A flexible integrated forward/reverse logistics model with random path

Dem Fachbereich Produktiontechnik
der
UNIVERSITÄT BREMEN

zur Erlangung des Gardes eines
Doktors der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte

DISSERTATION

von

M.Sc. Elham Behmanesh

Gutachter: Prof. Dr. Jürgen Pannek
Prof. Christopher Irgens (University of Strathclyde, UK)

Tag der mündlichen Prüfung: 04.02.2019

Abstract

A flexible integrated forward/reverse logistics model with random path

M.Sc. Elham Behmanesh
Doctor of Engineering
Graduate Department of Production Engineering
University of Bremen
2018

This dissertation focuses on the structure of a particular logistics network design problem, one that is a major strategic issue for supply chain design and management. Nowadays, the design of the supply chain network must allow for operation at the lowest cost, while providing the best customer service and accounting for environmental protection. Due to business and environmental issues, industrial players are under pressure to take back used products. Moreover, the significance of transportation costs and customer satisfaction spurs an interest in developing a flexible network design model. To this end, in this study, we attempt to include this reverse flow through an integrated design of a forward/reverse supply chain network design, that avoids the sub-optimal solutions derived from separated designs. We formulate a cyclic, seven-stage, logistics network problem as an NP-hard mixed integer linear programming (MILP) model. This integrated, multi-stage model is enriched by using a complete delivery graph in forward flow, which makes the problem more complex.

As these kinds of problems belong to the category of NP-hard problems, traditional approaches fail to find an optimal solution in sufficiently short time. Furthermore, considering an integrated design and flexibility at the same time makes the logistics network problem even more complex, and makes it even less likely, if not impossible, for a traditional approach to provide solution within an acceptable time frame. Hence, researchers develop efficient non-traditional techniques for the large-term operation of the whole supply chain. These techniques provide near optimal solutions particularly for large scale test problems. In our case within this thesis, to find a near optimal solution, we apply

a Memetic Algorithm with a neighborhood search mechanism and a novel chromosome representation called "extended random path direct encoding method" which includes two segments. Chromosome representation is one of the main issues that can affect the performance of a Memetic Algorithm. To illustrate the performance of the proposed Memetic Algorithm, LINGO optimization software as commercial package serves as a comparison for small size problems. We show that the proposed algorithm is able to efficiently find a good solution for the flexible, integrated, logistics network. Each algorithm has some parameters that need to be investigated to provide the best performance. In this regard, the effect of different parameters on the behavior of the proposed meta-heuristic algorithm is surveyed first. Then, the Taguchi method is adapted to identify the most important parameters and rank the latter. Additionally, Taguchi method is applied to identify the optimum operating condition of the proposed Memetic Algorithm to improve the results. In this study, four factors that are defined inputs of the proposed Memetic Algorithm, namely: population size, cross over rate, local search iteration, and number of iterations are considered. The analysis of the parameters and the improvement in results are both illustrated by a numerical case studies. Finally, to show the performance of the Memetic Algorithm, a Genetic Algorithm - as a second meta-heuristic algorithm option - is considered as regards large size cases.

Zusammenfassung

A flexible integrated forward/reverse logistics model with random path

M.Sc. Elham Behmanesh
Doctor of Engineering
Graduate Department of Production Engineering
University of Bremen
2018

Die Dissertation behandelt die Struktur eines logistischen Netzwerkproblems, was wichtige strategische Eigenschaften in der Lieferkette (Supply Chain) darstellt. Das Design heutiger Supply-Chain-Netzwerke muss es erlauben, mit niedrigsten Kosten den bestmöglichsten Kundenservice sicherzustellen und dabei Umweltstandards einzuhalten. Aufgrund von ökonomischen und ökologischen Vorgaben stehen insbesondere Unternehmen in der Pflicht, bestimmte Produkte am Ende des Lebenszyklus zurückzunehmen. Die Bedeutung von Transportkosten und Kundenzufriedenheit motiviert die Entwicklung eines flexiblen Netzwerkdesignmodells. Zu diesem Zweck wird in dieser Arbeit der Rückwärtsfluss in das Forward-/Reverse-Supply-Chain-Modell mit integriert, um eine suboptimale Lösung durch ein seperierendes Modell zu vermeiden. Dazu stellen wir ein zyklisches, siebenstufiges logistisches Netzwerkproblem als NP-hartes gemischt ganzzahliges Problem (MILP) auf. Dieses integrierte, mehrstufige Modell wird durch die vollständige Modellierung des Vorwärtsflusses erweitert, was das Problem komplexer macht. Da dieses Problem zur Klasse der NP-harten Probleme gehört, versagen traditionelle Methoden zur optimalen Lösungssuche in kurzer Zeit. Erschwerend trägt das integrierte Design und die Flexibilität zur erhöhten Komplexität des logistischen Netzwerkmodells dazu bei, dass in den meisten Szenarien keine optimalen Lösungen mit traditionellen Methoden in akzeptabler Zeit zu finden sind. Daher entwickelte die Forschung effiziente, nicht auf herkömmlichen Verfahren basierte Techniken, um die gesamte hochskalierte Supply-Chain-Kette steuern zu können. Diese Techniken bieten nahezu optimale Lösungen speziell für große Testprobleme. Um eine fast optimale Lösung zu finden, wenden wir einen memetischen

Algorithmus mit einer Nachbarschaftssuche und einer neuen Darstellung von Chromosomen, die im Verfahren als "Extended random path direct encoding method", die zwei Segmente enthält. Die passende Darstellung der Chromosome beeinflusst stark die Leistungsfähigkeit von memetischen Algorithmen. Um die Leistungsfähigkeit des entwickelten memetischen Algorithmus zu zeigen, dient die LINGO Optimierungssoftware als kommerzielles Paket zum Vergleich für kleine Problemgrößen. Wir zeigen in dieser Arbeit, dass der entwickelte Algorithmus effizient eine gute Lösung für die Problemstellung des integrierten flexiblen Netzwerks findet. Um die höchste Leistungsfähigkeit des Algorithmus' festzustellen, ist eine Justierung der Parameter erforderlich. Daher wird der Effekt verschiedener Parameter auf das Verhalten des Algorithmus' zunächst untersucht und die Taguchi-Methode angepasst, um die wichtigsten Parameter zu finden und zu sortieren. Dann wird zusätzlich die Taguchi-Methode zur Identifizierung der optimalen Arbeitsbedingungen des entwickelten memetischen Algorithmus angewendet, um die Ergebnisse zu verbessern. In der Studie werden vier Faktoren betrachtet, die als Eingabeparameter für den memetischen Algorithmus dienen: Populationsgröße, Cross Over-Rate, Anzahl lokaler Suche und Anzahl von Iterationen. Die Analyse der Parameter und die Verbesserung der Ergebnisse werden numerisch dargestellt. Am Schluss, für große Szenarien, wie sie auch in der realen Welt vorkommen, wird ein genetischer Algorithmus zum Vergleich benutzt, um die Effizienz des entwickelten memetischen Algorithmus zu zeigen.

Acknowledgements

I would like to acknowledge my supervisor Professor Jürgen Pannek. It was fantastic to have the opportunity to work with him. Not only did he always help and guide me throughout the work, but also helped me, as a friend, to realize that "if there is one step backward, this is not the end and will be recovered by several steps forward". He created a great work atmosphere that leads to more success. His expertise was essential to the completion of this research. More importantly, I have been taught innumerable lessons and insight regarding academic research, which have shaped my present, and hopefully my future academic endeavours.

Also, I am always thankful to Prof. Haasis, Prof. Thoben and Prof. Buer for their perceptive comments along the way.

During this work I have collaborated with many colleagues of the Dynamic in Logistics (DIL) research group as well as International Graduate School (IGS). I want to thank everyone in the DIL research group, it was great sharing office space with all of you during my PhD journey. You have been like a second family to me. I also want to further thank all my friends in the IGS group who always took the time to listen to me. Additionally, I am grateful for the financial support of the Deutscher Akademischer Austauschdienst (DAAD) as well as the University of Bremen. I would also like to thank the International Graduate School (IGS) for their significant contribution, with a special mention to Dr. Ingrid Rügge for her unfailing support and assistance from beginning to end, including the preparation of this journey.

My special thanks and respect go to my parents, Eftekhar Azimisaravi and Valiollah Behmanesh. My mother, Eftekhar, who is the most strongest woman I have ever seen, who tried her best to spread happiness in any situation during my life, and who always supported me and helped me to realize my dreams. My father, Valiollah, who always shows his trust in me. He leads me to know how capable and strong I am to seek my dreams. Thank you to my older brother, Iman, who was like a real case of "where there is a will there is a way" in my life. (Who was a practical model of how to create my own life as I want). I would like to thank my little brother, Omid, who just like the meaning of his name, he is our family's hope, who was there whenever I needed him, and allows me to know I can always count on him.

And finally, last but no means least, I thank the love of my life, Hamed Sadeghi, whose

love and confidence is a constant source of inspiration and encouragement. With you Hamed, I realized I have whatever I want in my life, all remains, just help us to enjoy more.

Contents

List of Tables

List of Figures

Abbreviations

3PL	3th party logistics
ACA	Ant colony algorithm
CR	Cross over
CP	Cross over probability
DE	Differential evolution
DOE	Design of experiment
EA	Evolutionary Algorithm
EC	Evolutionary computation
ECC	Evolutionary computation community
Fc-Tp	Fixed cost transportation problem
FL	Forward logistics
GA	Genetic algorithm
HC	Hill climbing
hGA	hybrid genetic algorithm
IDCNIP	Integer discrete continuous programming
MA	Memetic algorithm
MILP	Mixed integer linear programming
MINLP	Mixed integer non-linear programming
MIP	Mixed integer programming
Mu	Mutation
MVEP	Mixed variable evolutionary programming
NP	Non-deterministic polynomial-time hardness
OA	Orthogonal array
pb-GA	Priority based genetic algorithm
Pop	Population
PSO	Particle swarm optimization
RL	Reverse logistics
SA	Simulated annealing

SMIP	Stochastic mixed integer programming
SNR	Signal to noise ratio
TS	Tabu search
TP	Transportation problem
WMX	Weight mapping cross over

To my wonderful parents,
Eftekhar *and* ***Valiollah****,*
who have raised me to be the person I am
and my love,
Hamed*,*
who has helped me to be better than the person I was
supposed to be.

Chapter 1

Introduction

A supply chain is a network of suppliers, manufacturers, warehouses, distributions, and retailers. It is organized to produce and distribute products, and the respective design issue is to determine the number, location, and capacity of the facilities as well as the quantity of the flow between them [11]. The aim of supply chain design is minimization of total cost [11; 46] or profit maximization [116], while satisfying customer requirements. Each supply chain network involves decisions on three different levels: operational, tactical, and strategic, which correlates to short-, mid-, and long-term decisions respectively. As long-term strategic decisions take priority over tactical and operational ones, network configuration comes first in any supply chain network design and needs to be optimized for long-lasting efficient operation of the entire supply chain. In the open-loop supply chain, which is the traditional (forward) system, products do not return to their source. Figure 1.1 shows the traditional supply chain network including suppliers, plants, distribution centers, and retailers and customers. In this network, raw materials are shipped from suppliers to plants and the final products are transferred from plant to customers through distribution centers and retailers. But in today's world, companies cannot ignore reverse distribution due to strict rules of end-of-life issues [155], economic impact, as well as accounting for reverse flow when industries can use returned products. On the other hand, companies have to offer something to increase customer satisfaction and keep themselves competitive. Considering all these factors in a supply chain network, a comprehensive study at the strategic level is needed for any supply chain network.

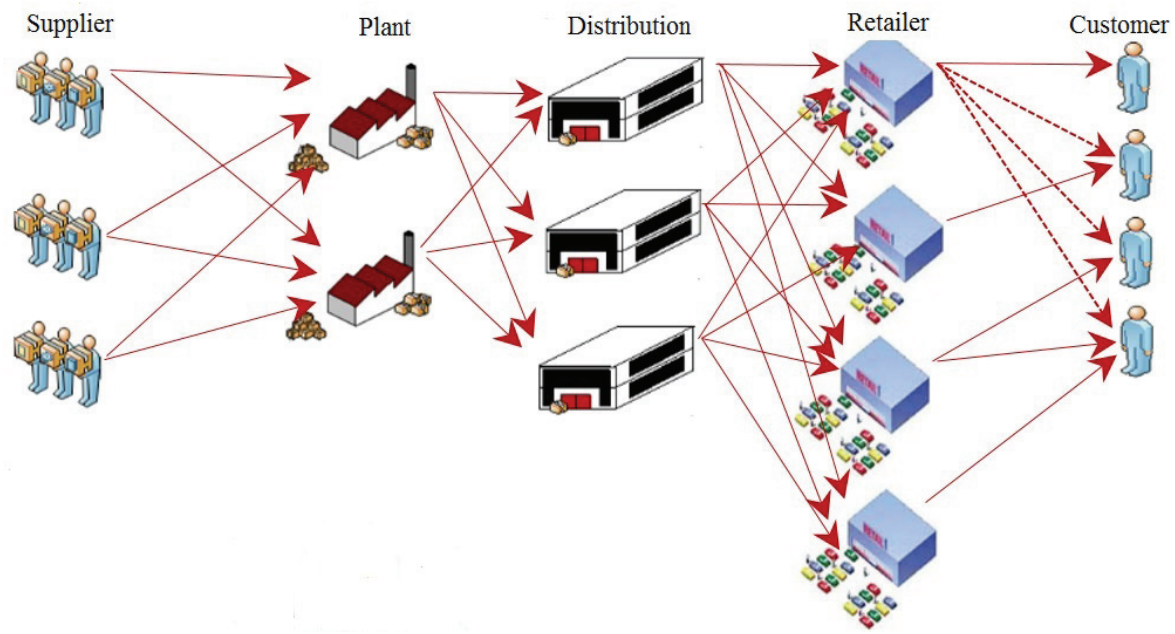


Figure 1.1: Traditional supply chain network.

1.1 Motivation and problem statement

In many countries, depending on environmental laws, companies are forced to equip their facilities in order to make the recovery of used products possible. As an example, in the last decade governmental legislation forced firms to collect, recover, and recycle, e.g., all electronic goods in Europe, Japan, China and many parts of US and Canada as well as so to the safe disposal of their end-of-life (EOL) products [155]. The importance of this issue leads some countries to ratify an annual recovery rate [12]. Also, there are some pick-up policies for logistics networks that reveal this procedure differs for each country [12]. In addition to environmental damages from ignoring reverse flow, shortages in resources are the second problem [59]. As a consequence, closed loop supply chains have become a pressing topic for supply chain partners.

Environmental factors are not the only ones that spur researchers or manufacturers to consider reverse logistic networks. In fact, the reason that recovery systems are attractive to companies is economic impact. The reverse activity represents added economic value that can be obtained by processing the returned products and capturing the remaining value in the used products. Indeed, one ton of electronic computer waste contains more gold than seventeen tons of material extracted from a gold mine [10]. It has been shown

that the unit cost of production can decrease by 60 to 40 percent by considering reverse activities in some industries [50]. In another example, Xerox Europe made more than 80 million dollars by applying a take back strategy for used products strategy in 1997 [126].

By considering reverse distribution, some facilities need to be established, such as collection/inspection centers to collect and inspect the products under expert supervision, as well as disposal center to ensure a safe disposal for non-recyclable items. Also, some sections in plants need to be updated for recycling or recovery of used products. Therefore, job creation can be considered as the other aspect of a closed loop supply chain network. We can summarize the benefits of the extended supply chain network as follows:

- Protects the environment
- Enhances economic impact
- Avoids shortages in resources
- Creates job opportunities

The reverse activities are classified in five steps [61] as follows:

- **Step 1**: Collect used products from customer.
- **Step 2**: Assess condition of the returned products by inspection and/or separation center.
- **Step 3**: Reprocess of used products into usable products.
- **Step 4**: Dispose of unrecoverable returns. It has to be noted that, in some cases, recovering is not economical. These conditions can be considered as unrecoverable returns as well.
- **Step 5**: Reuse the recovered items.

Depending on the condition of the returned products the recovery process may include repairing, reassembling, remanufacturing, cannibalizing, cleaning, replacing or recycling [205].

According to the above explanation, reverse distribution is important and needs to be investigated in each supply chain network. Most researchers in this area focused on reverse flows only [175], while considering the reverse and forward flow at the same time

can reveal better results [62; 155]. Figure 1.2 illustrates the proposed extended supply chain network. In this study, the reverse flow including collection/inspection center as well as disposal center is added to the traditional network. In the reverse flow, used products are collected by collection centers and after inspection, sorting, and disassembly, the recoverable products are shipped to the plant for further operation and returned to the cycle while scrapped products are transferred to disposal centers for safe disposal. It is important to note that the remanufacturing process can occur in a separated facility or in the plants where new products are produced. The same idea may be considered for collection and inspection centers. It is clear that establishing several facilities at the same location can reduce the cost of the supply chain network, in comparison with a separated design.

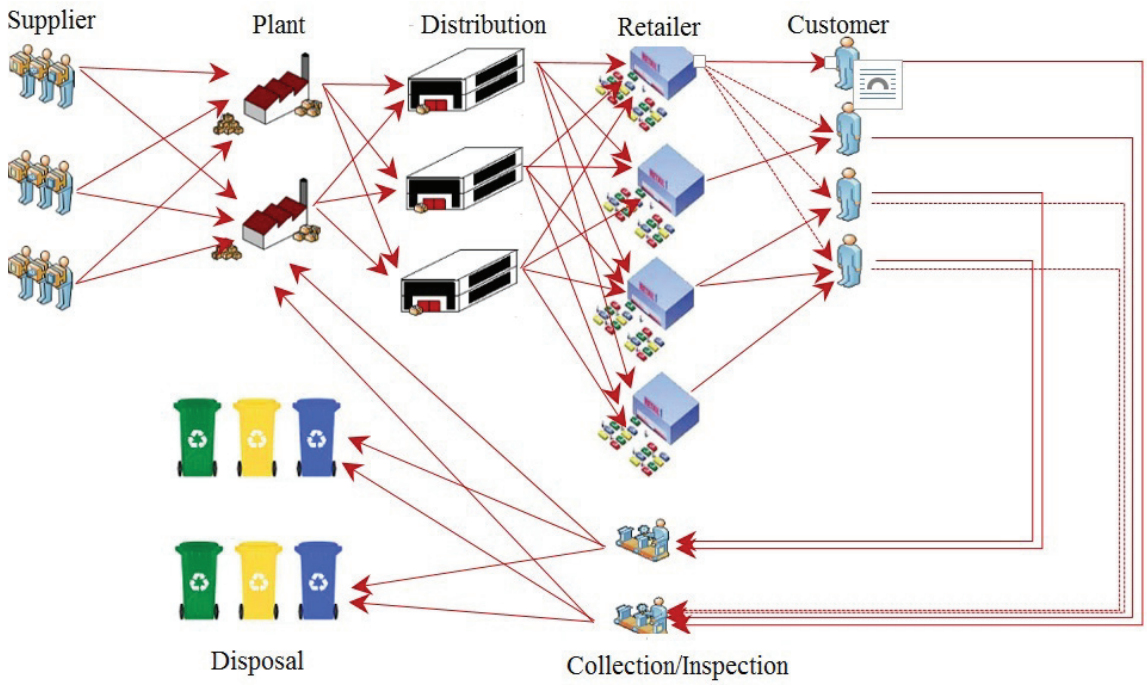


Figure 1.2: Extended supply chain network.

Fast, punctual, and accurate delivery of products plays an important role in customer satisfaction. The ability of a supply chain to satisfy the customer's expected delivery time is called "supply chain responsiveness" [172; 98], and result in greater profit for the related supply chain. Also, a shorter path can reduce transportation cost, and thus the total cost of the network. Some researchers proposed models to optimize the supply chain

network with regards to both topics, cost efficiency and responsiveness, simultaneously [172; 9]. However, most of these researchers limited themselves to only considering shipments between consecutive stages or just indirect shipment mechanisms [172; 9; 51]. Yet, efficiency can be improved by all possible delivery ways (flows between facilities that are not sequential), which allows us to skip some stages. Although a flexible delivery path can increase the efficiency of the supply chain network, the resulting problem will be more complex, cf. Figure 1.3 for a sketch of the delivery graph. It shows a fully capacitated graph between plant to customer in the forward flow.

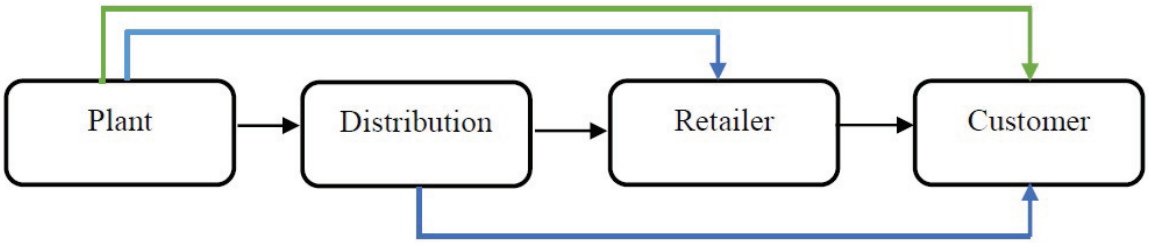


Figure 1.3: Framework of the proposed flexible supply chain network.

The benefits of the flexible supply chain can be summarized as follows:

- Enhances economic impact
- Reduces transportation cost
- Increases customer satisfaction
- Decreases delivery time

1.2 Research objectives

In real world applications, we often seek an answer to the question: *How can we find an efficient assignment strategy to satisfy the source and destination requirement with the aim of minimizing the total cost?* So, how to manage a logistic system efficiently has become a key issue for many companies in reducing their cost and delivery time as well as increasing customer satisfaction. On the other hand, well-managed reverse logistics allows us to cope with environmental factors by reducing, recovering, and reusing items, while also saving cost. Although industrial players are under pressure to take back

products after their use, most of logistics networks are not equipped to handle returned products.

To manage a logistic system efficiently, flexible and productive network design is one of particular interest.

As the next step, we need to model the network mathematically. The model has to address the problem as a holistic system and avoid splitting it into forward and reverse networks. Furthermore, the model should be capable of coping with the flexibility of the network. The mathematical model is aimed at minimizing the total cost of the network, including transportation and operation costs. Therefore, the next question will be: *How can we model the flexible, integrated, forward/reverse logistics network mathematically?*

In general, network design problems belong to the class of NP-hard problems and applying a flexible, integrated, logistics model makes the problem more complex. These conditions lead us to formulate the subsequent question: *How can we tackle this flexible, integrated network as an NP-hard problem?*

However, this increase in size, makes solving these complicated network design problems efficiently a major challenge. In the last two decades, Evolutionary Algorithms (EAs) have been applied in this attempt and obtained noticeable success [118]. While we can consider this ability of EAs as an advantage, most of the available Evolutionary Algorithms do not show good performance if the problem size increase [118]. Therefore, it is critical to present a modern Evolutionary Algorithm to study large scale optimization problems and answer these significant questions: *Which algorithm may be suitable in terms of accuracy and efficiency as a solution methodology for the proposed flexible, integrated, forward/reverse supply chain network?*

A Memetic algorithm with specialized encoding, initialization, and local search operator is considered to optimize the design of the proposed supply chain network. Each algorithm has some parameters that need to be investigated to increase efficiency. The parameters involved in this methodology include: population size, number of iteration, cross over rate, and number of local search iterations. Therefore, we face these questions: *What is the effect of these different parameters of the solution methodology on the results? What is the most important parameter and the importance of the order?*

Based on the aforementioned descriptions, within this study we consider an integrated forward/reverse multi-stage, single product, single period logistics network, which is enriched by a full delivery graph in forward flow, representing an NP-hard mixed integer linear programming model. Figure 1.4 shows the framework of this problem graphically.

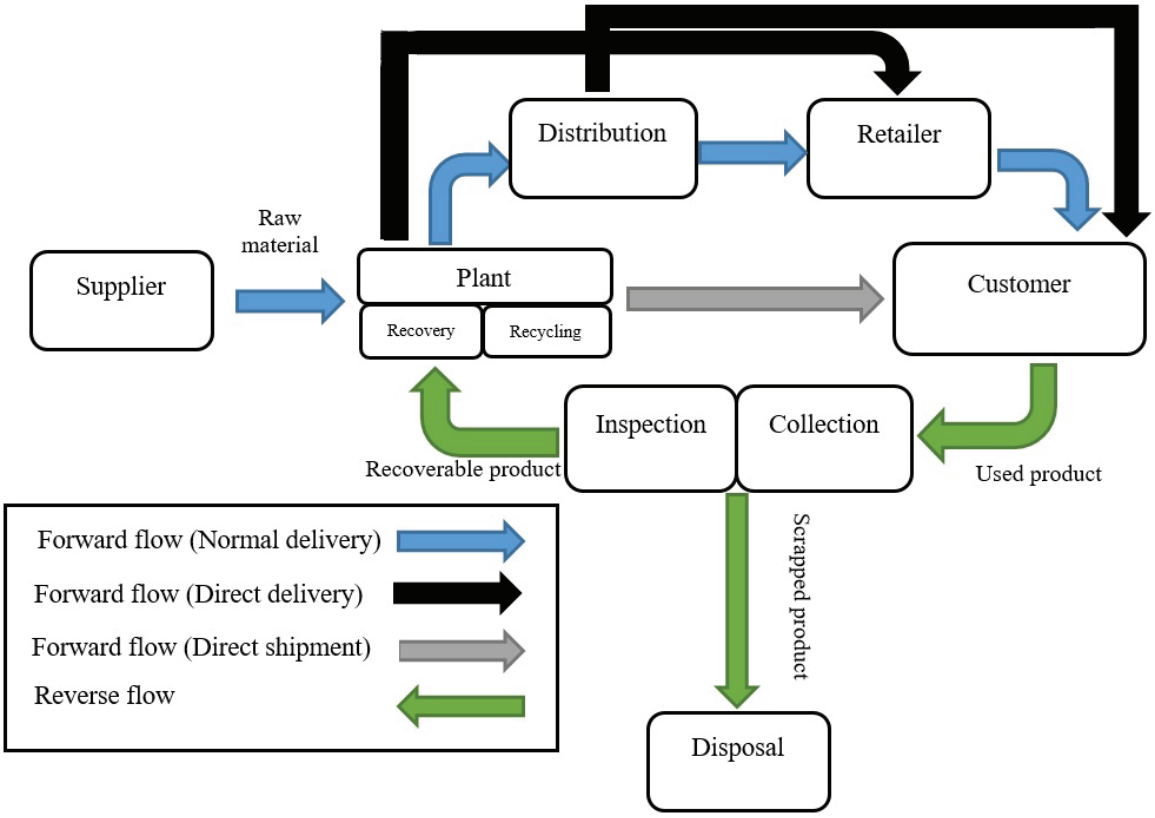


Figure 1.4: Framework of the proposed flexible supply chain network.

1.3 Research approach-an overview

The objectives provide an accurate description of the specific actions we will take in order to reach the aims. In this section, the proposed research objectives are classified into eight components as follows:

- **1. Overall characteristics:** In this study we address a seven stages closed-loop supply chain network including suppliers, plants, distribution centers, retailers, and customers in forward flow and collection/inspection centers and disposal centers in

backward flow. This network is operated on a "single-product" and "single-period" model.

- **2. Integration design of forward/reverse logistics network:** Reverse distribution can take place through a separated design or forward and reverse flow can be considered simultaneously as an integrated design. In this study, the integrated design is selected for two main reasons: The first reason is to avoid sub optimal solutions derived from separated designs. The second reason is take into account "product life cycles" for better utilization of manufacturing resources. This idea can be considered as a good way to increase the efficiency of the proposed supply chain network.

- **3. Flexible delivery path:** In the proposed model, we are aiming to connect the supply chain network more productively and flexibly by applying three different delivery paths that allow us to skip some stages. Although there are some cases in the real world that are applying these delivery paths separately, the idea behind considering these three delivery paths is having a fully capacitated graph between plants and customers in forward flow. Within this situation we can decrease the transportation cost by skipping some stages and also increase customer satisfaction by being close to customers and reducing lead delivery time.

- **4. Mathematical modeling:** After designing the network, we need to model our network mathematically. This model has to be capable of addressing the problem holistically, without splitting forward and reverse flow. Also, this model needs to be capable of handle three different delivery paths, as explained before. The mathematical model is aimed at minimizing the total cost including transportation and operation cost, and dealing with long-term decisions. In long-term decisions, determining the optimal number and capacity of facilities and also obtaining the optimal distribution network is addressed. The mathematical model is considered based on linear integer programming.

- **5. Developing a suitable solution methodology:** Generally, any network design problem belongs to a category of NP-hard problems. Therefore, traditional methods distribution failed to solve them in appropriate time. Additionally, the presented problem is a multi-choice problem. It means we are dealing with different subjects such as delivery paths, handling returned products, and meeting the demands of customers at the same time. Furthermore, by increasing the problem size, the problem will be even harder to solve. So we apply a meta-heuristic algorithm,

(Memetic Algorithm in particular) to tackle this NP-hard problem. To achieve the mentioned objectives, one of the contribution of this work is the development of a practical solution method based on the previously developed methods. To generate a population of individuals, we extend the "random path direct encoding method" [67] according to the characteristics of the proposed model. The first version of this method applied for a forward network problem with three echelons without any reverse flow. We aim to update the "random path direct encoding method" named as "extended random path direct encoding method" by adding two more echelons in forward flow, as well as reverse . Also, as we have flexibility in the network, the second segment is added to this method. Developing a solution methodology by considering a novelty in chromosome representation is one of the main focus of this study.

- **6. Parameter analyzing:** This section is divided into three parts and focuses on parameter analyzing. In the first part, the parameters involved in the solution methodology are introduced. Based on the numerical research, we realize that the introduced parameters are important and need to be investigated further. Therefore, the effect of each parameter is discussed separately. In the second part, a ranking between these parameters is derived to show which one has the stronger effect and needs more focus in special cases. Finding an order of effectiveness is the main contribution of this part. As we are dealing with a strategic problem, there is no time limitation. But still, having some information regarding the parameters can help us to improve our results, especially for the other types of concerns (tactical level). In this regard, Taguchi method is adopted to find the optimal value of parameters.

- **7. Programming** Our implementation was written in MATLAB R2015b and run on the PC with Intel CoreTM i5 2.40GHz with 12 GB RAM.

- **8. Validation:** The validation is divided into two parts. First, small size test problems are solved by a traditional linear programming software package, LINGO optimization software, to obtain the optimal solution. The results are considered in comparison to the results obtained from the proposed Memetic Algorithm. So if the optimal solution can be found by the Memetic Algorithm, we can trust it for large size problems. As the second part, the strategy of comparing between two meta-heuristic algorithms is taken to show the efficiency of the proposed Memetic Algorithm. Genetic Algorithms (GAs) are the most famous Evolutionary Algorithms (EAs), which are inspired by natural evolution and selection. Their main

application is in the field of optimization. The same strategy applied for initial population-, cross over and selection operator can be used for a classical Genetic Algorithm as well. These reasons lead us to employ a classical Genetic algorithm as the second meta-heuristic algorithm to compare with the proposed Memetic Algorithm. On the other hand, we can show how a local search engine can lead the algorithm to obtain better results.

1.4 Thesis outline

The structure of the thesis is as follows:

Chapter 2, entitled, "state of the art", is divided into two parts. In the first part, we aim to answer the question: "How can we find an efficient assignment strategy to satisfy the source and destination requirement with the aim of minimizing total cost". To this end, a description of a forward and reverse logistics network is provided individually first. These definitions can help us to grasp their characteristics as well as existing background, separately. In fact, by a comprehensive literature review we try to realize that "in which direction they need more effort?". Furthermore, the integrated design is carefully explained in this chapter. As the considered integrated network in this work is enriched by an additional feature, the rest of this chapter clarifies this feature. This feature is adding flexibility by considering different delivery paths into the network. It needs to be mentioned that this flexibility is referring to delivery paths in the forward distribution and employed as a full delivery graph from plant to customer. The second part is focused on solution methodology for the described problem with the mentioned complexity. The problem presented in this study is an NP-hard problem. Therefore, answering the following question is required here: "How to tackle with this NP-hard problem?". After finding the direction, we need to present a suitable solution methodology and find an answer for the question: "Which algorithm may be suitable as a solution methodology for the proposed problem?". As the main solution methodology considered for this study is a nature-based algorithm, historical background is first provided. Basic concepts and definitions of the solution methodology is presented afterward. To demonstrate that the proposed algorithm can support a variety of problems, four different categories are illustrated to address different application of the used algorithm. Further, we aim to compare our algorithm with another algorithm for the validation part. So, a brief background regarding the second solution methodology and its application is first provided.

Chapter 3, under the title "Modeling of the integrated, flexible logistics network", focuses on the mathematical model of the presented network and answers the question: "How can we model the problem to be capable of accounting for flexible delivery path and also integrated design simultaneously?". First of all, the main characteristics of the network that need to be considered in the formulation is described. Secondly, a general model is proposed to give an overall idea. Assumptions and notations, used in the formulation of the MILP model, are expressed thereafter. Utilizing the notation and assumptions, the structure of the MILP model is formulated. The proposed model, presented in this chapter, have been used for a published paper in **the 7th IFAC Conference on Management and Control of Production and Logistics, Bremen, Germany,** (E. Behmanesh and J. Pannek 2016), [20] as well as **the Logistics Research journal, Springer,** (E. Behmanesh and J. Pannek 2016), [19].

Chapter 4, "Proposed solution approach", gives exposition on the solution methodology. A comprehensive explanation regarding updating the algorithm according to the characteristic of the proposed problem is presented in the last section of this chapter. A novel extension for the population of individuals generation's method is obtained as well as updating the local search strategy. They are defined as the main contribution of this chapter. The overall procedure of proposed Memetic Algorithm is presented in this chapter. Some parts of the findings of Chapter 4 have been presented and published in **the 7th IFAC Conference on Management and Control of Production and Logistics, Bremen, Germany,** (E. Behmanesh and J. Pannek 2016), [20].

Chapter 5, " Effect of various parameters of solution methodology", investigates the effect of different parameters of the proposed Memetic Algorithm and aims to answer the question: "What is the effect of different parameters of the solution methodology on the results". To this end, the various parameters of the solution method are introduced first. These parameters include population size, number of iteration, number of local search iteration, and cross over rate. The effect of each parameter on the results of the solution methodology is studied through out this chapter. The initial findings of Chapter 5 have been presented in **"28th European Conference on Operation Research, Poznan, Poland"**, (E.Behmanesh and J. Pannek 2016), first and afterward have been published in the **"Mathematical Problems in Engineering journal"** (E. Behmanesh and J. Pannek 2017), [24].

Chapter 6, entitled "Parameter analysis", is designed to identify the optimum operating condition of the proposed Memetic Algorithm. In this regard, Taguchi method is adopted to find the best level of the four considered factors, namely: population size, cross over rate, local search iteration, and number of iteration. Applying the Taguchi method requires six steps, which are undertaken in this chapter, through several test problems. First, a comprehensive explanation regarding Taguchi method is provided. Afterward, the results are generated based on the step-by step procedures involved in the Taguchi method. Some of the results obtained in this chapter will be submitted in a relevant **journal**, by the title of "Taguchi analysis for improving optimization of integrated forward/reverse logistics" (E. Behmanesh and J. Pannek 2017), ([23]).
Also in this chapter, we endeavour to answer the question: "What is the most important parameter and the importance of the order?". In this regard, the order of mentioned parameters was found by applying the Taguchi method, which shows the most effective parameter and ranks accordingly. The extended version of the obtained results, regarding effectiveness order, have been presented and published in **the 6th International Conference on Dynamics in Logistics Bremen, Germany, 2018**, (E. Behmanesh and J. Pannek), ([22]).

In **Chapter 7**, "Numerical results of applying the proposed Memetic Algorithm", aims at answering the question: "How to show accuracy and efficiency of the proposed algorithm?". The quality of the approach is assessed in two parts. For the first part, we generated several test problems as numerical settings. Comparing the respective results obtained by the proposed Memetic Algorithm for test cases and solutions obtained by LINGO optimization software are the main contribution of this part. The computational results showed that the proposed algorithm is able to efficiently find good solutions for the flexible integrated logistics network. The initial findings of this part have been published in **the Logistics Research journal, Springer,** (E. Behmanesh and J. Pannek 2016), [19].
The second part is a comparison between two different meta-heuristic algorithms. This comparison is between the results obtained by the proposed Memetic Algorithm and a classical Genetic algorithm. To illustrate the tradeoffs between these, four comparison criteria are considered. These are generally focused on two aspects: objectives function value and CPU time. One of them is responsible for comparing the results obtained by the proposed Memetic Algorithm and the optimal solution of a commercial package. The same procedure is considered with separated criteria, to show the performance of the classical Genetic algorithm. The third criteria is designed to display the difference of the

proposed MA and GA and compare them with respect to the objective function value. And finally, the CPU times associated with the MA and GA is compared by the last criteria.
The aim of this chapter is to illustrate the efficiency of the proposed solution methodology, particularly in large size test problems. Therefore, the reminder of this chapter consists of numerical results based on the above explanation. The initial results of this part have been presented **in the 20th International Conference on Engineering Optimization and Industrial Applications, London, United Kingdom, 2018,** and some final results will be submitted to a high impact **journal**, under the title of "Memetic and Genetic algorithm for an integrated logistics network with flexible delivery path" (E. Behmanesh and J. Pannek 2018), [21].

Finally, the conclusions of this thesis and directions for future research are given in **Chapter 8**.

The problem addressed in this work is an integrated design of forward and reverse logistics enriched by three different delivery paths for a seven stage, single period, single product closed-loop supply chain network. The proposed model presents a general network and covers the previously described cases in the literature with less complexity. Furthermore, considering a full delivery graph in forward flow allows us to solve the conflicting goals of profit and responsiveness that otherwise may incur a greater cost [172].
Regarding the contributions of this study, we suggest a practical model as a decision support system that shows how contribution between different facilities can help us to control the entire of the supply chain network. Supporting this idea, this function cannot be filled by a person or a single facility. It needs the contribution of all related facilities to decrease the overall network cost and increase the efficiency of the network. On the other hand, network design problems belong to the category of NP-hard problems, therefore, developing an efficient solution methodology is still a critical requirement in this area. In addition, the flexible, integrated, forward/reverse logistic network suggested in this study is not a case-based network and because of its generic nature it can support a variety of industries such as electronic and digital equipment industries, and automotive industries. Therefore, progress in this area can bring multilateral profit.

Chapter 2

State of the art

2.1 Introduction

Logistics is the organization, planning, and realization of the flow of goods along the entire product life cycle in order to meet the requirements of customers or corporations while minimizing cost. Logistics network design addresses a select subset of facilities to be opened, as well as designing the distribution network strategy. The network model is the main part of the logistics network [153]. A set of nodes that are connected by transportation links is a usual way to model a logistic problem. These nodes may include suppliers, plants, distribution centers, retailers, and customers in forward flow. A supplier is a person or company that provides plants with raw materials. A plant is a place where a manufacturing process takes place, which is defined as the process of converting raw materials, components or parts into finished goods. Distribution centers are responsible for stocking goods for distribution to retailers, wholesaler or other possibilities. A retailer is a business or person that sells products to the consumer. In contrast with a supplier who is connected with another business, retailers are involved with the final customer directly.

In recent years, growing environmental concern pushed companies to pay more attention to product take-back, product recovery, and the re-distribution of end-of-life products. In this regard, Reverse logistics (RL), which refers to the return distribution activities, has an important role in modern industries. The possible nodes in backward flow can include: collection centers, inspection centers, and disposal centers. Collection centers manage used products by collecting returned products from customers. In inspection centers, if the materials or items are in proper quantity and condition, they will be sent to the plant for remanufacturing, reusing, recycling or other processes. The scraped products need to be transferred to disposal centers to be safely disposed of.

2.2 Network characteristics, process and challenges

Focusing on the concept of a logistics network, three categories appeared in the literature. Papers belonging to the first category are involved with forward logistics networks, while those in the second category are interested in the reverse flow only. The third category adopts an integrated approach that considers both forward and reverse flow simultaneously, in order to avoid the suboptimal solutions resulting from separated designs [62]. Therefore, we split our literature review to the research area of network design with forward, reverse and integrated logistics network.

2.2.1 Forward logistics network

In previous studies, the design of forward and reverse logistics networks were typically separated. In the field of forward logistics (FL), many models were developed as part of a traditional logistics network design. The common mixed integer linear programming (MILP) model aims for the choice of facilities to be open and the distribution network design to satisfy the demand with minimum cost.

Yeh [219] developed a MILP model for a supplier-production-distribution networks. He revised existing mathematical models presented by other researchers.
Altiparmak et al. [9] designed a multi-objective mixed integer nonlinear model (MINM) for a single product of a plastic company in a forward logistics network.
While most researchers in this area limited themselves to consider a single capacity level for each facility, Amiri [11] proposed a MILP model for a multi-stage and multi-capacity levels forward network design. The aim of this study was to determine the number, location, and optimal capacity of facilities as well as the best strategy for distribution.
Syarif et al. [201] also proposed a MILP formulation, though in their case for a fixed charge and multi-stage transportation problem for a single commodity forward supply chain model.
A two-echelon facility location problem was studied by Tragantalerngsak et al. [206]. In this work, each facility in the second echelon was limited in capacity and could only be supplied by one facility in the first echelon. Also, each customer is serviced by only one facility of the second echelon.
A different two-stage distribution-planning problem was addressed by Gen et al. [71] to minimise the total cost including transportation and opening costs.
A multi-period, multi-product fuzzy production and distribution planning supply chain model was presented by Aliev et al. [8]. They considered a trade-off between the filtrated

fuzzy customer demand and the profit.
Fallah-Tafti et al. [57] developed a multi-objective closed-loop supply chain network. Uncertainty from inexact cost coefficients and customer demands were considered in this study.
A four-echelon supply chain network with shortages was promulgated by Khalifehzadeh et al. [97]. The aim of the study was minimizing the operation cost of the whole supply chain and maximizing the reliability of the system through a multi-objective mathematical model.
Hinojosa et al. [84] introduced a dynamic supply chain with some characteristics, like allowing the carry over of stock in warehouses between consecutive periods. The objective was to minimize the total costs, including transportation and inventory holding costs, for products as well as fixed and operating costs for facilities to meet the demands of different products specified over different time periods at various customer locations.
Lin et al. [113] formulated a mixed integer linear model by including the direct shipment and direct delivery of logistics network as well as inventory. The authors's idea was having more productivity and flexibility can increase the efficiency of the supply chain network.

2.2.2 Reverse logistics network

Due to legislative changes regarding End-Of-Life (EOL)[155] issues as well as economic factors [10], considering the forward logistic network and omitting any reverse flow is impractical. A reverse logistics network describes a relationship between the end-users and facilities where new products are produce. When these two stages are considered simultaneously in a network it is called a closed loop network, otherwise it is called an open loop network [187].

A general review of the current developments in reverse logistics was presented by Pokharel and Mutha [175]. They identified three factors, that differ for reverse logistics and a traditional supply chain: 1: Most logistics networks are not equipped to handle returned product movement; 2: Reverse distribution costs may be higher than moving the original product from plant to customer; 3: Returned products may not be transported, stored, or handled in the same manner as in the regular channel [189].
Üster et al. [208] considered a semi-integrated network design problem and solved it by an exact method based on Benders decomposition. In this model, there is a limitation that any customer can only be connected to a single supply source, which renders the

model impractical.
Lites and Dekker [116] studied a stochastic mixed integer programming (SMIP) model by considering demand uncertainties and quality factors as important issues in the reverse design. The objective of the proposed model is to maximize the total profit in a sand recycling network. They developed their model for different situations dealing with several scenarios.
In a study by Jayaraman et al. [92], an analytical model to minimize reverse distribution costs was developed for operations of hazardous products. These products are first transferred to collection centers and then sent to refurbishing facilities. This MILP model limited supply of customer demand to a single distribution center. In addition, there was a tight bound on the number of collection and refurbishing sites. The presented model is aimed to find the optimal number and location of collection and refurbishing centers, as well as the flow of the hazardous products.
Another recent paper that consider reverse strategy is due to Min et al. [141]. The authors designed a MINLP model to minimise the overall reverse logistics costs, including spatial and temporal consolidation of returned products. In this study, the number of products returned to users is known. A mixed-integer nonlinear model is provided in order to find the optimal number and locations of collection points as well as centralized return centers.
Salema et al. [188] found focusing on generality in reverse network is a critical need. A mixed integer model is proposed for a reverse logistics network based on the following conditions: 1. Capacities are limited, 2. multi-product supply chains are considered, 3. uncertainty in product demands and returns are considered. The aim of this study is the minimization of total cost while determining the number of products and establishing a network for each product are considered in advance.
Aras et al. [12] probe a nonlinear model to assign the locations of collection centers in a reverse logistics network. In contrast to other studies, they proposed a model that is capable of defining the optimal buying price of used products to maximize the total profit.
Du and Evans [51] developed a bi-objective reverse logistics network by third party logistics (3PL) providers for post-sale services. The objectives of the model included the minimization of the tardiness and the costs for location and capacity decisions in a logistics network.
Alumur et al. [10] proposed a new multi-period, multi commodity reverse logistics model and used a reverse bill of products to receive component commonality.

There are other studies on reverse logistics, that are limited to specific applications,

such as carpet recycling by Louwers et al. [120] and Realff et al. [178], battery recycling by Schultmann et al. [191] as well as Kannan et al. [94], tire recycling by Figueiredo and Mayerele [60], paper recycling by Pati et al. [167], plastic recycling by Huysman et al. [86], bottle recycling by Shen et al. [193], sand recycling by Listes and Dekker [116]. Notable work with a remanufacturing focus was presented by Krikke et al. [102] on copiers and Srivastava [195] on appliances and personal computers. Currently, no general model for reverse logistics exists.

2.2.3 Integrated logistics network

In recent years, some researches started to integrate forward and reverse networks to close products cycles.
Lee and Dong [108] proposed an MILP model, which is capable of managing the forward and reverse flows at the same time for end-of-lease computer products.
Ko and Evans [101] developed a MINLP model considering 3-PLS service providers to design an integrated forward/reverse logistics network.
Fleischmann et al. [62] analyzed two cases of photocopier remanufacturing and paper recycling and proved that optimizing the forward and return network simultaneously is more efficient than a sequential design of both networks at the point of cost saving.
El-Sayed et al. [53] gave a mathematical model to solve an integrated forward-reverse logistics network design under statistic demand. Experimental work from the proposed model illustrated that there is a direct relationship between the total expected profit and demand mean and return ratio for a given capacity of the network.
Lu and Bostel [121] designed a two-level location problem as a MILP model with three types of facility (producers, remanufacturing centers, and intermediate centers). This model considers forward and reverse flows and their interactions simultaneously. The focus of this research was on remanufacturing to reduce costs of production and raw materials.
Pishvaee et al. [171] focused on an MILP model to integrate reverse logistics activities into the forward supply chain. To deal with uncertainty, they presented a scenario-based stochastic optimization model to minimise the total cost including fixed opening costs, transportation cost, processing costs, and penalties for non-utilised capacities.
Salema et al. [187] focused on designing and planning supply chains with reverse distribution. In this study, the authors focused on the strategic and tactical levels of the supply chain. Tactical decisions are summarized in production, storage, and distribution planning and a network design is considered for strategic decisions.

Pishvaee et al. [172] proposed a linear multi-objective programming model to improve the total cost as well as responsiveness of the integrated forward/reverse logistics network. Within this work, they allowed for a hybrid distribution-collection facility.

The overview of the literature on logistics network design is presented in Table 2.1. The characteristics of the logistics network proposed in this dissertation is separated in Table 2.1 to show the similarities and differences compared to previous studies.
The proposed integrated forward/reverse logistics network applied in this study is faced with an additional characteristic that makes the problem more complicated. This additional feature is about flexibility in delivery path for forward distribution and presented below.

2.2.4 Flexibility in delivery path

Customer satisfaction has an important role in the success of a business. In this regard, companies are trying to increase customer satisfaction that requires supply chain responsiveness [172; 98]. Fast and on time delivery is one of the main requirement of each consumer. While all logistics networks aim to minimize the total cost, the purpose of reducing delivery time often conflicts with the goal of decreasing logistics costs [225]. To deal with the issue of cost efficiency and network responsiveness simultaneously, researchers have proposed models to optimize both in the supply chain network [172; 9]. However, results are typically limited to shipments between consecutive stages or just indirect shipment mechanisms [172; 9; 51]. Lin et al. [113], formulated a MILP model by including direct shipment and direct delivery as well as inventory control for a three stages forward logistics network. Pishvaee and Rabbani [173] studied the responsiveness of a three stage forward logistics network when (1) direct shipment between plant to customer is allowed and (2) direct shipment is forbidden. They proposed two mixed integer programming models for these conditions and proved that both of these problems can be modeled by a bipartite graph.

Based on the above review, extending the following restrictions can be considered a potential field of research:

- Flexibility in delivery paths as a measure to shorten the delivery time is typically ignored for forward logistics network and completely ignored for integrated forward/reverse logistics networks.
- The total number of echelons in most of the developed integrated forward/reverse

logistics network models is not more than five echelons.

- It is still a critical need to develop an efficient solution to cope with NP-hard problems as well as a general model to be applicable to a wide range of industries.

Reverse distribution has received growing attention throughout this decade due to governmental legislation on environmental protection as well as resource shortages [59; 155]. Therefore, management of product return flows is becoming an essential part of each supply chain. However, considering reverse distribution through a separated design, resulted in sub-optimal solution, [62]. To avoid this, the integrated design is suggested [108; 210].

According to the above reviewed literature, various facility location models based on mixed integer programming (MIP) were considered for determining maximal profit, optimal number, and the capacity of facilities as well as the optimal flow between them.

To fill the presented gaps and mentioned requirements, the problem addressed in this work as a flexible integrated logistics network includes integrated design of forward and reverse logistics as well as flexibility in delivery paths for a closed-loop supply chain network with seven echelons. Figure 2.1 illustrates the framework of this research graphically. The model aims to minimize total cost by finding the optimal number and capacity of facilities as well as flow between them.

Due to the complexity of the proposed network, it is essential to develop an efficient solution methodology.

2.3 Solution method

Generally speaking, network design problems belong to category of NP-hard problems [52; 75; 74; 93] and therefore, traditional methods can not be appropriate to be applied to them. There are three main options to cope with NP-hard problems: probabilistic algorithms, approximation algorithms, and meta-heuristic algorithms. Approximation algorithms are effective algorithms that are capable of finding good results close to optimal solution. By applying this kind of algorithms, distance from the results to the optimum must be known. Since upper bounds and lower bounds are difficult to calculate in the proposed problem, approximation algorithms are not appropriate. On the other hand, probabilistic algorithms guarantee finding a solution close to the optimum with high

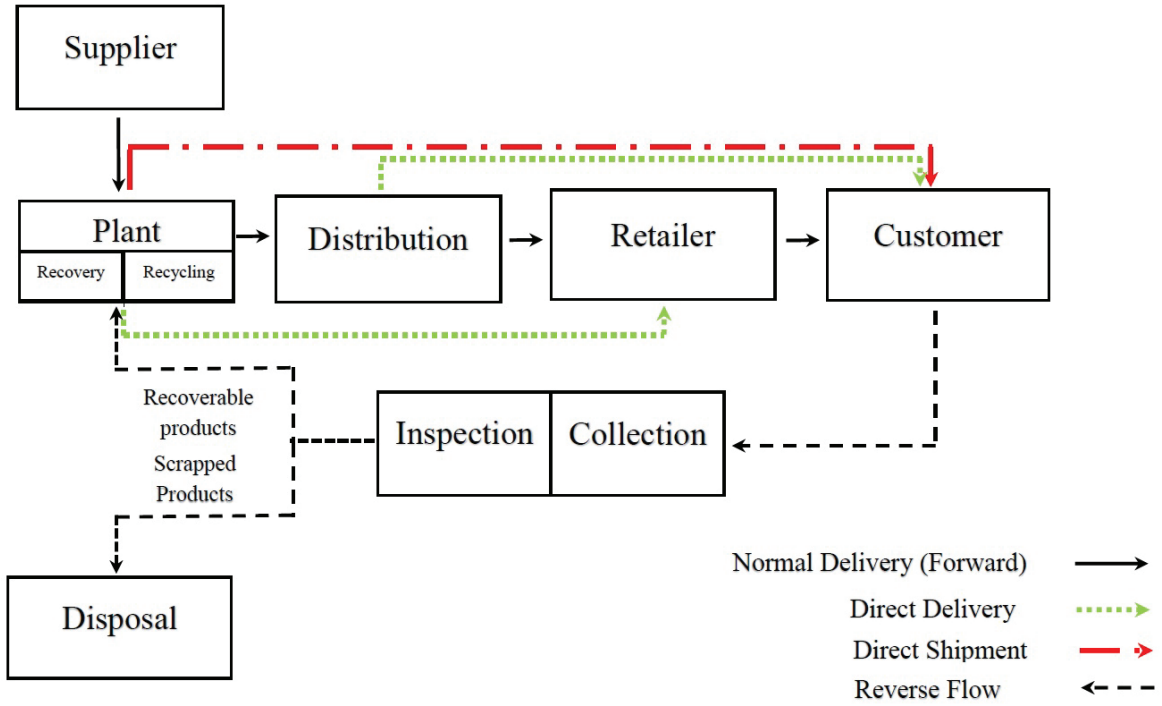


Figure 2.1: Framework of the flexible integrated forward/reverse logistics network.

probability. They are recommended as a solution methodology for NP-hard problems when parameters are not exact. Since meta-heuristic algorithms can reduce the search space and increase the solution quality, they got researcher's attention. Meta-heuristics are divided into two categories including "single-point based algorithms" such as Tabu Search (TS), Simulated Annealing (SA) and Hill Climbing (HC), and "population based algorithms" such as the Genetic Algorithm (GA), Ant Colony Algorithm (ACA) and Particle Swarm Algorithm (PSA). In single-based meta-heuristics, the algorithm is started to search in the search space by a single point, while in population-based algorithms, a population of solutions is considered.

To cope with the complexity of network design problems, many solution methodologies have been presented. Yeh [219] developed an efficient hybrid heuristic, which was enriched by applying three different local search techniques for a mixed integer linear program. Altiparmak et al. [9] solved a mixed integer linear program with a genetic algorithm based on a priority-based encoding method. A heuristic solution approach based Lagrangian relaxation was presented by Amiri [11] to cope with a mixed integer linear programming model. Syarif et al. [201] considered a spanning tree-based GA using

Prüfer number representation to solve a mixed integer linear programming model. Some comparison between results obtained by this method and LINDO showed the efficiency of the proposed method. Gen et al. [71] presented a priority-based genetic algorithm (pb-GA) with a new decoding and encoding method for a distribution-planning problem. Also, they introduced a new crossover operator called Weight Mapping Crossover (WMX) and analysed the effect of the latter on the computational performance. They showed the efficiency of the proposed method with regards to solution quality and computing time in comparison to two different GA approaches. Jayaraman et al. [92] presented a new heuristic solution method for a mixed integer linear programming. The algorithm has three components: random selection of potential collection and refurbishing sites, heuristic concentration, and heuristic expansion. In another study, Min et al. [141] designed a mixed integer non-linear programming model. To solve the mathematical model a GA is adapted. Du and Evans [51] considered a hybrid-scatter search method to solve a bi-objectives reverse logistics network. Ko and Evans [101] proposed a GA-based heuristic to solve an NP-hard problem presented as a mixed integer non-linear programming. Lu and Bostel [121] developed a location problem as a mixed integer linear programming. The model was solved using Lagrangian heuristics, which requires lower and upper bounds of the objective function. Olivares-Benitez et al. [159] formulated by a bi-objective mixed integer linear model for a two-echelon, single-product supply chain design network. For the solution methodology, a meta-heuristic algorithm based on greedy functions, scatter search, and path relinking was considered. A heuristic aimed at finding order quantities in a supply chain problem is presented by Petrovic et al. [169]. Fuzzy sets are applied in order to simulate uncertain demands and delivery due-dates. Performance of supply chains based on uncertain parameters was investigated using a special simulating tool. A supply chain network with the aim of optimizing inventory allocation and transportation routing was proposed by Kristianto et al. [104]. A fuzzy shortest path into two-stage programming to find the global optimum solution was the main contribution of this study. Khalifehzadeh et al. [97] solved a four-echelon supply chain network with shortage problems using a comparative particle swarm optimization algorithm. A Genetic Algorithm provided by Bahrampour et al. [15] as a solution methodology to study a three-stage, multi-product supply chain network model. Pishvaee et al. [172] proposed a solution algorithm based on a new dynamic search strategy using three different local searches to find the set of non-dominated solutions. The authors compared their pareto-optimal solutions to recent GA results. Pishvaee and Rabbani [173] worked on a forward logistics network. To tackle this NP-hard problem, a novel heuristic solution method was considered based on a new chromosome representation derived from graph theory. Al-

though, typically, larger models are required to represent real supply chains, researchers developed many heuristics [219; 92; 51; 11] and meta-heuristics, such as Genetic Algorithm [75; 94; 9; 74; 93; 140; 212; 49], Simulated Annealing [103; 174; 91], Tabu search [108; 200], Memetic Algorithm [172; 148], and Scatter Search [51]. However, there is still a critical need to increase the efficiency of solution approaches in this area, especially, when the complexity of the model increases ([128]).

Recently, Evolutionary Algorithms (EAs), have been successfully applied to solve hard optimization problems [201; 3; 181; 216; 209]. EAs are population-based global search methods based on the mechanism of natural selection in biological systems. They are able to consider many points in the search space simultaneously and multiple searches in different areas of the fitness landscape can be conducted at the same time, while conventional search methods use a single point. This ability provides a high chance of global convergence and renders the approach more powerful than traditional methods. Therefore, EAs present a good potential for global exploration.

Among them, Genetic Algorithm (GA) is recognized as the most well-known class of Evolutionary Algorithms. Genetic Algorithms are stochastic algorithms inspired by Darwin's theory of evolution. Holland, in 1975, introduced a Genetic Algorithm to cope with combinatorial problems for the first time, and since then they have rapidly been implemented as one of the most powerful and efficient stochastic solution search procedures for solving several network design problems [67; 69]. Genetic Algorithms are not dealing with decision variables and they do not need any domain knowledge of the problem. They use a structure to employ genetic information in order to find new search directions. They are based on coding and objective function for evaluating fitness. A population of feasible solutions is generated. Typically, these solutions are in the form of a string or chromosome. A selection strategy is applied to choose parents from the population. Genetic Algorithms are implemented through genetic operators. The main genetic operators observed in nature are reproduction, cross-over, and mutation [82; 69]. Offspring solutions are created with some of the characteristics of each parent based on these genetic operators. These operators (mutation and cross over) are named as the input variables of GA. It should also be noted that the crossover operator is usually applied as the main Genetic operator. It has the main role in the performance of a Genetic Algorithm, while a mutation operator is used as a background operator. Without cross over, all we have is local mutation. This means that, changes will happen slowly and it will be very hard to get our population out of a local optimum. We can make a pretty huge

jump from either of the parents by applying cross over. Mutation operators concentrate on the surroundings of a solution and make random small changes in chromosomes. It is not applied to make a big move in an unknown region but to move slowly. Therefore, there is a very low possibility that a particular random mutation will be useful [99].
Applying a population of solutions in GA allows for searching in multiple directions. On the other hand, lack of enough search intensification is often considered a disadvantage of a pure GA as there is no operator to go deep once a good area in the search space is found. In this regard, Moscato and Norman first introduced the so-called Memetic Algorithm (MA) [149]. The word "meme" is a neologism, introduced by Richard Dawkins for the first time [47]. Memetic Algorithms are hybrid EAs combining local searches with a population-based strategy that is designed to perform global optimization. For this reason, Memetic Algorithms are known under different names, such as "hybrid genetic algorithm", "genetic local search algorithm", "modified genetic algorithm" and "adapted genetic algorithm"[149; 221; 146; 131].

GAs are based on parental features while MAs consider non-parental features as well. A meme in a Memetic Algorithm can be an idea, behavior, activity, concept or style that spreads from person to person. It is considered as a unit of information that reproduces itself while people exchange ideas [47]. Unlike genes, memes are normally modified through individual learning during an individual's lifetime [47] and then passed on to the next generation. Individual learning in this context is usually obtained by local search strategy. For this reason, "Memetic evolution" is a combination of Evolutionary Algorithms with local search strategies that use the concept of meme and gene at the same time. This reveals a capability to balance global and local searches. MAs attempt to explore new regions in the solution space in order to make full use of the exploration (to search the global optimum). Then they try to converge to an optimum solution by exploiting the local search ability. This ability helps the algorithm to improve on the best solution it has found so far by searching nearby the current solution. Hence, [114] indicates that the solutions obtained by Memetic Algorithms are better than those obtained by different solving methods, such as Simulated Annealing (SA), Mixed Variable Evolutionary Programming (MVEP), and Integer Discrete Continuous Programming (IDCNIP). Therefore, the proposed Memetic Algorithm is appropriate for solving the proposed mixed-integer optimization problem. Memetic Algorithm, as stochastic algorithms, are capable of obtaining higher-quality solutions through additional local search space exploitation than Genetic Algorithm [114].

Recently, Memetic Algorithms (MAs) obtained increasing attention from the evolutionary computation community (ECC) and have shown to be promising and effective, with considerable success for solving difficult optimization problems in numerous applications domains [80; 87; 204]. Therefore, Memetic Algorithms have become a popular approach for various engineering optimization problems [127; 6; 204; 110].

The MA typically contains five components [225]:

- A representation of feasible solutions to the problem.
- A way to generate a population (an initial set of feasible potential solutions).
- An evaluation function: Evaluation is association of each solution with a fitness value to see whether it will survive. The fitness of each solution is related to the objective function value [16]. The fitness function to be used depends on the given problem. In this study, the objective function is minimizing the total cost which is including transportation and operation costs.
- Memetic operators, including a reproduction, crossover, and local search. By applying a reproduction operator, solutions are copied through the selection of the more proper solutions. Crossover combines the features of two parent chromosomes (feasible solutions) to create two offsprings by exchanging existing properties of the parents. Local search methods are responsible for improving the solutions to a problem by looking for the best solution in the neighborhood of the current solution. The idea is to create improvement by making some small changes. At each iteration, the local search operator ideally improves the solution obtained by the reproduction and crossover operators [145].
- Parameter values which include "population size", as applicable to the following question, "how many individuals should be in the population", "crossover rate" that was raised with the question, "what is the probability that the individual will crossover?", "number of local search iteration" that was proposed with the question "How many time should the local search be run?" and "number of iteration" that was raised by the question, "How many time should we repeat this procedure?"[118].

Figure 2.2 shows the classifications of solution method clearly. Memetic Algorithms (MAs) have been successfully applied to network design problems [20; 114]. The basic feature of MA is its capacity for multi-directional and global search by generating a

population of solutions as well as local search to improve intensification of the search. In this chapter, we aim to provide a background to MAs and focusing on practical applications thereof.

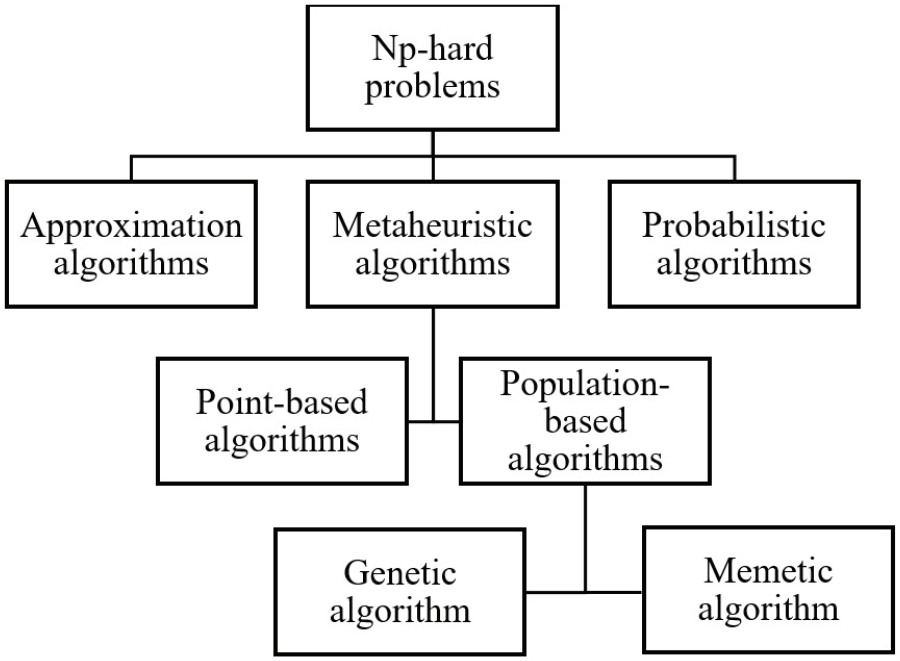


Figure 2.2: Classification of solution method.

2.3.1 Historical background of Memetic Algorithm

Moscato and Norman introduced Memetic Algorithm for the first time in 1989 [148]. The idea was to avoid "biologically constraints" that may restrict progress, as well as to consider cultural evolution. Ten years later, MAs population-based algorithms have become a successful optimization approach, which is applied successively in a variety of problem domains and particularly for NP-hard optimization problems [20; 114].
Traditional population-based Evolutionary Computation (EC) methods, such as Genetic Algorithms (GAs), Differential Evolution (DE), and Particle Swarm Optimization (PSO) try to optimize a problem by generating a population of feasible solutions and creating new candidate solutions by combining existing ones. In contrast, MA intrinsically employs all available knowledge of the problem under study.
The word "Meme" was introduced in the book "The Selfish Gene" [47] by Richard Dawkins, in 1976. "Meme" is an abbreviation of a Greek word, "mimema" (meaning

"something imitated", American Heritage Dictionary, [137]). The role of memes in cultural evolution works like genes in genetic evolution. According to Dawkin's description:

"Examples of memes are tunes, ideas, catch-phrases, clothes fashions, ways of making pots or of building arches. Just as genes propagate themselves in the gene pool by leaping from body to body via sperms or eggs, so memes propagate themselves in the meme pool by leaping from brain to brain via a process which, in the broad sense, can be called imitation."[47]

Building on the statement above, in cultural evolution information is not transmitted unchanged between individuals. The characteristics of a meme shows that communication can process and change information. These changes happen by incorporating heuristics, approximation algorithms, local search techniques, specialized recombination operators, truncated exact methods, etc. Most traditional evolutionary computation can be considered as a search strategy with the aim of optimizing through competition, while most MAs try to optimize by cooperation and competition. The success of MAs can be explained by applying the different search approaches they incorporate to cover diversification as well as intensification.

2.3.2 Basic concepts and definitions of Memetic Algorithm

Before proceeding to the description of an updated MA proposed in this study, some basic concepts and definitions are provided in this section. We extended the definition vocabulary from [149] in this section as well as the next section.
According to [149], *"an algorithm is a detailed step-by-step procedure for solving a computational problem."* A computational problem denotes a class of algorithmically doable tasks, is a problem with an input domain set of instances. For each instance, there is a set of feasible solutions. A feasible solution is a set of values for the decision variables that satisfies all of the constraints in an optimization problem. An algorithm should be capable of solving a problem and giving at least one solution that satisfies the requirements of the problem or, in some cases, indicate that no feasible solution exists.
In optimization, finding a solution that is aimed at minimizing or maximizing a given function is expected where defines as a certain feasible solution either an optimal solution. For each instance, there is a finite set of solutions. Each instance from this set reveals a value obtained by a given objective function. Also, this objective function is responsible for preferring one or more solutions between several feasible solutions, when

necessary.
The solution selected from a not empty finite set of solutions is called "best solution" when no other solution can be found that improves the value of the objective function. In the optimization, the best solution is the one, that minimizes or maximizes the objective function. In this regard, to apply a search algorithm, we consider a set as a search space for a combinational problem. In optimization problems at least one optimal solution from a finite set of solutions is represented by one element in search space. The role of the search space is to define a region including a set of possible solutions for an optimization problem, where the search algorithm will work. In a few cases, some configurations in the search space may results in an infeasible solution. In this situation, the algorithm needs to be prepared to deal with this issue. If these capabilities are present, we have a valid representation of the problem.

2.3.3 Designing a Memetic Algorithm

It is possible to design a general scheme for a Memetic Algorithm. A Memetic Algorithm is a population-based algorithm and therefore the first step is to generate an initial, feasible population. This population can be provided randomly or through an efficient mechanism according to the problem. A population-based algorithm maintains an entire set of candidate solutions, each solution corresponding to a unique point in the search space of the problem. It can be considered as a procedure of modifying current candidates in order to have a set of best candidates. To this end, in each iteration, a new population is produced using the old population. Old population is defined as the current population of the previous iteration. Reproduction take place according to the different mechanisms used in the population algorithm. The newly selected configuration turns into the current generation in the next step. There are two strategies for this selection: "plus strategy" and "comma strategy". In the plus strategy, the next population is selected from old population and new population, while in comma strategy the next population is selected just from new population. However, typically, keeping a fraction of old population and new generation will results in a better configuration as it can cover more area of the search space.

The algorithm is controlled by a fitness value related to the optimization problem. This procedure is repeated till a certain termination condition is met. Traditionally, criteria are a pre-specified number of iterations. However, applying a number of iteration which no further improvement in the last i iterations have not been found, is more efficient.

Each population-based search algorithm involves the recombination operators that are engaged by two aspects: transmit the parental features or non-parental features. Also, some operators can focus on both of these aspects at a same time.

Crossover, as one the basic operators, is responsible for transmitting the pure parental feature to off-springs. This operator is designed to prevent extremely fast convergence to a suboptimal solution but not go to deep to modify the current sub-optimal solution in order to have more intensification.

Locality is one of the key issues of local searches and it is about navigating the search space by iteratively stepping from one solution to one of its neighbours. The quality of each neighbour is assessed by the objective function, which is problem dependent. The most important issue that needs to be considered when selecting the class of changes to be applied for local search methods is the ability of the proposed method to find a sequence of changes that can achieve all other configurations in a neighborhood without skipping. The aim of each local search method is to detect a local optimum, which is the best of all present neighbors within the scale of fitness value. The whole process is repeated until a termination condition is satisfied. The purpose of local searches are to transmit new features from the neighbourhood. In this regard, "mutation based" local search mechanisms are applied to NP-hard problems and have obtained good quality solutions [55]. In MAs, it is very common to apply local searches right after inserting a cross over operator.
Selecting the particular type of operators is dependent on the characteristics of the problem as well as the representation method. No general advice is provided for it since some limitations are included in any cases [149].

2.3.4 Application of Memetic Algorithm

In this section, we focus on the application of Memetic Algorithms to show how this algorithm widely used throughout different areas. As Memetic Algorithms have been successfully employed to several facets of real world problems, we split the application into four different categories provided in Table 2.2 to Table 2.5. It should be noted that many researchers applied some algorithms similar to Memetic Algorithms but named them differently. Other names used such as hybrid Genetic algorithm, modified Genetic algorithm, adapted Genetic algorithm or Genetic local search algorithm in the literature

can be considered as a Memetic Algorithm.
More applications include, but are not limited to: medicine [77], economics [162], oceanography [152], mathematics [215], imaging science and speed processing [183]. New applications are being developed constantly.

A supply chain is usually represented by a network that is called a supply chain network. It contains some nodes that are connected by some arcs to each other. In recent years many studies have been done to design different supply chain networks shown in Table 2.3. These are NP-hard problems and by adding more features such as flexibility in delivery paths and integration in design forward and reverse flow, they get even more complex, with a larger search space. On the other hand, with the increase of the problem scale, the traditional techniques are facing the challenge to effectively and efficiently solve these complicated network design problems in acceptable CPU time. In this regard, developing a non-traditional solution methodology is needed in the area of network design problems.

Memetic Algorithms are rarely employed as the solution methodology for supply chain network problems since they are a very recent form of meta-heuristics, but lately they have been shown to be promising and effective with considerable success in network design problems [65; 219; 172; 158].

One application of Memetic Algorithms in a supply chain network was presented by Pishvaee et al. [172] and aimed to minimize total cost and maximize responsiveness of the network. A Memetic Algorithm based on a new dynamic search strategy was applied to find the set of non-dominated solution.
A mixed integer linear programming (MILP) is proposed by Lee and Dong [108] for an integrated forward/reverse design of a supply chain network for a specific product. A Tabu search-based Memetic Algorithm is performed to solve the problem.
Another study by Jamshidi et al. [88] looked at an NP-hard multi-objective supply chain problem. The aim of the research was minimizing the total cost including transportation, holding, and backorder costs. They utilized a Memetic Algorithm with a novel decoding method and priority-based algorithm for the coding of the solution chromosomes.
Yeh [220] worked on a multi-stage supply chain network problem. The main purpose of the study was to develop an efficient and effective algorithm to find a near optimal solution. A Memetic Algorithm was applied to find the strategy that can reveal the lowest cost of the physical distribution flow.

A mixed integer non-linear production-distribution planning model is discussed by Fahimnia et al. [56] using a Memetic Algorithm. The real data of an automotive company is considered as a real world case study for this research.

This dissertation is focused on a seven stages integrated forward/reverse supply chain network with flexible delivery paths. The aim of this study is minimizing the total cost of the network which is including transportation and operation cost. We use a Memetic Algorithm with a novelty in decoding method and search strategy for the solution methodology. From the computational point of view, we incorporate the graph structure in the chromosome representation, thereby avoiding different model and solution methodologies as, e.g., considered by Pishvaee and Rabbani [172].

2.4 Summary and conclusion

The content of this chapter is contained in two main segments. First, we were dealing with finding an efficient assignment strategy to satisfy all requirements from source and destination by the aim of minimizing total cost. These requirements include increasing customer satisfaction, reducing network cost and delivery time. Customer satisfaction creates a positive effect on any organization's profitability. Satisfied customers form the foundation of any successful business as it leads to repeat purchase, brand loyalty, and positive word of mouth. In this regard, except normal delivery that connects each stage to another one, two additional delivery paths were applied. This full delivery graph in forward flow allows us to solve the conflicting goals profit and responsiveness, which otherwise may lead to greater cost ([172]). On the other hand, competition and marketing motives, economic impact, and concerns with the environment forced companies to accept the concept of reverse logistics. It was observed that integrated forward and reverse networks reveal better results compared to separated designs. This work addresses the issue of flexible integrated, multi-stage, forward/reverse logistics network designs that aims to minimize the total cost of opening facilities and transportation, as well as optimal capacities of facilities and assigning product flows between them in the proposed network. The presented network is not a case-based network and because of its generic features, it can support other industries such as electronic and digital equipment industries and vehicle industries. But network design problems are NP-hard and adding flexibility in delivery path and integration design makes the problem more complex and search space even larger. Developing an efficient solution methods is still a critical need in this area. In the second segment we focused on the available options to tackle these NP-hard

problems. Meta-heuristic algorithms were selected as the proper solution methodology as they reduce the search space and increase the quality of solutions. After that we need to know "which meta-heuristic algorithm may be suitable for the proposed flexible integrated network". Genetic Algorithms (GAs) had been applied for many years as the most successful algorithms in the area of network design problems. But lack of enough intensification was always the main weakness of GAs. Memetic Algorithms were introduced to cover this weakness while keeping the benefits of GAs. In this step we need a mathematical model of the problem to understand the system behavior and ability. This model must be able to model the problem as a system in total without splitting it to two part: forward and reverse. Also it should be capable to consider three different delivery paths. Therefore the next question we are facing is "How can we model the problem mathematically?"

Table 2.1: Reviewed articles in logistics network design

Reference	type of networks(s)	Modeling	Objective	Solution method	Network organization	Other Characteristics
Yeh [219]	Forward	MILP	Min cost	Hybrid heuristic	Supplier production distribution	-
Altiparmark et al. [9]	Forward	MINLP	Min cost max customer service min capacity utilization	GA	Supplier production distribution	Multi objective supply chain network
Amiri [11]	Forward	MILP	Min cost	LR-based heuristic	Production distribution	Multi capacity Network
Syarif et al. [201]	Forward	MILP	Min cost	GA	Supplier production distribution	-
Tragntalerngsak et al. [206]	Forward	MILP	Min cost	LR-based branch and bound algorithm	Production distribution customer	-
Gen et al. [71]	Forward	MILP	Min cost	GA	Production distribution	-
Hinojosa et al. [84]	Forward	MILP	Min cost	LR-based heuristic	Plant warehouses	Multi commodity multi period
Lin et al. [113]	Forward	MILP	Min cost	Hybrid evolutionary algorithm	Plant distribution retailer	Inventory control different delivery paths
Krikke et al. [102]	Reverse	MILP	Min cost	Exact method	Collection-inspection recovery distribution	-
Üster et al. [208]	Reverse	MILP	Min cost	Exact method	Collection-inspection recovery production distribution	Multi products
Listes and Dekker [116]	Reverse	SMIP	Max profit	Exact method	Collection-inspection recycling	Uncertainties in demand and returned products
Jayaraman et all. [92]	Reverse	MILP	Min cost	Heuristic	Collection-inspection recovery	-
Salema et al. [188]	Reverse	SMIP	Min cost	Exact method	Factories warehouses disassembly	Uncertainties in demand and returned products-multi products
Aras et al. [12]	Reverse	MINLP	Max profit	Tabu search based heuristics	Collection-inspection recovery	Considering buying price of used products
Du and Evans [51]	Reverse	MILP	Min tardiness Min cost	Hybrid-scatter search	Recovery production distribution	Multi objective supply chain network
Alumur et al. [10]	Reverse	MILP	Max profit	Exact method	Inspection disassembly recycling remanufacturing refurbishing	Multi period multi commodity considering the reverse-bill of products
Min et al. [141]	Reverse	MINLP	Min cost	GA	Collection-inspection recovery	Multiple time network
Lu and Bostel [121]	Integrate	MILP	Min cost	Lagrangian relaxation based heuristic	Collection-inspection recovery disposal production distribution	
Lee and Dong [108]	Integrate	MILP	Min cost	Tabu search based heuristics	Collection-inspection recovery production distribution	-
Ku and Events [101]	Integrate	MINLP	Min cost	GA	Recovery production distribution	-
Fleischmann et al. [62]	Integrate	MILP	Min cost	Exact method	Collection-inspection recovery production distribution	Exact method -
El-Sayed et al. [53]	Integrate	SMILP	Max profit	Exact method	Supplier distribution disassembly redistribution	Uncertainties in demand and returned products-multi echelon multi period
Pishvaee et al. [172]	Integrate	MILP	Max cost max responsiveness	MA	Collection-inspection recovery distribution disposal production	Multi objective supply chain network
Pishvaee et al. [171]	Integrate	SMILP	Min cost	Exact method	Production distribution collection recovery disposal	Uncertainties in demand and returned products
This thesis	Integrate	MILP	Min cost	MA	Supplier plant distribution retailer collection-inspection disposal	Flexible delivery path

Table 2.2: Application of Memetic Algorithm

Classical NP optimization problem	
Problem	references
Bin packing	[179; 130; 166; 192]
Graph partitioning	[27; 28; 43; 134]
Max independent set	[4; 83; 186]
Min generalized assignment	[41; 119]
Min graph coloring	[44; 63; 122]
Min number partitioning	[43]
Min travelling salesman	[73; 43]
Multidimensional Knapsack	[42; 180; 176; 184]
Non-linear integer programming	[203; 207]
Parallel machine scheduling	[40; 142; 129]
Quadratic assignment	[26; 131; 43; 133]
Set covering	[17; 89]
Set partitioning	[109]
Single machine scheduling	[107; 139; 64; 96; 124]

Table 2.3: Application of Memetic Algorithm

Other combinational optimization problem	
Problem	references
Degree-constrained minimum spanning tree problem	[199; 36]
Flowshop scheduling	[150; 151; 35; 164]
Frequency allocation	[95]
Kauffman NK landscapes	[132]
Maintenance scheduling	[29; 30; 31]
Network design	[65; 219; 172; 158]
Open shop scheduling	[39; 58; 111; 54]
Partial shape matching	[163]
Placement problems	[196; 85; 105; 190]
Production planning	[48; 143]
Project scheduling	[156; 177; 7; 106]
Rostering	[147; 34]
Task allocation	[78; 117]
Timetabling	[90; 146; 32; 33]
Transportation problems	[68; 157; 224]
Uncapacitated hub location	[1; 125]
Vehicle routing	[223; 154; 144]
Warehouse scheduling	[214]

Table 2.4: Application of Memetic Algorithm

Machine learning and robotics	
Problem	references
Analysis of time series	[160]
Manipulator motion planning	[165]
Neural network training	[197]
Path planning	[226]
Pattern classification	[138]
Pattern recognition	[5]
Time optimal control	[37]

Table 2.5: Application of Memetic Algorithm

Electronics and engineering	
Problem	references
Aeronautic design	[25]
Analogue network synthesis	[76]
Circuit design	[123]
Computer aided design	[18]
Power planning	[218]
Semiconductor manufacturing	[100]
Service restoration	[14]
Structure optimization	[217]
System modeling	[213]
Traffic control	[168]
Trim loss minimization	[161]

Chapter 3

Modeling of the integrated, flexible logistics network

3.1 Introduction

To manage a logistic system efficiently is a key issue for many companies as it is connected with cost, delivery time, customer satisfaction, and environmental protection. To achieve these aims, flexible and productive networks based on integration designs are of particular interest. From the published literature [71; 121; 171; 187], it is understood that the integrated forward/reverse logistics problems are different from traditional logistics models. Due to their size and complexity, these problems require more efforts to analyze. Moreover, the significance of transportation costs and customer satisfaction spurs an interest in developing a flexible network design model that makes the problem more complex. This study proposes an integrated logistics network model with three kinds of delivery paths which present a fully capacitated graph in forward flow. Minimizing the total costs, reveals a mixed integer linear program (MILP). Then, the next step will be modeling the presented integrated forward/reverse logistics network. Therefore, this chapter presents a comprehensive mathematical model of an integrated, flexible logistics network. This model will represent our problem as a complete system without splitting into forward and reverse parts and will include three different delivery paths which allow us to skip some stages.

The initial version of the proposed model, presented in this chapter, have been used for a published paper in the 7th IFAC Conference on Management and Control of Production and Logistics, Bremen, Germany, (E. Behmanesh and J. Pannek 2016), [20] as

well as the Logistics Research journal, Springer, (E. Behmanesh and J. Pannek 2016), [19].

3.2 Problem definition

In this work the integrated design of forward/reverse flows is considered to avoid sub-optimal solutions derived from separated designs. As illustrated in Figure 3.1, in the forward flow, new products are shipped from plant to customer through distribution and retailer centers in a pull manner to meet the demand. Customer locations are assumed to be predetermined and fixed. Also, three kinds of shipments are used in the proposed network to enhance the logistic network's efficiency and flexibility, including:

Normal Delivery: Products are transported from one echelon to another.

Direct Shipment: Products are transported from plants to customers directly.

Direct Delivery: Products are transported from distribution centers to customers or from plants to retailers directly.

The three different types of arrows and colors considered in Figure 3.1 show the differences of these delivery paths graphically. They cover all possible alternatives for the proposed model in forward flow, that is given as a fully capacitated graph from plant to customer.
In the reverse flow, returned products are collected by collection/inspection centers and, after inspection, the recoverable products are shipped to the recovery facilities, and scrapped products are shipped to disposal centers in a push manner for safe disposal.
The objective function is to minimize the total cost consisting of transportation and processing costs. We assume that the numbers of suppliers and their capacities as well as number of customers and their demands are known. The numbers of potential plants, distribution centers, retailers, collection/inspection centers and disposal centers as well as their maximum capacities are known in advanced. Additionally, the return rate, recovery rate and disposal rate are also known. The problem is aiming to select the subset of plants, distribution centers, retailers collection/inspection centers and disposal centers to be opened as well as to configure the distribution network such that it satisfies all capacities and demand requirements and minimizes the total cost.

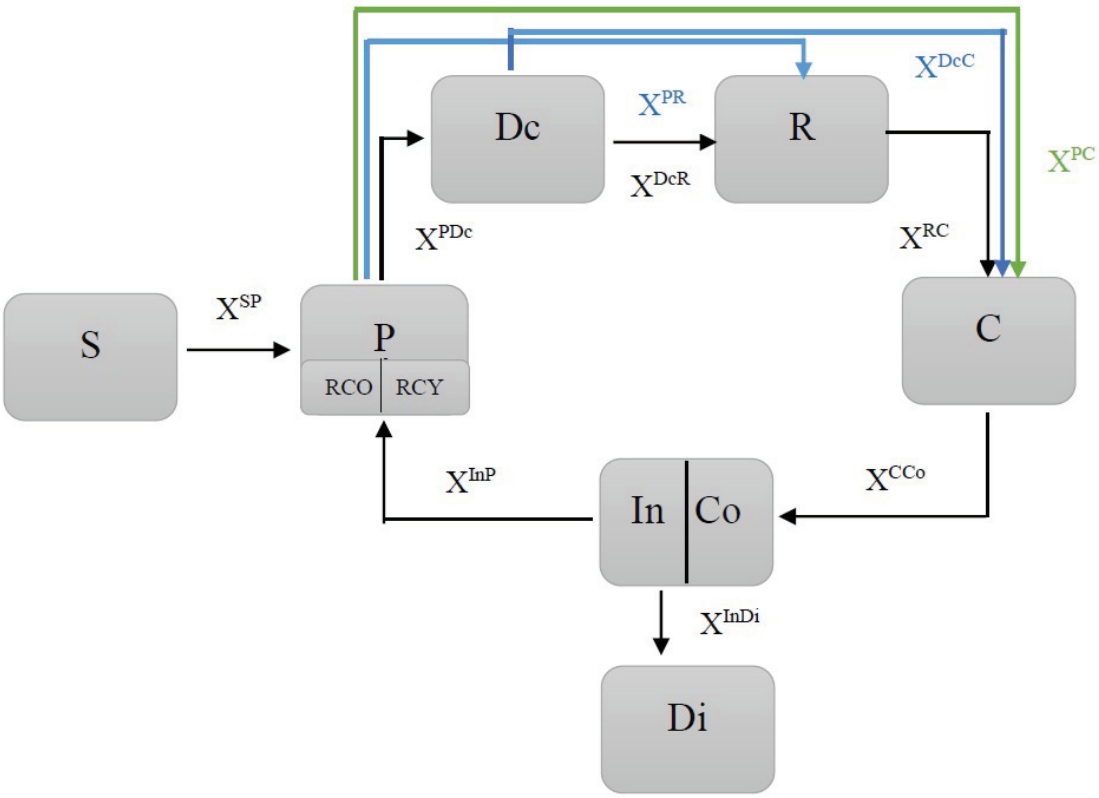


Figure 3.1: Underlying structure of MILP.

3.3 Model generalization

To support the presentation of the proposed mathematical model, we consider the general model of our problem. To this end, we consider $G = (N, E)$ to be a digraph where $N = \{1, 2, ..., n\}$ denotes the set of all nodes and $E = \{(i, j)|i, j \in N\}$ the set of all edges in the closed-loop network. The cost for node $i \in N$ are denoted by c_i, and the unit transportation cost on edge $(i, j) \in E$ are given by c_{ij}. The respective decision variables $y_i \in \{0, 1\}$ and $x_{ij} \in \mathbb{N}_0$ represent whether a stage $i \in N$ is used and which quantity is transported between node i and j. To determine the optimal distribution network and capacity of each node, we minimize the transportation and operation cost of the proposed

network, which reveals the following mixed integer minimization problem:

$$\begin{aligned} \min_{x_{ij},y_i} & \sum_{(i,j)\in E} c_{ij}x_{ij} + \sum_{i\in N} c_i y_i \\ \text{s.t.} & \sum_{(i,j)\in E} a_i x_{ij} \leqslant b_i y_i \\ & x_{ij} \geq 0,\ y_i \in \{0,1\} \end{aligned} \tag{3.1}$$

Next, we specialize this model to reflect the problem properties.

3.4 Assumption of the proposed model

The previously described flexible forward/reverse logistics network setting represents an integrated supply chain with seven echelons consisting of suppliers S, plants P, distribution centers Dc, retailers, R and customers C in forward flow, as well as collection/inspection centers Co and disposal centers Di in reverse flow, cf. Figure 3.1 for a schematic sketch. We like to point out that in accordance with Figure 3.1, we consider a hybrid manufacturing-recovery-recycling facility as well as a hybrid collection-inspection facility. Establishing several facilities at the same location can decrease the price of the whole network in comparison with separated design [108; 172].

To adapt problem (3.1), we impose the following assumptions:

- There are seven echelons: suppliers, plants, distribution centers, retailers, customers, collection/inspection centers, and disposal centers.
- The set of nodes is given by $N = S \cup P \cup Dc \cup R \cup C \cup Co \cup Di$.
- There are no edges between facilities of the same stage, the delivery graph is complete in forward flow and the return graph is simple, i.e. $E = (S \times P) \cup (P \times Dc) \cup (P \times R) \cup (P \times C) \cup (Dc \times R) \cup (Dc \times C) \cup (R \times C) \cup (C \times Co) \cup (Co \times Di) \cup (Co \times P)$.
- The demands of each customer are deterministic and must be satisfied.
- The number of facilities per stage and respective capacities are limited.
- All cost parameters (fixed and variable) are known in advance.

- The transportation rates are perfect and there are no storages. Moreover, the return rate p_m^{C} as well as the disposal and recovery rates p_n^{Co} and $(1 - p_n^{\mathrm{Co}})$ are fixed. All returned products from each customer must be collected.
- The inspection cost per item for the returned products are included in the collection cost.
- The un-recyclable returned products will be sent to the disposal center. The remaining products are returned to the original plant.
- The required recycled materials are assumed to be of the same quality as the raw materials bought from suppliers and any plant chooses the raw material from the collection/inspection center over suppliers.
- Customers have no special preference. It means, price is the same in all facilities including: plants, distribution centers, and retailers.

3.5 Notation

To support the presentation of the proposed mathematical model, we first provide a verbal description of the model as follows:

$$\text{Cost} = \text{transportation costs} + \text{Fixed opening cost.} \tag{3.2}$$

The notation given in Tables 3.1-3.4 are used in the formulation of the MILP model.

Table 3.1: Model indicates

Indices	
i	Index of Supplier (i=1,2,...,I)
j	Index of Plant (j=1,2,...,J)
k	Index of Distribution Center (k=1,2,...,K)
l	Index of Retailer (l=1,2,...,L)
m	Index of Customer (m=1,2,...,M)
n	Index of Collection/Inspection Center (n=1,2,...,N)
o	Index of Disposal Center (o=1,2,...,O)

Table 3.2: Model parameters

Parameters	
I	Number of Suppliers
J	Number of Plants
K	Number of Distribution Centers
L	Number of Retailers
M	Number of Customers
N	Number of Collection/Inspection Centers
O	Number of Disposal Centers
$\overline{S}_i$	Bound of products supplied by Supplier i
$\overline{P}_j$	Bound of products produced by Plant j
$\overline{Dc}_k$	Capacity of Distribution Center k
$\overline{R}_l$	Capacity of Retailer l
D_m	Demand of Customer m
$\overline{Co}_n$	Capacity of Collection/Inspection Center n
$\overline{Di}_o$	Capacity of Disposal Center o
p_m^C	Recovery percentage of Customer m
P_n^{Co}	Disposal percentage of Collection/Inspection Center n

Table 3.3: Model parameters

Parameters	
c_{ij}^{SP}	Unit cost of transportation from Supplier to Plant
c_{jk}^{PDc}	Unit cost of transportation from Plant to Distribution Center
c_{kl}^{DcR}	Unit cost of transportation from Distribution Center to Retailer
c_{lm}^{RC}	Unit cost of transportation from Retailer to Customer
c_{mn}^{CCo}	Unit cost of transportation from Customer to Collection/Inspection Center
c_{no}^{CoDi}	Unit cost of transportation from Collection/Inspection Center to Disposal Center
c_{nj}^{CoP}	Unit cost of transportation from Collection/ Inspection Center to Plant
c_{jl}^{PR}	Unit cost of transportation from Plant to Retailer
c_{km}^{DcC}	Unit cost of transportation from Distribution Center to Customer
c_{jm}^{PC}	Unit cost of transportation from Plant to Customer
c_{j}^{P}	Fixed cost of operating Plant j
c_{k}^{Dc}	Fixed cost of operating Distribution Center k
c_{l}^{R}	Fixed cost of operating Retailer l
c_{n}^{Co}	Fixed cost of operating Collection/Inspection Center n
c_{n}^{Di}	Fixed cost of operating Disposal Center o

Table 3.4: Model variables

Integer	
X_{ij}^{SP}	Amount of products transported from Supplier i to Plant j
X_{jk}^{PDc}	Amount of products transported from Plant j to Distribution Center k
X_{kl}^{DcR}	Amount of products transported from Distribution Center k to Retailer l
X_{lm}^{RC}	Amount of products transported from Retailer l to Customer m
X_{mn}^{CCo}	Amount of products transported from Customer m to Coll/Ins Center n
X_{no}^{CoDi}	Amount of products transported from Collection/Inspection Center n to Disposal Center o
X_{nj}^{CoP}	Amount of products transported from Collection/Inspection Center n to Plant j
X_{jl}^{PR}	Amount of products transported from Plant j to Retailer l
X_{km}^{DcC}	Amount of products transported from Distribution Center k to Customer m
X_{jm}^{PC}	Amount of products transported from Plant j to Customer m

Binary	
X_j^P	$\begin{cases} 1 & \text{if Plant j is activated for production and remanufacturing} \\ 0 & \text{otherwise} \end{cases}$
X_k^{Dc}	$\begin{cases} 1 & \text{if Distribution Center k is activated for distribution} \\ 0 & \text{otherwise} \end{cases}$
X_l^R	$\begin{cases} 1 & \text{if Retailer l is activated for distribution} \\ 0 & \text{otherwise} \end{cases}$
X_n^{Co}	$\begin{cases} 1 & \text{if Collection/Inspection Center n is activated for collectiong/ inspecting} \\ 0 & \text{otherwise} \end{cases}$
X_o^{Di}	$\begin{cases} 1 & \text{if Disposal Center o is activated for safe disposal} \\ 0 & \text{otherwise} \end{cases}$

3.6 Mathematical formulation

Utilizing the notation of Tables 3.1-3.4, the structure of the MILP model can be presented as follows. The objective of this model is to minimize the total cost of the proposed supply chain. The first part of the cost function represents the variable costs, which are given by the direct multiplication of the transportation cost and the quantity transferred from the origin to the destination. The fixed costs of the network is given by the second part of cost function. This reveals the cost function as follows:

$$
\begin{aligned}
min \sum_{i=1}^{I}\sum_{j=1}^{J} c_{ij}^{SP}\cdot X_{ij}^{SP} + \sum_{j=1}^{J}\sum_{k=1}^{K} c_{jk}^{PDc}\cdot X_{jk}^{PDc} + \sum_{k=1}^{K}\sum_{l=1}^{L} c_{kl}^{DcR}\cdot X_{kl}^{DcR} \\
+ \sum_{l=1}^{L}\sum_{m=1}^{M} c_{lm}^{RC}\cdot X_{lm}^{RC} + \sum_{m=1}^{M}\sum_{n=1}^{N} c_{mn}^{CCo}\cdot X_{mn}^{CCo} + \sum_{n=1}^{N}\sum_{o=1}^{O} c_{no}^{CoDi}\cdot X_{no}^{CoDi} \\
+ \sum_{n=1}^{N}\sum_{j=1}^{J} c_{nj}^{CoP}\cdot X_{nj}^{CoP} + \sum_{j=1}^{J}\sum_{l=1}^{L} c_{jl}^{PR}\cdot X_{jl}^{PR} + \sum_{k=1}^{K}\sum_{m=1}^{M} c_{km}^{DcC}\cdot X_{km}^{DcC} \\
+ \sum_{j=1}^{J}\sum_{m=1}^{M} c_{jm}^{PC}\cdot X_{jm}^{PC} + \sum_{j=1}^{J} c_{j}^{P}\cdot X_{j}^{p} + \sum_{k=1}^{K} c_{k}^{Dc}\cdot X_{k}^{Dc} + \sum_{l=1}^{L} c_{l}^{R}\cdot X_{l}^{R} \\
+ \sum_{n=1}^{N} c_{n}^{Co}\cdot X_{n}^{Co} + \sum_{o=1}^{O} c_{o}^{Di}\cdot X_{o}^{Di}. \qquad (3.3)
\end{aligned}
$$

To shorten notation, we omit the bounds on the parameters in the following presentation of the model constraints.

For one, the capacities in each node is limited.

$$\sum_{j=1}^{J} X_{ij}^{SP} \leqslant \overline{S}_i; \quad \forall i \in I \tag{3.4}$$

$$\sum_{k=1}^{K} X_{jk}^{PDc} + \sum_{l=1}^{L} X_{jl}^{PR} + \sum_{m=1}^{M} X_{jm}^{PC} \leqslant X_j^P \cdot \overline{P}_j; \quad \forall j \in J \tag{3.5}$$

$$\sum_{l=1}^{L} X_{kl}^{DcR} + \sum_{m=1}^{M} X_{km}^{DcC} \leqslant X_k^{Dc} \cdot \overline{D}_k; \quad \forall k \in K \tag{3.6}$$

$$\sum_{m=1}^{M} X_{lm}^{RC} \leqslant X_l^R \cdot \overline{R}_l; \quad \forall l \in L \tag{3.7}$$

$$\sum_{j=1}^{J} X_{nj}^{CoP} + \sum_{o=1}^{O} X_{no}^{CoDi} \leqslant X_n^{Co} \cdot \overline{Co}_n; \quad \forall n \in N \tag{3.8}$$

$$\sum_{n=1}^{N} X_{no}^{CoDi} \leqslant X_o^{Di} \cdot \overline{Di}_o; \quad \forall o \in O \tag{3.9}$$

Secondly, the demands of customers have to be satisfied.

$$\sum_{j=1}^{J} X_{jm}^{PC} + \sum_{k=1}^{K} X_{km}^{DcC} + \sum_{l=1}^{L} X_{lm}^{RC} = D_m; \quad \forall m \in M \tag{3.10}$$

Additionally, according to assumption, only a fraction p_m^{C} is returned by customers and a fraction p_n^{Co} of the returned products has to be disposed off. Apart from these dissipativities, the supply chain network is conservative, i.e. in-flow and out-flow in each node

must be identical. These conditions reveal:

$$\sum_{i=1}^{I} X_{ij}^{SP} + \sum_{n=1}^{N} X_{nj}^{CoP} = \sum_{k=1}^{K} X_{jk}^{PDc} + \sum_{l=1}^{L} X_{jl}^{PR} + \sum_{m=1}^{M} X_{jm}^{PC}; \quad \forall j \in J \tag{3.11}$$

$$\sum_{j=1}^{J} X_{jk}^{PDc} = \sum_{l=1}^{L} X_{kl}^{DcR} + \sum_{m=1}^{M} X_{km}^{DcC}; \quad \forall k \in K \tag{3.12}$$

$$\sum_{j=1}^{J} X_{jl}^{PR} + \sum_{k=1}^{K} X_{kl}^{DcR} = \sum_{m=1}^{M} X_{lm}^{RC}; \quad \forall l \in L \tag{3.13}$$

$$P_m^C \cdot (\sum_{j=1}^{J} X_{jm}^{PC} + \sum_{k=1}^{K} X_{km}^{DcC} + \sum_{l=1}^{L} X_{lm}^{RC}) = \sum_{n=1}^{N} X_{mn}^{CCo}; \quad \forall n \in N \tag{3.14}$$

$$\sum_{o=1}^{O} X_{no}^{CoDi} = P_n^{Co} \cdot \sum_{m=1}^{M} X_{mn}^{CCo}; \quad \forall n \in N \tag{3.15}$$

$$\sum_{j=1}^{J} X_{nj}^{CoP} = 1 - P_n^{Co} \cdot \sum_{m=1}^{M} X_{mn}^{CCo}; \quad \forall n \in N \tag{3.16}$$

Finally, we require the decision variables to be non-negative or binary, respectively.

$$X_{ij}^{SP}, X_{jk}^{PDc}, X_{kl}^{DR}, X_{lm}^{RC}, X_{mn}^{CCo}, X_{no}^{CoDi}, X_{nj}^{CoP}, X_{jl}^{PR}, X_{km}^{DcC}, X_{jm}^{PC} \in \mathbb{N} \cup \{0\} \tag{3.17}$$

$$X_j^P, X_k^{Dc}, X_l^R, X_n^{Co}, X_o^{Di} \in \{0,1\}; \forall j \in J, k \in K, l \in L, n \in N, o \in O \tag{3.18}$$

3.7 Summary and conclusion

After designing the network, modeling the network mathematically comes second. This chapter has primarily focused on developing a general model for the presented flexible, integrated, forward/reverse logistics network based on the mixed integer linear programming. Then, assumptions, limitations, and notations were provided to adapt the problem. Afterwards, the mathematical formulation was presented to describe the model mathematically. The main features of the mathematical model are: capability to model the problem as a complete system without splitting it into forward and reverse parts, considering three different delivery paths that reveal four different options for each customer, and finding the best path. It is aimed at minimizing the total cost including transportation and operation cost, and dealing with long- and short-term decisions. Long-term decisions are involved in determining the optimal number and capacity of facilities and

short-term decisions aim at optimizing the distribution network.

As this is an NP-hard problem, and applying a flexible integrated model makes the problem more complex. Although the problem can be formulated into an integer linear program, it is not possible to compute a suitable solution for large size problems in acceptable time. Therefore, finding a proper solution methodology is a critical need in this area.

Chapter 4

Proposed solution approach

4.1 Introduction

So far, we formulated the cyclic seven stage logistics network design. Our model is an NP-hard problem [52; 75; 74; 93] and it can be reformulated into an integer linear program, but we cannot compute a suitable solution for large-scaled problems within an acceptable time. Three main options exist: probabilistic algorithms, approximation algorithms, and meta-heuristic algorithms. Since meta-heuristic algorithms can reduce the search space and increase the solution quality, they are selected as the solution methodology for the proposed network. According to [20; 114], Memetic Algorithms (MAs) are appropriate for this kind of problem and showed great success. The basic feature of MA is a multi-directional and global search by generating a population of solutions as well as local search to improve intensification of the search.
In this chapter we provide an adaptation of a Memetic Algorithm according to the characteristic of the proposed problem. The proposed Memetic Algorithm with a novel chromosome representation including two segments and a neighborhood search mechanism is the suggested solution methodology to deal with the proposed flexible integrated forward/reverse logistics network. The chapter ends by providing an overall procedure for the proposed Memetic Algorithm.

The initial content of this chapter have been published in the 7th IFAC Conference on Management and Control of Production and Logistics, Bremen, Germany,(E. Behmanesh and J. Pannek 2016), [20].

4.2 General formulation

Genetic Algorithms (GAs), as the most famous Evolutionary Algorithms (EAs), have been applied to network design problems for many years [69]. They are one of the most applicable and powerful techniques known to be problem-independent. Their strategy is based on solving the problems through coding space. From this point of view, they are not only techniques or algorithms, they are a kind of art to cope with difficult-to-solve problems in industrial engineering and computer communication network systems. Recent studies showed a great advancement of GAs in solving network design problems [69].

However, the pure GA often suffers from lack of capability for enough search intensification [172]. In other words, GAs are not equipped with a process to get closer to optimal solutions [112]. In this regard, to improve the intensification of the search, Moscato and Norman [149], introduced a Memetic Algorithm (MA) that was enriched with a local search operator. After that, MAs have become famous as population-based heuristic search techniques for optimization problems like GAs, with additional local search engine to refine individuals and make the algorithm stronger [20; 114].

Given the successful application of Memetic Algorithm to network design problems, we develop a Memetic Algorithm to tackle a flexible integrated forward/reverse logistics network problem. This research can be considered as an application of MAs to some difficult-to-solve network design problems. Within this work, we design a Memetic Algorithm to solve a flexible integrated forward/reverse logistics network. According to the reviewed literature and above explanation, two major issues affect the performance of Memetic Algorithm [212], i.e. the chromosome representation and the Memetic operators.

4.2.1 Chromosome representation

Memetic Algorithms are known to be problem-independent, and the chromosome representation is one of critical reasons for this. A chromosome must have the necessary gene information for solving the problem. Selecting a proper chromosome representation greatly affects the performance of meta-heuristic algorithms. Therefore, the first step of applying MA to a specific problem is to decide how to design a chromosome.

The first approach in chromosome representation in GAs was a mapping between potential solutions and binary representations including repair procedures [69]. In this approach it was necessary to adapt the problem into a binary representation. These lim-

itations made the approach insufficient for providing successful applications particularly for complex problems. To overcome these weaknesses, non-binary implementations for solution representations as the second approach have been developed for specific problems. In this approach, adaptation are considered for GAs, instead of the problem. This adaptation contains a modification of the chromosome representation of a potential solution and applying the proper operators. An encoding method is either direct or indirect. In the direct encoding method, the whole solution for the problem is used as a chromosome, while in indirect encoding, only the required part of a solution is used. After that, a decoding procedure is applied to generate solutions according to chromosomes. In the third approach, adaptation algorithms, as well as the problem is considered. In this approach, the algorithm is trying to create a solution based a on combination of some items under consideration [69]. This approach has recently been successfully implemented in the area of industrial engineering and obtained significant success [38]. Within this work, the chromosome is divided into two segments. For the first segment of the chromosome representation, we are using the second approach in a direct way so that computational time can be greatly cut down, and for the second segment, we adapt the third approach to define different delivery paths.

The tree-based representation is known to be one way for representing network problems while using the second approach. Different methods have been developed to encode trees. One of them is matrix-encoding, which was developed by Michalewicz [136]. In this method, the solution is presented by a $|K| \cdot |J|$ matrix where $|K|$ and $|J|$ are the number of sources and depots, respectively. Although this solution approach has a simple representation, applying this method requires the development of a special crossover and mutation operator for obtaining a feasible solution as well as huge amount of memory. Another tree-based representation is the Prüfer number. The use of the Prüfer number representation for solving various network problems was introduced by Gen and Cheng [66]. It requires an array of the length $|K| + |J| - 2$ with $|K|$ sources and $|J|$ depots. Since this method may compute infeasible solutions [93], a repair mechanism has been developed. In this regard, Jo et al. [93] presented the procedure for repairing infeasible chromosomes. Later, Gen et al. [72] introduced determinant encoding using priority that does not need any repair mechanism to guarantee the feasibility of solutions. Solutions are encoded as arrays of size $|K| + |J|$, in which the position of each cell represents the sources and depots and the value in cells represent the priorities.

From the literature [212], we have found that both Prüfer and determinant encoding

are applied for the encoding of the spanning tree problem. However, as the determinant encoding overcomes the bottlenecks of Prüfer encoding [2; 182], such as infeasible representation, complex calculation, and low locality, we utilize determinant encoding in our study. In the following, encoding and decoding are discussed.

4.2.2 Random path-based direct encoding method

Previous researchers in this area presented networks with integrated design or different delivery paths and also suggested different algorithms as solution methods [172; 9; 51]. But they rarely linked these additional features to solution method. In this study we show how the integrated problem can be merged with a Memetic Algorithm using direct encoding. We additionally show how applying the extended random path direct encoding method can capture the complexity of a full graph and reduce the size of the encoding and thereby computational time by developing a two segment approach.
The delivery and recovery path can be conventionally determined by applying the random path direct encoding method introduced by Lin et. al. [67]. In this method the decision variables are directly coded as integer-valued numbers. Using this method computation time can be greatly cut down. The other major advantages obtained through the use of this encoding schema are: simplified and effective representation of search space, shrinking the encoding drastically and possibility to use repeated numbers during applying operators.
One gene in a chromosome is characterized by two factors: locus, the position of the gene within the structure of chromosome, and allele, the value the gene takes. In this method, each gene is initialized with a random value from its domain and it contains M groups where M is the total number of customers. Each group represents a delivery path in forward flow as well as recovery path in reverse flow. Due to existence of three different delivery paths in the proposed problem, we extended the random path-based direct encoding method by adding a second segment into the chromosome. The proposed approach uses a novel encoding schema composed of both binary and integer values.

4.2.3 Extended random path-based direct encoding

Although applying the new delivery paths improves the flexibility and efficiency of the supply chain network, it makes the problem more complex. In Figure 4.1 the representation of the extended random path-based direct encoding method in two segments is shown.

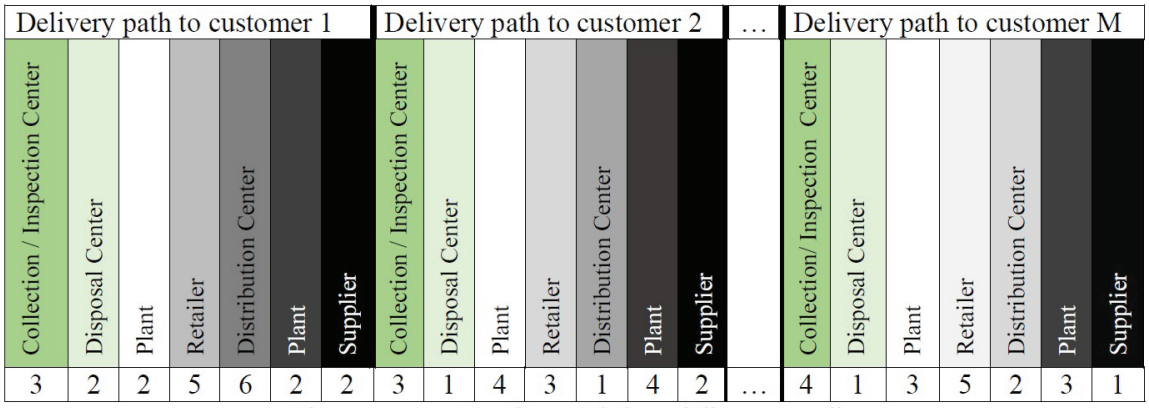


Figure 4.1: Representation of extended random path-based direct encoding method.

The first segment is encoded by using the random path-based direct encoding method that shows the delivery path for each customer. The second segment of the chromosome contains two parts: the first part with J locus including the guide information regarding plant assignments in the network, and the second part of length K containing the information of the Distribution centers. As shown in Figure 4.1, the length of chromosome is $(7*M)+J+K$ where M, J and K are the total number of customers, plants and distribution centers respectively. Each sequence of seven subsequent genes forms a group. Each group encodes four potential delivery paths through plant, distribution center, and retailer to customer as well as a recovery path from customer through collection/inspection to disposal center or plant. The first three alleles of a group represent the reverse flow of the network, while the next four alleles of that group show the forward flow from supplier to customers. As an illustration, a randomly assigned ID to these facilities in the reverse and forward flow is shown in Figure 4.1. Each locus in the second part is assigned an integer in the set $\{0, 2\}$ for plants due to the existence of three delivery options for each plant in the network. Regarding Distribution centers, an integer from $\{0, 1\}$ is chosen to represent the two respective delivery options. The second segment is involved by determining the sort of delivery path for the selected plant as well as distribution center in the first segment.

It should be noted that applying this encoding approach might generate infeasible solutions, which violate the facility capacity constraint, hence a repairing procedure is needed. If the total demand of a depot from a source exceeds its capacity, the depot will be assigned to another source with sufficient product supply so that the transportation cost between that source and the depot is the lowest. The procedure of encoding by extended random path-based direct encoding is shown in Algorithm 1 below.

Algorithm 1 Initialization by extended random path-based direct encoding

Input: Number of customers M
Number of collection/inspection centers N
Number of disposal centers O
Number of plants J
Number of retailers L
Number of distribution centers K
Number of suppliers I

Step 1: ▷ (first segment)

for $i = 0 : M - 1$ **do**
  $ch_k[7 * i + 1] \leftarrow random(1, N)$
  $ch_k[7 * i + 2] \leftarrow random(1, O)$
  $ch_k[7 * i + 3] \leftarrow random(1, J)$
  $ch_k[7 * i + 4] \leftarrow random(1, L)$
  $ch_k[7 * i + 5] \leftarrow random(1, K)$
  $ch_k[7 * i + 6] \leftarrow ch_k[7 * i + 3]$
  $ch_k[7 * i + 7] \leftarrow random(1, I)$
end for

Step 2: ▷ (second segment, plant delivery path)

for $i = 0 : J - 1$ **do**
  $ch_k[7 * M + i] \leftarrow random(0, 2)$
end for

Step 3: ▷ (second segment, Dc delivery path)

for $i = 0 : K - 1$ **do**
  $ch_k[7 * M + J + i] \leftarrow random(0, 1)$
end for

Output: Chromosome $ch_k[\cdot]$

Remark 1: According to the assumptions presented in Chapter 3, returned products have to be directed to the original plant. To follow this limitation, the third and sixth position of first segment of the chromosome representation for any customer, should be identical.

Remark 2: The state for the third and sixth position of first segment belong to the

same plant and are always identical. Therefore, 6 has been considered as the number of each unit, instead of 7, for the implementation part. It means that for given problem with M customers, we have the length of an encoding $6 * M$ instead of $7 * M$ for the first segment of chromosome representation.

4.2.4 Extended random path-based direct decoding

Decoding is the mapping from chromosomes to candidate solutions of the problem. As an example, Figure 4.2 represents an instance of a delivery and recovery path in our model.

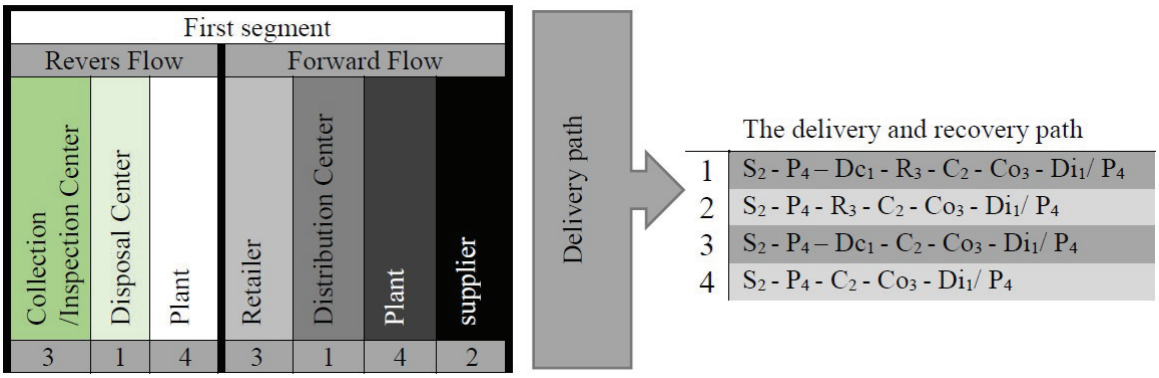


Figure 4.2: Delivery path for a sample of gene unit.

In each gene unit, four delivery paths can be designed by applying normal delivery, direct shipment, and direct delivery. All of them are form a neighborhood. For instance, we can obtain the neighborhood given in Algorithm 1 from the sample of gene unit shown in Figure 4.2 that shows the delivery path to customer 2. In this sample, consider a network design problem composed of two suppliers, four plants, four distribution centers, five retailers in forward flow and four collection/inspection centers and two disposal centers in backward flow. We start with supplier 2 and continue via plant 4, distribution center 1 and retailer 3 in forward flow as well as collection/inspection center 3, disposal center 1 and plant 4 in the reverse flow. Due to construction, four different delivery paths are possible, cf. Figure 4.2. The delivery and recovery path 1 occurs if normal delivery is chosen for all stages. By skipping distribution centers, path number 2 is selected. Similarly, path number 3 is chosen if retailers are skipped. Last, if direct shipment is selected, the delivery path number 4 will be implemented. An important difference between the traditional random path-based direct encoding method and the method adopted in this paper is that we include the delivery path information through

the second segment. The detailed decoding procedure is shown in Figure 4.3. Each locus in this segment is assigned to an integer in the range of $\{0, 1, 2\}$ for plants and $\{0, 1\}$ for distribution centers. Here, we encode Normal Delivery for plants and distribution centers by $P_j = 0$ and $Dc_k = 0$ respectively, where j and k denote the ID of the plant and of the distribution center. Moreover, $P_j = 0$ and $Dc_k = 1$ as well as $P_j = 1$ represent Direct Delivery and $P_j = 2$ Direct Shipment. The paths displayed in Figure 4.3 correspond to respective choices, i.e. we have

$$\begin{aligned}
\text{Path 1} &\iff P_j = 0,\ Dc_k = 0 \\
\text{Path 2} &\iff P_j = 1,\ Dc_k \in \{0, 1\} \\
\text{Path 3} &\iff P_j = 0,\ Dc_k = 1 \\
\text{Path 4} &\iff P_j = 2,\ Dc_k \in \{0, 1\}.
\end{aligned}$$

It should be noted that because the amount of returned products shipped to each one should be known for decoding the forward flow, decoding of the forward flow is impossible until the reverse flow is decoded.

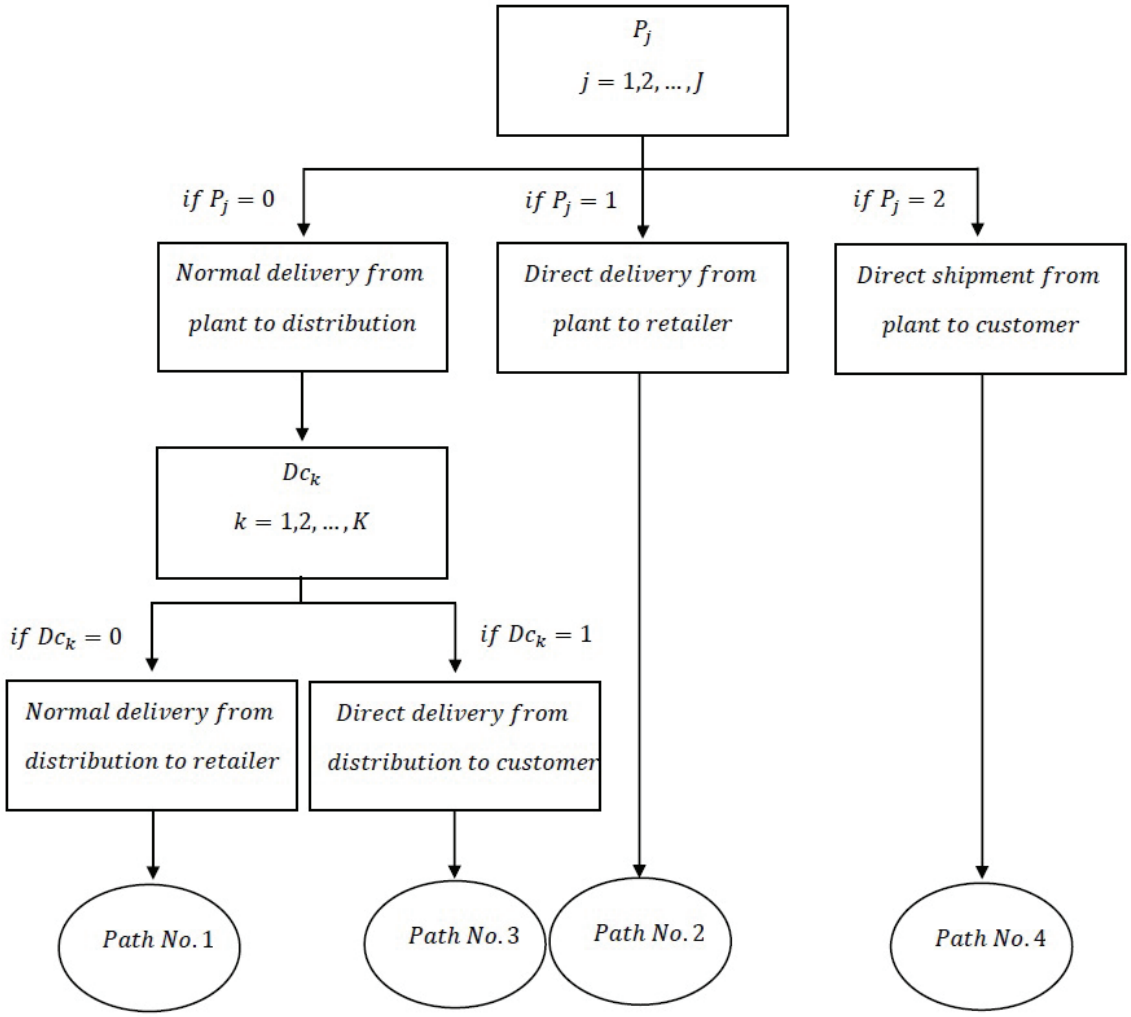


Figure 4.3: Presentation of the second segment of the extended random path-based direct encoding.

4.2.5 Evaluation

In nature, fitter individuals have higher chances of survival to the next generation. By applying nature-based algorithms it is necessary to equip the algorithm with a mechanism that leads the fitter individuals to survive. The evaluation process as a measurement tool is responsible for calculating a fitness value for each individual. This information is important in comparing individuals.

In our study, the evaluation assigns a fitness value to each individual based on its achievement of the objective function carried out using (3.3), thereby inducing a measurement.

We apply the cost function as the fitness value. This fitness value is computed for the decoded chromosome to analyze the accuracy and efficiency of the proposed MA.

4.2.6 Selection

Selection is one of the main stages of a Memetic Algorithm, while it plays the role of adviser in Evolutionary Algorithms. By selection, individuals are chosen from a population for later reproduction. Usually, selection is applied according to the fitness value for each individual. We use the objective value obtained from (3.3) for each individual's fitness value. The most popular types of selection are roulette wheel, tournament, random, and steady state [112]. In tournament selection, several tournaments are played among a few individuals. The individuals are chosen at random from the population. Random selection and not attending all individuals in tournaments are the disadvantages of this method. In random selection, an equal chance is considered for all individuals, i.e., there is no preference for fitter individuals to survive. In steady state selection, just a few good chromosomes are used for creating new offspring and there is no chance for other remaining chromosomes.
In the present work, we adopt the well-known roulette wheel selection for generating the next generation of chromosomes [115]. The roulette wheel strategy is a probabilistic selection based on fitness. Therefore, fitter individuals have a higher chance of mating and propagating their features to the next generation. The roulette wheel selection in contrast the other well-known types of selection mechanism tries to decrease the role of randomness by considering the fitness value of individuals. On the other hand, it considers a chance for all individuals while applying a selection pressure to the more fit in the population, evolving fitter individuals over time. Therefore, evaluation and selection play a very significant role in the evolutionary algorithms.
Applying roulette wheel selection is summarized in five steps as follows:

- Calculate the fitness function for any individual $[fit(1), fit(2), ..., fit(popsize)]$.
- Calculate the selection probability for any individual $[p(1), p(2), ..., p(popsize)]$ (Individual i will have a probability $p(i) = \frac{fit(i)}{\sum_{i=1}^{popsize} fit(i)}$ to be chosen.).
- Calculate the cumulative probability for any individual $[cp(1), cp(2), ..., cp(popsize)]$ (Individual i will have a cumulative probability $cp(i) = \sum_{i=1}^{popsize} p(i)$).
- Choose a random number R between 0 and 1.

- Select an individual according to :
  if $R \leq cp(1)$ then individual (1) is selected.
  if $cp(i-1) < R \leq cp(i)$ then individual (i) is selected.

Also the pseudocode of the roulette wheel selection is available in Algorithm 2.

Algorithm 2 Pseudocode of the roulette wheel selection

Input: Number of population n
for $i = 1 : n$ **do**
    Calculate the fitness function $fit(i)$
end for
for $i = 1 : n$ **do**
    Calculate the selection probability $p(i)$
end for
for $i = 1 : n$ **do**
    Calculate the cumulative probability $cp(i)$
end for
Loop until cross over population is full
for $j = 1 : 2$ **do**
    $R_j \leftarrow \text{random}(1, 0)$
    if $R_j \leq cp(1)$ **then**
        Individual (1) is selected
    end if
    if $cp(i-1) < R_j \leq cp(i)$ **then**
        Individual (i) is selected
    end if
    Create offspring
end for
End loop
Output: Offsprings

4.2.7 Cross over

Selection alone is not enough and cannot produce any new individuals for the population. *Crossover* is the crucially important recombination of both good parents' feature to explore new solutions within the search space. The process of applying cross over is started by fixing the probability for crossover (parameter p_c). This parameter is implemented as an input for the algorithm and gives us the number of chromosomes ($p_c * popsize$) to which the cross over operator will be applied. There are several types of crossover operations developed in the literature, including: one-point, two-point, uniform, blending, and position-based cross over, cf. [71; 69; 112]. Based on the characteristics of the chromosome and in order to generate new feasible individuals, two-point cross over is adapted

since other types of crossover methods, may increase the chance of generating infeasible solutions. The other major advantage of this method is its ability to create an extensive search in the evolutionary process. Two point crossover method applies the steps shown in Algorithm 3. This method can guarantee the generation of feasible offspring given feasible parents. Accordingly, the characteristics of the two-point cross over method, is that the genes from the same position are swapped. So, feasible genes of each parent are exchanged with feasible genes from the other one. Therefore the new obtained offsprings are still feasible. On the other hand, two-point crossover creates an extensive search in the evolutionary process. In this type of crossover, two selected chromosomes, called parents, are needed. As the next step, randomly taking two positions from parents are required. Two new chromosomes, called offspring, are created by exchanging the data of two parents. In practical terms, this operator creates a new supply chain configuration by exchanging facilities at a chose stage. This idea is illustrated in Figure 4.4 by an example.

After applying the crossover procedure, our next step is merging the population which includes the crossover population (offsprings) and the initial population (candidate's parents) to make them ready for the next step. Mutation and crossover operators in GA can help the algorithm to avoid getting trapping in a local optimal. In MA, local search has an important role to find optimal or near optimal solutions.

Algorithm 3 Pseudocode of the two point crossover for the proposed model

Input: Number of population n
Two parents chromosome p_1, p_2
Crossover probability P_c

1: **for** k=1: $(2 * round(P_c * n/2))$**do**
2: Generate two random crossover points a, b
3: Swap data between the two points of parents
4: **end for**

Output: Two offsprings ch_1, ch_2

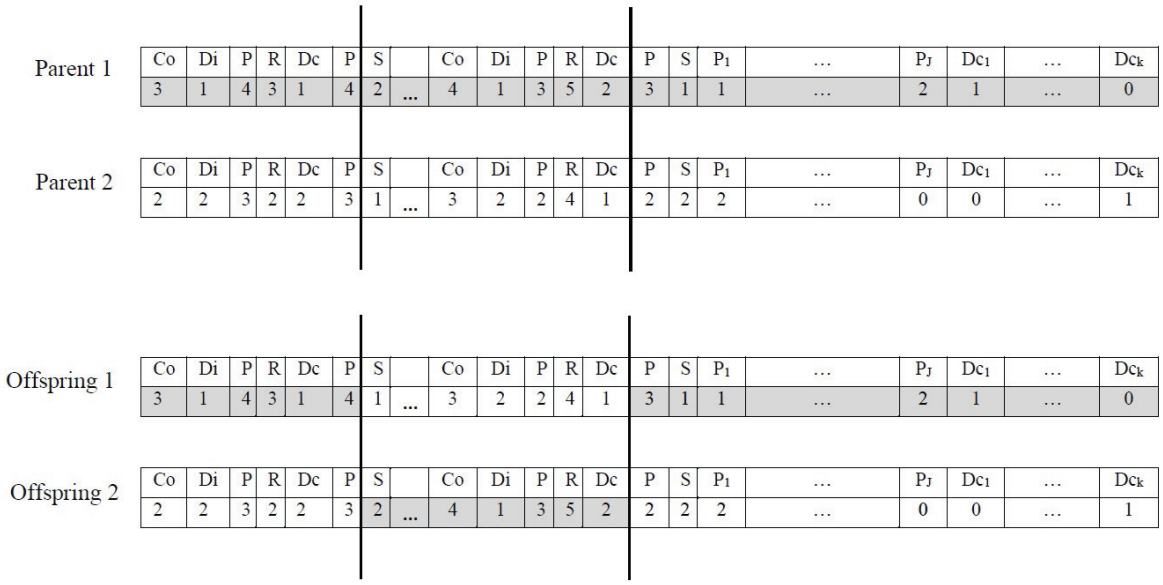


Figure 4.4: Two point cross over method.

4.2.8 Local search

Genetic Algorithms do not have access to a process to get closer to optimal solutions. By applying the cross over operator and selecting the best individuals among initial populations and the cross over population, the similarity among the individuals of the new population is increased. This reveals the similarity in the fitness values of current individuals as well. To overcome this weakness, one possible alternative is adding new individuals into the current population. In this regard, MA came with a *local search* strategy to modify individuals by searching in the neighborhood of an incumbent solution [67]. This neighborhood has the advantages of the incumbent solution and aiming to find an improved solution by making some small changes. Memetic Algorithm mainly focuses on the improvement for search operators. In this algorithm, a good combination between exploration and exploitation will increase the chance of finding better solutions.

The local search, as the vital component in the framework of MAs, is responsible for searching the neighborhood to improve the current solution. It is always employed on the output of the crossover operator [145]. After crossover, the population is merged and sorted according to its fitness value. To apply a local search process, three main requirements are needed: (1) an initial incumbent solution, (2) a definition of a neighborhood for an incumbent solution, and (3) a process for selecting the new incumbent solution. At each iteration, a neighborhood of the current solution is created based on one or more

types of changes. In the next step, the best solution from this neighborhood is replaced by the current solution [145] and then the neighborhood search starts again according to the number of local search iterations. If no further improvement can be found, the new solution is called a local optima for the given neighborhood. It means there is no better solution in the neighborhood of the current solution. There is a direct relation between the size of a neighborhood and the quality of the obtained local optima [99]. Considering a larger size of a neighborhood has an important role to increase the chance of finding a good solution but needs more computation time [99].

In this study, for *local search technique* we apply hill climbing-based random mutation method based on the characteristics of the chromosomes, to generate more adapted offspring, reduce the chance of producing infeasible solutions and improve the chance of finding better solutions. The following four major steps describe this method in details:

- Specify a position randomly to apply mutation for a given incumbent solution.
- Create a neighborhood by replacing the selected gene with feasible possibilities using random mutation operator [135]. The procedure of random mutation operator is simply selecting one position in the current individual and exchange the related selected gene by a regenerated randomly new number [221].
- Select the best member of neighborhood according to the fitness value as the offspring. Figure 4.5 illustrates an example of the local search based-mutation method while the number of the chosen gene in the whole network is 4.
- Repeat this procedure according to the number of local search iterations. This number is considered as an input parameter of MA, and was first proposed by Hart [79]. It refers to the number of iterations of local searches during each generation cycle. For different problems, and according to the size of problems, the values of local search iterations need to be changed [118]. In this work, an adaptive local search engine is applied to adjust the number of local search iterations dynamically during the process according to the size of the considered test problems. As the number of local search iterations for the test problems should be determined by their size, we set the number of local search iterations increasingly according to the number of retailers (L) for all test problems. The number of retailer is selected as it has the highest number among all facilities for each test problem.

The detailed procedure is shown in Algorithm 4. The individual showing the best fitness is selected.

Algorithm 4 Pseudocode of the local search for the proposed model

Input: One parent A
Number of customers M
Number of collection/inspection centers N
Number of disposal centers O
Number of plants J
Number of retailers L
Number of distribution centers K
Number of suppliers I

Randomly select position a in chromosome of A
$b \leftarrow \text{mod}(a, 7)$
$$X \leftarrow \begin{cases} I & \text{if } b = 0 \\ N & \text{if } b = 1 \\ O & \text{if } b = 2 \\ J & \text{if } b \in \{3, 6\} \\ L & \text{if } b = 4 \\ K & \text{if } b = 5 \end{cases}$$
$n \leftarrow \text{round}(\text{random}(30/100 * X, 70/100 * X))$
for $i = 1 : n$ **do**
    $A_i \leftarrow A$
    $c \leftarrow \text{random}(1, X)$
    $A_i(a) \leftarrow c$
    Evaluate fitness function for A_i
end for
Select best chromosome among n new instances

Output: One offspring

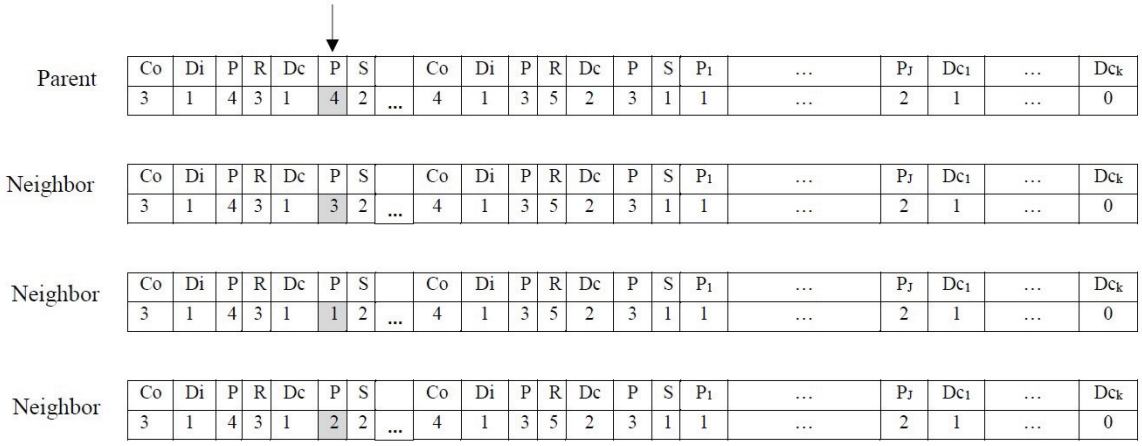


Figure 4.5: Local search method.

4.2.9 Termination criteria

The termination criteria can be characterized in terms of pre-determined computing time, pre-determined number of generations or number of generations with no improvement [12]. In this work, we are dealing with a strategic problem. Therefore, there is no time limitation. Thus, the proposed algorithm in this study is terminated by two termination criteria. The first one is the total number of iterations, which is controlled by the parameter "maxit". The second criterion is the maximum number of iterations without improvement. If the best solution (incumbent) does not improve within this number, the algorithm will be stopped. This number is given by, and thus the criterion is controlled by, the parameter "it".

4.2.10 Overall procedure of proposed Memetic Algorithm

The utilized algorithm in this study is designed to establish the optimum link between the opened facilities and aims to minimize the cost while satisfied demand. The pseudocode of the proposed solution methodology is presented by Algorithm 5.

Algorithm 5 Pseudocode of the proposed Memetic algorithm

Input: Number of population n
Number of crossover population m
for $k = 1 : n$ **do**
  Encode $ch_k[\cdot]$ by Algorithm 1
  Evaluate $ch_k[\cdot]$ according to fitness function
end for
Set $i \leftarrow 0$
while termination condition not satisfied **do**
  for $k = n + 1 : n + m$ **do**
    Select two parents via roulette wheel
    Generate $ch_k[\cdot]$ by Crossover Algorithm 3
  end for
  Merge $ch[\cdot] = \bigcup_{k=1:n+m} ch_k[\cdot]$
  Evaluate and sort $ch[\cdot]$ by fitness value
  Select first n elements of $ch[\cdot]$
  Obtain $\overline{ch}[\cdot]$ via Algorithm 4 with $ch_1[\cdot]$
  if fitness value of $\overline{ch}[\cdot]$ is better than of $ch_1[\cdot]$ **then**
    $ch_1[\cdot] \leftarrow \overline{ch}[\cdot]$
  end if
  $i \leftarrow i + 1$
end while
Output: Chromosome of optimal solution $ch_1[\cdot]$

Note that as we apply only one crossover and search step before selecting the next generation, our method belongs to the class of steady state MA. Combining the aforementioned components, the flow of the proposed algorithm is as follows:

- **Step 1:** (Data reading) Set parameters and read in the data of a given instance.
- **Step 2:** (Initialization) Create an initial random population by the method of *Section 4.2.5*. Repeat this process according to the population size denote by N. Each individual in this population represents a supply chain configuration for the presented problem, (Algorithm 1).
- **Step 3:** (Evaluation) Calculate the fitness value of each individual in the population using the objective functions presented in (3.3), (Algorithm 5, step 3 and 12).
- **Step 4:** (Selection) Apply selection operator using roulette wheel selection presented in *Section 4.2.6*, (Algorithm 2).
- **Step 5:** (Cross over) For each selected two parents, apply two-point crossover operation (defined in *Section 4.2.7*) to create two offsprings by respecting to the crossover probability Pc, (Algorithm 3).
- **Step 6:** (Merging, sorting) The new population obtained by the cross over operator is merged with the initial population. The fitness values of all new individuals are calculated and the new merged population is sorted according to their fitness value, (Algorithm 5, step 11 and 12).
- **Step 7:** (Segregation) The best N individuals are reserved as the initial population for the next generation, (Algorithm 5, step 13).
- **Step 8:** (Local search) The first individual that is the best is selected for the local search operator (defined in *section 4.2.8*). If an improvement in the fitness value occurred, the new individual is exchanged for the current one, otherwise the previous one is kept as a member of population for the next generation, (algorithm 4). This step is repeated according to the number of local search iterations.
- **Step 9:** (Termination) The algorithm runs until one of the stopping conditions is satisfied. If the pre-specified stopping conditions are not satisfied (defined in *Section 4.2.9*) the algorithm needs to go back to Step 4, (Algorithm 5, step 6).

The procedure is displayed via a flowchart in Figure 4.6 to show the overall steps of the solution method.

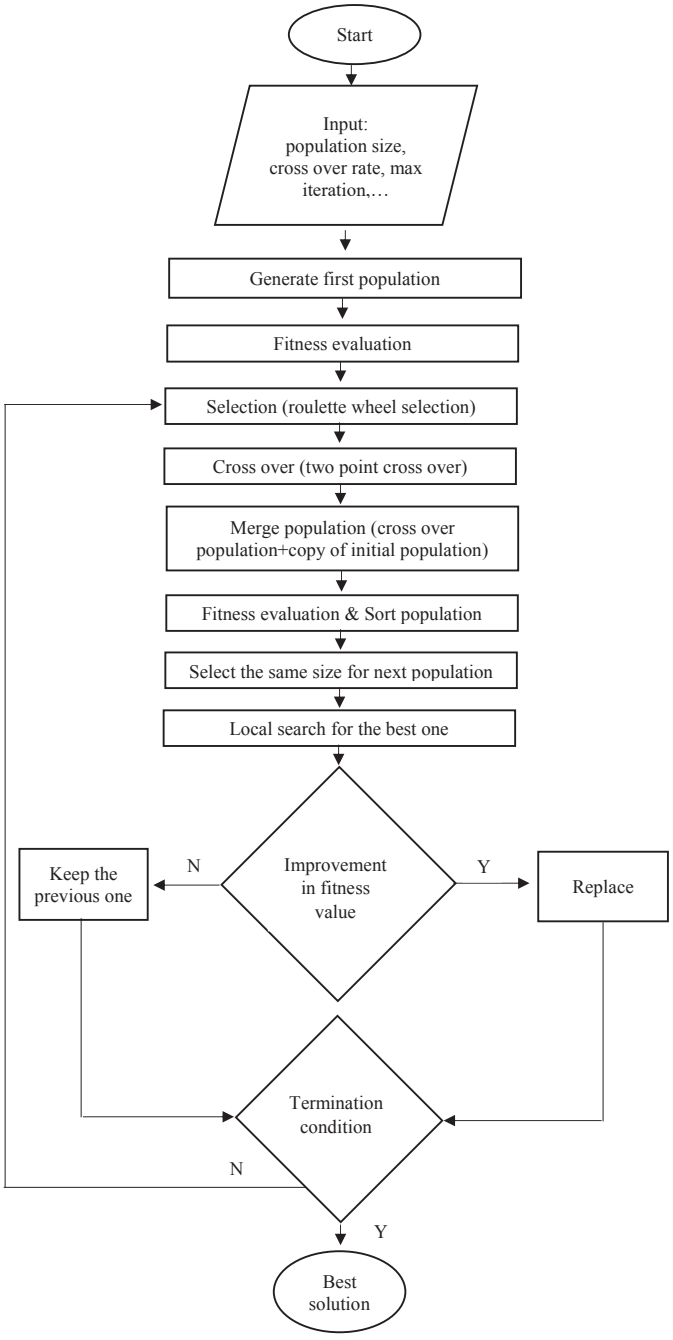


Figure 4.6: The flow diagram of the proposed MA.

4.3 Summary and conclusion

In this chapter, we focused on presenting a solution methodology for a comprehensive mixed integer linear programming formulation for a seven stage closed-loop network design problem. We applied the extended direct delivery path representation-based Memetic Algorithm. Since the basic problem is NP-hard, the combination with flexibility in the delivery path makes the search space of the problem much larger, more complex and even harder to solve. Because existing methods are unable to solve this problem, we proposed a Memetic Algorithm approach to compute a near optimal solution for large size network design problems. Also in this chapter, we introduced a new chromosome representation for MA to enhance its search ability for the proposed flexible model. We explained the details of the steps involved in the applied algorithm. Each algorithm has some parameters that need to be investigated to reveal the best performance. In this regard having some information about the parameters involved in the presented Memetic Algorithm helps us to lead the algorithm in more efficient way.

Chapter 5

Effect of various parameters of solution methodology

5.1 Introduction

Network design problems are NP-hard and developing an efficient solution methodology is still a critical need in this area. In this regard, we adapted a Memetic Algorithm, belonging to the class of meta-heuristic algorithms, with a neighborhood search mechanism and a novelty in population generation to find a near optimal solution for large size problems. In order to increase the efficiency of the proposed algorithm, some information about the parameters applied in the Memetic Algorithm, such as number of iteration, local search iteration, population size, and cross over rate in format of a parameter analysis are required. To this end, studying the effect of these parameters is the main contribution of this chapter.

The initial results of this chapter have been presented in the 28th European conference on operation research, Poznan, Poland, 2016 (E. Behmanesh and J. Pannek 2016) and some of the final results of this chapter have been published in the "Mathematical Problems in Engineering" journal, (E. Behmanesh and J. Pannek 2017) [24].

5.2 Various parameters of solution methodology

There are some studies regarding the effect of different parameters of meta-heuristic algorithms applied for logistics network design problems.

Lee and Dong [108] proposed a MILP model, which can simultaneously manage the forward and reverse distributions at the same time for end-off lease computer products. Regarding the solution methodology, they used a Memetic Algorithm based on a Tabu Search. They observed that the maximum number of iterations does not make a significant impact to the performance of the proposed heuristics in all the presented five test problems. From their result, it is clear that about 80 percent of the progress towards the optimum is obtained within 20 iterations. They also presented some results with different number of iterations within a neighborhood search. Since this work was focusing on a specific case study for the recovery of end-of-lease computer products, the lower bound of total cost in the distribution network was available and adapted for the purpose of solution comparison. The computation times for the heuristic presented in this study increases reasonably as the problem size increases.

Syarif et al. [201] proposed a MILP formulation for a fixed-charge and multi-stage transportation problem. A spanning tree based Genetic Algorithm was applied to solve this problem. Based on their results, the cross over rate is suggested to be equal to 0.4, the mutation rate is fixed as 0.2 and the population size is set to 100. To have more information about the proposed algorithm, they divided each test problem into three numerical experiments by giving different population size. They concluded that since the search space for large size problems is so huge, it is very important to set the experiment with reasonable population size and maximum generation in order to ensure increased chances for good results.

Wang and Hsu [212] presented a closed-loop model with a spanning-tree based Genetic Algorithm. In this study, the cross over rate is fixed as 0.4, the mutation rate 0.2, and for the population they considered different numbers to observe its influence with respect to problem size. Moreover, several conditions are settled for number of generations, computing time and fitness convergence. Fitness convergence appears when all chromosomes in the population have the same fitness value. The authors stop the evolutionary process in their implementation when the number of iterations without improvement in fitness value was 10. At the same time, they imposed the maximal number of generations to be 750 as another stopping criterion. From simulations, the authors realized that in the proposed algorithm increasing the population size improves accuracy for large size problems only slightly while the required computation times was huge. Therefore, they concluded that using large population size for large size test problems is not sufficient.

The extended version of a two stage transportation problem was developed by Gen et al. [71] to minimize total cost, including opening cost of distribution centers and shipping cost from plant to distribution center as well as distribution center to customers. As solution methodology, a priority-based Genetic Algorithm, was presented using a new cross over method called Weight Mapping Crossover (WMX). Simulations were carried out in two stages. In the first stage, the authors investigated the effect of different combinations of cross over and mutation operators. The aim of this stage was to find the best combination among the 8 possibilities given by 4 cross over and 2 mutation methods. In the second stage, each test problem was divided into three numerical experiments to assess the effect of population size and number of generations on the performance of the proposed Genetic Algorithm. In addition, to show the performance of the priority-based Genetic Algorithm, another chromosome representation, called the spanning tree-based Genetic Algorithm using Prüfer number was considered. Statistical analysis using different population sizes and numbers of generations showed the improved performance of the priority-based Genetic Algorithm considering the average performance. Yet, it required more computation time in comparison to the spanning-tree based Genetic Algorithm. Regarding the second stage, the results showed an improvement in the quality of solution with increasing the population size as well as number of generations. The authors mentioned that a trade-off between solution quality and computation time exists.

Experiments on Memetic Algorithms have shown that there are two major impact sources affecting their performance [212]. One of them is chromosome representation, and the second one determined as the Memetic operators. Here, we utilize our chromosome representation from [19] and analyze the effect of various parameters on the Memetic operators regarding both performance and computing time. We particularly focus on four factors that are defined as the given inputs to the algorithm. These factors include:

- number of iterations,
- population size,
- number of local search iterations, and
- cross over rate.

5.3 Design of experiments

Since the logistics network framework in this study differs from previous studies, we generated six small and medium size test problems with 128, 209, 234, 468, 1006 and 1780 decisions variables, cf. Table 5.1, to assess the effect of parameters of the developed MA, mentioned in *Section 5.2*. Other parameters of the logistics network were generated randomly using uniform distributions, cf. Table 5.2. To show the performance of the proposed MA, we employed the Branch-and-Bound Algorithm from LINGO15 to solve the optimization problem. Respective results are included in Table 5.1. Each test problem has been solved 20 times to test the robustness of the results.

Table 5.1: Settings of test problems

Problem	1	2	3	4	5	6
Supplier	2	2	2	2	3	4
Plant	2	4	3	4	6	8
Distribution	5	6	8	10	15	20
Retailer	8	10	9	16	24	32
Customer	2	2	3	4	6	8
Collection/Inspection center	2	2	3	4	6	8
Disposal	1	1	2	2	2	4
Total number of facilities	22	27	30	42	62	84
Total number of possible routes	110	218	294	432	966	1728
Solution by LINGO	2905	2345	2335	1160	4100	11365

5.4 Termination condition

Traditionally [185], termination conditions for Genetic and Memetic Algorithms include execution time, bounds on the number of generations or the evaluations of the fitness function, or may be triggered if the chance of achieving significant changes in the next generations is excessively low. For the first two alternatives, we require some knowledge about the problem or solution. In contrast to that, the third one does not require such knowledge and may be implemented, e.g., by imposing a bound on the number of iterations without improvement.

Table 5.2: Parameters values used in the test problems

Parameters	Range
$b_j, j \in S$	Uniform (200,1100)
$b_j, j \in P$	Uniform (100,1000)
$b_j, j \in Dc$	Uniform (50,900)
$b_j, j \in R$	Uniform (50,850)
$b_j, j \in D$	Uniform (100,500)
$b_j, j \in Co$	Uniform (20,100)
$b_j, j \in Di$	Uniform (20,100)
p_j^{C}	10 %
p_j^{Co}	50 %
c_{ij}	Uniform (1,3)
$c_j, j \in P$	Uniform (100,2500)
$c_j, j \in Dc$	Uniform (100,2100)
$c_j, j \in R$	Uniform (100,400)
$c_j, j \in Co$	Uniform (100,500)
$c_j, j \in Di$	Uniform (50,400)

5.5 Computational result

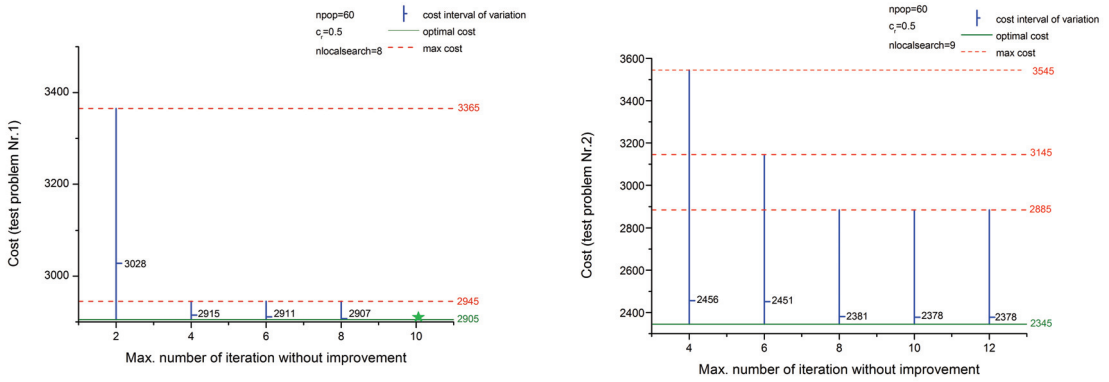
First of all, we note that our implementation was written in MATLAB R2015b and run on the PC with Intel® Core™ i5 2.40GHz with 12 GB RAM. For the implementation, six test problems are considered by different sizes using Table 5.1. This section includes four parts and each part aims to investigate effect of one parameter on the overall system.

5.5.1 Number of iteration without improvement

During our simulations, we observed that a specific maximal number of iterations without improvement cannot be chosen uniformly for all test problems. We found for larger size problems, with bigger search space we need to apply more iterations to enhance the chance of finding good results. Hence, in a first step we determined the number of iterations without improvement for each test problems specifically. To this end, we considered several instances and selected the largest candidate for which significant improvement may be achieved.

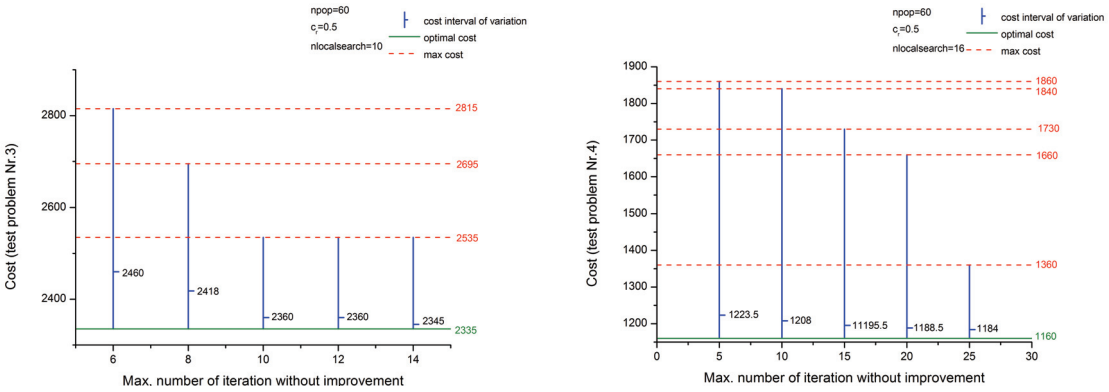
To assess the effect of the termination criterion on the proposed MA, we imposed a maximum iteration number of 200 and varied the bound on the number of iterations without improvement for the test problems. As the criterion appears to be depending on the size of the problems, we considered different ranges for the bound, cf. Figures 5.1(a),

5.1(b), 5.2(a), 5.2(b), 5.3(a), and 5.3(b). Table 5.3 is available to present more details.



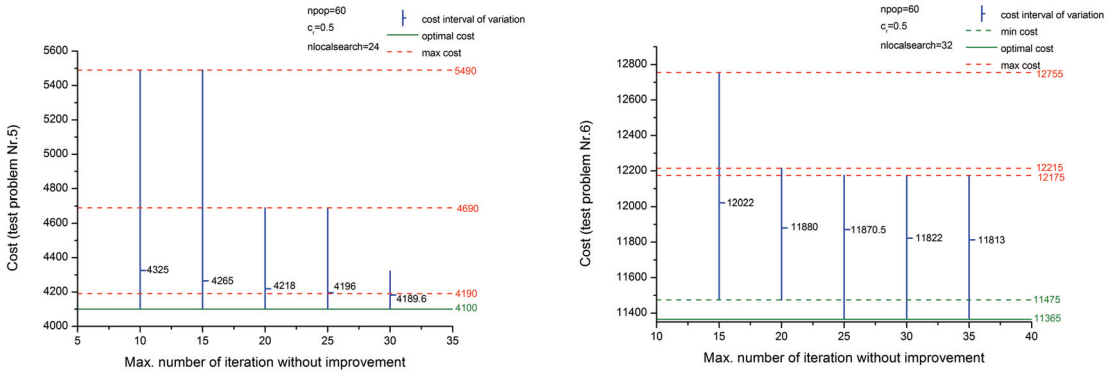
(a) Test problem No.1 (b) Test problem No.2

Figure 5.1: Gap of the proposed MA for different bounds on the number of iterations without improvement (Test problem No.1 and 2)



(a) Test problem No.3 (b) Test problem No.4

Figure 5.2: Gap of the proposed MA for different bounds on the number of iterations without improvement (Test problem No.3 and 4)



(a) Test problem No.5

(b) Test problem No.6

Figure 5.3: Gap of the proposed MA for different bounds on the number of iterations without improvement (Test problem No.5 and 6)

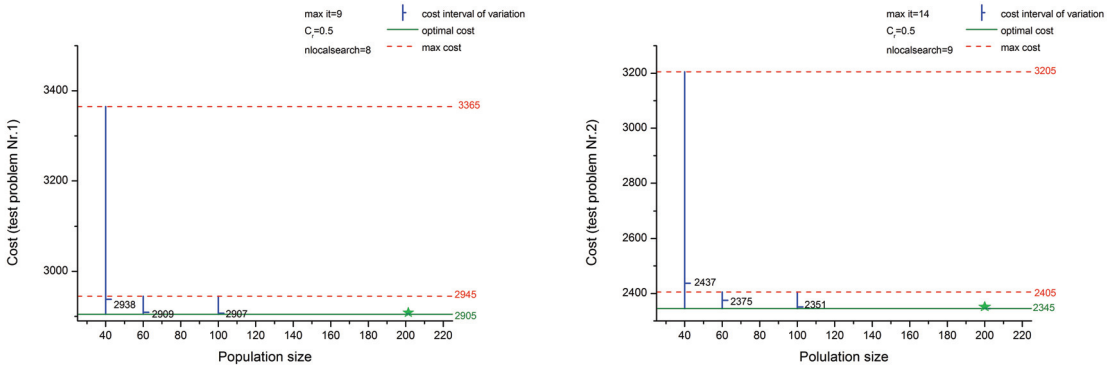
Table 5.3: Sensitivity analysis for maximum number of iterations without improvement

Test problem number 1			
Max number of iteration without improvement	Min-cost	Max-cost	Ave-cost
2	2095	3365	3028
4	2905	2945	2915
6	2905	2945	2911
8	2905	2945	2907
10	2905	2905	2905
Test problem number 2			
Max number of iteration without improvement	Min-cost	Max-cost	Ave-cost
4	2345	3545	2456
6	2345	3145	2451
8	2345	2885	2381
10	2345	2885	2378
12	2345	2885	2378
Test problem number 3			
Max number of iteration without improvement	Min-cost	Max-cost	Ave-cost
6	2335	2815	2460
8	2335	2695	2418
10	2335	2535	2360
12	2335	2535	2360
14	2335	2535	2345
Test problem number 4			
Max number of iteration without improvement	Min-cost	Max-cost	Ave-cost
5	1160	1860	1223
10	1160	1840	1208
15	1160	1730	1195
20	1160	1660	1188
25	1160	1360	1184
Test problem number 5			
Max number of iteration without improvement	Min-cost	Max-cost	Ave-cost
10	4100	5490	4325
15	4100	5490	4265
20	4100	4690	4218
25	4100	4690	4196
30	4100	4320	4282
Test problem number 6			
Max number of iteration without improvement	Min-cost	Max-cost	Ave-cost
15	11475	12755	12022
20	11475	12215	11880
25	11365	12175	11870
30	11365	12175	11822
35	11365	12175	11813

These bar charts allow us to identify the critical range for our braking criteria. For example, given test problem 1, choosing the maximum number of iterations without improvement to be 10 is advantageous from an performance point of view as the algorithm can find the optimal solution each time. But we consider this bound to be set to 6 to assess the effect of other parameters.
Additionally, we computed the minimum, maximum, and average number of iterations and selected the average number of iterations to fix the number of iteration for further assessment. Note that the minimum number of iteration is not sufficient and maximum number of iterations is too large to assess the effect of other parameters. Strictly speaking, the number of iterations should be high enough as to obtain good results, but not too high to avoid influencing other parameters. Setting the maximal number of iterations without improvement at 6 reveals the average number of iterations for test problem number 1 to be 9. Similarly, 8, 14, 20, 25 and 30 are selected as the maximal number of iteration without improvement for test problems numbers 2 to 6. According to these selections, we obtain 14, 22, 39, 43, and 60 as the average number of iterations respectively.

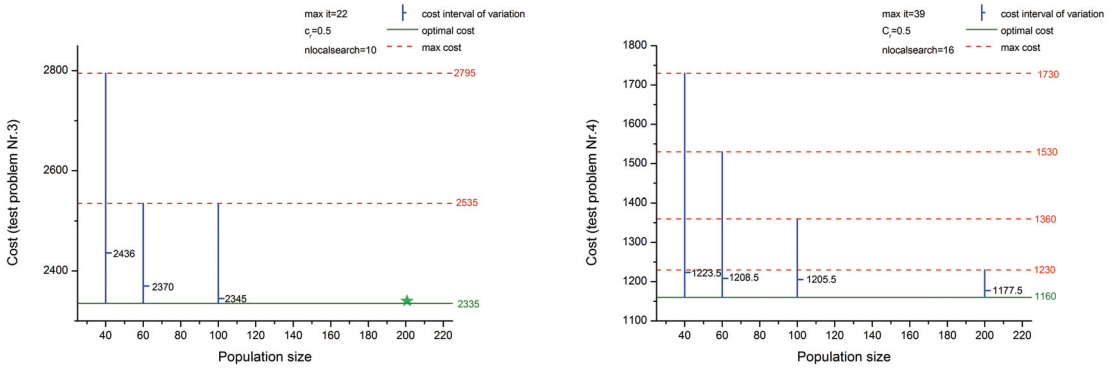
5.5.2 Number of population

Population size is often specified by the user and remains fixed [170]. In most recent studies, the population size is considered as a problem-independent, but algorithm-dependent, variable, and 50 to 100 individuals are typically chosen. It is found that an improper choice of the population size may lead the algorithm to be less efficient [170]. However, there is a clear relation between the population size and the convergence speed. To see the effect of population size on the proposed Memetic Algorithm, four different population size settings with 40, 60, 100, and 200 individuals were considered, cf. Figures 5.4(a), 5.4(b), 5.5(a), 5.5(b), 5.6(a), and 5.6(b). More details are available in Table 5.4.



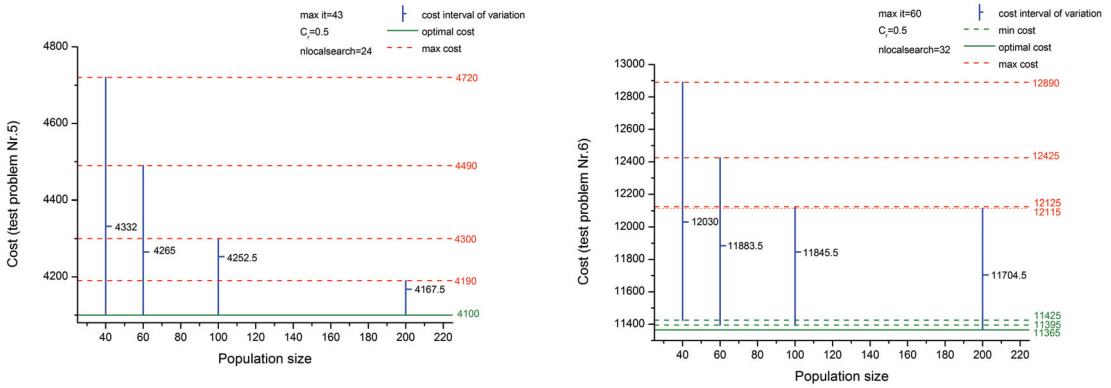
(a) Problem No.1 (b) Problem No.2

Figure 5.4: Gap of the proposed MA for different population sizes



(a) Problem No.3 (b) Problem No.4

Figure 5.5: Gap of the proposed MA for different population sizes



(a) Problem No.5 (b) Problem No.6

Figure 5.6: Gap of the proposed MA for different population sizes

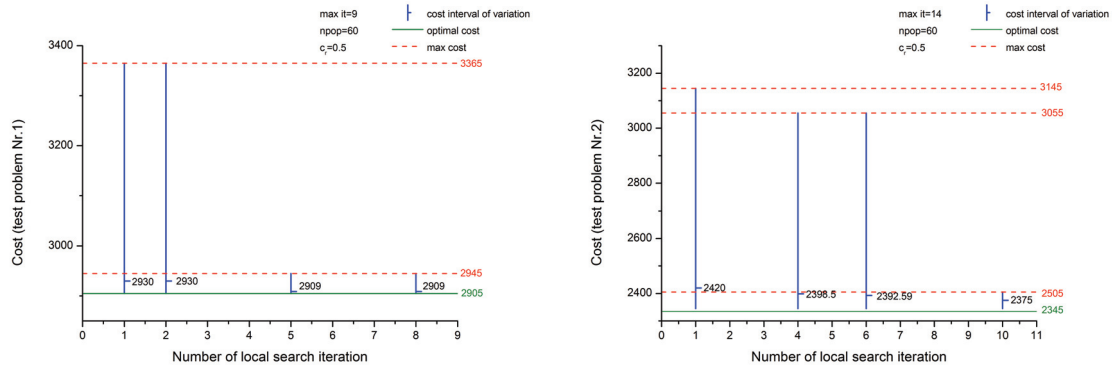
Table 5.4: Sensitivity analysis based on different population size

Test problem number 1			
Population size	Min-cost	Max-cost	Ave-cost
40	2095	3365	2938
60	2905	2945	2909
100	2905	2945	2907
200	2905	2905	2905
Test problem number 2			
Population size	Min-cost	Max-cost	Ave-cost
40	2345	3205	2437
60	2345	2405	2375
100	2345	2405	2351
200	2345	2345	2345
Test problem number 3			
Population size	Min-cost	Max-cost	Ave-cost
40	2335	2795	2436
60	2335	2535	2370
100	2335	2535	2345
200	2335	2335	2335
Test problem number 4			
Population size	Min-cost	Max-cost	Ave-cost
40	1160	1730	1223
60	1160	1530	1208
100	1160	1360	1205
200	1160	1230	1177
Test problem number 5			
Population size	Min-cost	Max-cost	Ave-cost
40	4100	4720	4332
60	4100	4490	4265
100	4100	4300	4252
200	4100	4190	4167
Test problem number 6			
Population size	Min-cost	Max-cost	Ave-cost
40	11425	12890	12030
60	11395	12425	11883
100	11395	12125	11845
200	11365	12115	11704

5.5.3 Number of local search iteration

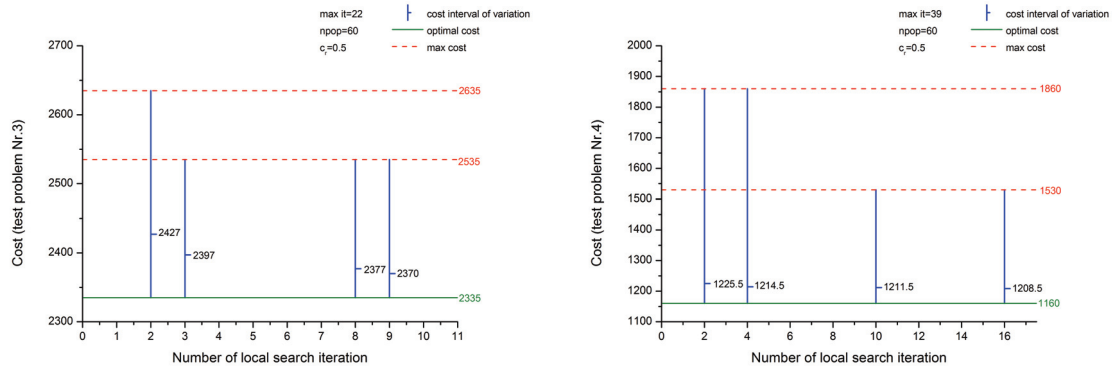
Based on our simulations, we observed that a predefined number of local search iterations may render the method to be ineffective for larger test problems. To determine the effect of the number of local search iterations on the performance of the proposed MA, we imposed different bounds for the test problems according to their size, cf. Figures 5.7(a), 5.7(b), 5.8(a), 5.8(b), 5.9(a) and 5.9(b). Table 5.5 shows more details for respective results.

We increased the number of local search iterations in accordance with the size of test problems. Since the number of echelons in each stage is increased by enlarging the size of test problems, we linked the number of local search iterations to the number of disposal centers O, Plant J, Distribution centers K and retailers L respectively.



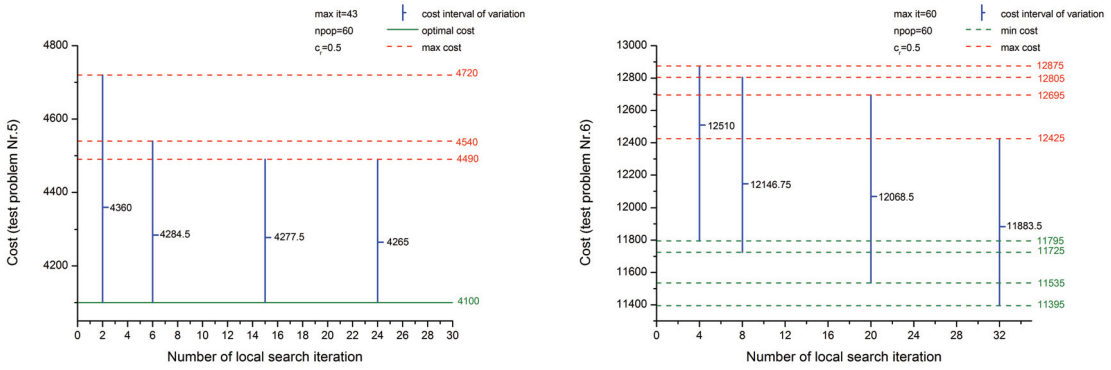
(a) Problem No.1 (b) Problem No.2

Figure 5.7: Gap of the proposed MA for different number of local search iterations



(a) Problem No.3 (b) Problem No.4

Figure 5.8: Gap of the proposed MA for different number of local search iterations



(a) Problem No.5 (b) Problem No.6

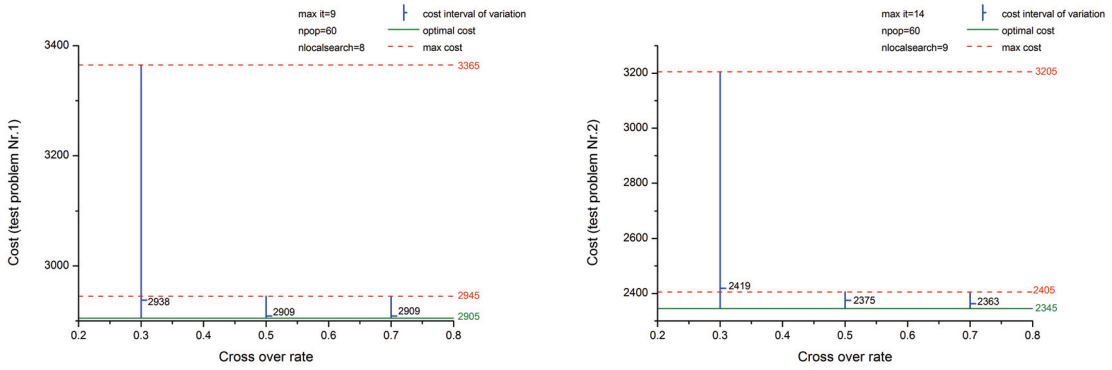
Figure 5.9: Gap of the proposed MA for different number of local search iterations

Table 5.5: Sensitivity analysis based on different number of local search iterations

Problem number 1			
Number of local search iterations	Min-cost	Max-cost	Ave-cost
1	2905	3365	2930
2	2905	3365	2930
5	2905	2945	2909
8	2905	2945	2909
Problem number 2			
Number of local search iterations	Min-cost	Max-cost	Ave-cost
1	2345	3145	2456
4	2345	3055	2451
6	2345	3055	2381
10	2345	2405	2378
Problem number 3			
Number of local search iterations	Min-cost	Max-cost	Ave-cost
2	2335	2635	2427
3	2335	2535	2397
8	2335	2535	2377
9	2335	2535	2370
Problem number 4			
Number of local search iterations	Min-cost	Max-cost	Ave-cost
2	1160	1860	1225
4	1160	1860	1214
10	1160	1530	1211
16	1160	1530	1208
Problem number 5			
Number of local search iterations	Min-cost	Max-cost	Ave-cost
2	4100	4720	4360
6	4100	4540	4284
15	4100	4490	4277
24	4100	4490	4265
Problem number 6			
Number of local search iterations	Min-cost	Max-cost	Ave-cost
4	11795	12875	12510
8	11725	12805	12145
20	11535	12695	12068
32	11395	12425	11883

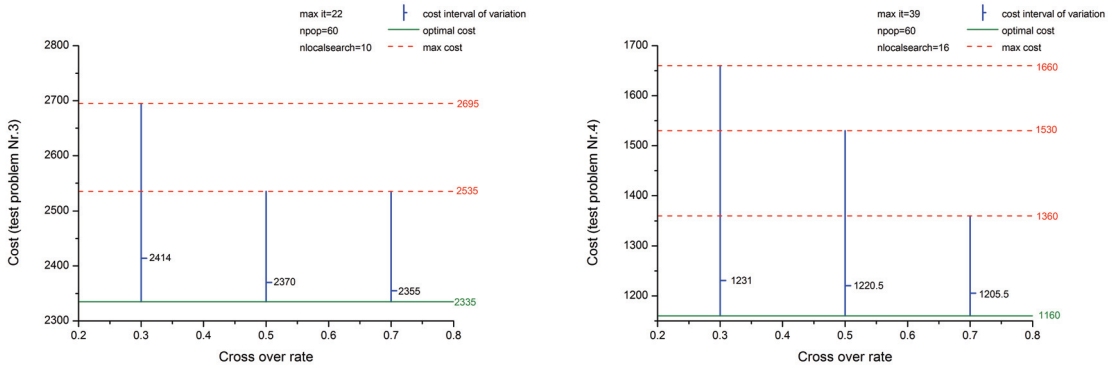
5.5.4 Cross over rate

Cross over is a process of taking more than one parent chromosome to produce an offspring chromosome. There is no single best setting to choose the cross over rate. It depends on the algorithm and specific problem to which it is applied. Lowering the crossover rate will leave more individuals unchanged in the next generation, while increasing the cross over rate leads to an increased search space for the method. To recognize the effect of cross over rates on the proposed Memetic Algorithm, three different cross over rates, 0.3, 0.5, and 0.7 were considered as low, medium, and high cross over rates respectively, cf. Figures 5.10(a), 5.10(b), 5.11(a), 5.11(b), 5.12(a) and 5.12(b). Table 5.6 covers all details of the obtained results.



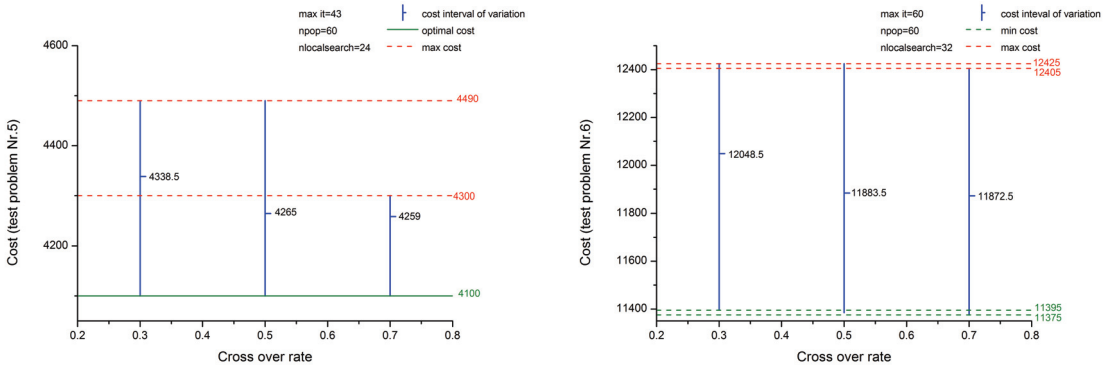
(a) Problem Nr.1 (b) Problem Nr.2

Figure 5.10: Gap of the proposed MA for different cross over rates



(a) Problem Nr.3 (b) Problem Nr.4

Figure 5.11: Gap of the proposed MA for different cross over rates



(a) Problem Nr.5 (b) Problem Nr.6

Figure 5.12: Gap of the proposed MA for different cross over rates

Table 5.6: Sensitivity analysis based on different cross over rate

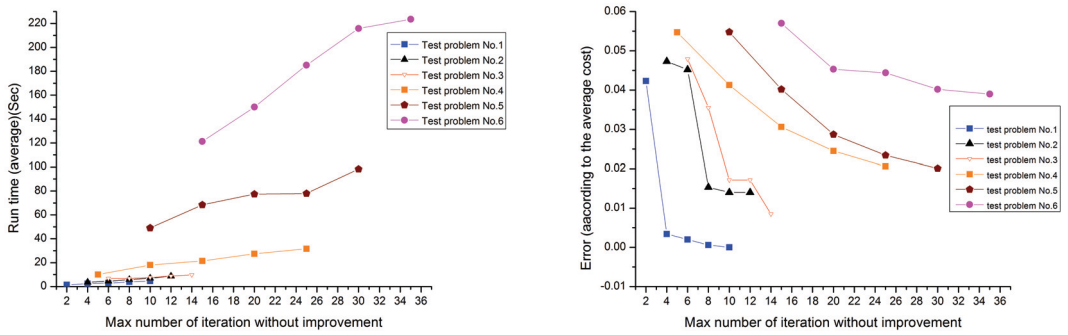
Test problem number 1			
Cross over rate	Min-cost	Max-cost	Ave-cost
0.3	2095	3365	2938
0.5	2905	2945	2909
0.7	2905	2945	2909
Test problem number 2			
Cross over rate	Min-cost	Max-cost	Ave-cost
0.3	2345	3205	2419
0.5	2345	2405	2375
0.7	2345	2405	2363
Test problem number 3			
Cross over rate	Min-cost	Max-cost	Ave-cost
0.3	2335	2695	2414
0.5	2335	2535	2370
0.7	2335	2535	2355
Test problem number 4			
Cross over rate	Min-cost	Max-cost	Ave-cost
0.3	1160	1660	1231
0.5	1160	1530	1220
0.7	1160	1360	1205
Test problem number 5			
Cross over rate	Min-cost	Max-cost	Ave-cost
0.3	4100	4490	4338
0.5	4100	4490	4265
0.7	4100	4300	4259
Test problem number 6			
Cross over rate	Min-cost	Max-cost	Ave-cost
0.3	11395	12425	12048
0.5	11385	12425	11883
0.7	11375	12405	11872

5.6 Result analysis

To analyze the proposed Memetic Algorithm, the error percentage of its solution is calculated according to

$$\text{Error percent} = \frac{\text{MA}_{\text{average answer}} - \text{LINGO}_{\text{answer}}}{\text{LINGO}_{\text{answer}}} \tag{5.1}$$

Figure 5.13(a) shows the variations in runtime for increasing the number of iterations without improvement. This behavior is similar for all six test problems and directly connected to additional iterations of the algorithm.



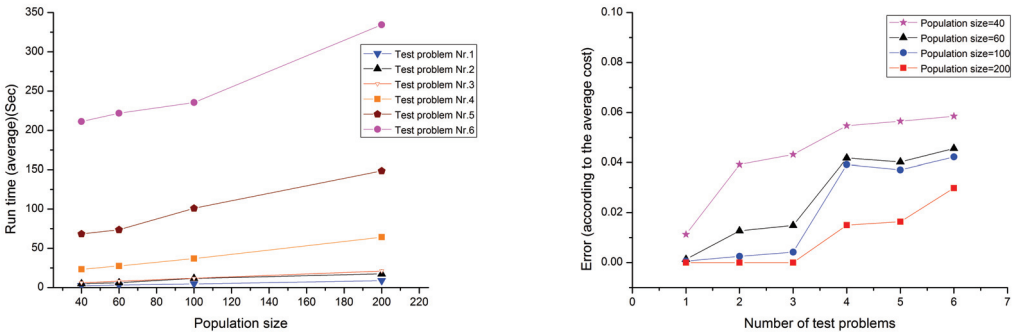
(a) Runtime of the proposed MA for different number of iterations without improvement

(b) Gap of the proposed MA for different number of iterations without improvement

Figure 5.13: Effect of different numbers of iterations without improvement

Figure 5.13(b) shows a general relation between number of iterations without improvement and the error percentage for all six test problems. We observe that the error percentage is decreasing for increasing bounds. More iterations led the algorithm to improved results and a decreasing error percentage.
As Figures 5.13(a) and 5.13(b) show, the solution gaps are shrinking for enlarged numbers of iterations without improvement. Since the latter incurs higher computing times, this parameter should be increased only up to a tolerable computing time.

To assess the effect of the population size, Figures 5.14(a) and 5.14(b) allow us to conclude that the error percentage is decreasing for enlarged populations, again at the cost of increased computing time. The comparison between with population sizes indicates that the two population sizes 60 and 100 have an acceptable performance with respect to the solution gap criterion and runtime. Although population size 200 has a better efficiency regarding solution, the required computation times are huge.

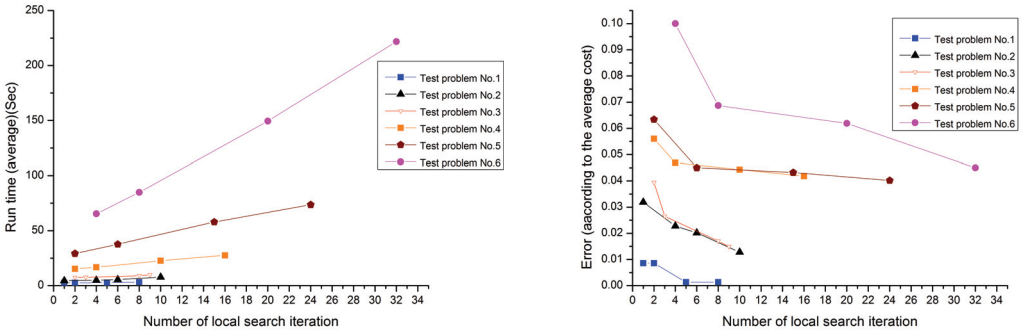


(a) Runtime of the proposed MA for different population size

(b) Gap of the proposed MA for different population sizes

Figure 5.14: Effect of population sizes

Regarding the effect of the number of local search iterations, from Figures 5.15(a) and 5.15(b) we observe a general decrease in the error percentage by increasing the local search iteration parameter as well as a general growth in runtime. The increased computation requirements are due to the additional local search steps. Because of the beneficial



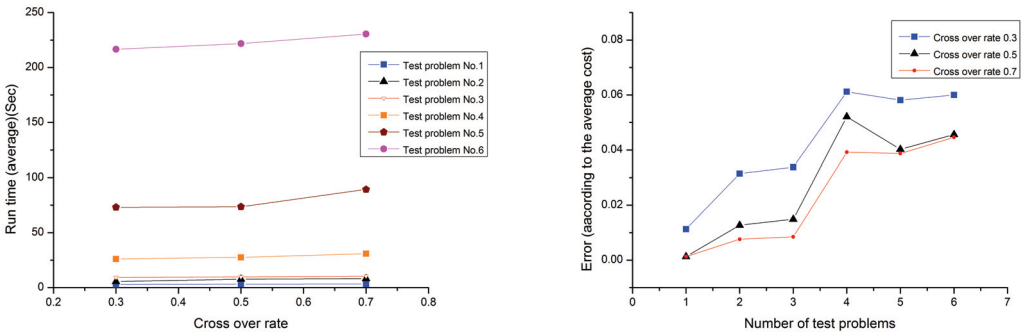
(a) Runtime of the proposed MA for different number of local search iterations

(b) Gap of the proposed MA for different number of local search iterations

Figure 5.15: Effect of different number of local search iterations

impact of this parameter on the quality of the solution, we can improve the latter at the cost of computing time. From Figures 5.16(a) and 5.16(b) we observe that increasing the cross over rate causes the error percentage to decrease and the runtime to increase. The latter is due to the additional cross over operations.

The comparison between different cross over rates illustrate that the two cross over rates 0.5 and 0.7 have an acceptable performance in terms of both solution gap and com-



(a) Runtime of the proposed MA for different cross over rates

(b) Gap of the proposed MA for different cross over rates

Figure 5.16: Effect of different cross over rates

putation time.

Consequently, increasing or decreasing each parameter is based on the manager's emphasizing either the accuracy or efficiency of the algorithm. For example, if the managers set the acceptable error rate in advance, the population size of 60 can be used in all test problems with maximum error rate of 4.5 %. On the other hand, by sacrificing efficiency, more accurate results and lower error rates can be obtained by the variable population size approach.

5.7 Summary and conclusions

The chapter focused on the effect of four different parameters on the behavior of the proposed Memetic Algorithms. First, the parameters involved in the solution methodology were introduced. Through numerical research, we realized that all these parameters can significantly effect the results. These parameters include: population size, cross over rate, local search iteration, and number of iterations. Then, the effect of each parameter is shown separately. From our experiments, we observed that, although the large population size, cross over rate, and number of local search iterations can improve the solution, they incur high computational time costs. The trade-off between these is to detect suitable parameters in the consideration of the error percent and time. In summary, the proposed algorithm has demonstrated its performance in terms of both efficiency and accuracy.

Following the above discussion, if we set the acceptable error rate as well as computa-

tional time [108; 212], the respective parameters can be determined, which is the main idea pursued in the next chapter.

Chapter 6

Parameter analysis

6.1 Introduction

Having more information regarding the parameters involved in each evolutionary algorithms helps us to increase the performance of the algorithm. However, very little effort is spent on studying the effect of parameters on performance. After assessing the effect of each parameter of the proposed Memetic algorithm separately in the previous chapter, in this chapter we aim to focus on the optimum operation condition. The problem of interest in this study deals with the strategic level. Therefore, there is no time limitation. Still doing some further analysis regarding the parameters can help us to improve our results, especially for the other types of problems (say at the tactical level). To identify the optimal value of parameters of the proposed Memetic Algorithm, the Taguchi method is adopted. In this study, four factors are considered namely: population size, cross over rate, local search iteration, and number of iteration.

The initial content of this chapter will be submitted to a relevant journal, by the title of "Taguchi analysis for improving optimization of integrated forward/reverse logistics" (E. Behmanesh and J. Pannek 2018) [23].

6.2 Parameter analysis methods

There are many methods for studying the process parameters [13]. "One factor at a time" and "Design of Experiment (DOE)" are the most popular approaches being implemented by researchers in this field.

- **One factor at a time**: "One factor at a time" also known as "monothetic analysis"

is a method involving one specific condition instead of multiple factors simultaneously. The experiments are repeated by changing any other one factor.

- **Design of Experiment**: "Design of Experiment (DOE)" is a statistical approach, capable to determine the relationship between factors involved in a complex and multi-variable process with few trials.

Although some researchers have shown that "One factor at a time" can be more accurate some times [45], it requires more running and computing time for the same estimation. There are two major approaches to DOE [13]:

- **Full factorial design**: A "full factorial design", also known as a "fully crossed design", is able to consider two or more factors. And each factor has a number of possible levels, with all possible combinations, generated by those levels across all those factors taken into account. This method allows to record the effect of each factor on the response variable individually. Using this method is time-consuming, particulary for large numbers of factors. By applying this method with k factors and l levels for each factor, l^k trials are needed.

- **Taguchi method**: According to the statistical theory, not every combination of parameters need to be checked [222]. To overcome this problem, Taguchi [202] came up with his eponymous fractional replicated designs using Orthogonal Arrays (OAs) [198]. In this method, a small set from all the possibilities is selected. By a comparison between these two methods, it is found that Taguchi's method is more efficient than the "full factorial design" approach due to same results with lesser number of experiments to be conducted [13].

As Taguchi's method is more efficient [13], we applied it in this research. A summary of this section is given in Figure 6.1.

6.3 Improving optimization using Taguchi

In order to improve the operation condition of the Memetic Algorithm regarding the selected parameters, we applied the following steps from the Taguchi analysis [13]:

1. Identify the objective function to be optimized.

2. Identify the independent factors and numbers of levels for each.

3. Select a suitable orthogonal array and construct the simulation matrix.

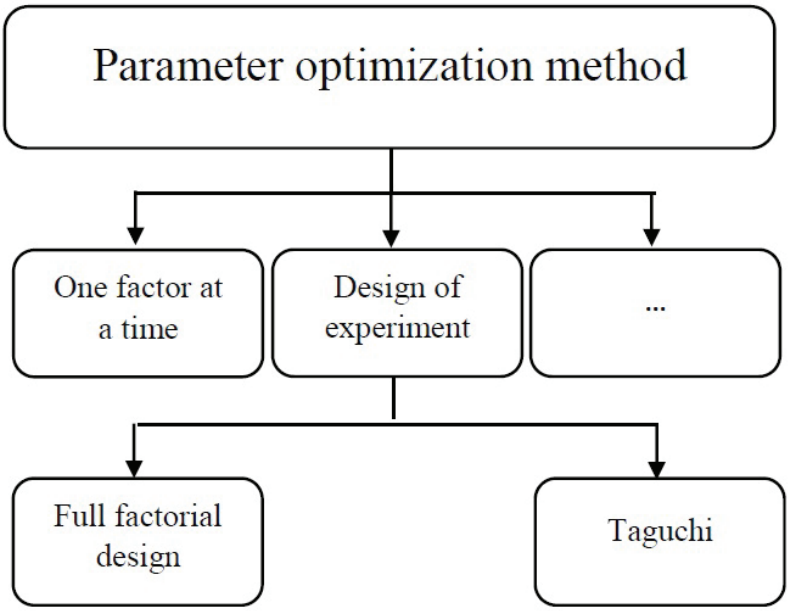


Figure 6.1: The flow diagram of the proposed parameter optimization methods.

4. Conduct the simulation matrix.

5. Examine and analyze data and predict the best parameter levels.

6. Confirm the obtained results from step 5 by simulation.

To apply the presented steps, we generated three medium size test problems (Table 6.1).

Table 6.1: Setting of test problems

	Test No.1	Test No.2	Test No.3
Supplier	2	3	4
Plant	4	6	8
Distribution	10	15	20
Retailer	16	24	32
Customer	4	6	8
Collection/Inspection	4	6	8
Disposal	2	2	4
Optimal cost	1160	4100	11365
Number of facilities	42	62	84
Number of routs	456	966	1728

Since the framework of the proposed logistics network in this work is not the same as in previous studies, all parameters were generated randomly using uniform distributions

as given in Table 5.2:

6.3.1 Identify the objective function to be optimized

Taguchi suggested the use of a loss function for measurement. It is recommended to transform the value of the loss function into a signal to noise ratio $S/N = SNR$ [198]. Since the signals have a very wide dynamic range, they are mostly expressed using:

$$S/Nratio = -10 \log \left[\frac{\sum_{i=1}^{n} f_i^2}{2} \right]. \tag{6.1}$$

where f_i denotes the fitness value for run number i and n defines the number of repeated experiments. Three different categories are classified in order to analyze the S/N ratio: "nominal is better", "larger is better", and "smaller is better" [194; 211]. As the aim of this study is minimizing the total cost, "smaller is better" is considered. The best results are obtained when the total cost is minimized. This condition results in considering a minus in (6.1) as the S/N ratio has "the larger the better" characteristic.

6.3.2 Identify the independence factors and number of level for each

According to a comprehensive review [170; 108; 24], there are four independent factors that need to be investigated regarding the proposed algorithm. These factors are: population size, cross over rate, local search iteration, and number of iterations. As all of these factors have a noticeable effect on the results of the proposed model, it is important to assess all for the efficiency criterion [24]. Three different levels for each factor are considered according to the information presented in Table 6.2.

Table 6.2: Selected parameters and levels

Parameters(factors)	Level 1	Level 2	Level 3
Population size	60	100	140
Cross over rate	0.3	0.5	0.7
Local search iteration	4	8	16
Number of iteration	20	30	40

6.3.3 Select a suitable orthogonal array and construct the matrix

The Taguchi method deals with two main tools: the "S/N ratio" and a special set of arrays called the "orthogonal array". Compared to many statistical designs, the orthogonal array shows a significant success [81]. This method allows us to reduce the number of experiments while a more reliable estimation for parameter optimization can be reached. Previous studies presented many orthogonal arrays according to the problem they are facing. Still, selecting a proper orthogonal array is difficult, as the reported orthogonal arrays in the literature are not able to cover all possibilities [81]. To overcome this problem, some standard orthogonal arrays have been designed according to the number of factors and levels. In order to choose an appropriate orthogonal array, a minimum number of experiments need to be carried out first, based on the total amount of freedom [222]:

$$N_{Taguchi} = 1 + \sum_{i=1}^{n_f} (L_i - 1) \tag{6.2}$$

where n_f denotes the number of factors and L defines the number of levels of each factor. For the salient case, there are four different factors with 3 levels each, the degree of freedom is obtained $1 + (4 * 2) = 9$. It means at least 9 runs need to be carried out and another selection of an orthogonal array with more runs may be used to increase accuracy.

Here, we utilized Minitab software to find an automatic design of the orthogonal array as well as for the analysis of experiments. Minitab is a statistics package to analyze data and interpret the results. Minitab software suggested using L_9 orthogonal array, i.e. 9 instead of $3^4 = 81$ experiments. The details of the proper orthogonal array are shown in Table 6.3:

Table 6.3: The orthogonal array $L_9(3^4)$

trial	Population size	Crossover rate	Local search iteration	Number of iteration
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

6.3.4 Conduct the matrix simulation

In accordance with the presented orthogonal array mentioned in Table 6.3, simulations were carried out with their factors and levels. The scheme with the specified values of the factors is demonstrated in Table 6.4. Each of these 9 experiments is repeated 10 times to generate more reliable averaged results.

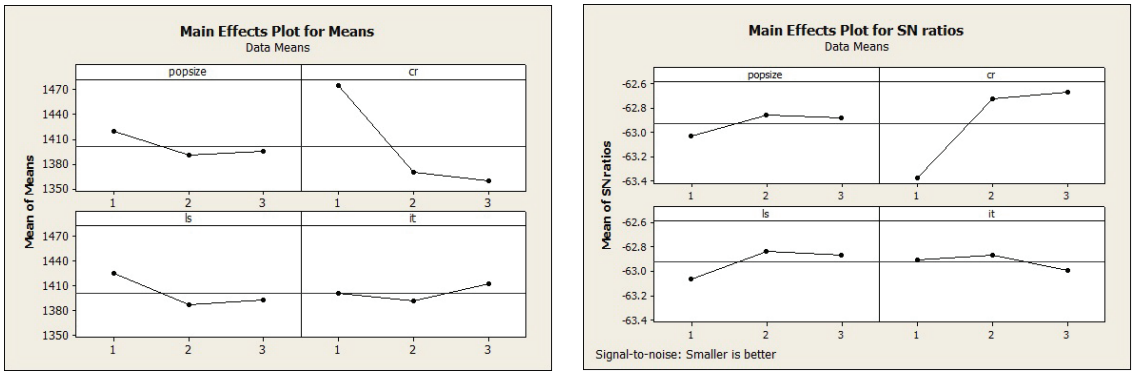
Table 6.4: Orthogonal array with the specified values

trial	Population size	Crossover rate	Local search iteration	Number of iteration
1	60	0.3	4	20
2	60	0.5	8	30
3	60	0.7	16	40
4	100	0.3	8	40
5	100	0.5	16	20
6	100	0.7	4	30
7	140	0.3	16	30
8	140	0.5	4	40
9	140	0.7	8	20

6.3.5 Examine and analyze data to predict the best parameter levels

According to the obtained results presented in Chapter 5, we realized that all the proposed parameters are important and need to be investigated further. Since we are dealing with a strategic problem, there is no time limitation. Still having some information regarding the best values of the parameters can help us to improve our results, applicable to other types of proceeding when time limitation is considered in advance (tactical level). On the other hand, as there is no similar case available in literature, here Taguchi's method is adapted for the generated test problems presented in Table 6.1. We generated three medium size test problems and an error percent is fixed as 0.05 that is acceptable for most of logistics problems [212]. It means we applied the parameters obtained from [24] that lead us to an error percent equal to or less than 0.05 percent. Once these parameters were determined, a high number of iterations was considered. In the next step, several runs were conducted to find the average number of iterations that the algorithm needs to find good results (that the error percent is equal to or less than 0.05). The specific time that these results were obtained, was recorded. The goal is discovering the best value of the parameters once the algorithm is forced to find results, in half of the obtained specific time, using Minitab software.

Figures 6.2(a) to 6.4(b) display the effect of the parameters on the objective function value and S/N ratio respectively.

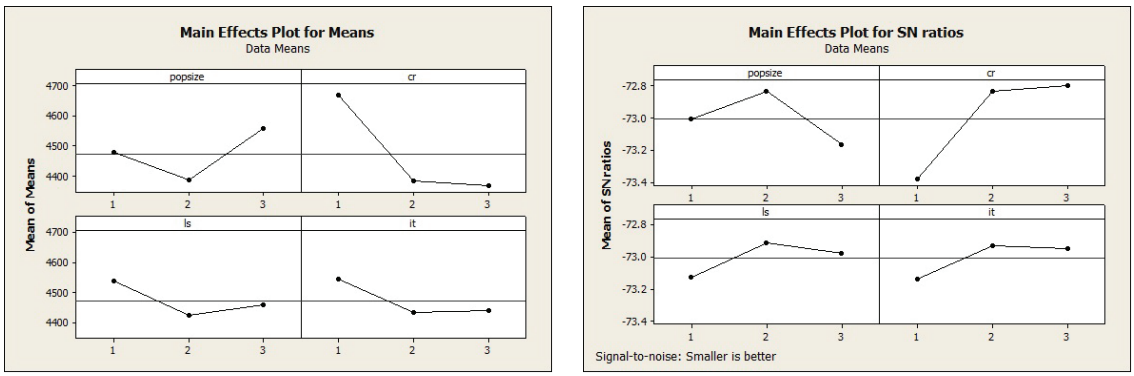


(a) Effect of parameters on total cost

(b) Effect of parameters on S/N ratio

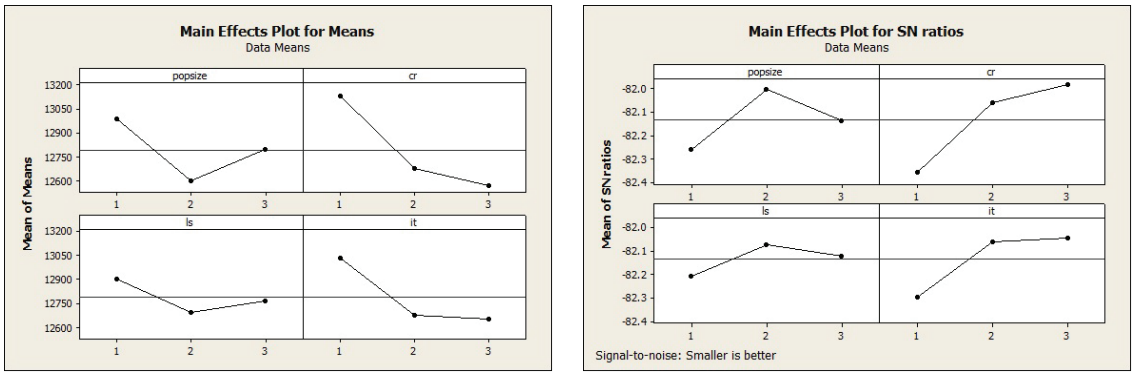
Figure 6.2: Problem No.1

Since the S/N ratio has the "larger is better" characteristic, the analysis of the results lead to the population size at level 2, cross over rate at level 3, local search iteration at



(a) Effect of parameters on total cost (b) Effect of parameters on S/N ratio

Figure 6.3: Problem No.2



(a) Effect of parameters on total cost (b) Effect of parameters on S/N ratio

Figure 6.4: Problem No.3

level 2 and number of iterations at level 2.

6.3.6 Confirmation simulation

Once the best level of the parameters were chosen according to the mentioned condition, the final step needed to be implemented. The idea of the confirmation experiment is to validate the results obtained from the experiments. A confirmation experiment was executed 10 times with the obtained combination of parameters. The actual value of cost obtained by Matlab, was calculated based on the experiments. The predicted value obtained by Minitab software, was computed according to statistical data. All results are provided in Table 6.5. The results show, the predicted value of cost obtained with

Table 6.5: Confirmation experiment data
(parameter combination Population size level 2, Cross over rate level 3, Local search iteration level 2, Number of iteration level 2)

trial	Predicted value	Actual value	Error percent
1	1324.67	1325	0.0002
2	4188.5	4248	0.014
3	12168.7	12375	0.016

the best level of parameters is very close to the value of the actual cost. This validates the presented Taguchi method for predicting the optimal levels of the parameters.

6.4 Summary and conclusions

In the current chapter, we applied the Taguchi method to design the optimal parameters of the proposed Memetic Algorithm. From the confirmation experiments, a good agreement between the predicted results and the actual results is observed. Using the Taguchi method in determining the optimal level of parameters is a new contribution of this study.

We would like to note that in another comprehensive research work we applied the Taguchi method to identify the most important parameters as well as the ranking amongst these parameters. We validated the obtained results in numerical experiments. The results show parameter population size has the largest effect on the outcome of the experiment, while local search iteration has the least effect. The obtained results were presented and published in the 6th International conference on Dynamics in Logistics,

(E. Behmanesh and J. Pannek) Bremen, Germany, 2018 [22].

After doing parameter analysis and obtaining enough information regarding the parameters involved in the proposed Memetic Algorithms, the verification of the proposed solution methodology using numerical experiments is the next task.

Chapter 7

Numerical results of applying the proposed Memetic Algorithm

7.1 Introduction

Once the parameters analysis of the proposed Memetic Algorithm is completed and the proper value for each parameter is selected, the algorithm is ready for the next stage: implementation and verification. The goal of this chapter is to evaluate the performance of the proposed Memetic Algorithm. Thus, the computational results obtained through experiments are considered. To illustrate the performance of the proposed Memetic Algorithm, LINGO optimization software serves as the basis of comparison for small size problems. In the large size cases that we are dealing with in real world, a classical Genetic Algorithm is presented to compare the results and show the efficiency of the Memetic Algorithm.

Some parts of the initial results of this chapter have been presented in the 20th International Conference on Engineering Optimization and Industrial Applications, London, United Kingdom, 2018 and also published in the Logistics Research journal, Springer, (E. Behmanesh and J. Pannek 2016), [19]. In addition, some of the final results of this chapter will be submitted to a high impact journal, under the title of "Memetic and Genetic algorithm for an integrated logistics network with flexible delivery path" (E. Behmanesh and J. Pannek 2018), [21].

7.2 Proposed Memetic algorithm

In this study we considered a Memetic Algorithm with a local search engine and a new chromosome representation. Evaluating the performance of our Memetic Algorithm is divided into two parts in order to solve the test instances: The first one is applied for small size problems using a commercial package (LINGO optimization software), and the second is employing a classical Genetic Algorithm as a second meta-heuristic algorithm. The procedure of the classical Genetic Algorithm is displayed through a flowchart in Figure 7.1 to clearly show the overall steps involved. To have a comparison between these two, the same structure presented in Chapter 4 is considered for both algorithms. The only difference is that the local search engine is replaced by a mutation operator in the proposed Memetic Algorithm. Thereafter, we consider some explanations regarding the mutation operator's application for the classical Genetic Algorithm.

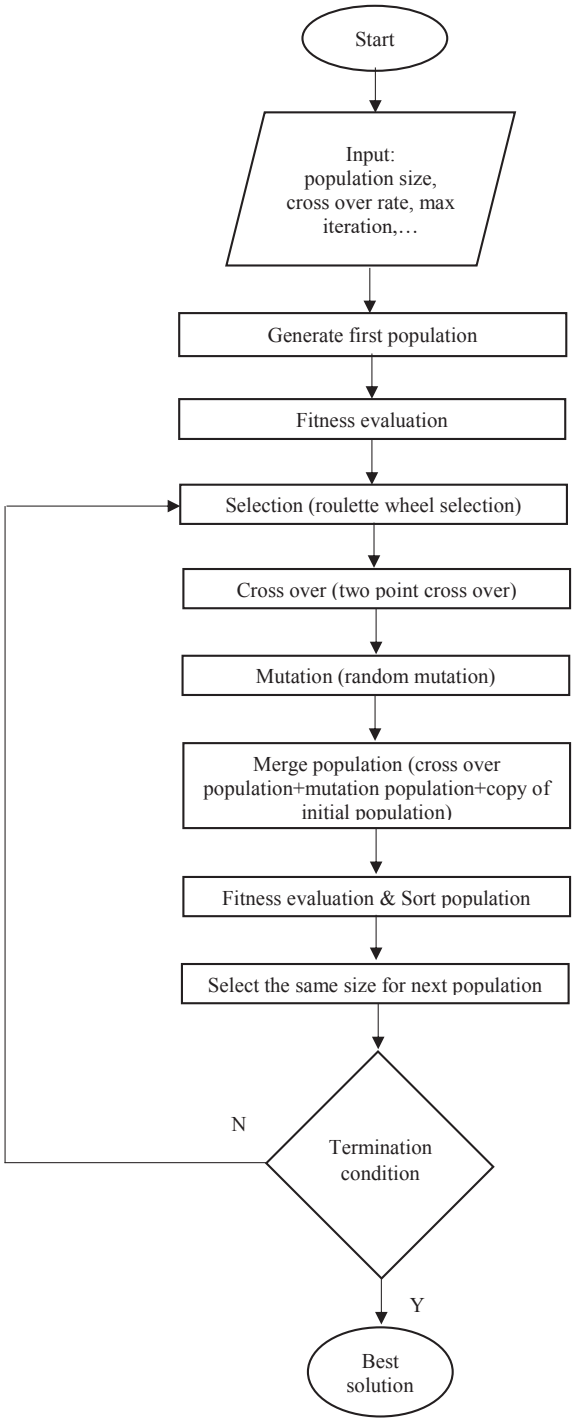


Figure 7.1: The flow diagram of the classical GA.

The mutation operator can create offspring from pairs of individuals and aims to vary the solution to avoid GA lead to a local optima. Then the mutation operator is repeated according to the mutation rate. For applying this operator, a single parent is selected and a random change is enacted on the selected parent by modifying one or more gene values. As this operator causes a very small change, a very low rate is used most of the time [70].

For the proposed Genetic Algorithm, a random mutation operator is used that simply selects a gene at random and replaces it with a random number from a feasible range. This method is selected according to the characteristic of the chromosomes. Figure 7.2 gives the illustration of the proposed random mutation.

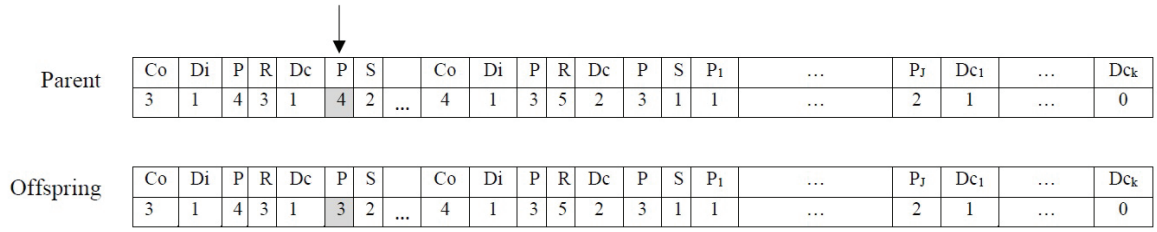


Figure 7.2: Mutation method.

7.3 Numerical setting

Within this section, 13 numerical examples with different sizes are carried out to show the performance of the proposed MA. The considered logistics network in this study is not exactly the same as in previous studies. Therefore, the size of the presented numerical examples are selected randomly as shown in Table 7.1. Performance of the proposed MA is proved via two different parts.
In the first part, we employed LINGO17 to provide optimal results for small size problems. To assess the accuracy and efficiency of the developed MA, we generated various test problems of different sizes are compared the outcomes obtained by our MA from Algorithm 5 with those of a branch-and-bound algorithm from LINGO17.
In the second part, a comparison between the classical GA and the proposed MA from Algorithm 4 is considered under the same condition. Also, the structure of the GA applied in this study is the same by the proposed MA, however, the local search mechanism is replaced by a classical mutation operator.

Table 7.1: Settings of test problems

Problem	Supplier	Plants	Distribution	Retailers	Customers	Collection	Disposal
1	2	3	3	4	3	2	1
2	2	2	5	8	2	2	1
3	2	4	6	10	2	2	1
4	2	3	8	9	3	3	2
5	2	4	10	16	4	4	2
6	3	6	15	24	6	6	2
7	4	8	20	32	8	8	4
8	6	12	30	48	12	12	6
9	6	14	32	54	14	14	6
10	8	16	40	64	16	16	8
11	10	18	36	80	18	18	10
12	10	20	40	84	20	20	10
13	12	24	40	96	24	24	12

The first seven test problems are small sized and the number of decisions variables are 98, 128, 209, 234, 468, 1006, and 1780 respectively. The remaining problems are large sized. Other parameters are generated randomly using uniform distributions according to the information presented in Table 5.2. The proposed MA and GA were developed in the MATLAB 2016. To test the robustness of the method each test problem has been implemented 10 times.

The test problems are solved by the proposed MA under a constant population size of 100, while three different population sizes: 100, 200, and 300 are considered for the classical GA. Regarding stopping conditions, we imposed a maximum iteration number of 200 as well as a maximum number of iterations without improvement of 6, 8, 10, 12, 20, 25, and 30 for our small size problems, respectively. For the large size problems, we increased the latter bound by 5. Also, to standardize all test problems, we set the number of local search iterations to be equal to the number of retailers L for each test problem. Therefore, the local search iteration number will be proper to the size of the test problems. The cross over rate of 0.4 is fixed for both algorithms, as well as the mutation rate of 0.2 which were known as better parameter settings for GAs [70].

Table 7.2: Results obtained by LINGO

Problem	Problem size	number of involved facilities	Solution	Ave time (s)
1	$2 \cdot 3 \cdot 3 \cdot 4 \cdot 3 \cdot 2 \cdot 1$	18	1920	0.05
2	$2 \cdot 2 \cdot 5 \cdot 8 \cdot 2 \cdot 2 \cdot 1$	22	2905	0.1
3	$2 \cdot 4 \cdot 6 \cdot 10 \cdot 2 \cdot 2 \cdot 1$	27	2345	0.12
4	$2 \cdot 3 \cdot 8 \cdot 9 \cdot 3 \cdot 3 \cdot 2$	30	2335	0.12
5	$2 \cdot 4 \cdot 10 \cdot 16 \cdot 4 \cdot 4 \cdot 2$	42	1160	0.14
6	$3 \cdot 6 \cdot 15 \cdot 24 \cdot 6 \cdot 6 \cdot 2$	62	4100	0.16
7	$4 \cdot 8 \cdot 20 \cdot 32 \cdot 8 \cdot 8 \cdot 4$	84	11365	0.17
8	$6 \cdot 12 \cdot 30 \cdot 48 \cdot 12 \cdot 12 \cdot 6$	126	-	
9	$6 \cdot 14 \cdot 32 \cdot 54 \cdot 14 \cdot 14 \cdot 6$	140	-	
10	$8 \cdot 16 \cdot 40 \cdot 64 \cdot 16 \cdot 16 \cdot 8$	168	-	
11	$10 \cdot 18 \cdot 36 \cdot 80 \cdot 18 \cdot 18 \cdot 10$	190	-	
12	$10 \cdot 20 \cdot 40 \cdot 84 \cdot 20 \cdot 20 \cdot 10$	204	-	
13	$12 \cdot 24 \cdot 40 \cdot 96 \cdot 24 \cdot 24 \cdot 12$	232	-	

7.4 Computational result

To evaluate the performance of the proposed MA and GA, LINGO17 is adapted to solve the optimization problem. Obtained results are presented in Table 7.2. Although LINGO provides optimal results for small size problems quickly, Table 7.2 indicates that LINGO is inappropriate for solving the large size problems and it runs out of memory.

According to the results presented in Table 7.3, the proposed MA is able to provide good solutions for the small size problems, which allows us to trust the algorithm for large size problems as well. The results obtained by the classical GA with three different sizes are presented in Table 7.4 to 7.6. We report the worst reward (*maxcost*), the best reward (*mincost*), and the average reward (*avecost*), which show the robustness of the obtained results.

To indicate the tradeoffs between the proposed MA and the classical GA, four comparison criteria are considered. First, the gap between the mean of the objective function values associated with the MA and the optimal solution obtained by LINGO optimization software (i.e., $F - Gap_{MA-LINGO}$), using formula (7.1). Secondly, the same procedure for the classical GA is determined (i.e., $F - Gap_{GA-LINGO}$), using formula (7.2). Third, the gap between the mean of the objectives value' associated with the MA and GA (i.e.,

$F - Gap_{MA-GA}$), using formula (7.3), and finally the relative gap between the mean of the CPU time associated with the MA and GA (i.e., $T - Gap_{MA-GA}$), using formula (7.4). The comparison is focused on two aspects: objectives function value and CPU time.

$$F - Gap_{MA-LINGO} = \frac{\mathrm{F_{MA}} - F^*_{\mathrm{LINGO}}}{F^*_{\mathrm{LINGO}}} \tag{7.1}$$

$$F - Gap_{GA-LINGO} = \frac{\mathrm{F_{GA}} - F^*_{\mathrm{LINGO}}}{F^*_{\mathrm{LINGO}}} \tag{7.2}$$

$$F - Gap_{MA-GA} = \frac{\mathrm{F_{GA}} - \mathrm{F_{MA}}}{\mathrm{F_{MA}}} \tag{7.3}$$

$$T - Gap_{MA-GA} = \frac{\mathrm{T_{MA}}}{\mathrm{T_{GA}}} \tag{7.4}$$

Table 7.3: Results for the proposed MA with $n = 100$ and cross over rate= 0.4 over 10 runs

Test problem	Min cost	Max cost	Ave cost	Ave time (s)
1	1920	1920	1920	4.06
2	2905	2905	2905	5.11
3	2345	2405	2351	10.34
4	2335	2535	2355	32.7
5	1160	1340	1185	35.05
6	4100	4600	4222	97.6
7	11365	12095	11814	261.5
8	16588	18189	17300.5	2070
9	12358	15133	13330.6	2633.3
10	21585	26332	23457.8	3970
11	22056	23088	22400	10200
12	27388	29171	27982.3	14666.6
13	29039	31607	29982.3	26333.3

Table 7.4: Results for GA with $n = 100$, cross over rate= 0.4 and mutation rate= 0.2 over 10 runs

Test problem	Min cost	Max cost	Ave cost	Ave time (s)
1	1920	1920	1920	1.75
2	2905	2905	2905	2
3	2345	2405	2369	3.1
4	2335	2535	2415	7.59
5	1160	1530	1225	10.2
6	**4190**	4720	4350	20.2
7	**11805**	12895	12292.3	50.1
8	18324	21974	19529.6	392
9	15430	16059	15706	583.3
10	25538	31389	28033.2	828
11	25488	27574	26660	1003.3
12	31746	34649	33681.33	1983.3
13	34103	43757	37667.2	2000

Table 7.5: Results for GA with $n = 200$, cross over rate= 0.4 and mutation rate= 0.2 over 10 runs

Test problem	Min cost	Max cost	Ave cost	Ave time (s)
1	1920	1920	1920	2.5
2	2905	2905	2905	3.2
3	2345	2405	2363	6.1
4	2335	2535	2395	13.36
5	1160	1360	1194	18.9
6	4100	4720	4279	42.3
7	**11775**	12425	12071	110.5
8	16956	18659	17742.5	744.5
9	13444	15321	14259	1126.6
10	22399	27525	25597.4	1507
11	25275	27503	26038	2000
12	30985	34602	32486	3183.3
13	33596	37277	35051.2	4010

Table 7.6: Results for GA with $n = 300$, cross over rate= 0.4 and mutation rate= 0.2 over 10 runs

Test problem	Min cost	Max cost	Ave cost	Ave time (s)
1	1920	1920	1920	4
2	2905	2905	2905	4.4
3	2345	2405	2357	9.5
4	2335	2535	2375	26.1
5	1160	1340	1192	27.4
6	4100	4720	4239	84.1
7	**11475**	12315	11933	203.3
8	16588	18625	17410	1333
9	13166	14293	13885	2016.6
10	21686	26396	24471.6	2590
11	24827	25544	25081.3	3700
12	30142	31871	31154	6456.2
13	33006	35807	34807	8410

Table 7.7: Comparison of results from LINGO17 and the proposed MA and GA

$OFV - GAP$

	$MA - LINGO$	$GA - LINGO$		
Test problem	$popsize = 100$	$popsize = 100$	$popsize = 200$	$popsize = 300$
1	0	0	0	0
2	0	0	0	0
3	0.0025	0.0102	0.0076	0.0051
4	0.0085	0.0342	0.0256	0.0171
5	0.021	0.056	0.052	0.043
6	0.0297	0.0658	0.063	0.06
7	0.0395	0.0815	0.077	0.076
average (%)	1.4	3.5	3.2	2.8

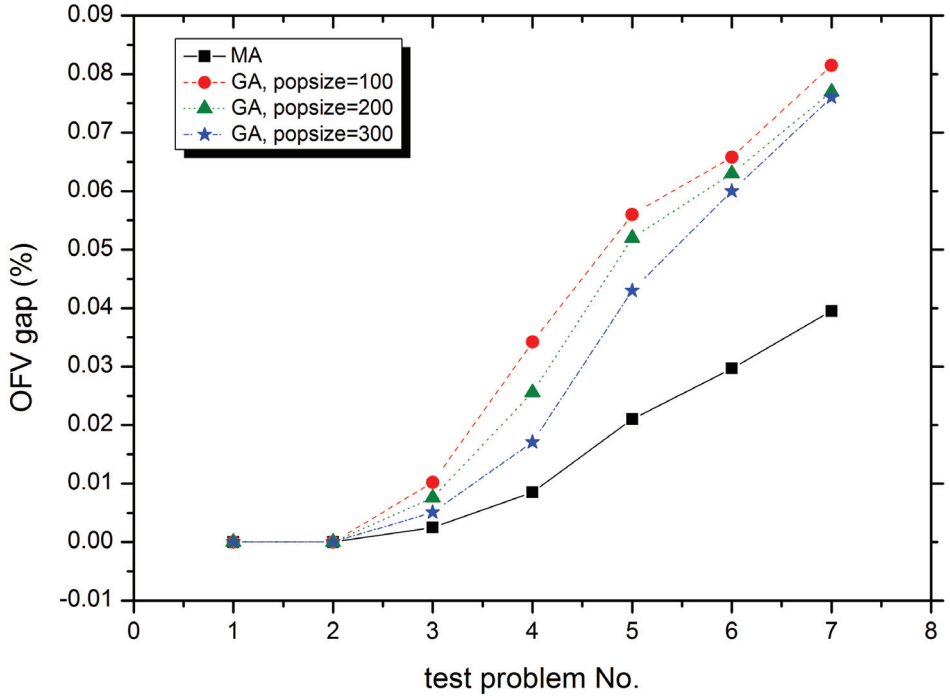


Figure 7.3: Comparison of results from LINGO17 and the proposed MA and GA

Table 7.3 to 7.6 present the comparison between the proposed MA and the classical GA. By assessing the results presented in Table 7.7, we can observe the average $F-Gap_{GA-LINGO}$ calculated by formula (7.2) for the first 7 test problems that are small sized is 3.5 percent, which is almost triple that of the $F-Gap_{MA-LINGO}$ result calculated by formula (7.1) reported by the MA (i.e., 1.4). This indicates the high accuracy of the proposed MA. Although the operation time is higher compared to LINGO, our implementation allows us to derive results for the large size problems. Hence, the proposed MA demonstrated that it can prepare sufficiently accurate solutions within efficient computation time for our integrated forward/reverse logistics problem with flexible delivery. By increasing the population size to 200 and 300 this number decreased to 3.2 and 2.8 respectively.

From another point of view, as is shown in bold in Table 7.4, (4190 for test problem number 6 and 11805 for test problem number 7) the classical GA is not able to find the optimal solution for test problem number 6 and 7, which our proposed MA is able to find (Table 7.3). By increasing the population size to 200, the GA can find optimal solution

for test problem number 6, but the problem still remains for test problem number 7 even by increasing the population size to 300 (Table 7.5 and 7.6).

A similar comparison for $F - Gap$ is provided in Figure 7.3 between the proposed MA and the classical GA based on the LINGO optimization software listed in Table 7.7. Excepting test problems no.1 and 2, which are both too small size problems, the proposed MA present a good quality solution in comparison with the GA. When the MA and GA have the same size of 100, the $F - Gap$ of the GA is always equal or higher than MA. By increasing population size to 200 and 300 an improvement, in the GA is observed, however, they are not really comparable with the results obtained by the proposed MA. On the other hand, increasing population size improves accuracy for the problems only slightly.

Table 7.8: Comparison of objective function value between the MA and GA

	$F - GAP_{(MA-GA)}$		
Test problem	$popsize = 100$	$popsize = 200$	$popsize = 300$
1	0	0	0
2	0	0	0
3	0.007	0.005	0.002
4	0.025	0.016	0.008
5	0.033	0.03	0.02
6	0.035	0.032	0.03
7	0.04	0.036	0.035
8	0.12	0.051	0.038
9	0.17	0.069	0.041
10	0.19	0.091	0.043
11	0.2	0.16	0.11
12	0.2	0.16	0.11
13	0.256	0.169	0.160
average (%)	9.8	6.3	4.6

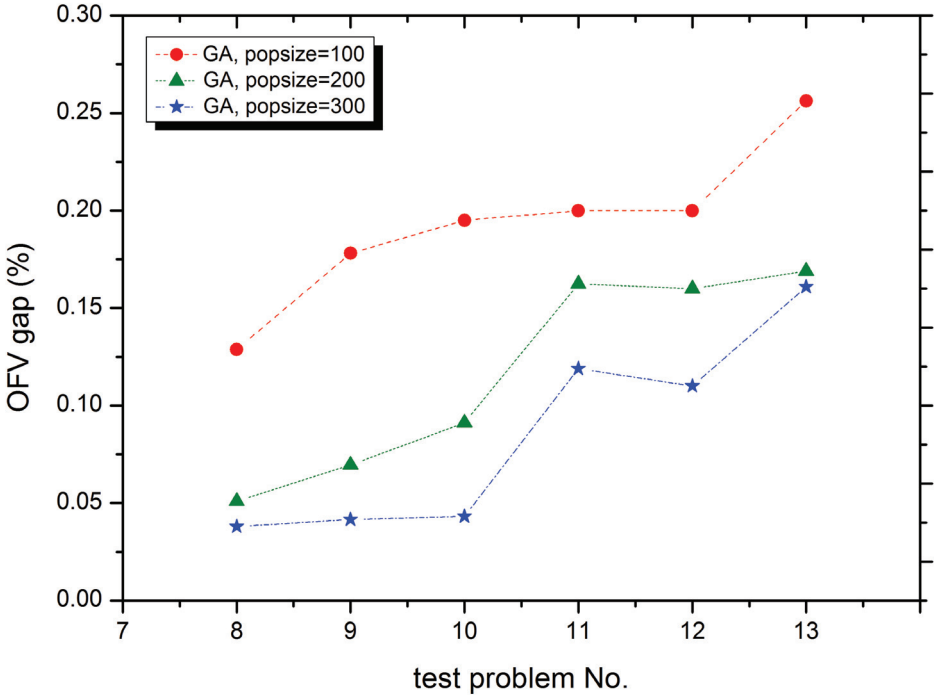


Figure 7.4: Comparison between the proposed MA and classical GA

The results in Table 7.3 and 7.4 are based on the equal population size of 100 for the MA and GA. The average $F - Gap_{MA-GA}$ presented in Table 7.8 obtained through the mentioned condition calculated by formula (7.3) is 9.8%, which shows a noticeable difference in the objective function value between these two algorithms. To improve the performance of the GA, population size is increased to 200. According to the results in Table 7.8 and 7.9, this change causes the average $F - Gap_{MA-GA}$ to be decreased to 6.3% while the average $T - Gap_{MA-GA}$ calculated by formula (7.4) was almost doubled. But still, a significant gap can be seen between the $F - Gap$ of the MA and GA. To have more improvement, the population size of the GA is increased up to 300. The results show the average $F - Gap_{MA-GA}$ is reduced to 4.6 but the average $T - Gap_{MA-GA}$ was again doubled.

A same comparison for $F - Gap$ is shown in Figure 7.4 between the proposed MA and the classical GA with three different population size, provided in Table 7.8. Although increasing in population size resulted in good improvement in the performance of the

GA, it does not show the GA as a comparable algorithm to the proposed MA.

From another aspect, Figure 7.5 provides an example of the proposed MA and GA convergence during 100 iteration related to problem No.7. This figure reveals that the convergence of the MA is very sharp in comparison with the GA. The proposed MA found the optimal solution after 70 iteration, while the GA could not find the optimal solution in 100 iteration, even by increasing population size.

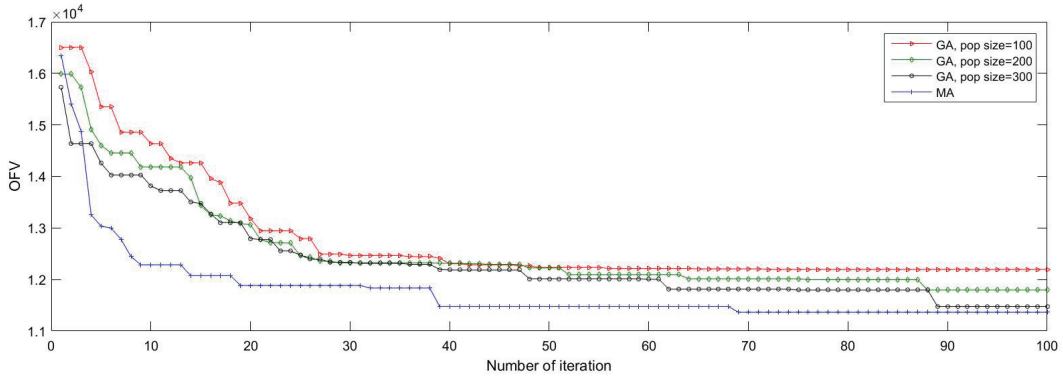


Figure 7.5: Comparison of the convergence between the proposed MA and classical GA

Table 7.9: Comparison of CPU time between the MA and GA

	$CT - GAP_{(MA-GA)}$		
Test problem	$popsize = 100$	$popsize = 200$	$popsize = 300$
1	2.32	1.62	1.015
2	2.555	1.59	1.16
3	3.33	1.69	1.08
4	4.3	2.44	1.25
5	3.43	1.854	1.27
6	4.83	2.30	1.16
7	5.219	2.36	1.28
8	5.28	2.78	1.55
9	4.51	2.33	1.30
10	4.79	2.63	1.532
11	10.1	5.1	2.756
12	7.3	4.6	2.27
13	13.16	6.5	3.13
average	5.47	2.9	1.5

Generally speaking, the obtained results show that the GA cannot find an appropriate near optimal solution in large size problems for the proposed flexible, integrated, forward/reverse logistics network without enriching the algorithm with a powerful local search engine. For the presented network with a complex nature, lack of intensification of the classical GA prevents the algorithm from finding a good solution. We showed that the proposed Memetic Algorithm is able to efficiently find a good solution for small and large size problems.

7.5 Summary and conclusion

In this thesis we focused on a comprehensive mixed integer linear programming formulation for a seven stages closed-loop network design problem. We applied the extended random path direct delivery path representation-based Memetic Algorithm that has been developed in the preceding chapters. Within this chapter, we treated several numerical examples to verified correctness of the proposed method as well as to confirm the effectiveness of that using a commercial package and a classical GA. On the other hand, we

applied the commercial solver, to show the performance of the classical GA. Considering the scale of the test problems, the results display that the proposed Memetic Algorithm can effectively detect solutions that are close to optimal. Also the $OFV - Gap$ between the GA and MA is significantly larger, which shows the efficiency and capability of the proposed MA. At the end, the results confirm that the classical GA cannot obtain a good solution without applying a powerful local search engine, because of the complex nature of the proposed flexible, integrated, forward/reverse logistics network. In contrast, our MA approach produced high-quality solutions. Therefore we believe the presented method will be an efficient method to solve this kind of multi-stage logistics network design problems.

Chapter 8

Recapitulation

In this chapter, the main conclusions are summarized and outlook for future research is presented.

8.1 Conclusions

The focus of this thesis was on an integrated, flexible, multi-stages, single period and single product, forward/reverse logistics network design problem that is formulated as a mixed integer linear programming problem. The aim of the study is to minimize the total cost, including transportation and operation cost, by finding the optimal number and capacities of facilities as well as assigning products flows between them in the proposed network. The considered flexible, integrated, forward/reverse logistics network can support other industries, such as electronic and digital equipment industries and vehicle industries, therefore developing an efficient solution methodology has multilateral profit.

On the other hand, since network design problems belong to the class of NP-hard problems, and solving this kind of problem is still a critical need in this area, many research directions still require intensive future work.

In this study, we suggest a practical model as a decision support system that shows how contribution between different facilities can help to control the entirety of the supply chain and increase the efficiency of the network.

The major conclusions of this work can be summarized by directly responding to the questions posed in Chapter 1 as follows:

How to find an efficient assignment strategy to satisfy the source and destination requirements with the aim of minimizing the total cost?

Reducing network cost and delivery times as well as increasing customer satisfaction are the main requirements of each network and need to be considered before designing the network. Additionally, reverse logistics has become a particular point of interest recently, due to environmental concerns as well as cost saving by reusing returned products.

To manage logistics system efficiently in terms of cost and delivery time as well as increase customer satisfaction, flexibility and productively needs to be added to the system.

In this regard, we employed two additional delivery paths. Excepting normal delivery, which is established from any stage to another one, direct delivery and direct shipment are added to the network. In direct delivery skipping distribution center or retailer is defined and in direct shipment, a direct path from plant to customer is designed. Each of them, are applied separately in the real word, but considering all of them at the same time allows us to skip some stages and make the paths shorter. Considering three different delivery paths reveals a fully capacitated graph between plant and customer in the proposed network. Increasing customer satisfaction and decreasing delivery time are the advantages of applying three different delivery path.

We need to look at to the reverse distribution as industrial players are forced, but not equipped to manage the reverse flow. A reverse distribution can be considered independently through a separated design. By applying this approach, sub-optimal solutions may be generated. Combinations of forward and reverse flow in the content of integration design not only avoids the network falling into sub-optimal solutions, but also lets the facilities share a number of resources that could help to increase the performance of each supply chain network.

Once we assigned the framework of the considered strategy for the network, we need to describe the system using mathematical concepts and language. Therefore, the next question will be *How we can model the flexible, integrated, forward/reverse logistics network mathematically?* In this regard, we applied a mixed integer linear programming to model the proposed problem.

Flexible logistics network and integrated design can improve the flexibility and efficiency

of the supply chain network but make the problem more complex. On the other hand, network design problems belong to the class of NP-hard problems. Therefore, the next question was,

How to tackle this flexible integrated network as an NP-hard problem?

The basic problem is a network design problem, which is NP-hard and the combination with flexibility in delivery path and integration in design makes the search space of the problem much larger and more complex. Since traditional method fail to cope with this problem, particularly for large size problems, meta-heuristic algorithms are considered to tackle the proposed NP-hard problem for large sizes. Meta-heuristic algorithms are capable of reducing the search space and increasing the quality of solutions. By answering the above question, the next question is raised:

Which algorithm may be suitable, in the sense of accuracy and efficiency, as a solution methodology for the proposed flexible, integrated, forward/reverse supply chain network?

Meta-heuristic algorithms are divided into two main parts. The first one are "single-point based algorithms" such as: Tabu search, Simulated annealing, and hill climbing. By applying these algorithm, search in the search space is started and continued using a single point. The second group are well-known as "population based algorithms" such as Genetic Algorithms, Ant colony algorithms and Particle swarm algorithms. These algorithms search in the search space for a population of solutions, instead of single solution. As the quality of solutions by the multi-directional search features of population-based algorithms are better than those generated by single-point based algorithms, population-based algorithms are selected for this study.

Genetic Algorithms (GAs) were often one of the top options when facing network design problems [69]. But lack of enough intensification was always a weakness of the GAs. In this regard, a Memetic Algorithm with specialized encoding, initialization, and local search operator is adapted to optimize the design of the proposed supply chain network. To verify the correctness of the proposed method, two directions are considered using numerical experiments. First for small and medium size test problems, we apply a commercial package (LINGO optimization software in our study), which uses branch and bound to find optimal results. Once the optimum results are obtained, we can check the efficiency of the proposed algorithm by calculating an error percentage. The error

percentage can show the gap between the results obtained by the proposed MA and an optimal solution. The obtained error percentage proved that we can trust our MA for large size problems. Furthermore, we showed that the method can solve larger size problems that cannot be solved by LINGO. The other option was considered to show the accuracy and efficiency of the proposed Memetic algorithm for large size problems.

For large size problem, a classical Genetic algorithm is considered with the same structure by the proposed MA, to contrast their performance. The only difference between these two algorithms is the local search engine, which is replaced by a mutation operator in the Memetic Algorithm.

Once we are using any algorithm, we need to collect information regarding the parameters involved. This helped us to pursue our method effectively. Therefore, the following questions were raised regarding improving optimization based on parameter analysis:

How can different parameters effect on the results of the algorithm? What is the most important parameter and the importance of the order?

As the first step, we assessed the effect of the parameters of the solution methodology, such as, population size, cross over rate, number of iterations, and number of local search iterations. As we are dealing with a strategic problem and there is no time limitation, we observed that large population sizes, large numbers of local search iterations, and high cross over rates can improve the solution. However, computational time is concurrently increased. So, if a predefined acceptable error rate as well as computation time are given [108; 212], according to the demands of the manager, the respective parameters can be determined.

In the second step, the Taguchi method was adapted to design the optimal parameters of the proposed Memetic Algorithm. The analysis of the results lead to the specific number for the mentioned parameters based on some conditions. The above results helped us to improve optimization when we are dealing with other aspects of the problems like considering tactical level.

In the last step of analyzing parameters, we applied the Taguchi method to find the most important parameter and rank the latter. The results showed that population size has the most effect while number of local search iterations has the least effect on the

results.

8.2 Outlook

The contents of this thesis has successfully addressed an efficient approach for the proposed flexible integrated supply chain network. But there are still areas that need further improvement. For this reason, future work is recommended below:

Other objective functions, such as responsiveness, tardiness, and robustness can be considered as other goals in designing the proposed flexible, integrated, forward/reverse logistics network design problem. The implementation of these ideas needs an updated version of the algorithm that is capable of solve multi-objective models.

To be close to real world applications, adding inventory facility to the network and also considering multi-product, multi-capacity, and multi-period networks can be pursued.

Uncertainty in demand, capacity of facilities, and recovery rates can be considered and examined in a more analytical way to make the model closer to reality.

The priority-based representation can be applied to the proposed flexible, integrated, forward/reverse logistics network, and making a comprehensive comparison between the priority-based representation and the proposed extended random path direct encoding method can be considered.

The efficiency of the proposed algorithm can be increased based on some changes in the structure of the algorithm, such as applying different strategies for cross over, selection, chromosome representation, and local searches or in details such as applying local searches on some random chromosomes or adding new population in between of implementation to create some new search directions. Moreover, for each iteration, processing of the population itself can be paralleled by having several processors working on subpopulations of solutions simultaneously.

The applied extended random path-based direct encoding and combinatorial local search method can be used in other meta-heuristic algorithms, such as cloud theory-based simulated annealing algorithm. Afterwards, a comparison between the results obtained by the new algorithms and the proposed Memetic Algorithm can be done.

Many research directions still require intensive work in the area of closed-loop logistics network design problems. Moreover, since network design problems belong to the class of NP-hard problems, developing efficient solution methods is still a critical need in this area.

Bibliography

[1] H.S. Abdinnour. A hybrid heuristic for the uncapacitated hub location problem. *European Journal of Operational Research*, 106(2-3):489–499, 1998.

[2] F.N. Abuali, R.L. Wainwright, and D.A. Schoenefeld. Determinant factorization: a new encoding scheme for spanning trees applied to the probabilistics minimum spanning tree problem. In LJ. Eshelman, editor, *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 470–477, 1995.

[3] A. Afkar, M.M. Kaleibar, and A. Payani. Geometry optimization of double wishbone suspension system via genetic algorithm for handling improvement. *Journal of Vibroengineering*, 14:827–883, 2012.

[4] C.C. Aggarwal, J.B. Orlin, and R..P. Tai. Optimized crossover for the independent set problem. *Operation Research*, 42(2):226–234, 1997.

[5] J. Aguilar and A. Colmenares. Resolution of pattern recognition problems using a hybrid genetic/random neural network learning algorithm. *Pattern Analysis and Applications*, 1(1):52–61, 1998.

[6] Y. Ahn, J. Park, C.G. Lee, and J.W. Kim. Novel memetic algorithm implemented with ga (genetic algorithm) and mads (mesh adaptive direct search) for optimal design of electromagnetic system. *IEEE Transportation*, 46(6):1982–1985, 2010.

[7] I.M. Ali, S.M. Elsayed, T. Ray, and R.A. Sarker. Memetic algorithm for solving resource constrained project scheduling problems. In *IEEE Congress on Evolutionary Computation*, Sendai, Japan, 2015.

[8] R.A. Aliev, B. Fazlollahi, B.G. Guirimov, and R.R. Aliev. Fuzzy-genetic approach to aggregate productiondistribution planning in supply chain management. *Information Sciences*, 177:4241–4255, 2007.

[9] F. Altiparmak, M. Gen, L. Lin, and T. Paksoy. A genetic algorithm approach for multi-objective optimization of supply chain networks. *Computers and Industrial Engineering*, 51:197–216, 2006.

[10] S.A. Alumur, S. Nickel, F. Saldanha da Gamad, and V. Verter. Multi-period reverse logistics network design. *European Journal of Operational Research*, 220(1):67–78, 2012.

[11] A. Amiri. Designing a distribution network in a supply chain system: formulated and efficient solution procedure. *European Journal of Operational Research*, 171: 567–576, 2006.

[12] N. Aras, D. Aksen, and A.G. Tanugur. Locating collection centers for incentive dependent returns under a pick-up policy with capacitated vehicles. *European Journal of Operational Research*, 191:1223–1240, 2008.

[13] S. Athreya and Y.D. Venkatesh. Application of taguchi method for optimization of process parameters in imroving the surface roughness of lathe facing operation. *International Refereed Journal of Engineering and Science*, 1(3):13–19, 2012.

[14] A. Augugliaro, L. Dusonchet, and E. Riva-Sanseverino. Service restoration in compensated distribution networks using a hybrid genetic algorithm. *Electric Power Systems Research*, 46(1):59–66, 1998.

[15] P. Bahrampour, M. Safari, and M. Baghban Taraghdari. Modeling multi product multi stage supply chain network design. *Procedia Economics and Finance*, 36: 70–80, 2016.

[16] B.M. Baker and M.A. Ayechew. A genetic algorithm for vehicle routing problem. *Computers and Operations Research*, 30:787–800, 2003.

[17] J. Beasley and P. C. Chu. A genetic algorithm for the set covering problem. *European Journal of Operation Research*, 94(2):393–404, 1996.

[18] B. Becker and R. Drechsler. Ofdd based minimization of fixed polarity reed-muller expressions using hybrid genetic algorithms. In *IEEE International Conference on Computer Design: VLSI in Computers and Processor*, pages 106–111, Los Alamitos, CA, 1994.

[19] E. Behmanesh and J. Pannek. A memetic algorithm with extended random path encoding for a closed-loop supply chain model with flexible delivery. *Logistics Research*, 9(22):1–12, 2016.

[20] E. Behmanesh and J. Pannek. Modeling and random path-based direct encoding for a closed loop supply chain model with flexible delivery paths. In *the 7th IFAC Conference on Management and Control of Production and Logistics*, 2016.

[21] E. Behmanesh and J. Pannek. Memetic and genetic algorithm for an integrated logistics network with flexible delivery path. *To be submitted*, 2018.

[22] E. Behmanesh and J. Pannek. Ranking parameters of a memetic algorithm for a flexible integrated logistics network. In *6th International Conference on Dynamics in Logistics (LDIC 2018)*, 2018.

[23] E. Behmanesh and J. Pannek. Taguchi analysis for improving optimization of integrated forward/reverse logistics. *To be submitted*, 2018.

[24] E. Behmanesh and J. Pannek. The effect of various parameters of solution methomethod on a flexible integrated supply chain model. *Mathematical Problems in Engineering*, pages 1–14, 2018.

[25] A.H.W. Bos. Aircraft conceptual design by genetic/gradient-guided optimization. *Engineering Applications of Artificial Intelligence*, 11(3):377–382, 1998.

[26] D. Brown, C. Huntley, and A. Spillane. A parallel genetic heuristic for the quadratic assignment problem. In *3rd International Conference on Genetic Algorithms*, pages 406–415, 1989.

[27] T.N. Bui and B.R. Moon. Genetic algorithm and graph partitioning. *IEEE Transactions on Computers*, 45(7):841–855, 1996.

[28] T.N. Bui and B.R. Moon. GRCA: A hybrid genetic algorithm for circuit ratio-cut partitioning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(3):193–204, 1998.

[29] E.K. Burke and A.J. Smith. A memetic algorithm to schedule grid maintenance. In *International Conference on Computational Intelligence for Modelling Control and Automation*, pages 122–127, Vienna, 1999.

[30] E.K. Burke and A.J. Smith. A multi-stage approach for the thermal generator maintenance scheduling problem. In *Congress on Evolutionary Computation*, pages 1058–1092, Washington D.C., 1999.

[31] E.K. Burke and A.J. Smith. Hybrid evolutionary techniques for the maintenance scheduling problem. *IEEE Transactions on Power Systems*, 15(1):122 –128, 2000.

[32] E.K. Burke, K.S. Jackson, J.H. Kingston, and R.F. Wear. Automated timetabling: The state of the art. *The Computer Journal*, 40(9):565–571, 1997.

[33] E.K. Burke, J.P. Newall, and R.F. Weare. Initialisation strategies and diversity in evolutionary timetabling. *Evolutionary Computation*, 6(1):81–103, 1998.

[34] E.K. Burke, P.I. Cowling, P.D. Causmaecker, and G.V. Berghe. A memetic approach to the nurse rostering problem. *Applied Intelligence*, 15:199–214, 2001.

[35] S. Cavalieri and P. Gaiardelli. Hybrid genetic algorithm for a multiple-objective scheduling problem. *Journal of Intelligent Manufacturing*, 9(4):361–367, 1998.

[36] C. Cerrone, R. Cerulli, and A. Raiconi. Relations, models and a memetic approach for three degree-dependent spanning tree problems. *European Journal of Operational Research*, 232:442–453, 2014.

[37] N. Chaiyaratana and A.M.S. Zalzala. Hybridisation of neural networks and genetic algorithms for time-optimal contro. In *Congress on Evolutionary Computation*, pages 389–439, Washington D.C, 1999.

[38] R. Cheng and M. Gen. An evolution program for the resource-constrained project scheduling problem. *Computer Integrated Manufactoring*, 11(3):274–287, 1998.

[39] R. Cheng, M. Gen, and Y. Tsujimura. A tutorial survey of job-shop scheduling problems using genetic algorithms. ii. hybrid genetic search strategies. *Computers and Industrial Engineering*, 37(1-2):51–55, 1999.

[40] R.W. Cheng and M. Gen. Parrallel machine scheduling problems using memetic algorithms. *Computer and Industrial Engineering*, 33(3-4):761–764, 1997.

[41] P.C. Chu and J. Beasley. A genetic algorithm for the generalised assignment problem. *Computer and Operation Research*, 24:17–23, 1997.

[42] P.C. Chu and J.E. Beasley. A genetic algorithm for the multidimentional knapsack problem. *Journal of Heuristics*, 4:63–86, 1998.

[43] D. Corne, M. Dorigo, F. Glover, D. Dasgupta, P. Moscato, R. Poli, and K. Price. *New ideas in optimization*. McGraw-Hill, 1999.

[44] D. Costa, N. Dubuis, and A. Hertz. Embedding of a sequential procedure within an evolutionary algorithm for coloring problems in graphs. *Journal of Heuristics*, 1(1):105–128, 1995.

[45] C. Daniel. One at a time plans. *Journal of the America Statistical Association*, 68: 353–360, 1973.

[46] A. Dasci and V. Verter. A continuous model for proproduct-distribution system des. *European Journal of Operational Research*, 129:278–298, 2001.

[47] R. Dawkins. *The Selfish Gene*. Oxford University Press, 1976.

[48] N. Dellaert and J. Jeunet. Solving large unconstrained multilevel lot-sizing problems using a hybrid genetic algorithm. *International Journal of Production Research*, 38(5):1083–1099, 2000.

[49] B. Dengiz, F. Altiparmak, and A. E. Smith. Local search genetic algorithm for optimal design of reliable networks. *IEEE Transactions on Evolutionary Computation*, 1(3):179–188, 1997.

[50] S. Dowlatshahi. Developing a theory of reverse logistics. *Interfaces*, 30:143–155, 2000.

[51] F. Du and G.W. Evans. A bi-objective reverse logistics network analysis for post-sale service. *Computers and Operations Research*, 35:2617–2634, 2008.

[52] C. Eckert and J. Gottlieb. *Direct representation and variation operators for the fixed charge transportation problem, in parallel problem solving from nature*. Springer Berlin Heidelberg, 2002.

[53] M. El-Sayed, N. Afia, and A. El-Khatboty. A stochastic model for forward-reverse logistics network design under risk. *Computers and Industrial Engineering*, 58: 423–431, 2010.

[54] B.E. Engin, M.O. Sümbül, O. Engin, M.E. Baysal, and A. Sarucan. A memetic algorithm to solve the open shop scheduling problem. In *6th International Conference on Modeling, Simulation, and Applied Optimization*, Istanbul, Turkey, 05 2015.

[55] C. Ersoy and S.S. Panwar. Topological design of interconnected lan/man networks. *IEEE Journal on Selected Areas in Communications*, 11(8):1172–1182, 1993.

[56] B. Fahimnia, R.Z. Farahani, and J. Sarkis. Integrated aggregate supply chain planning using memetic anetworks ‗ a performance analysis case study. *International Journal of Production Research*, 51(18), 2013.

[57] A. Fallah-Tafti, R. Sahraeian, R. Tavakkoli-Moghaddam, and M. Moeinipour. An interactive possibilistic programming approach for a multi-objective close-loop supply chain network under uncertainty. *International Journal of Systems Science*, 45 (3):283–299, 2014.

[58] J. Fang and Y. Xi. A rolling horizon job shop rescheduling strategy in the dynamic environment. *International Journal of Advanced Manufacturing Technology*, 13(3): 227–232, 1997.

[59] M. Fen. The opportunities, challenges and tendency of reverse logistics. In Liangzhong Jiang, editor, *Proceedings of the 2011 International Conference on Informatics, Cybernetics, and Computer Engineering (ICCE2011)*, pages 137–143. Springer Berlin Heidelberg, 2012.

[60] J. Figueiredo and S. Mayerle. Designing minimum-cost recycling collection networks with required throughput. *Transportation Research Part E*, 44:731–752, 2008.

[61] M. Fleischmann, H.R. Krikke, R. Dekker, and S.D.P. Flapper. A characterization of logistics networks for product recovery. *Omega*, 28:653–666, 2000.

[62] M. Fleischmann, P. Beullens, J.M. Bloemhof-Ruwaard, and L. Wassenhove. The impact of product recovery on logistics network design. *Production and Operations Management*, 10:156–173, 2001.

[63] C. Fleurent and J.A. Ferland. Genetic and hybrid algorithms for graph coloring. *Annals of Operations Research*, 63:437–461, 1997.

[64] P. Franca, A. Mendes, and P. Moscato. A memetic algorithm for the total tardiness single machine scheduling problem. *European Journal of Operational Research*, 132 (1):224–242, 2001.

[65] B.L. Garcia, P. Mahey, and L.J. Leblanc. Iterative improvement method for a multiperiod network design problem. *European Journal of Operational Research*, 110(1):150–165, 1998.

[66] M. Gen and R. Cheng. *Genetic algorithms and engineering design.* New York: Wiley, 1997.

[67] M. Gen and R. Cheng. *Genetic algorithms and engineering optimization.* New York: Wiley, 2000.

[68] M. Gen, K. Ida, and L. Yinzhen. Bicriteria transportation problem by hybrid genetic algorithm. *Computers and Industrial Engineering*, 35(1-2):363–366, 1998.

[69] M. Gen, R. Cheng, and S.S. Oren. Network design techniques using adapted genetic algorithms. *Advanced in Engineering Software*, 32:731–744, 2001.

[70] M. Gen, A. Kumar, and J.R. Kim. Recent network design techniques using evolutionary algorithms. *International Journal of Production*, 98:251–261, 2005.

[71] M. Gen, F. Altiparmak, and L. Lin. A genetic algorithm for two-stage transportation problem using priority-based encoding. *Operations Research-Spectrum*, 28: 337–354, 2006.

[72] M. Gen, R. Cheng, and L. Lin. *Network model and optimization: Multiobjective genetic algorithm approach*. Springer, 2008.

[73] M. Gorges-Schleuter. *Genetic algorithm and population structures-A massively parallel algorithm*. PhD thesis, University of Dortmund, Germany, 1991.

[74] J. Gottlieb and L. Paulmann. Genetic algorithms for the fixed charge transportation problem. In *IEEE World Congress on Evolutionary Computation*, pages 330–335, 1998.

[75] K.H. Govindan, H. Soleimani, and D. Kannan. Reverse logistics and closed supply chain: A comprehensive review to explore the future. *European Journal of Operational Research*, 240(3):603–626, 2015.

[76] J.B. Grimbleby. Hybrid genetic algorithms for analogue network synthesis. In *Congress on Evolutionary Computation*, pages 1781–178, Washington D.C., 1999.

[77] O.C.L. Haas, K.J. Burnham, and J.A. Mills. Optimization of beam orientation in radiotherapy using planar geometry. *Physics in Medicine and Biology*, 43(8): 2179–2193, 1998.

[78] A.B. Hadj-Alouane, J.C. Bean, and K.G. Murty. A hybrid genetic/optimization algorithm for a task allocation problem. *Journal of Scheduling*, 2(4):189–201, 1999.

[79] W. Hart. *Adaptive global optimization with local search*. PhD thesis, University of California, San Diego, CA, 1994.

[80] W.E. Hart, N. Krasnogor, and J.E. Smit. *Recent advances in memetic algorithms*. Springer-Verlag, 2005.

[81] A.S. Hedayat and N.J.A. Sloane. *Orthogonal arrays: Theory and Applications*. Springer Verlag: New York, 1999.

[82] M.H.A. Hendriks, B. Voeten, and L. Kroep. Human resource allocation in a multi-project r and d environment. *International Journal of Production Management*, 17 (2):181–188, 1999.

[83] M. Hifi. A genetic algorithm-based heuristic for solving the weighted maximum indeepencent set and some equivalent problem. *Journal of the Operational Research Socirty*, 48(6):612–622, 1997.

[84] Y. Hinojosa, J. Kalcsics, S. Nickel, J. Puerto, and S. Velten. Dynamic supply chain design with inventory. *Computers and Operations Research*, 35(2):373–391, 2008.

[85] E. Hopper and B. Turton. A genetic algorithm for a 2d industrial packing problem. *Computers and Industrial Engineering*, 37(1-2):375–378, 1999.

[86] S. Huysman, S. Debaveya, T. Schaubroeck, S. De Meester, F. Ardente, F. Mathieux, and J. Dewulf. The recyclability benefit rate of closed-loop and open-loop systems: A case study on plastic recycling in flanders. *Resources , Conservation and Recycling*, 101:53–60, 2015.

[87] H. Ishibuchi, T. Yoshida, and T. Mura. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Transactions on Evolutionary Computation*, 7(2):204–223, 2003.

[88] R. Jamshidi, S.M.T. Fathemi Ghomi, and B. Karimmi. Multi-objective green supply chain optimization with a new hybrid memetic algorithm using the Taguchi method. *Scientia Iranica Journal*, 19(6):1876–1886, 2012.

[89] A. Jaszkewicz. A comparative study of multiple-objective metaheuristics on the bi-objective set covering problem and the pareto memetic algorithm. *Annals of Operations Research*, 131:135–158, 2004.

[90] S.N. Jat and S. Yang. A memetic algorithm for the university course timetabling problem. In *20th IEEE International Conference on Tools with Artificial Intelligence*, 2008.

[91] V. Jayaraman and A. Ross. A simulated annealing mathodology to distribution network design and management. *European Journal of Operational Research*, 144: 629–645, 2003.

[92] V. Jayaraman, RA. Patterson, and E. Rolland. The design of reverse distribution networks: model and solution procedure. *European Journal of Operational Research*, 150:128–149, 2003.

[93] J.B. Jo, L. Yinzhen, and M. Gen. Nonlinear fixed charge transportation problem by spanning tree-based genetic algorithm. *Computer and Industrial Engineering*, 53:290–298, 2007.

[94] G. Kannan, S. Pokharel, and P.S. Kumar. A hybrid approach using ism and fuzzy tosis for the selection of reverse logistics provider. *Resource, Conservation and Recycling*, 54(1):28–36, 2009.

[95] I.E. Kassotakis, M.E. Markaki, and A.V. Vasilakos. A hybrid genetic approach for channel reuse in multiple access telecommunication networks. *IEEE Journal on Selected Areas in Communications*, 18(2):234–243, 2000.

[96] P. Kaweegitbundit. A single machine scheduling problem with earliness and tardiness penalties using memetic algorithm. *Advanced Materials Research*, pages 2353–2357, 2011.

[97] S. Khalifehzadeh, M. Seifbarghy, and B. Naderi. A four-echelon supply chain network design with shortage: Mathematical modeling and solution methods. *Journal of Manufacturing Systems*, 35, 2015.

[98] W. Kilibi, A. Martel, and A. Guitouni. The design of robust value-creating supply chain networks: a critical review. *European Journal of Operational Research*, 203: 283–293, 2010.

[99] K.W. Kim, Y.S. Yun, J.M. Yoon, M. Gen, and G. Yamazaki. Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling. *Computation in Industry*, 56:143–160, 2005.

[100] T.S. Kim and G.S. May. Intelligent control of via formation by photosensitive bcb for mcm-l/d applications. *IEEE Transactions on Semiconductor Manufacturing*, 12:503515, 1999.

[101] H.J. Ko and G.W. Evans. A genetic-based heuristic for the dynamic integrated forward/reverse logistics network for 3pls. *Computers and Operations Research*, 34:346–366, 2007.

[102] H.R. Krikke, V. Harten, and A. Schuur. Reverse logistic network redesign for copiers. *OR Spectrum*, 21:381–409, 1999.

[103] S. Krikpatrick. Optimizing by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 34(5):975–986, 1984.

[104] Y. Kristianto, A. Gunasekaran, P. Helo, and Y. Hao. A model of resilient supply chain network design: A two-stage programming with fuzzy shortest path. *Expert Systems with Applications*, 41(1):39–49, 2014.

[105] R.M. Krzanowski and J. Raper. Hybrid genetic algorithm for transmitter location in wireless networks. *Computers, Environment and Urban Systems*, 23(5):359–382, 1999.

[106] S. Kumanam and K. Raja. Multi-project scheduling using a heuristic and memetic alogrithm. *Journal for Manufacturing Science and Production*, 10(3-4):249–256, 2011.

[107] C.Y. Lee. Genetic algorithm for single machine job scheduling woth common due data and symmetric penalties. *Journal of the operation Research Society of Japan*, 37(2):83–95, 1994.

[108] D. Lee and M. Dong. A heuristic approach to logistics network design for end-of lease computer product recovery. *Transportation Research Part E*, 44:455–474, 2007.

[109] D. Levine. *Meta-heuristics: Theory and Applications*. Springer, 1996.

[110] B. Li, Z. Zhou, W. Zou, and D. Li. Quantum memetic evolutionary algorithm-based low-complexity signal detection for underwater acoustic sensor networks. *IEEE Transactions on Evolutionary Computation*, 42(5):626–640, 2012.

[111] C.F. Liaw. A hybrid genetic algorithm for the open shop scheduling problem. *European Journal of Operational Research*, 124(1):28–42, 2000.

[112] C.C. Lin, K.C. Chen, and W.J. Chuang. Motion planning using a memetic evolution algorithm for swarm robots. *International Journal of Advanced Robotic Systems*, 9:1–9, 2012.

[113] L. Lin, M. Gen, and X. Wang. Integrated multistage logistics network design by using hybrid evolutionary algorithm. *Computers and Industrial Engineering*, 56: 854–873, 2009.

[114] Y.C. Lin. Mixed-integer constrained optimoptimization on memetic algprithm. *Journal of Applied Research and Technology*, 11:242–250, 2013.

[115] A. Lipowski and D. Lipowska. Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications*, 391(6):2193–2196, 2012.

[116] O. Listes and R. Dekker. A stochastic approach to a case study for product recovery network design. *European Journal of Operational Research*, 160:268–287, 2005.

[117] C. Liu and A. Kroll. Memetic algorithms for optimal task allocation in multi-robot systems for inspection problems with cooperative tasks. *Soft Computing*, 19(3): 567–584, 2015.

[118] C. Liu and B. Li. Memetic algorithm with adaprive local search depth for large scale global optimization. In *IEEE Congress on Evolutionary Computation*, 2014.

[119] H.R. Lourenco. Adaptive search heuristics for the generalized assignment problem. *Mathware and Soft Computing*, 7:1–15, 2000.

[120] D. Louwers, B. Kip, E. Peters, F. Souren, and S. Flapper. A facility location allocation model for reusing carpet materials. *Computers and Industrial Engineering*, 36:855–869, 1999.

[121] Z. Lu and N. Bostel. A facility location model for logistics systems including reverse flows: the case of remanufacturing activities. *Computers and Operations Research*, 34:299–323, 2007.

[122] Z. Lu and J.K. Hao. A memetic algorithm for graph coloring. *European Journal of Operational Research*, 203:241–250, 2010.

[123] C.J.A.T. Machado and J.B. Cunha. A memetic algorithm for logic circuit design. In *International Conference on Dynamical Systems and Control*, pages 598–603, 2005.

[124] R. Maheswaran, S.G. Ponnambalam, and C. Aravindan. A meta-heuristic approach to single machine scheduling problems. *The International Journal of Advanced Manufacturing Technology*, 25:772–776, 2005.

[125] M. Maric, Z. Stanimirovic, and P. Stanojevic. An efficient memetic algorithm for the uncapacitated single allocation hub location problem. *Soft Computing*, 17(3): 445–466, 2013.

[126] I. Maslennikova and D. Foley. Xerox's approach to sustainability. *Interfaces*, 30: 226–233, 2000.

[127] Y. Mei, K. Tang, and X. Yao. Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem. *IEEE Transactions on Evolutionary Computation*, 15(2):151–165, 2011.

[128] M.T. Melo, S. Nickel, and F. Saldanha da Gama. Facility location and supply chain management - a review. *European Journal of Operational Research*, 196:401–412, 2009.

[129] A.S. Mendes, F.M. Muller, P.M. Franca, and P. Moscato. Comparing meta-heuristic approach for parallel machine scheduling problems with sequence-dependent setup times. In *15th International Conference on CAD/CAM Robotics and Factories of the Future*, 1999.

[130] P.M. Francaand A.S. Mendes and P. Moscato. Memetic algorithm to minimize tardiness on a single machine with sequsequence-depen setup times. In *5th International Conference of the Decision Sience Institute*, pages 1708–1710, 1999.

[131] P. Merz and B. Freisleben. A genetic local search approach to the quadratic assignment problem. In *Seventh International Conference on Genetic Algorithms*, pages 465–472, 1997.

[132] P. Merz and B. Freisleben. On the effectiveness of evolutionry search in high-dimensional nk-landscapes. In *IEEE International Conference on Evolutionary Computation*, pages 741–745, 1998.

[133] P. Merz and B. Freisleben. A comparison of memetic algorithms, tabu search, and ant colonies for the quadratic assignment problem. In *Congress on Evolutionary Computation*, pages 2063–2070, 1999.

[134] P. Merz and B. Freisleben. Fitness landscape, memetic algorithms, and greedy operators for graph bipartitioning. *Evolutionary Computation*, 8(1):61–91, 2000.

[135] Z. Michalewicz. *Genetic algorithms + data Structures = evolution Programs*. Springer-Verlag, 1996.

[136] Z. Michalewicz, G.A. Vignaux, and M. Hobbs. A non-standard genetic algorithm for the transportation problem. *Operations Research Society of America*, 3:307–316, 1991.

[137] H. Mifflin. *The american heritage dictionary of the english language*. Houghton Mifflin Harcourt, 2000.

[138] M. Mignotte, C. Collet, P. Perez, and P. Bouthemy. Hybrid genetic optimization and statistical model based approach for the classification of shadow shapes in sonar imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):129–141, 2000.

[139] D.M. Miller, H.C. Chen, J. Matson, and Q. Lin. A hybrid genetic algorithm for the single machine scheduling peroblem. *Journal of Heuristics*, 5(4):437–454, 1999.

[140] H. Min and H. Ko. The dynamic design of a reverse logistics network from the perspective of third-party logistics service providers. *International Journal of Production Economics*, 113:176–192, 2008.

[141] H. Min, C.S. Ko, and H.J. Ko. The spatial and temporal consolidation of returned product in a closed-loop supply chain network. *Computers and Industrial Engineering*, 51(2):309–320, 2006.

[142] L. Min and W. Cheng. Identical parallel machine scheduling problem for minimizing the makespan using genetic algorithm combined with simulated annealing. *Chinese Journal of Electronics*, 7(4):317–321, 1998.

[143] X.G. Ming and K.L. Mak. A hybrid hopfield network-genetic algorithm approach to optimal process plan selection. *International Journal of Production Research*, 38(8):1823–1839, 2000.

[144] T.T. Minh, T.V. Hoai, and T.T.N. Nguyet. A memetic algorithm for waste collection vehicle routing problem with time windows and conflicts. In *International Conference on Computational Science and Its Applications*, pages 485–499, 2013.

[145] R.T. Moghaddam, N. Safaei, and F. Sassani. A memetic algorithm for the flexible flow line scheduling problem with processor blocking. *Computers and Operations Research*, 36:402–414, 2009.

[146] A. Monfroglio. Hybrid genetic algorithms for timetabling. *International Journal of Intelligent Systems*, 11(8):477–523, 1996.

[147] A. Montfroglio. Hybrid genetic algorithms for a rostering problem. *Software Practice and Experience*, 26(7):851–862, 1996.

[148] P. Moscato and C. Cotta. An introduction to memetic algorithms. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 19:131–148, 2003.

[149] P. Moscato and C. Cotta. *A gentle introduction to memetic algorithms*, volume 57. Springer, 2003.

[150] T. Murata and H. Ishibuchi. Performance evaluation of genetic algorithm for flowshop scheduling problems. In *The First IEEE Conference on Evolutionary Computation*, pages 812–817, 1994.

[151] T. Murata, H. Ishibuchi, and H. Tanaka. Genetic algorithm for flowshop scheduling problems. *Computer and Industrial Engineering*, 30(4):1061–1071, 1996.

[152] M. Musil, M.J. Wilmut, and N.R. Chapman. A hybrid simplex genetic algorithm for estimating geoacoustic parameters using matched-field inversion. *IEEE Journal of Oceanic Engineering*, 24(3):358–369, 1999.

[153] R.T. Nakatsu. Designing business logistics networks using model-based reasoning and heuristic-based searching. *Expert Systems with Applications*, 29:735–745, 2005.

[154] J. Nalepa and M. Blocho. Adaptive memetic algorithm for minimizing distance in the vehicle routing problem with time windows. *Soft Computing*, 20(6):2309–2327, 2016.

[155] J. Quariguasi Frota Neto, G. Walther, J. Bloemhof, A.E.E. Van Nunen, and T. Spengler. From closed-loop to sustainable supply cchain: the weee case. *International Journal of Production Research*, 48(15):4463–4481, 2010.

[156] A.L. Nordstrom and S. Tufekci. A genetic algorithm for the talent scheduling problem. *Computers and Operations Research*, 21(8):927–940, 1994.

[157] A.G.N. Novaes, J.E.S. De-Cursi, and O.D. Graciolli. A continuous approach to the design of physical distribution systems. *Computers and Operations Research*, 27 (9):877–893, 2000.

[158] R.O. Oladele and O.T. Oladele. Comparative evaluation of genetic algorithm and memetic algorithm for optimal topology design of computer networks. *Journal of Computer Science and Its Application*, 21(2), 2014.

[159] E. Olivares-Benitez, R.R. Mercado, and J.L. Gonza. A metaheuristic algorithm to solve the selection of transportation channels in supply chain design. *International Journal of Production Economics*, 145:161–172, 2013.

[160] R. Ostermark. A neuro-genetic algorithm for heteroskedastic time-series processes: empirical tests on global asset returns. *Soft Computing*, 3(4):206–220, 1999.

[161] R. Ostermark. Solving a nonlinear non-convex trim loss problem with a genetic hybrid algorithm. *Computers and Operations Research*, 26(6):623–635, 1999.

[162] R. Ostermark. Solving irregular econometric and mathematical optimization problems with a genetic hybrid algorithm. *Computational Economics*, 13(2):103–115, 1999.

[163] E. Ozcan and C.K. Mohan. Steady state memetic algorithm for partial shape matching. In *International Conference on Evolutionary Programming*, 1998.

[164] Q.K. Pan, L. Wang, H.Y. Sang, J.Q. Li, and M. Liu. A high performing memetic algorithm for the flowshop scheduling problem with blocking. *IEEE Transactions on Automation Science and Engineering*, 10(3):741–756, 2013.

[165] M. Pastorino. Reconstruction algorithm for electromagnetic imaging. *IEEE Transactions on Instrumentation and Measurement*, 53:692–699, 2004.

[166] K. Patel and M. Pamchal. One dimension multi objective bin packing problem using memetic algorithm. *Journal of Computer Science and Information Technology*, 3 (2):761–766, 2014.

[167] R. Pati, P. Vrat, and P. Kumar. A goal programming model for paper recycling system. *Omega*, 36:405–417, 2008.

[168] A. Paz, V. Molano, E. Martinez, C. Gaviria, and C.Arteaga. Calibration of traffic flow models using a memetic algorithm. *Transportation Research Part C: Emerging Technologies*, 55:432–443, 2015.

[169] D. Petrovic, R. Roy, and R. Petrovic. Supply chain modeling using fuzzy sets. *International Journal of Production Economics*, 59:443–453, 1999.

[170] A.P. Piotrowski. Review of differential evolution population size. *Swarm and Evolutionary Computation*, 32:1–24, 2017.

[171] M. S. Pishvaee, F. Jolai, and J. Razmi. A stochastic optimization model for integrated forward/reverse logistics network design. *Journal of Manufacturing Systems*, 28:107–114, 2009.

[172] M. S. Pishvaee, R. Zanjirani Farahani, and W. Dullaert. A memetic algorithm for bi-objective integrated forward/reverse logistics network design. *Computers and Operations Research*, 37:1100–1112, 2010.

[173] M.S. Pishvaee and M. Rabbani. A graph theoretic-based heuheuris algorithm for responsive supply chain network design with direct and indirect shipment. *Advances in Engineering Software*, 42:57–63, 2011.

[174] M.S. Pishvaee, K. Kianfar, and B. Karimi. Reverse logistics network design simulated annealing. *The International Journal of Advanced Manufacturing Technology*, 47(1):269–281, 2010.

[175] S. Pokharel and A. Mutha. Perspectives in reverse logistics: A review. *Conservation and Recycling*, 53(4):175–182, 2009.

[176] J. Puchinger, G.R. Raidl, and M. Gruber. Cooperating memetic and branch-and-cut algorithms for solving the multidimensional knapsack problem. In *The 6th Metaheuristics International Conference*, University of Vienna Vienna, Austria, Europe, 2005.

[177] E. Ramat, G. Venturini, C. Lente, and M. Slimane. Solving the multiple resource constrained project scheduling problem with a hybrid genetic algorithm. In T. Back, editor, *The Seventh International Conference on Genetic Algorithms*, pages 489–496, 1997.

[178] M. Realff, J. Ammons, and D. Newton. Robust reverse production system design for carpet recycling. *IIE Transactions*, 36:767–776, 2004.

[179] C. Reeves. Hybrid genetic aalgorithm for bin-packing and related problems. *Annals of Operations Research*, 63(5-8):371–396, 1996.

[180] A. Rezoug, D. Boughaci, and M. Badr-El-Den. Memetic algorithm for solving the 0-1 multidimensional knapsack problem. In *17th Portuguese Conference on Artificial Intelligence*, pages 293–304, 2015.

[181] M.J. Richard, M. Bouazara, L. Khadir, and G.Q. Ca. Structural optimization algorithm for vehicle suspensions. *Transactions of the Canadian Society for Mechanical Engineering*, 35:1–17, 2011.

[182] F. Rothlauf and D.E. Goldberg. Pruefer numbers and genetric algorithms: a lesson on how the low locality of an encoding can harm the performance of gas. In *6th International Conference on Parallel Problem Solving from Nature*, 2000.

[183] C.F. Ruff, S.W. Hughes, and D.J. Hawkes. Volume estimation from sparse planar images using deformable models. *Image and Vision Computing*, 17(8):559–565, 1999.

[184] S. Runggeratigul. Local search and memetic aagorithms for knapsack problems. In *The Fifth Metaheuristics International Conference*, Kyoto, Japan, 08 2003.

[185] M. Safe, J. Carballido, I. Pomzoni, and N. Brig. On stopping criteria for genetic algorithms. In *Advances in Artificial Intelligence SBIA 2004*, pages 330–335, 2004.

[186] A. Sakamoto, X.Z. Liu, and T. Shimamoto. A genetic approach for maximum independent set problem. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E80A(3):551–556, 1997.

[187] M.I. Salema, A.P. Barbosa-Povoa, and A.Q. Novais. A strategic and tactical model for closed-loop supply chain. *OR Spectrum*, 31(3):573–599, 2009.

[188] M.I.G. Salema, A.P. Barbosa-Povoa, and A.Q. Novais. An optimizing model for the design of a capacitated multi-product reverse logistis network with uncertainty. *European Journal of Operational Research*, 179:1063–1077, 2007.

[189] J. Sarkis, N. Darnal, G. Nehman, and J. Priest. The role of supply chain management within the industrial ecosystem. *The 1995 IEEE International Symposium on Electronics and the Environment*, pages 229–234, 1995.

[190] V. Schnecke and O. Vornberge. Hybrid genetic algorithms for constrained placement problems. *IEEE Transactions on Evolutionary Computation*, 1(4):266–277, 1997.

[191] F. Schultmann, B. Engels, and O. Rentz. Closed-loop supply chain for spent batteries. *Interfaces*, 33:57–71, 2003.

[192] E. Segredo, C. Segura, and C. Loen. Memetic algorithms and hyperheuristics applied to a multiobjectivised two-dimensional packing problem. *Journal of Global Optimization*, 58(4):769–794, 2014.

[193] L. Shen, E. Worrel, and M.K. Patel. Open-loop recycling: A lca case study of pet bottle-to-fibre recycling. *Resource, Conservation and Recycling*, 55(1):34–52, 2010.

[194] S.K. Shula, M.K. Tiwari, H. Wan, and R. Shankar. Optimization of the supply chain network:simulation, taguchi, and psychoclonal algorithm embedded approach. *Computers and Industrial Engineering*, 58:29–39, 2010.

[195] S. Srivastava. Network design for reverse logistics. *Omega*, 36:535–548, 2008.

[196] P. Subbaraj and P. Sivakumar. *Multithreaded Memetic Algorithm for VLSI Placement Problem. Swarm, Evolutionary, and Memetic Computing.* Springer, Berlin, Heidelberg, 2011.

[197] B. Subudhi, D. Jena, and M.M. Gupta. Memetic differential evolution trained neural networks for nonlinear system identification. In *the 3rd International Conference on Industrial and Information Systems*, 2008.

[198] K. Subulan and A.S. Tasan. Taguchi method for analyzing the tactical planning model in a closed-loop supply chain considering remanufacturing option. *International Journal of Advanced Manufacturing Technology*, 66:251–269, 2013.

[199] M. Sun and H. Wang. The memetic algorithm for the minimum spanning tree problem with degree and delay constraints. In *the 15th International Conference on Advanced Communication Technology (ICACT)*, 2013.

[200] C.S. Sung and S.H. Song. Integrated service network design for a cross-docking supply chain network. *Journal of Operation Research Society*, 54:1283–1295, 2003.

[201] A. Syarif, Y.S. Yun, and M. Gen. Study on multi-stage logistic chain network: a apanning tree-based genetic algorithm approach. *Computers and Industrial Engineering*, 43(1):299–314, 2002.

[202] G. Taguchi and S. Konishi. *Taguchi methods, orthogonal arrays and linear graphs.* Tools for Quality American Supplier Institute, American Supplier Institute, 1987.

[203] T. Taguchi, T. Yokota, and M.Gen. Reliability optimal design problem with interval coefficients using hybrid genetic algorithms. *Computer and Industrial Engineering*, 35(1-2):373–376, 1998.

[204] M. Tang and X. Yao. A memetic algorithm for vlsi floorplanning. *IEEE Trannportation System*, 37(1):62–69, 2007.

[205] M. Thierry, M. Salomon, J.V. Nunen, and L.N.V. Wassenhove. Strategic issues in product recovery management. *California Management Review*, 37:114–135, 1995.

[206] S. Tragantalerngsak, S. Holt, and J. Ronnqvist. An exact method for two-echelon, single-source, capacitated facility location problem. *European Journal of Operational Research*, 123:473–489, 2000.

[207] M. Urselmann, S. Barkmann, G. Sand, and S. Engell. A memetic algorithm for global optimization in chemical process synthesis problems. *IEEE Transactions on Evolutionary Computation*, 15(5), 2011.

[208] H. Üster, G. Easwaran, E. Akali, and S.etinkaya. Benders decomposition with alternative multiple cuts for a multi-product closed-loop supply chain network design model. *Naval Research Logistics (NRL)*, 54(8):890–907, 2007.

[209] A. Vargas-Martnez and L.E. Garza-Castanon. Combining artificial intelligence and advanced techniques in fault-tolerant contro. *Journal of Applied Research and Technology*, 9(2):202–222, 2011.

[210] S. Verstrepen, F. Cruijssen, and M. Brito. An exploratory analysis of reverse logistics in flanders. *European Journal of Transport and Infrastructure Research*, 7 (4):301–316, 2007.

[211] V.P. Vinay and R. Sridharan. Taguchi methood for parameter design in aco algorithm for distribution-allocation in a two-stage supply chain. *International Journal of Advanced Manufacturing Technology*, 64:1333–1343, 2013.

[212] H.F. Wang and H.W. Hsu. A closed-loop logistic model with a spanning- tree based genetic algorithm. *Computers and Operations Research*, 37:376–389, 2010.

[213] L. Wang and J. Yen. Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and kalman filter. *Fuzzy Sets and Systems*, 101(3):353–362, 1999.

[214] J.P. Watson, S. Rana, L.D. Whitley, and A.E. Howe. The impact of approximate evaluation on the performance of search algorithms for warehouse scheduling. *Journal of Scheduling*, 2(2):79–98, 1999.

[215] X. Wei and F. Kangling. A hybrid genetic algorithm for global solution of nondifferentiable nonlinear function. *Control Theory and Applications*, 17(2):180–183, 2000.

[216] F. Yaman and A. E. Yilmaz. Impacts of genetic algorithm parameters on the solution performance for the uniform circular antenna array pattern synthesis problem. *Journal of Applied Research and Technoloy*, 8(3):378–39, 2010.

[217] J. Yan, M. Gong, L. Ma, S. Wang, and B. Shen. Structure optimization based on memetic algorithm for adjusting epidemic threshold on complex networks. *Applied Soft Computing*, 49:224–237, 2016.

[218] D. Yang, S. Hong, H. Cheng, and L. Yao. A novel dynamic reactive power planning methodology to enhance transient voltage stability. *International Transactions on Electrical Energy Systems*, 27(10):1–17, 2017.

[219] W. Yeh. A hybrid heuristic algorithm for the multistage supply chain network problem. *International Journal of Advanced Manufacturing Technology*, 26:675–685, 2005.

[220] W. Yeh. An efficient memetic algorithm for the multi-stage supply chain network problem. *International Journal of Advanced Manufacturing Technology*, 29:803–813, 2006.

[221] Y. Yun, C. Moon, and D. Kim. Hybrid genetic algorithm with adaptive local search scheme for solving multistage-based supply chain problems. *Computers and Industrial Engineering*, 56:821–838, 2009.

[222] M. Zandieh, M. Amiri, B. Vahdani, and R. Soltani. A robust parameter design for multi-response problems. *Journal of Computational and Applied Mathematics*, 230:463–476, 2009.

[223] Z. Zhang, O. Che, B. Cheang, A. Lim, and H. Qin. A memetic algorithm for the multiperiod vehicle routing problem with profit. *European Journal of Operational Research*, 229(3):573–584, 2013.

[224] Z. Zhang, M. Liu, and A. Lim. A memetic algorithm for the patient transportation problem. *Omega*, 54:60–71, 2015.

[225] G. Zhou, H. Min, and M. Gen. A genetic algorithm approach to the bi-criteria allocation of customers to warehouse. *International Journal of Production Economics*, 86:35–45, 2003.

[226] Z. Zhu, F. Wang, S. He, and Y. Sun. Global path planning of mobile robots using a memetic algorithm. *International Journal of Systems Science*, 46(11):1982–1993, 2013.