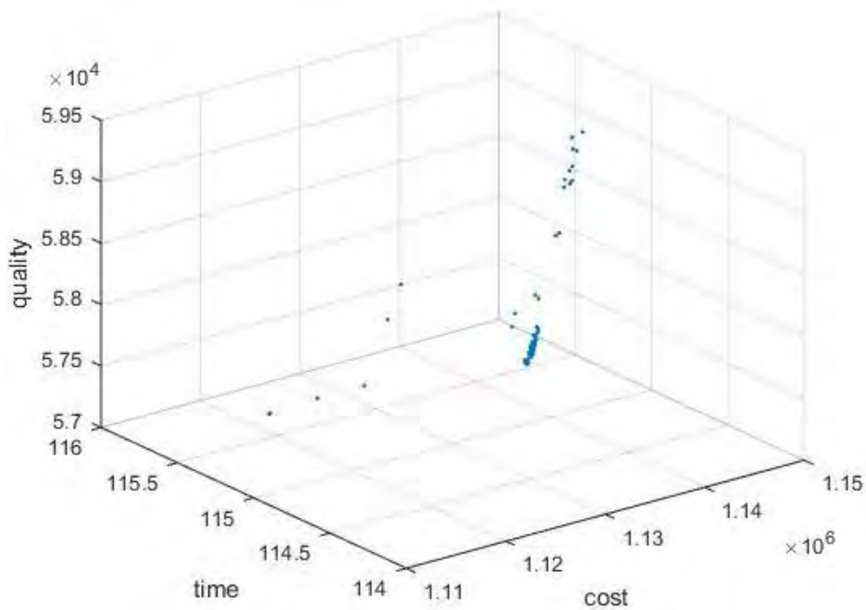


Figure 6.8: Pareto Optimal Solutions Distribution

Solution taken from the Pareto front generated for optimum parameter condition and list of decision variables for each schemes, DCs and customers are shown here.

Table 6.2: Value of Decision Variable for a Set of Solution (Scheme 1)

Scheme 1					
Supplier	$\mathbf{S^{p}_{i,x}}$	$\mathbf{W^{p}_{i,x\_a}}$	**DC**	$\mathbf{y_d}$	$\mathbf{W^{p}_{a,d}}$
A1	1	80	1	1	11
A2	1	75	2	1	32
A3	1	68	3	1	99
B1	1	13	4	1	81
B2	1	92			
B3	1	118			
C1	1	114			
C2	1	108			
C3	1	1			
D1	1	2			
D2	1	97			
D3	1	124			
E1	1	27			
E2	1	45			
E3	1	151			
F1	1	79			
F2	1	44			
F3	1	100			
G1	1	115			
G2	1	1			
G3	1	107			
H1	1	27			
H2	1	96			
H3	1	100			

Table 6.3: Value of Decision Variable for a Set of Solution (Scheme 2)

Scheme 2					
Supplier	**$S^p_{i,x}$**	**W^p_{i,x_a}**	**DC**	**y_d**	**$W^p_{a,d}$**
A1	1	183	1	1	189
A2	1	110	2	1	118
A3	1	283	3	1	150
B1	1	288	4	1	119
B2	1	16			
B3	1	272			
C1	1	440			
C2	1	110			
C3	1	26			
D1	1	305			
D2	1	53			
D3	1	218			
E1	1	227			
E2	1	80			
E3	1	269			
F1	1	266			
F2	1	212			
F3	1	98			
G1	1	47			
G2	1	413			
G3	1	116			
H1	1	2			
H2	1	312			
H3	1	262			

Table 6.4: Value of Decision Variable for a Set of Solution (Scheme 3)

Scheme 3					
Supplier	$\mathbf{S^p_{i,x}}$	$\mathbf{W^p_{i,x\_a}}$	**DC**	$\mathbf{y_d}$	$\mathbf{W^p_{a,d}}$
A1	1	1	1	0	0
A2	0	0	2	0	0
A3	0	0	3	1	1
B1	1	1	4	0	0
B2	0	0			
B3	0	0			
C1	1	1			
C2	0	0			
C3	0	0			
D1	1	1			
D2	0	0			
D3	0	0			
E1	1	1			
E2	0	0			
E3	0	0			
F1	1	1			
F2	0	0			
F3	0	0			
G1	1	1			
G2	0	0			
G3	0	0			
H1	1	1			
H2	0	0			
H3	0	0			

Table 6.5: Value of Decision Variable for Quantity of Finished Products from DC to Customer

DC to Customer $\mathbf{W^p_{d,c}}$							
Customer / DC		Customer					
		1	2	3	4	5	6
DC	a	0	0	0	0	0	200
	b	0	150	0	0	0	0
	c	150	0	0	100	0	0
	d	0	0	100	0	100	0

Chapter-7

CONCLUSION & RECOMMENDATIONS

7.1 CONCLUSION

The objective of this research is to formulate a multi-echelon multi-objective supply chain model selecting the optimum quality supplier while assembling and distributing the product volumes to the rightful customers along with distribution center allocation regarding the goal of fulfilling and grasping some new research opportunities in the field of supply chain network optimization. Considering the vacuity in the field of supply chain network regarding this criteria of selecting supplier and DC allocation, the design was formed.

In this multi-criteria decision solving, a new mPaGA approach based on PaGA is selected to take out the effectiveness regarding the experimental criteria. This method usually represents weaker sorting algorithm as well as creates a large space of solutions. Besides it cannot easily incorporate specific information and weaker at identifying local optima, it can formulate various proper solutions. To find the exact result regarding the parameters of biomimetic population sorting Taguchi helps blindly.

The proposed GA approach yields an efficient model for high quality of parts with cost and time reduction and overall degree of DM satisfaction with determined goal values. Moreover, the proposed approach provides a systematic framework that facilitates the decision-making process, enabling a DM to interactively modify the imprecise data and related model parameters until a satisfactory solution is obtained. Consequently, the proposed approach is expected to be suitable for making real world multi-stage multi-objective supply chain location decisions and supplier selection in an uncertain environment.

The proposed GA approach not only provides more computational efficiency and more flexible solution, but also supports decision-making in an uncertain environment. Different Genetic Algorithm options have been considered to solve a real world typical supply chain problem in this study for solving multiple objective problems.

Moreover, a loss function is incorporated using Taguchi in decision making process for calculating the loss in monetary terms for the case of poor service performance of the suppliers. After applying the algorithm, the minimized ordering cost, purchase cost and transportation costs of the total network is being taken out successfully. The most effective suppliers with optimum service performance is sorted out along with minimum cost and delivery time.

With these, it has opened a new research opportunity for the optimization of sustainable supply chain designs in industries. Even though this new model has some limitations for not calculating the losses from assembling in the factory to reaching the distribution centers and the customers preceding, the applications of it can be found in the area of supply chain management and in designing the optimized supply chain network preliminarily. Besides being a model dedicated to strategic supply chain designs, with certain modifications it can be used for optimizing or restructuring existing supply chains problems. It is supposed to be an engineering and management tool, rather than being a scientific one. Therefore, global corporations with robust supply chains, who are tending to implement effectiveness, efficiency and responsiveness in supply chain network can definitely be benefitted from this research.

7.2 RECOMMENDATIONS

There is a broad scope for future research in the arena of multi-echelon supply chain optimization. The following recommendations are being given for future improvements:

- As it is solved by Genetic Algorithm it can be compared with other searching algorithms. Roulette wheel selection method and single point crossover have been used in this research. Other selection and crossover methods can be used and compared with the proposed algorithm to observe the result.

- In many problems, GAs may have a tendency to converge towards local optima or even arbitrary points rather than the global optimum of the problem. This means that it does not "know how" to sacrifice short-term fitness to gain longer-term fitness. This problem may be alleviated by using a different fitness function, increasing the rate of mutation, or by using selection techniques that maintain a diverse population of solutions. Another possible technique would be to simply replace part of the population with randomly generated individuals, when most of the population is too similar to each other.

- GAs cannot effectively solve problems in which the only fitness measure is a single right/wrong measure (like decision problems), as there is no way to converge on the solution (no hill to climb). In these cases, a random search may find a solution as quickly as a GA. However, if the situation allows the success/failure trial to be repeated giving (possibly) different results, then the ratio of successes to failures provides a suitable fitness measure.

- For specific optimization problems and problem instances, other optimization algorithms may find better solutions than genetic algorithms (given the same amount of computation time). Alternative and complementary algorithms include NSGA, particle swarm optimization, evolution strategies, evolutionary programming, simulated annealing, Gaussian adaptation, hill climbing, and swarm intelligence (e.g.: ant colony optimization) and methods based on integer linear programming. The question of which, if any, problems are suited to genetic algorithms (in the sense that such algorithms are better than others) is open and controversial. Issues related to Pareto optimality, such as a solution maintenance strategy and performance measurement, are also topics worthy of future study.

- Due to lack of available resources especially the relevant information, the case study has not reached the estimated level which was expected. The demand distribution is based on historical data avoiding seasonal effects and other abnormal conditions were included. This type of analysis need to be deseasonalized and trend corrected with the available forecasting models like ‘Hault’s Model’ or ‘Winter’s Model’ which could not be done in this study.

- Moreover, it was assumed that the supplier can instantly supply the distribution center which is not possible. The optimization proposed here needs that demand data be updated daily this can only be possible when the echelons are in real time reliable data communication network.

- The losses of time in monetary terms regarding supplier of each part, the losses of products and time while assembling and transferring the assembled goods to distribution centers and customers should be added further to minimize more cost and time as well. To facilitate the economy of scale price discount in different echelons, shortage cost and lost sales can be considered.

- Moreover, the equilibrium constraints considered for the proposed equations are not likely to be matched in case of real life problems in different workspaces. The equity of the demand and supply while selecting the quantitative parts from the supplier might be uncertain. These forecasted values should be more analyzed and categorized further.

- Uncertainty of parameters such as costs, demands, inventory capacity, transportation capacity and production capacity of the problem can be extended as a fuzzy model and may be solved by using fuzzy multi-objective meta-heuristic algorithms.

REFERENCES

[1] Aiello, G., Scalia, G. L. and Enea, M., "A Multi Objective Genetic Algorithm for The Facility Layout Problem Based Upon Slicing Structure Encoding", Expert Systems with Applications, Vol. 39, No. 12, pp. 10352-10358, 2012.

[2] Altiparmak, F., Gen, M., Lin, L. and Paksoy, T., "A Genetic Algorithm Approach for Multi-Objective Optimization of Supply Chain Networks", Computers and Industrial Engineering, Vol. 51, No. 1, pp. 196–215, 2006.

[3] Amini, M. and Li, H., "Supply Chain Configuration for Diffusion of New Products: An Integrated Optimization Approach", Omega, Vol. 39, pp. 13–22, 2011.

[4] Amiri, A., "Designing A Distribution Network in A Supply Chain System: Formulation and Efficient Solution Procedure", European Journal of Operational Research, Vol. 171, No. 2, pp. 567–576, 2006.

[5] Amodeo, L., Chen, H. and Hadji, A. E., "Multi-Objective Supply Chain Optimization: An Industrial Case Study," Workshop on Applications of Evolutionary Computing, Springer, Berlin, Heidelberg, pp. 732-741, 2007.

[6] Arntzen, B.C., Brown, G.G., Harrison, T.P. and Trafton, L.L., "Global Supply Chain Management at Digital Equipment Corporation", Interfaces, Vol. 21, No. 5, pp. 69-93, 1995.

[7] Aytug, H., Khouja, M. and Vergara, F. E., "Use of Genetic Algorithms to Solve Production and Operations Management: A Review", International Journal of Production Researches, Vol. 41, No.17, pp. 3955–4009, 2003.

[8] Beamon, B. M., "Supply Chain Design and Analysis: Model and Methods, International Journal of Production Economics, Vol. 55, No.3, pp. 281-294, 1998.

[9] Bhatnagar, R., Chandra, P. and Goyal, S. K., "Models for Multi-Plant Coordination", European Journal of Operational Research, Vol. 67, No. 2, pp. 141-160, 1993.

[10] Canel, C., Khumawala, B. M., Law, J. and Loh, A., "An Algorithm for the Capacitated, Multi-Commodity Multi-Period Facility Location Problem," Computers and Operations Research, Vol. 28, pp. 411-427, 2001.

[11] Chakraborti, N., Mishra, P., Aggarwal, A., Banerjee, A. and Mukherjee, S. S., “The Williams and Otto Chemical Plant Re-Evaluated Using A Pareto-Optimal Formulation Aided by Genetic Algorithms”, Applied Soft Computing, Vol. 6, No. 2, pp. 189–97, 2006.

[12] Chan, F. T. S., Chung, S. H. and Choy, K. L., “Optimization of Order Fulfillment in Distribution Network Problems”, Journal of Intelligent Manufacturing, Vol. 17, No. 3, pp. 307–319, 2006.

[13] Chan, F. T. S., Chung, S. H. and Wadhwa, S., “A Heuristic Methodology for Order Distribution in A Demand Driven Collaborative Supply Chain”, International Journal of Production Research, Vol. 42, No.1, pp. 1–19, 2004.

[14] Che, Z. H. and Chiang, C. J., “A Modified Pareto Genetic Algorithm for Multi-Objective Build-to-Order Supply Chain Planning with Product Assembly”, Advances in Engineering Software, Vol. 41, pp. 1011–1022, 2010.

[15] Che, Z., "Using Fuzzy Analytic Hierarchy Process and Particle Swarm Optimisation for Balanced and Defective Supply Chain Problems Considering WEEE/RoHS Directives", International Journal of Production Research, Vol. 48, No. 11, pp. 3355-81, 2010.

[16] Chen C. L. and Lee, W. C., "Multi-Objective Optimization of Multi-Echelon Supply Chain Networks with Uncertain Product Demands and Prices," Computers and Chemical Engineering, Vol. 28, pp. 1131-1144, 2004.

[17] Cho, B. R. and Leonard, M. S., “Identification and Extensions of Quansiconvex Quality Loss Functions”, International Journal of Reliability, Quality and Safety Engineering, Vol. 4, pp. 191-204, 1997.

[18] Chopra, S. and Meindle, P., “Supply chain management”, Singapore: Pearson Education Inc., 2003.

[19] Christopher, M., “Logistics and Supply Chain Management”, 2nd edition, Prentice Hall, Norfolk, 2004.

[20] Coello, C. A. C., “A Comprehensive Survey of Evolutionary-Based Multi-Objective Optimization Techniques”, Knowledge and Information systems, Vol. 1, No. 3, pp. 269–308, 1999.

[21] Deb, K., “Multi-Objective Optimization Using Evolutionary Algorithms”, Vol. 16, John Wiley and Sons, 2001.

[22] Deb, K., Pratap, A. and Agarwal, S., “A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II”, IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, 2002.

[23] Dickson, G. W., “An Analysis of Supplier Selection System and Decision”, International Journal of Engineering and Science Research, Vol. 2, No. 1, pp 5–17, 1966.

[24] Dimopoulos, C. and Zalzala, A. M. S., “Recent Developments in Evolutionary Computation for Manufacturing Optimization: Problems, Solutions and Comparisons”, IEEE Transactions on Evolutionary Computation, Vol. 4, No. 2, pp. 93–113, 2000.

[25] Ding, H., Benyoucef, L. and Xie, X., “A Simulation Optimization Methodology for Supplier Selection Problem”, International Journal of Computer Integrated Manufacturing Vol. 18, No. 2–3, pp. 210–224, 2005.

[26] Dogan, K. and Goetschalckx, M., “A Primal Decomposition Method for the Integrated Design of Multi-Period Production-Distribution Systems,” IIE Transactions, Vol. 31, No. 3, pp. 1027-1036, 1999.

[27] Erol I. and Ferrell, W. G., "A Methodology to Support Decision Making Across The Supply Chain of An Industrial Distributor," International Journal of Production Economics, Vol. 89, pp. 119-129, 2004.

[28] Farahani, R. Z. and Elahipanah, M., “A Genetic Algorithm to Optimize the Total Cost and Service Level for Just-In-Time Distribution in A Supply Chain”, International Journal of Production Economics, Vol. 111, No. 2, pp. 229-243, 2008.

[29] Farahani, R. Z., SteadieSeifi, M. and Asgari, N., “Multiple Criteria Facility Location Problems: A Survey”, Applied Mathematical Modelling, Vol. 34, No. 7, pp. 1689–1709, 2010.

[30] Fei, Y., Li, N. N. and Han, Z. G., “Multi-Objective Optimization Method and Its Application Based on Pareto Sets”, Hoisting and Conveying Machinery, Vol. 9, pp. 13–5, 2006.

[31] Ferrio, J. and Wassick, J., “Chemical Supply Chain Network Optimization”, Computers and Chemical Engineering, Vol. 32, No. 11, pp. 2481-2504, 2008.

[32] Festervand, T. A., Kethley, R. B. and Waller, B. D., “The Marketing of Industrial Real Estate: Application of Taguchi Loss Functions”, Journal of Multi-Criteria Decision Analysis, Vol. 10, No. 4, pp. 219–228, 2001.

[33] Fujita, K., Hirokawa, N. and Akagi, S., “Multi-Objective Optimal Design of Automotive Engine Using Genetic Algorithm”, Proceedings of DETC’98, ASME Design Engineering Technical Conferences, 1998.

[34] Gen, M. and Cheng, R., “Genetic Algorithms and Engineering Optimization”, New York: Wiley, 2000.

[35] Gen, M. and Syarif, A., “Hybrid Genetic Algorithm for Multi-Time Period Production/Distribution Planning”, Computers and Industrial Engineering, Vol. 48, pp. 799–809, 2005.

[36] Gen, M., Cheng, R. and Lin, L., “Network Models and Optimization: Multi-Objective Genetic Algorithm Approach”, London: Springer, 2008.

[37] Geoffrion, A. M. and Powers, R. F., “Twenty Years of Strategic Distribution System Design: An Evolutionary Perspective,” Interfaces, Vol. 25, No. 5, pp. 105-127, 1995.

[38] Georgiadis, M. C., Tsiakis, P., Longinidis, P. and Sofioglou, M. K., “Optimal Design of Supply Chain Networks Under Uncertain Transient Demand Variations”, Omega, Vol. 39, pp. 254–72, 2011.

[39] Gheidar-Kheljani, J., Ghodsypoura, S. H. and O'Brien, C., “Optimizing Whole Supply Chain Benefit Versus Buyer's Benefit Through Supplier Selection”, International Journal of Production Economics, Vol. 121, pp. 489–493, 2009.

[40] Glickman, T. S. and White, S. C., "Optimal Vendor Selection in a Multiproduct Supply Chain with Truckload Discounts", Transportation Research Part E: Logistics and Transportation Review, Vol. 44, No. 5, pp. 684-695, 2008.

[41] Goetschalckx, M., Vidal, C. J. and Dogan, K., “Modelling and Design of Global Logistics Systems: A Review of Integrated Strategic and Tactical Models and Design Algorithms”, European Journal of Operations Research, Vol. 143 No. 1, pp. 1-18, 2002.

[42] Graves, S.C. and Willems, S. P., “Optimizing the Supply Chain Configuration for New Products”, Management Science, Vol. 51, No. 8, pp. 1165–1180, 2005.

[43] Griffis, S. E., Bell, J. E. and Closs, D. J., “Metaheuristics in Logistics and Supply Chain Management”, Journal of Business Logistics, Vol. 33, No. 2, pp. 90–106, 2012.

[44] Gunasekaran, A. and Ngai, E. W. T., "Modeling and Analysis of Build-to-Order Supply Chains", European Journal of Operational Research, Vol. 195, No. 2, pp. 319-334, 2009.

[45] Gunasekaran, A., “The build-to-Order Supply Chain (BOSC): A Competitive Strategy for 21st Century”, Journal of Operations Management, Vol. 23, No.5, pp. 419–22, 2005.

[46] Gunasekaran, A., Lai, K. H. and Cheng, C. T. E, “Responsive Supply Chain: A Competitive Strategy in a Networked Economy, Omega, Vol. 36, No. 4, pp. 549–64, 2008.

[47] Ha, S. H. and Krishnan, R., "A Hybrid Approach to Supplier Selection for the Maintenance of A Competitive Supply Chain", Expert Systems with Applications, Vol. 34, No. 2, pp. 1303-1311, 2008.

[48] Harrison, A. and Hoek, R., Logistics Management and Strategy, 2nd edition, Prentice Hall, Essex, 2005.

[49] Hindi, K. S. and Basta, T., “Computational Efficient Solution of a Multiproduct, Two-stage Distribution-Location Problem,” Journal of Operational Research Society, Vol. 45, No. 11, pp. 1316-1323, 1994.

[50] Ho, W., Xu, X. and Dey, K. P., “Multi-Criteria Decision-Making Approaches for Supplier Evaluation and Selection: A Literature Review”, European Journal of Operational Research, Vol. 202, pp. 16–24, 2010.

[51] Ho, W., Xu, X. and Dey, K. P., “Multi-Criteria Decision-Making Approaches for Supplier Evaluation and Selection: A Literature Review”, European Journal of Operational Research, Vol. 202, pp. 16–24, 2010.

[52] Hosung, C., Rogers, R. L. and Hao, L., “Design of Electrically Small Wire Antennas Using a Pareto Genetic Algorithm, IEEE Antennas and Wireless Propagation, Vol. 53, No. 3, pp. 1038–46, 2005.

[53] Hosung, C., Rogers, R. L. and Hao, L., “Design of Electrically Small Wire Antennas Using A Pareto Genetic Algorithm”, IEEE Transactions on Antennas and Propagation, Vol. 53, No. 3, pp. 1038–46, 2005.

[54] Houlihan, J. B., “International Supply Chain Management”, International Journal of Physical Distribution and Materials Management, Vol. 15, No.1, pp. 22–38, 1985.

[55] Howard, M., Miemczykb, J. and Graves, A., “Automotive Supplier Parks: An Imperative for Build-To-Order”, Journal of Purchasing and Supply Management, Vol. 12, pp. 91–104, 2006.

[56] Hui, Y. V. and Leung, L. C., “Optimal Economic Tool Regrinding with Taguchi's Quality Loss Function”, Engineering Economist, Vol. 39, No. 4, pp. 313–331, 1994.

[57] Jayaraman, V. and Pirkul, H., “Planning and Coordination of Production and Distribution Facilities for Multiple Commodities”, European journal of operational research, Vol. 133, pp. 394-408, 2001.

[58] Jayaraman, V. and Ross, A., “A Simulated Annealing Methodology to Distribution Network Design and Management”, European Journal of Operational Research, Vol. 144, No. 3, pp. 629–645, 2003.

[59] Jones, T. C. and Riley, D. W., “Using Inventory for Competitive Advantage Through Supply Chain Management”, International Journal of Physical Distribution and Materials Management, Vol. 15, No. 5, pp. 16–26, 1985.

[60] Kadadevaramath, R.S., Jason, J. C. H., Shankar, L. B. and Kumar, R. K., “Application of Particle Swarm Intelligence Algorithms in Supply Chain Network Architecture Optimization”, Expert Systems with Applications, Vol. 39, No. 11, pp. 10160-10176, 2012.

[61] Kim, Y. J. and Cho, B. R., “The Use of Response Surface Designs in the Selection of Optimum Tolerance Allocation”, Quality Engineering, Vol. 13, pp. 47-54, 2000.

[62] Konak, A., Coit, D. W. and Smith, A. E., “Multi-Objective Optimization Using Genetic Algorithms: A Tutorial”, Reliability Engineering and System Safety, Vol. 91, No. 9, pp. 992–1007, 2006.

[63] Kopanos, G. M., Puigjaner, L. and Georgiadis, M. C., “Simultaneous Production and Logistic Operations Planning in Semicontinuous Food Industries”, Omega, Vol. 40, pp. 634–50, 2012.

[64] Krajewski, L., Wei, J. C. and Tang, L. L., “Responding to Schedule Changes in Build-To-Order Supply Chains”, Journal of Operations Management, Vol. 23, No.5, pp. 452–69, 2005.

[65] Lavanpriya, C., Manivel, M., and Lakshmanpriya, C., "A Case Study of Integrating the Taguchi Loss Function and Topsis Method to Select an Optimal Supplier in a Manufacturing Industry", The SIJ Transactions on Industrial, Financial & Business Management, Vol. 1, No. 1, pp. 18-22, 2013.

[66] Lee, Y. H., Kim, S. H. and Moon, C., "Production-Distribution Planning in Supply Chain Using A Hybrid Approach," Production Planning and Control, Vol. 13, pp. 35-46, 2002.

[67] Li, M., "Quality Loss Functions for the Measurement of Service Quality", The International Journal of Advanced Manufacturing Technology, Vol. 21, No. 1, pp. 29-37, 2003.

[68] Liao, C. N., “Supplier Selection Project Using an Integrated Delphi, AHP and Taguchi Loss Function”, Probstat Forum, Vol. 3, pp. 118–134, 2010.

[69] Liao, C., and Kao, H., "Supplier Selection Model Using Taguchi Loss Function, Analytical Hierarchy Process and Multi-Choice Goal Programming", Computers and Industrial Engineering, Vol. 58, No. 4, pp. 571-77, 2010.

[70] Liao, S., Hsieh, C. and Lai, P., "An Evolutionary Approach for Multi-Objective Optimization of the Integrated Location–Inventory Distribution Network Problem in Vendor-Managed Inventory", Expert Systems with Applications, Vol. 38, No. 6, pp. 6768-6776, 2011.

[71] Liu, S. and Papageorgiou, L.G., “Multi Objective Optimization of Production, Distribution and Capacity Planning of Global Supply Chains In The Process Industry”, Omega, Vol. 41, No. 2, pp. 369-382, 2013.

[72] Magdalena, R., "Supplier Selection for Food Industry: A Combination of Taguchi Loss Function and Fuzzy Analytical Hierarchy Process", The Asian Journal of Technology Management, Vol. 5, No. 1, pp. 13-22, 2012.

[73] Melachrinoudis, E., Messac, A. and Min, H., “Consolidating A Warehouse Network: A Physical Programming Approach”, International Journal of Production Economics, Vol. 97, pp. 1-17, 2005.

[74] Melo, M. T., Nickel, S. and Da-Gama, F. S., "Dynamic Multi-Commodity Capacitated Facility Location: A Mathematical Modeling Framework for Strategic Supply Chain Planning," Computers and Operations Research, Vol. 33, pp. 181-208, 2006.

[75] Melo, M. T., Nickel, S. and Da-Gama, F. S., "Facility Location and Supply Chain Management – A Review," European Journal of Operational Research, Vol. 196, pp. 401-412, 2009.

[76] Min, H. and Zhou, G., “Supply Chain Modeling: Past, Present and Future”, Computers and Industrial Engineering, Vol. 43, pp. 231-249, 2002.

[77] Munson, L., Hu, J. and Rosenblatt, M., “Teaching the Costs of Uncoordinated Supply Chains, Interfaces, Vol. 33, No. 3, pp. 24-39, 2003.

[78] Nozick, L. K. and Turnquist, M. A., “Integrating Inventory Impacts into a Fixed-charge Model for Locating Distribution Centers,” Transportation Research-E, Vol. 34, No. 3, pp. 173-186, 1998.

[79] Ordoobadi, S., “Application of Taguchi Loss Functions for Supplier Selection”, Supply Chain Management: International Journal, Vol. 14, No. 1, pp. 22–30, 2009.

[80] Owen, S. H. and Daskin, M. S., “Strategic Facility Location: A Review,” European Journal of Operational Research, Vol. 111, No. 3, pp. 423-447, 1998.

[81] Phillips, M. D. and Cho, B. R., “An Empirical Approach to Designing Product Specifications”, Quality Engineering, Vol. 11, pp. 91-100, 1998.

[82] Phillips, M. D. and Cho, B. R., “Modelling of Optimal Specification Regions”, Applied Mathematical Modelling, Vol. 24, pp. 327-341, 2000.

[83] Pinto, E. G., “Supply Chain Optimization Using Multi-Objective Evolutionary Algorithms”, Retrieved December, Vol. 15, 2004.

[84] Poulos, P. N., Rigatos, G. G, Tzafestas, S. G. and Koukos, A. K., “A Pareto-Optimal Genetic Algorithm for Warehouse Multi-Objective Optimization”, Engineering Applications of Artificial Intelligence, Vol. 14, No. 6, pp. 737–49, 2001.

[85] Poulos, P. N., Rigatos, G. G., Tzafestas, S. G. and Koukos, A. K., “A Pareto-Optimal Genetic Algorithm for Warehouse Multi-Objective Optimization”, Engineering Applications of Artificial Intelligence, Vol. 14, No. 6, pp. 737–4, 2001.

[86] Prasad, S., Tata, J. and Madan, M., “Build to Order Supply Chains in Developed and Developing Countries”, Journal of Operations Management, Vol. 23, No.5, pp. 551–68, 2005.

[87] Ripon, K. S. N., Glette, K., Khan, K. N., Hovin, M. and Torresen, J., “Adaptive variable Neighborhood Search for Solving Multi-Objective Facility Layout Problems with Unequal Area Facilities”, Swarm and Evolutionary Computation, Vol. 8, No. 1, pp. 1-12, 2013.

[88] Ripon, K. S. N., Khan, K. N., Glette, K., Hovin, M. and Torresen, J., “Using Pareto-Optimality for Solving Multi-Objective Unequal Area Facility Layout Problem”, Proceedings of 13th Annual Conference on Genetic and Evolutionary Computation, Dublin, Ireland, pp. 681-687, 2011.

[89] Roy, B., “Problems and Methods with Multiple Objective Functions”, Mathematical Programming, Vol. 1, pp. 239-266, 1971.

[90] Sabri, E. H. and Beamon, B. M., “A Multi-Objective Approach to Simultaneous Strategic and Operational Planning in Supply Chain Design”, Omega, Vol. 28, pp. 581-598, 2000.

[91] Sadrzadeh, A., “A Genetic Algorithm with The Heuristic Procedure to Solve the Multi-Line Layout Problem”, Computers and Industrial Engineering, Vol. 62, No. 4, pp. 1055-1064, 2012.

[92] Salhi, S., Rand, G. K., “The Effect of Ignoring Routes When Locating Depots”, European Journal of Operational Research, Vol. 39, pp. 150–156, 1989.

[93] Sawik, T., "Single vs. Multiple Objective Supplier Selection in A Make to Order Environment", Omega, Vol. 38, No. 3, pp. 203-212, 2010.

[94] Sha, D. Y. and Che, Z. H., “Supply Chain Network Design: Partner Selection and Production/Distribution Planning Using a Systematic Model”, Journal of Operational Research Society, Vol. 57 No. 1, pp. 52–62, 2006.

[95] Sharma, S., and Balan, S., "An Integrative Supplier Selection Model Using Taguchi Loss Function, Topsis and Multi Criteria Goal Programming", Journal of Intelligent Manufacturing, Vol. 24, No. 6, pp. 1123-1130, 2013.

[96] Simaria, A. S. and Vilarinho, P. M., “A Genetic Algorithm Based Approach to the Mixed Model Assembly Line Balancing Problem of Type II”, Computers and Industrial Engineering, Vol. 47, No. 4, pp. 391–407, 2004.

[97] Simchi-Levi, D., Kaminsky, P. and Simchi-Levi, E., “Designing and Managing the Supply Chain: Concepts, Strategies, and Case Studies, New York: McGraw-Hill Education, 2002.

[98] Sivanandam, S.N. and Deepa, S. N., “Introduction to Genetic Algorithms”, New York: Springer Berlin Heidelberg, 2008.

[99] Stevens, G. C., “Integrating the Supply Chain”, International Journal of Physical Distribution and Logistics Management, Vol. 19, pp. 3–8, 1989.

[100] Syarif, A., Yun, S.Y. and Gen, M., “Study on Multistage Logistic Chain Network: A Spanning Tree-Based Genetic Algorithm Approach”, Computers and Industrial Engineering, Vol. 43, No. 1, pp. 299-314, 2002.

[101] Taguchi, G., Chowdhury, S. and Wu, Y., “Taguchi’s Quality Engineering Handbook”, John Wiley and Sons, Inc., Hoboken, New Jersey, 2005.

[102] Taguchi, G., Elsayed, E. A. and Hsiang, T. C., “Quality Engineering in Production Systems”, McGraw-Hill College, London, 1989.

[103] Taherkhani, M. and Seifbarghy, M., “Determination of Material Flows in a Multi-Echelon Assembly Supply Chain”, Journal of Optimization in Industrial Engineering, Vol. 5, No. 11, pp. 43-52, 2012.

[104] Taner, T. and Antony, J., “Applying Taguchi Methods to Health Care”, Leadership Health Services, Vol. 19, No. 1, pp. 26–35, 2006.

[105] Teeravaraprug, J. and Cho, B. R., “Designing the Optimal Process Target for Multiple Quality Characteristics”, International Journal of Production Research, Vol. 44, 37- 54, 2002.

[106] Truong, T. H. and Azadivar, F. “Optimal Design Methodologies for Configuration of Supply Chains. International Journal of Production Research”, Vol. 43, No. 11, pp. 2217–2236, 2005.

[107] Tsoukalas, L.H. and Uhrig, R. E., “Fuzzy and Neural Approach in Engineering”, New York: John Willey, 1997.

[108] Verderame, P. M. and Floudas, C. A., “Operational Planning Framework for Multisite Production and Distribution Networks”, Computers and Chemical Engineering, Vol. 33, pp. 1036–50, 2009.

[109] Voudouris, V. T. and Consulting, A., "Mathematical Programming Techniques to Debottleneck the Supply Chain of Fine Chemical Industries," Computers and Chemical Engineering, Vol. 20, pp. S1269-S1274, 1996.

[110] Wadhwa, V. and Ravindran, ARR., “Vendor Selection in Outsourcing”, Computational Operations Research, Vol. 34, No. 12, pp. 3725–3737, 2007.

[111] Weber, C. A. and Current, J. R., “A Multi-Objective Approach to Vendor Selection”, European Journal of Operational Research, Vol. 68, pp. 173–84, 1993.

[112] Weber, C. A., Current, J. R., and Benton, W. C., “Vendor Selection Criteria and Methods”, European Journal of Operational Research, Vol. 50, pp. 2–18, 1991.

[113] Wolpert, D. H. and Macready, W. G., “No Free Lunch Theorems for Optimization”, IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, pp. 67–82, 1997.

[114] Xu, J. Liu, Q. and Wang, R., “A Class of Multi-Objective Supply Chain Networks Optimal Model Under Random Fuzzy Environment and its Application to the Industry of Chinese Liquor”, Information Sciences, Vol. 178, pp. 2022–43, 2008.

[115] Yan, H., Yu, Z. and Cheng, T. C. E., “A Strategic Model for Supply Chain Design with Logical Constraints: Formulation and Solution”, Computer and Operation Research, Vol. 30, pp. 2135–2155, 2003.

[116] Yeh, W. C. and Chuang, M. C., “Using Multi-Objective Genetic Algorithm for Partner Selection in Green Supply Chain Problems”, Expert Systems with Applications, Vol. 38, pp. 4244–4253, 2011.

[117] Yeh, W. C., “An Efficient Memetic Algorithm for The Multi-Stage Supply Chain Network Problem”, International Journal of Advanced Manufacturing Technology, Vol. 29, pp. 803–813, 2006.

[118] Zeng, D. D., “Multi-Issue Decision Making in Supply Chain Management and Electronic Commerce. Ph.D. Dissertation”, Graduate School of Industrial Administration and Robotics Institute, Carnegie Mellon University, Pittsburgh, USA, December 1998.

[119] Zhou, G., Min, H. and Gen, M., “The Balanced Allocation of Customers to Multiple Distribution Centers in The Supply Chain Network: A Genetic Algorithm Approach”, Computers and Industrial Engineering, Vol. 43, pp. 251–261, 2002.

[120] Zhou, S. Q. and Xiang, J. W., “Multi-Objective Collaborative Optimization Based on BP Neural Network and Pareto Genetic Algorithm, Machine Design Research, Vol. 22, No. 5, pp.10–3, 2006.

[121] Zhu, X. J., Pan, D., Wang, A. L., Zhang, H. Q. and Ye, Q. T., “Multi-Objective Optimization Design with Mixed-Discrete Variables in Mechanical Engineering Via Pareto Genetic Algorithm”, Journal of Shanghai Jiaotong University, Vol. 34, No. 3, pp. 411–4, 2000.

[122] Zitzler, E., “Evolutionary Algorithms for Multi-Objective Optimization: Methods and Applications”, Switzerland: Swiss Federal Institute of Technology, 1999.

[123] Zitzler, E., Laumanns, M. and Bleuler, S., “A Tutorial on Evolutionary Multi-Objective Optimization”, Metaheuristics for Multi-Objective Optimization, Lecture Notes in Economics and Mathematical Systems, Vol. 535, pp. 3–37, Berlin: Springer, 2004.

APPENDICES

APPENDIX-A

Java Code for the Optimization Program

(Software Used: Java SE 8u141)

Java Command: From Supplier to DC

```
import java.io.*;
import java.util.*;
class Supplier{
	public int supplierId;
	public int supply_item;
	public int supplier_no;
	public int capacity;
	public int order_cost;
	public int procure_cost;
	public int q_grade;
	public int[] cost;
	public int[] time;
	public Supplier(){
		cost = new int[3];
		time = new int[3];
	}
	public Supplier(int id,int item, int no, int cap, int o_cost, int p_cost, int grade){
		supplierId = id;
		supply_item = item;
		supplier_no = no;
		capacity = cap;
		order_cost = o_cost;
		procure_cost = p_cost;
		q_grade = grade;
		cost = new int[3];
		time = new int[3];
	}
	public Supplier(Supplier s){
		supplierId = s.supplierId;
		supply_item = s.supply_item;
		supplier_no = s.supplier_no;
		capacity = s.capacity;
		order_cost = s.order_cost;
		procure_cost = s.procure_cost;
		q_grade = s.q_grade;
		cost = new int[3];
```

```
		time = new int[3];
		if((s.cost).length != (s.time).length){
			System.out.println("Cost and Time array length mismatch.");
		}
		for(int i=0;i<(s.cost).length;i++){
			cost[i] = s.cost[i];
			time[i] = s.time[i];
		}
	}
	public void setCost(int c0, int c1, int c2){
		cost[0] = c0;
		cost[1] = c1;
		cost[2] = c2;
	}
	public void setTime(int t0, int t1, int t2){
		time[0] = t0;
		time[1] = t1;
		time[2] = t2;
	}
	public String toString(){
		temp += "Supplier Id: "+supplierId+"\n";
		temp += "Supplier info: \n";
		temp += "Supply Item: "+supply_item+"\n";
		temp += "Supplier Id: "+supplier_no+"\n";
		temp += "Supplier Capacity: "+capacity+"\n";
		temp += "Order cost: "+order_cost+"\n";
		temp += "Procurement Cost: "+procure_cost+"\n";
		temp += "Q grade: "+q_grade+"\n";
		temp += "Scheme 1 details: "+cost[0]+"/"+time[0]+"\n";
		temp += "Scheme 2 details: "+cost[1]+"/"+time[1]+"\n";
		temp += "Scheme 3 details: "+cost[2]+"/"+time[2]+"\n";
		return temp;
	}
}
class schemeElement{
	public int stage;
	public int start;
	public int end;
	public int cost;
	public int time;
	public schemeElement(int stg,int s, int e, int c, int t){
		stage = stg;
		start = s;
		end = e;
		cost = c;
		time = t;
```

```
	}
	public schemeElement(schemeElement s){
		// Copying data from another stage .
		stage = s.stage;
		start = s.start;
		end = s.end;
		cost = s.cost;
		time = s.time;
	}
	public String toString(){
		String temp = "";
		temp += "Stage: "+stage;
		temp += " Start : "+start;
		temp += " End: "+end;
		temp += " Cost: "+cost;
		temp += " Time: "+time+"\n";
		return temp;
	}
}
class Scheme{
	public int schId ;
	public ArrayList<schemeElement> firstStage;
	public ArrayList<schemeElement> secondStage;
	public ArrayList<schemeElement> thirdStage;
	public Scheme(int id){
		schId = id;
		firstStage = new ArrayList<schemeElement>();
		secondStage = new ArrayList<schemeElement>();
		thirdStage = new ArrayList<schemeElement>();
	}
	public Scheme(Scheme s){
		schId = s.schId;
		firstStage = new ArrayList<schemeElement>(s.firstStage);
		secondStage = new ArrayList<schemeElement>(s.secondStage);
		thirdStage = new ArrayList<schemeElement>(s.thirdStage);
	}
}
class ChromosomeScheme{
	public int schemeId;
	public int numberOfSuppliers;
	public int numberOfCustomers;
	public int[] supplierGeneValues;
	public int[] customerGeneValues;
	public ChromosomeScheme(int id,int num_sup, int num_cus){
		schemeId = id;
		numberOfSuppliers = num_sup;
```

```
		numberOfCustomers = num_cus;
		supplierGeneValues = new int[num_sup];
		customerGeneValues = new int[num_cus];
	}
	public ChromosomeScheme(ChromosomeScheme c){
		schemeId = c.schemeId;
		numberOfSuppliers = c.numberOfSuppliers;
		numberOfCustomers = c.numberOfCustomers;
		supplierGeneValues = new int[c.numberOfSuppliers];
		customerGeneValues = new int[c.numberOfCustomers];
		System.arraycopy(c.supplierGeneValues, 0, supplierGeneValues, 0,
(c.supplierGeneValues).length);
		System.arraycopy(c.customerGeneValues, 0, customerGeneValues, 0,
(c.customerGeneValues).length);
	}
}
class Chromosome{
	public ArrayList<ChromosomeScheme> chromosomeParts;
	public int cost;
	public int time;
	public int quality;
	public Chromosome(int numberOfParts, int sup_num, int cus_num){
		chromosomeParts = new ArrayList<ChromosomeScheme>();
		for(int i=0;i<numberOfParts;i++){
			chromosomeParts.add(new ChromosomeScheme(i,sup_num,cus_num));
		}
	}

}
class ChromosomeComparator implements Comparator<Chromosome>{
	public int compare(Chromosome a, Chromosome b){
		double val_a = calculateFitnessValue(a);
		double val_b = calculateFitnessValue(b);
		if(val_a > val_b){
			return -1;
		}
		else if(val_b > val_a){
			return 1;
		}
		return 0;
	}
	public double calculateFitnessValue(Chromosome c){
		double ans = 20000000.0;
		ans += ( -10 * (double)c.cost );
		ans += (-8595 * (double)c.time);
		ans += (30 * (double)c.quality);
```

```
		return ans;
	}
}
class GenAlgoObject{
	public ArrayList<Chromosome> population;
	public static ArrayList<Supplier> supplierList;
	public ArrayList<Scheme> schemeList;
	public ArrayList<Integer> demands_list;
	public int demand;
	public double costCoEfficient;
	public double timeCoEfficient;
	public double qualityCoEfficient;
	public GenAlgoObject(int d, ArrayList<Supplier> s_list , ArrayList<Scheme> sch_list ,
ArrayList<Integer> d_list){
		population = new ArrayList<Chromosome>();
		supplierList = s_list;
		schemeList = sch_list;
		demands_list = d_list;
		demand = d;
		costCoEfficient = -0.4;
		timeCoEfficient = -0.3;
		qualityCoEfficient = 0.3;
	}
	public void initialPopulationGeneration(int popsize){
		int entry = 0;
		for(int i =0 ;; i++){
			if(entry == popsize){
				break;
			}
			Triplet demands = Utilities.threeWaySplitter(demand);
			ArrayList<Integer> d_l = new ArrayList<Integer>(demands_list);
			Chromosome c = new Chromosome(3,24,4);
			ChromosomeScheme cScheme0 = c.chromosomeParts.get(0);
			processChromosomeEntry(cScheme0,demands.u, d_l);
			ChromosomeScheme cScheme1 = c.chromosomeParts.get(1);
			processChromosomeEntry(cScheme1,demands.v, d_l);
			ChromosomeScheme cScheme2 = c.chromosomeParts.get(2);
			processChromosomeEntry(cScheme2,demands.w, d_l);
			c.cost = calculateCost(c);
			c.time = calculateTime(c);
			c.quality = calculateQualityGrade(c);
			population.add(c);
			entry++;
		}
		Collections.sort(population, new ChromosomeComparator());
	}
```

```
	public void processChromosomeEntry(ChromosomeScheme cScheme0, int
schemeOutput, ArrayList<Integer> d_l){
		for(int j=0; j < 24 ; j+= 3){
			Triplet allocations = Utilities.threeWaySplitter(schemeOutput);
			int remainingPrev = 0;
			Supplier s0 = supplierList.get(j);
			if(s0.capacity >= allocations.u){
				cScheme0.supplierGeneValues[j] = allocations.u;
			}
			else{
				remainingPrev = allocations.u - s0.capacity;
				cScheme0.supplierGeneValues[j] = s0.capacity;
			}
			Supplier s1 = supplierList.get(j+1);
			if(s1.capacity >= allocations.v+remainingPrev){
				cScheme0.supplierGeneValues[j+1] =
allocations.v+remainingPrev;
				remainingPrev = 0;
			}
			else if(s1.capacity >= allocations.v  && (s1.capacity <
allocations.v+remainingPrev)){
				cScheme0.supplierGeneValues[j+1] = s1.capacity;
				remainingPrev = remainingPrev - (s1.capacity - allocations.v);
			}
			else{
				cScheme0.supplierGeneValues[j+1] = s1.capacity;
				remainingPrev = remainingPrev + (allocations.v - s1.capacity) ;
			}
			Supplier s2 = supplierList.get(j+2);
			if(s2.capacity >= allocations.w+remainingPrev){
				cScheme0.supplierGeneValues[j+2] =
allocations.w+remainingPrev;
				remainingPrev = 0;
			}
			else if(s2.capacity >= allocations.w  && (s2.capacity <
allocations.w+remainingPrev)){
				cScheme0.supplierGeneValues[j+2] = s2.capacity;
				remainingPrev = remainingPrev - (s2.capacity - allocations.w);
			}
			else{
				cScheme0.supplierGeneValues[j+2] = s2.capacity;
				remainingPrev = remainingPrev + (allocations.w - s2.capacity) ;
			}
		}
		int[] d = Utilities.fourWaySplitter(schemeOutput);
		for(int j=0;j<4;j++){
```

```
				cScheme0.customerGeneValues[j] = 0;
			}
			for(int j=0;j<4;j++){
				int val = (int)d_l.get(j);
				int remainingDemands = 0;
				if(val >= d[j]){
					val = val - d[j];
					cScheme0.customerGeneValues[j] += d[j];
					d_l.set(j,new Integer(val));
				}
				else{
					cScheme0.customerGeneValues[j] += val;
					remainingDemands = d[j] - val;
					val = 0;
					d_l.set(j,new Integer(val));
					for(int k=0;k<4;k++){
						if(remainingDemands == 0){
							break;
						}
						if(k != j){
							int nval = (int)d_l.get(k);
							if(nval >= remainingDemands){
								nval = nval - remainingDemands;
								cScheme0.customerGeneValues[k] +=
remainingDemands;

								remainingDemands = 0;
								d_l.set(k,new Integer(nval));
							}
							else{
								remainingDemands = remainingDemands -
nval;

								cScheme0.customerGeneValues[k] += nval;

								d_l.set(k,new Integer(0));
							}
						}
					}
				}
			}
		}
		public int calculateCost(Chromosome c){
			int cost = 0;
			for(int i=0;i<c.chromosomeParts.size(); i++){
				ChromosomeScheme cs = c.chromosomeParts.get(i);
				for(int j=0; j<24; j++){
					Supplier s = supplierList.get(j);
```

```
                    if(cs.supplierGeneValues[j] > 0){
                        cost += (double)s.order_cost;
                    }
                    cost += (cs.supplierGeneValues[j] * s.cost[i] );
                    cost += (cs.supplierGeneValues[j] * s.procure_cost);
                }
                int totalProduction = 0;
                for(int k = 0; k<4 ; k++){
                    totalProduction += cs.customerGeneValues[k];
                }
                Scheme sch =  schemeList.get(i);
                for(int k=0; k<sch.firstStage.size() ; k++){
                    cost += (totalProduction * sch.firstStage.get(k).cost);
                }
                for(int k=0; k<sch.secondStage.size() ; k++){
                    cost += (totalProduction * sch.secondStage.get(k).cost);
                }
                for(int k=0; k<sch.thirdStage.size() ; k++){
                    cost += (totalProduction * sch.thirdStage.get(k).cost);
                }
            }
            return cost;
        }
        public int calculateTime(Chromosome c){
            int totalTime = 0;
            for(int i=0;i<3;i++){
                int schemeTime = 0;
                ChromosomeScheme cs = c.chromosomeParts.get(i);
                int supplierToFirstStage = 0;
                for(int j=0;j<24;j++){
                    if(cs.supplierGeneValues[j] > 0){
                        supplierToFirstStage = max(supplierToFirstStage,
supplierList.get(j).time[i]);
                    }
                }
                schemeTime += supplierToFirstStage;
                Scheme sch = schemeList.get(i);
                int firstToSecondStage = 0;
                for(int k=0;k<sch.firstStage.size();k++){
                    firstToSecondStage = max(firstToSecondStage,
sch.firstStage.get(k).time);
                }
                schemeTime += firstToSecondStage;
                int secondToThirdStage = 0;
                for(int k=0;k<sch.secondStage.size();k++){
```

```
                    secondToThirdStage = max(secondToThirdStage,
sch.secondStage.get(k).time);
                }
                schemeTime += secondToThirdStage;
                int thirdStageToCustomer = 0;
                for(int k=0;k<sch.thirdStage.size();k++){
                    thirdStageToCustomer = max(thirdStageToCustomer,
sch.thirdStage.get(k).time);
                }
                schemeTime += thirdStageToCustomer;
                totalTime = max(totalTime, schemeTime);
            }
            return totalTime;
        }
        public int calculateQualityGrade(Chromosome c){
            int qg = 0;
            for(int i=0;i<c.chromosomeParts.size();i++){
                ChromosomeScheme cs = c.chromosomeParts.get(i);
                for(int j=0; j<24; j++){
                    qg += (cs.supplierGeneValues[j] * supplierList.get(j).q_grade);
                }
            }
            return qg;
        }
        public static int lossCalculator(Chromosome c){
            int lossValue;
            int weight = 0;
            int weightedSum = 0;
            for(int i=0; i<c.chromosomeParts.size(); i++){
                ChromosomeScheme cs = c.chromosomeParts.get(i);
                for(int k=0;k<24;k++){
                    weight+= cs.supplierGeneValues[k];
                    weightedSum += cs.supplierGeneValues[k] *
supplierList.get(k).q_grade;
                }
            }
            double temp = (double)weightedSum/(double)(10*weight);
            temp = 64.0/ (temp*temp);
            lossValue = (int) temp;
            return lossValue;
        }
        public void singleSchemeMutationProcess(Chromosome c){
            Chromosome mutatedChromosome = new Chromosome(3,24,4);
            for(int i=0;i<3;i++){
                mutatedChromosome.chromosomeParts.set(i,new
ChromosomeScheme(c.chromosomeParts.get(i)));
```

```
        }
        Random r = new Random();
        int index = (int)Math.abs(r.nextInt(3));
        ChromosomeScheme toBeMutated =
mutatedChromosome.chromosomeParts.get(index);
        for(int i=0;i<24;i+=3){
            int temp_sum = 0;
            temp_sum += toBeMutated.supplierGeneValues[i];
            temp_sum += toBeMutated.supplierGeneValues[i+1];
            temp_sum += toBeMutated.supplierGeneValues[i+2];
            if(temp_sum == 0){
                continue;
            }
            Triplet t_x = Utilities.threeWaySplitter(temp_sum);
            int h = highestPriceIndicator(index,i,i+1,i+2);
            int l = lowestPriceIndicator(index,i,i+1, i+2);
            int[] val = {t_x.u, t_x.v, t_x.w};
            Arrays.sort(val);
            if(h == i){
                toBeMutated.supplierGeneValues[i] = val[0];
            }
            else if(l == i){
                toBeMutated.supplierGeneValues[i] = val[2];
            }
            else{
                toBeMutated.supplierGeneValues[i] = val[1];
            }
            if(h == i+1){
                toBeMutated.supplierGeneValues[i+1] = val[0];
            }
            else if(l == i+1){
                toBeMutated.supplierGeneValues[i+1] = val[2];
            }
            else{
                toBeMutated.supplierGeneValues[i+1] = val[1];
            }
            if(h == i+2){
                toBeMutated.supplierGeneValues[i+2] = val[0];
            }
            else if(l == i+2){
                toBeMutated.supplierGeneValues[i+2] = val[2];
            }
            else{
                toBeMutated.supplierGeneValues[i+2] = val[1];
            }
        }
```

```
			mutatedChromosome.cost = calculateCost(mutatedChromosome);
			mutatedChromosome.time = calculateTime(mutatedChromosome);
			mutatedChromosome.quality = calculateQualityGrade(mutatedChromosome);
			population.add(mutatedChromosome);
		}
		public void crossOverEquilibrium(Chromosome a, Chromosome b){
			Random r = new Random();
			int index = Math.abs(r.nextInt(3));
			ChromosomeScheme c_a = new
ChromosomeScheme(a.chromosomeParts.get(index));
			ChromosomeScheme c_b = new
ChromosomeScheme(b.chromosomeParts.get(index));
			int a_val = 0;
			int b_val = 0;
			for(int i=0;i<4;i++){
				a_val += c_a.customerGeneValues[i];
				b_val += c_b.customerGeneValues[i];
			}
			double a_ratio = (double)a_val/(double)b_val;
			double b_ratio = (double)b_val/(double)a_val;
			Chromosome crossA = new Chromosome(3,24,4);
			Chromosome crossB = new Chromosome(3,24,4);
			for(int i=0;i<3;i++){
				crossA.chromosomeParts.set(i,b.chromosomeParts.get(i));
				crossB.chromosomeParts.set(i,a.chromosomeParts.get(i));
			}
			crossA.chromosomeParts.set(index,c_b);
			crossB.chromosomeParts.set(index,c_a);
			equilibriumProcess(crossA, index, a_ratio);
			equilibriumProcess(crossB, index, b_ratio);
			crossA.cost = calculateCost(crossA);
			crossB.cost = calculateCost(crossB);
			crossA.time = calculateTime(crossA);
			crossB.time = calculateCost(crossB);
			crossA.quality = calculateQualityGrade(crossA);
			crossB.quality = calculateQualityGrade(crossB);
			population.add(crossA);
			population.add(crossB);
		}
		public void equilibriumProcess(Chromosome c, int x, double ratio){
			ChromosomeScheme cs = c.chromosomeParts.get(x);
			for(int i=0;i<24; i++){
				cs.supplierGeneValues[i] =
(int)Math.round((double)cs.supplierGeneValues[i] * ratio);
			}
			int demand_now = 0;
```

```
		for(int i=0;i<4; i++){
			cs.customerGeneValues[i] = 
(int)Math.round((double)cs.customerGeneValues[i] * ratio);
			demand_now += cs.customerGeneValues[i];
		}
		for(int i=0; i<24; i+=3){
			int supply_now = 
cs.supplierGeneValues[i]+cs.supplierGeneValues[i+1]+cs.supplierGeneValues[i+2];
			int diff = 0;
			if(supply_now<demand_now){
				diff = demand_now  - supply_now;
				int p = lowestPriceIndicator(x, i, i+1, i+2);
				cs.supplierGeneValues[p] = cs.supplierGeneValues[p] + diff;
			}
			else if(supply_now > demand_now){
				diff = supply_now - demand_now;
				if(cs.supplierGeneValues[i] - diff >=0){
					cs.supplierGeneValues[i] = cs.supplierGeneValues[i] - diff;
				}
				else if(cs.supplierGeneValues[i+1] - diff >=0){
					cs.supplierGeneValues[i+1] = cs.supplierGeneValues[i+1] 
- diff;
				}
				else if(cs.supplierGeneValues[i+2] - diff >=0){
					cs.supplierGeneValues[i+2] = cs.supplierGeneValues[i+2] 
- diff;
				}
			}
		}
		int[] deficits = new int[4];
		for(int i=0; i<4; i++){
			int temp_sum = 0;
			temp_sum += c.chromosomeParts.get(0).customerGeneValues[i];
			temp_sum += c.chromosomeParts.get(1).customerGeneValues[i];
			temp_sum += c.chromosomeParts.get(2).customerGeneValues[i];
			deficits[i] = temp_sum - demands_list.get(i);
		}
		for(int i=0;i<4; i++){
			int target = deficits[i];
			int k = 0;
			Random r = new Random();
			int iteration = 0;
			while(true){
				do{
					k = Math.abs(r.nextInt(3));
				}while(k != x);
```

```
ChromosomeScheme cx = c.chromosomeParts.get(k);
if(target > 0){
	if(cx.customerGeneValues[i] >= target){
		cx.customerGeneValues[i] =
cx.customerGeneValues[i] - target;
		for(int j=0; j<24; j+=3){
			int t = target;
			if(cx.supplierGeneValues[j]>=t){
				cx.supplierGeneValues[j] =
cx.supplierGeneValues[j]-t;
				continue;
			}
			else{
				t = t - cx.supplierGeneValues[j];
				cx.supplierGeneValues[j] = 0;
			}
			if(cx.supplierGeneValues[j+1]>=t){
				cx.supplierGeneValues[j+1] =
cx.supplierGeneValues[j+1]-t;
				continue;
			}
			else{
				t = t - cx.supplierGeneValues[j+1];
				cx.supplierGeneValues[j+1] = 0;
			}
			if(cx.supplierGeneValues[j+2]>=t){
				cx.supplierGeneValues[j+2] =
cx.supplierGeneValues[j+2]-t;
				continue;
			}
			else{
				t = t - cx.supplierGeneValues[j+2];
				cx.supplierGeneValues[j+2] = 0;
			}
		}
		break;
	}
	else{
		continue;
	}
}
else if(target < 0){
	int t = -target;
	cx.customerGeneValues[i] += t;
	for(int j=0; j<24; j+=3){
		int l = lowestPriceIndicator(k,j,j+1,j+2);
```

```
                    cx.supplierGeneValues[l] += t;
                }
            }
            iteration++;
            if(iteration == 50){
                break;
            }
        }
    }
}
public int lowestPriceIndicator(int s,int x,int y, int z){
    if(supplierList.get(x).cost[s] < supplierList.get(y).cost[s]){
        if(supplierList.get(x).cost[s] < supplierList.get(z).cost[s]){
            return x;
        }
        else{
            return z;
        }
    }
    else{
        if(supplierList.get(y).cost[s] < supplierList.get(z).cost[s]){
            return y;
        }
        else{
            return z;
        }
    }
}
public int highestPriceIndicator(int s, int x,int y, int z){
    if(supplierList.get(x).cost[s] > supplierList.get(y).cost[s]){
        if(supplierList.get(x).cost[s] > supplierList.get(z).cost[s]){
            return x;
        }
        else{
            return z;
        }
    }
    else{
        if(supplierList.get(y).cost[s] > supplierList.get(z).cost[s]){
            return y;
        }
        else{
            return z;
        }
    }
}
```

```
	public void evolutionProcess(int generationNUmber,int inPopSize,int cross_rate, int
mutate_rate){
		int presentSize = inPopSize -1;
		double c_r = (double)cross_rate/100.0;
		double m_r = (double)mutate_rate/100.0;
		int cross_number = (int)Math.round((double)presentSize*c_r);
		int mutate_number = (int)Math.round((double)presentSize*m_r);
		mutate_number = 50*mutate_number;
		for(int k=0; k<generationNUmber; k++){
			Random r = new Random();
			while(true){
				if(cross_number == 0 && mutate_number == 0){
					break;
				}
				if(cross_number > 0 ){
					int index1 = (int)Math.abs(r.nextInt(presentSize));
					int index2 = (int)Math.abs(r.nextInt(presentSize));

	crossOverEquilibrium(population.get(index1),population.get(index2));
					cross_number--;
				}
				if(mutate_number > 0 ){
					int index = (int)Math.abs(r.nextInt(presentSize));
					singleSchemeMutationProcess(population.get(index));
					mutate_number--;
				}
			}
			Collections.sort(population,new ChromosomeComparator());
			for(int j = population.size()-1; j>inPopSize-1;j--){
				population.remove(j);
			}
		}
	}
	public int max(int a,int b){
		return (a>b)?a:b;
	}
	public int min(int a,int b){
		return (a<b)?a:b;
	}
}
class Triplet{
	public int u;
	public int v;
	public int w;
	public Triplet(int a,int b, int c){
		u = a;
```

```
            v = b;
            w = c;
        }
        public Triplet(Triplet t){
            u = t.u;
            v = t.v;
            w = t.w;
        }
}
class Utilities{
        public static void printSupplierList(ArrayList<Supplier> s_list){
            for(int i=0;i<s_list.size(); i++){
                System.out.println(s_list.get(i));
            }
        }
        public static void demandInput(String filename,ArrayList<Integer> d_list){
            try{
                BufferedReader br = new BufferedReader(new FileReader(filename));
                String line = null;
                while((line = br.readLine()) != null){
                    d_list.add(Integer.parseInt(line));
                }
            }catch(IOException ioe){
                ioe.printStackTrace();
            }
        }
        public static int[] fourWaySplitter(int supply){
            Random r = new Random();
            int a = Math.abs(r.nextInt(supply));
            int b = Math.abs(r.nextInt(supply));
            int c = Math.abs(r.nextInt(supply));
            int d = Math.abs(r.nextInt(supply));
            int sum = a+b+c+d;
            double p1 = (double)a/(double)sum;
            double p2 = (double)b/(double)sum;
            double p3 = (double)c/(double)sum;
            double p4 = (double)d/(double)sum;
            int d1 = (int)Math.round((double)supply*p1);
            int d2 = (int)Math.round((double)supply*p2);
            int d3 = (int)Math.round((double)supply*p3);
            int d4 = (int)Math.round((double)supply*p4);
            if((d1+d2+d3+d4) > supply){
                int diff = (d1+d2+d3+d4) - supply;
                if(d1>diff){
                    d1 = d1-diff;
                }
```

```
			else if(d2>diff){
				d2 = d2-diff;
			}
			else if(d3 > diff){
				d3 = d3-diff;
			}
			else if(d4>diff){
				d4 = d4-diff;
			}
		}
		if((d1+d2+d3+d4) < supply){
			int diff =  supply - (d1+d2+d3+d4);
			d1 += diff;
		}
		if((d1+d2+d3+d4) != supply){
			System.out.println("Quadruplet mismatch.");
		}
		int[] result = new int[4];
		result[0] = d1;
		result[1] = d2;
		result[2] = d3;
		result[3] = d4;
		return result;
	}
	public static Triplet threeWaySplitter(int d){
		Random r = new Random();
		int a = Math.abs(r.nextInt(d)+1);
		int b = Math.abs(r.nextInt(d)+1);
		int c = Math.abs(r.nextInt(d)+1);
		int sum = a+b+c;
		double p1 = (double)a/(double)sum;
		double p2 = (double)b/(double)sum;
		double p3 = (double)c/(double)sum;
		int d1 = (int)Math.round((double)d*p1);
		int d2 = (int)Math.round((double)d*p2);
		int d3 = (int)Math.round((double)d*p3);
		if((d1+d2+d3) > d){
			int diff = (d1+d2+d3) - d;
			if(d1>d2){
				if(d1>d3){
					d1 = d1-diff;
				}
				else{
					d3 = d3-diff;
				}
			}
```

```
            else{
                if(d2>d3){
                    d2 = d2-diff;
                }
                else{
                    d3 = d3-diff;
                }
            }
        }
        else if((d1+d2+d3)<d){
            int diff = d - (d1+d2+d3);
            d1 = d1+ diff;
        }
        if((d1+d2+d3) != d){
            System.out.println("Triplet mismatch.");
        }
        return new Triplet(d1,d2,d3);
    }
    public static void supplierInput(String filename, ArrayList<Supplier> s_list){
        try{
            BufferedReader br = new BufferedReader(new FileReader(filename));
            String line;
            while((line = br.readLine()) != null){
                StringTokenizer tk = new StringTokenizer(line);
                int id = Integer.parseInt(tk.nextToken());
                int item = Integer.parseInt(tk.nextToken());
                int no = Integer.parseInt(tk.nextToken());
                int capacity = Integer.parseInt(tk.nextToken());
                int o_cost = Integer.parseInt(tk.nextToken());
                int p_cost = Integer.parseInt(tk.nextToken());
                int grade = Integer.parseInt(tk.nextToken());
                int c1 = Integer.parseInt(tk.nextToken());
                int t1 = Integer.parseInt(tk.nextToken());
                int c2 = Integer.parseInt(tk.nextToken());
                int t2 = Integer.parseInt(tk.nextToken());
                int c3 = Integer.parseInt(tk.nextToken());
                int t3 = Integer.parseInt(tk.nextToken());
                Supplier s = new Supplier(id,item, no, capacity, o_cost, p_cost,
grade);
                s.setCost(c1, c2, c3);
                s.setTime(t1, t2, t3);
                s_list.add(s);
            }
        }catch(IOException ioe){
            ioe.printStackTrace();
        }
```

```
	}
	public static void schemeInput(String filename, ArrayList<Scheme> schemes){
		try{
			BufferedReader br = new BufferedReader(new FileReader(filename));
			String line;
			int schemeIndex = 0;
			while((line = br.readLine())!= null){
				if(line.equals("Scheme Begins:")){
					Scheme s = new Scheme(schemeIndex);
					schemeIndex++;
					line = br.readLine();
					while(line.equals("Scheme Ends.") == false){
						StringTokenizer tk = new StringTokenizer(line);
						int _stage = Integer.parseInt(tk.nextToken());
						int _start = Integer.parseInt(tk.nextToken());
						int _end = Integer.parseInt(tk.nextToken());
						int _cost = Integer.parseInt(tk.nextToken());
						int _time = Integer.parseInt(tk.nextToken());
						schemeElement s_e = new schemeElement(_stage,
_start, _end, _cost, _time);
						if(_stage == 1){
							s.firstStage.add(s_e);
						}
						else if(_stage == 2){
							s.secondStage.add(s_e);
						}
						else if(_stage == 3){
							s.thirdStage.add(s_e);
						}
						else{
							System.out.println("NoStageError in the
scheme input section.");
						}
						line = br.readLine();
					}
					schemes.add(s);
				}
			}
		}catch(IOException ioe){
			ioe.printStackTrace();
		}
	}
	public static void chromosomePrinter(ArrayList<Chromosome> c_list){
		for(int i=0; i<c_list.size();i++){
			Chromosome c = c_list.get(i);
			System.out.println("Serial Number : "+i);
```

```
			for(int k=0;k<3;k++){
				ChromosomeScheme s0 = c.chromosomeParts.get(k);
				System.out.print("Supplier: ");
				for(int j=0; j<8;j++){
					int sum = 0;
					sum += s0.supplierGeneValues[(j*3)+0];
					sum += s0.supplierGeneValues[(j*3)+1];
					sum += s0.supplierGeneValues[(j*3)+2];
					System.out.print(sum+",");
				}
				System.out.println();
				System.out.print("Customer: ");
				int sum =0;
				for(int j=0;j<4;j++){
					System.out.print(s0.customerGeneValues[j]+" ");
					sum+= s0.customerGeneValues[j];
				}
				System.out.print("sum: "+sum+"\n\n");

			}
		}
	}
	public static void chromosomeFitnessPrinter(ArrayList<Chromosome> c_list){
		PrintWriter pw = null;
		try{
			pw = new PrintWriter(new FileWriter("data.txt"));
		}catch(IOException ioe){
			ioe.printStackTrace();
		}
		for(int i=0; i<c_list.size();i++){
			if(i==200){break;}
			Chromosome c = c_list.get(i);
			double val = 20000000.0 + (-10.0 *(double)c.cost)+(-
8595.0*(double)c.time)+(30.0*(double)c.quality);
			int lossval = GenAlgoObject.lossCalculator(c);
			if(i==0){
				try(FileWriter fw = new FileWriter("minitab data.txt", true);
				   BufferedWriter bw = new BufferedWriter(fw);
				   PrintWriter out = new PrintWriter(bw))
				{
					out.println("Serial: "+i+" cost: "+c.cost+" time: "+c.time+"
grade: "+c.quality+" loss:"+lossval);
				   out.println("Fitness value: "+val+"\n");
				   out.close();
				} catch (IOException e) {
				   e.printStackTrace();
```

```
                }
            }
            if(i%5 == 0){
                System.out.println("Serial: "+i+" cost: "+c.cost+" time: "+c.time+"
grade: "+c.quality+" loss:"+lossval+ " fitness value: "+val);
            }
            pw.println(c.cost+","+c.time+","+c.quality);
        }
        pw.close();
        PrintWriter pw1 = null;
        try{
            pw1 = new PrintWriter(new FileWriter("chromosome gene values.txt"));
        }catch(IOException ioe){
            ioe.printStackTrace();
        }
        for(int i=0; i<c_list.size();i++){
            if(i==200){break;}
            Chromosome c = c_list.get(i);
            pw1.println("Index: "+i);
            for(int k=0; k<c.chromosomeParts.size();k++){
                ChromosomeScheme cs_temp = c.chromosomeParts.get(k);
                for(int j=0; j<24; j++){
                    pw1.print(cs_temp.supplierGeneValues[j]+", ");
                }
                pw1.println();
                for(int j =0; j<4; j++){
                    pw1.print(cs_temp.customerGeneValues[j]+", ");
                }
                pw1.println();
            }
        }
        pw1.close();
    }
    public static void printSchemes(ArrayList<Scheme> listOfSchemes){
        for(int i = 0; i<listOfSchemes.size(); i++){
            Scheme s = listOfSchemes.get(i);
            System.out.println("Scheme Number: "+s.schId);
            for(int j=0; j<s.firstStage.size(); j++){
                System.out.print(s.firstStage.get(j));
            }
            System.out.println();
            for(int j=0; j<s.secondStage.size(); j++){
                System.out.print(s.secondStage.get(j));
            }
            System.out.println();
            for(int j=0; j<s.thirdStage.size(); j++){
```

```
                        System.out.print(s.thirdStage.get(j));
                    }
                    System.out.println();
                }
        }
}
class Driver{
        public static Utilities u ;
        public static ArrayList<Supplier> s_list;
        public static ArrayList<Scheme> scheme_list;
        public static ArrayList<Integer> d_list;
        public static GenAlgoObject ga;
        public static void main(String[] args) {
                u = new Utilities();
                s_list = new ArrayList<Supplier>();
                scheme_list = new ArrayList<Scheme>();
                d_list = new ArrayList<Integer>();
                u.supplierInput(args[0],s_list);
                u.schemeInput(args[1],scheme_list);
                u.demandInput(args[2],d_list);
                int sumDemand = 0;
                for(int i=0;i<d_list.size();i++){
                        sumDemand += d_list.get(i);
                }
                try(FileWriter fw = new FileWriter("minitab data.txt", true);
                   BufferedWriter bw = new BufferedWriter(fw);
                   PrintWriter out = new PrintWriter(bw))
                {
                   out.println("\ninitial size:"+args[3]+" generation number: "+args[4]+"
Crossover: "+args[5]+" Mutation: "+args[6]);
                   out.close();
                } catch (IOException e) {
                   e.printStackTrace();
                }
                ga = new GenAlgoObject(sumDemand, s_list, scheme_list, d_list);
                ga.initialPopulationGeneration(Integer.parseInt(args[3]));
                u.chromosomeFitnessPrinter(ga.population);
        ga.evolutionProcess(Integer.parseInt(args[4]),Integer.parseInt(args[3]),Integer.parseInt(ar
gs[5]),Integer.parseInt(args[6]));
                System.out.println("\nSolution space: \n");
                u.chromosomeFitnessPrinter(ga.population);
        }
}
```

Java Command: From DC to Customer

```
import java.io.*;
import java.util.*;
class Edge{
	public int dc;
	public int customer;
	public int cost;
	public int time;
	public Edge(int _dc, int _customer, int _cost, int _time){
		dc = _dc;
		customer = _customer;
		cost = _cost;
		time = _time;
	}
}
class EdgeComparator implements Comparator<Edge>{
	public int compare(Edge a, Edge b){
		if(a.cost > b.cost){
			return 1;
		}
		else if(a.cost < b.cost){
			return -1;
		}
		else{
			if(a.time > b.time){
				return 1;
			}
			else if(a.time < b.time){
				return -1;
			}
			else{
				return 0;
			}
		}
	}
}
class EdgeUtilities{
	public static void readInputData( ArrayList<Edge> e){
		try{
			BufferedReader br = new BufferedReader(new FileReader("customer
center data.txt"));
			int distributionCenterNumber = Integer.parseInt(br.readLine());
			int numberOfCustomers = Integer.parseInt(br.readLine());
			String line = null;
			for(int i=0; i<distributionCenterNumber; i++){
```

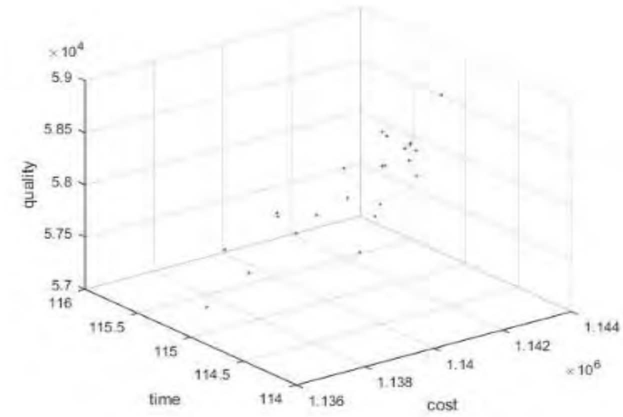
```
					line = br.readLine();
					StringTokenizer tk = new StringTokenizer(line);
					for(int k=0; k<numberOfCustomers; k++){
						e.add(new
Edge(i,k,Integer.parseInt(tk.nextToken()),Integer.parseInt(tk.nextToken())));
					}
				}
				br.close();
			}catch(IOException ioe){
				ioe.printStackTrace();
			}
	}
	public static void readDemand(ArrayList<Integer> dc_supply_list, ArrayList<Integer>
customer_demand_list){
			try{
				BufferedReader br = new BufferedReader(new FileReader("dc customer
demand data.txt"));
				int distributionCenterNumber = Integer.parseInt(br.readLine());
				int numberOfCustomers = Integer.parseInt(br.readLine());
				String line = br.readLine();
				StringTokenizer tk = new StringTokenizer(line);
				while(tk.hasMoreTokens()){
					dc_supply_list.add(Integer.parseInt(tk.nextToken()));
				}
				line = br.readLine();
				tk = new StringTokenizer(line);
				while(tk.hasMoreTokens()){
					customer_demand_list.add(Integer.parseInt(tk.nextToken()));
				}
				br.close();
			}catch(IOException ioe){
				ioe.printStackTrace();
			}
	}
	public static void printAllData(ArrayList<Edge> e, ArrayList<Integer> dc,
ArrayList<Integer> cus){
			for(int i=0; i<e.size();i++){
				Edge x = e.get(i);
				System.out.println(x.dc+" "+x.customer+" "+x.cost+" "+x.time);
			}
			System.out.println();
			for(int i=0;i<dc.size();i++){
				System.out.print(dc.get(i)+" ");
			}
			System.out.println();
			for(int i=0;i<cus.size();i++){
```

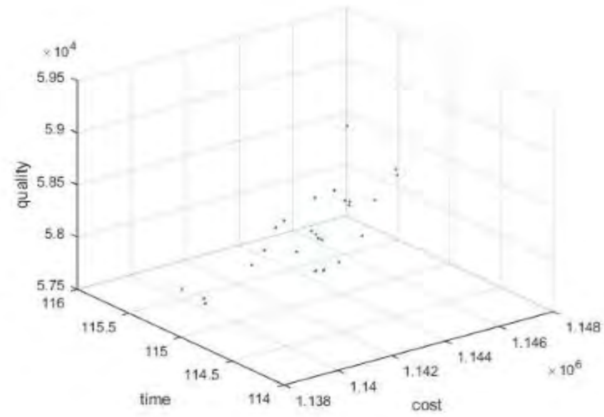
```
                        System.out.print(cus.get(i)+" ");
                }
                System.out.println();
        }
}
class EdgeDriver{
        public static ArrayList<Edge> edge_list = null;
        public static ArrayList<Integer> dc_supply_list = null;
        public static ArrayList<Integer> customer_demand_list = null;
        public static EdgeUtilities util = null;
        public static int numDC = 0;
        public static int numCus = 0;
        public static void process(){
                int totalCost = 0;
                int totalTime = 0;
                int[] available = new int[numDC];
                int[] remainingDemand = new int[numCus];
                int[][] answerSpace = new int[numCus][numDC];
                for(int i=0;i<numDC; i++){
                        available[i] = dc_supply_list.get(i);
                }
                for(int i=0; i<numCus; i++){
                        remainingDemand[i] = customer_demand_list.get(i);
                }
                for(int i=0;i<numCus;i++){
                        for(int j=0;j<numDC; j++){
                                answerSpace[i][j] = 0;
                        }
                }
                for(int k=0; k<edge_list.size(); k++){
                        Edge x = edge_list.get(k);
                        if(remainingDemand[x.customer]>0){
                                if(remainingDemand[x.customer] <= available[x.dc]){
                                        answerSpace[x.customer][x.dc] +=
remainingDemand[x.customer];
                                        totalCost+= (x.cost * remainingDemand[x.customer] );
                                        totalTime = max(totalTime,x.time);
                                        available[x.dc] = available[x.dc] -
remainingDemand[x.customer];
                                        remainingDemand[x.customer] = 0;
                                }
                                else if((remainingDemand[x.customer] > available[x.dc]) &&
(available[x.dc]>0)){
                                        answerSpace[x.customer][x.dc] += available[x.dc];
                                        totalCost+= (x.cost * available[x.dc] );
                                        totalTime = max(totalTime,x.time);
```

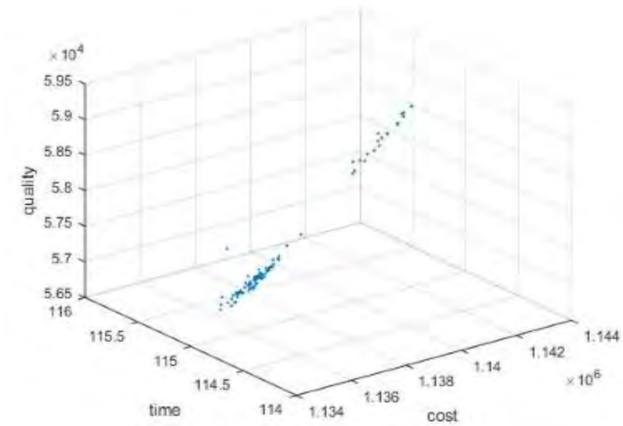
```
                                        remainingDemand[x.customer] =
remainingDemand[x.customer] - available[x.dc];
                                        available[x.dc] = 0;
                                }
                        }
                }
                System.out.println("Cost is: "+totalCost+"\nTime is: "+totalTime);
                for(int i=0;i<numCus; i++){
                        System.out.print("Customer NO: "+i+"-> ");
                        for(int j=0;j<numDC;j++){
                                System.out.print(answerSpace[i][j]+"  ");
                        }
                        System.out.println();
                }
        }
        public static int max(int a,int b){
                return (a>b)?a:b;
        }
        public static void main(String[] args) {
                edge_list = new ArrayList<Edge>();
                dc_supply_list = new ArrayList<Integer>();
                customer_demand_list = new ArrayList<Integer>();
                util = new EdgeUtilities();
                numDC = Integer.parseInt(args[0]);
                numCus = Integer.parseInt(args[1]);
                util.readInputData(edge_list);
                util.readDemand(dc_supply_list, customer_demand_list);
                Collections.sort(edge_list, new EdgeComparator());
                util.printAllData(edge_list, dc_supply_list, customer_demand_list);
                process();
        }
}
```

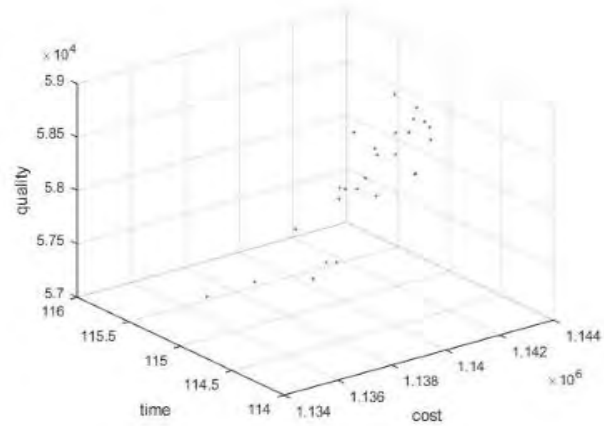
APPENDIX-B

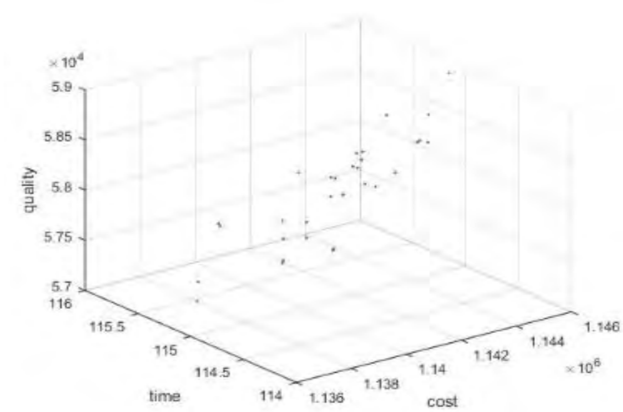
Illustration of Different Pareto Front for Different Generation

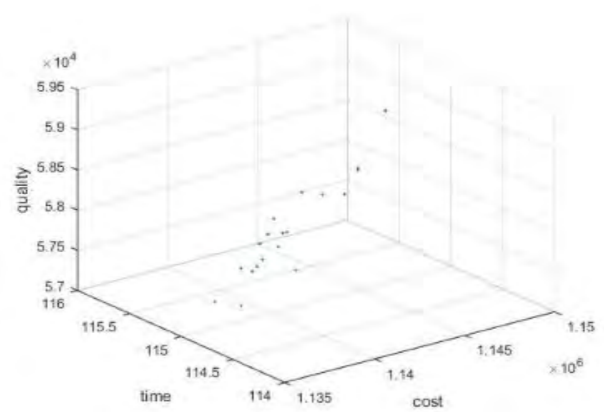


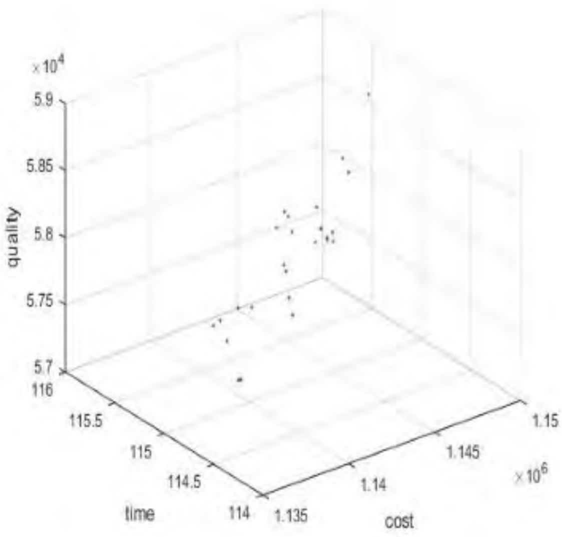


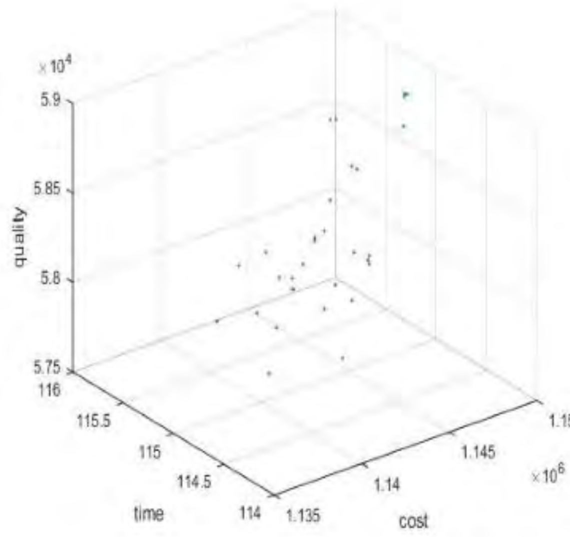


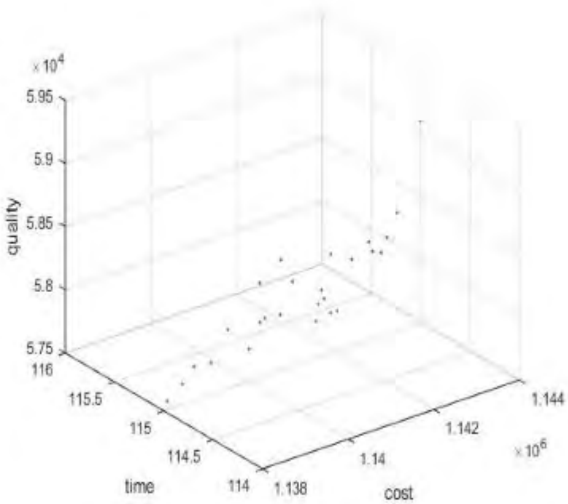


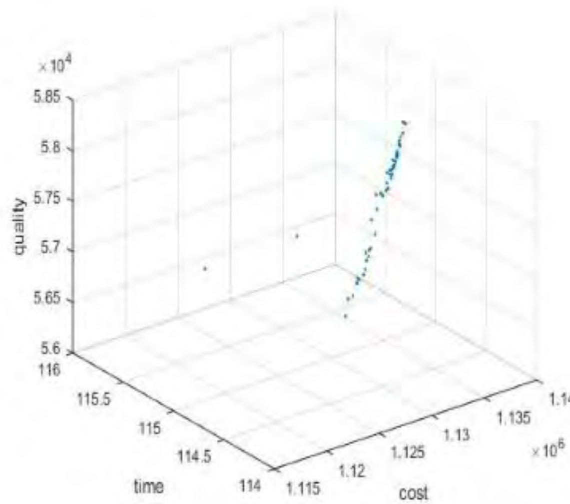


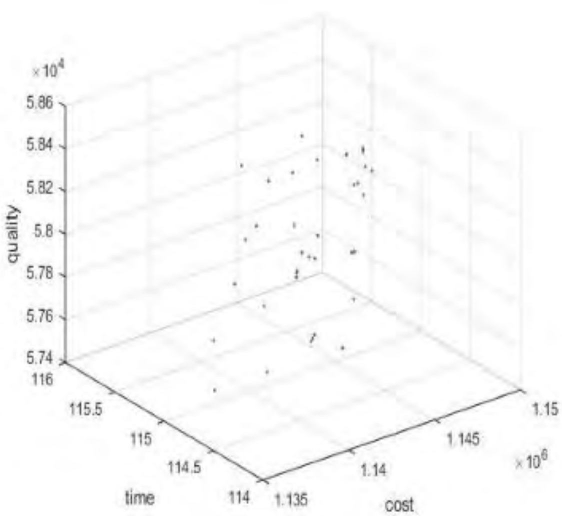


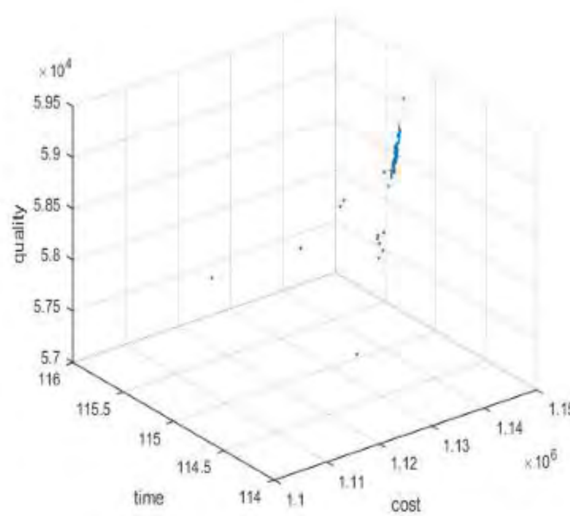

×10^4
5.9
5.85
5.8
5.75
5.7
quality
116
115.5
115
114.5
114
time
1.135
1.14
1.145
1.15
×10^6
cost


×10^4
5.9
5.85
5.8
5.75
quality
116
115.5
115
114.5
114
time
1.135
1.14
1.145
1.15
×10^6
cost


×10^4
5.95
5.9
5.85
5.8
5.75
quality
116
115.5
115
114.5
114
time
1.138
1.14
1.142
1.144
×10^6
cost


×10^4
5.85
5.8
5.75
5.7
5.65
5.6
quality
116
115.5
115
114.5
114
time
1.115
1.12
1.125
1.13
1.135
1.14
×10^6
cost


×10^4
5.86
5.84
5.82
5.8
5.78
5.76
5.74
quality
116
115.5
115
114.5
114
time
1.135
1.14
1.145
1.15
×10^6
cost


×10^4
5.95
5.9
5.85
5.8
5.75
5.7
quality
116
115.5
115
114.5
114
time
1.1
1.11
1.12
1.13
1.14
1.15
×10^6
cost