

The Fleet-Sizing-and-Allocation Problem: Models and Solution Approaches



CHEMNITZ UNIVERSITY
OF TECHNOLOGY

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur

(Dr.-Ing.)

vorgelegt

der Fakultät für Informatik
der Technischen Universität Chemnitz

von M.Sc. Moustafa Mohamed El-Ashry

Betreuer : Prof. Dr. rer. nat. habil. Dr. oec. Peter Köchel

Gutachter: Prof. Dr. rer. nat. habil. Dr. oec. Peter Köchel, TU Chemnitz.
Prof. Dr. Knut Richter, Europa University Viadrina, Frankfurt.
Prof. Dr. Peter Gluchowski, TU Chemnitz.

Acknowledgement

- All gratitude is due to “**ALLAH**” almighty who have guided and aided me to fulfill this work and who supported me in all my life. To Him I extend my heartfelt gratitude.
- I am grateful to my parents, who have supported me through not only my education but also my entire life, my sisters and my brother for their love and support. I am also grateful to my beloved wife, for her patience with me especially during the past few years, for her unwavering love, support and giving me the strength and comfortable atmosphere to finish this study, to my nice children **Yara, Sara, Mohamed and Eman** who will always be in my heart. I would like to dedicate this thesis to all my family.
- I am grateful to my promoter **Prof. Dr. Peter Köchel** for welcoming me as his student, for his kind supervision, continuous encouragement, valuable enthusiastic discussion and unfailing advice throughout the present work and for giving me the opportunity to study in an intellectually rich environment. He supported and helped me during my learning period in Germany. I am very lucky being one of his students.
- I would like to express my deep thanks to the members of my examining committee **Prof. Dr. Knut Richter** – Europa University Viadrina, Frankfurt (Oder), Germany and **Prof Dr. Peter Gluchowski** – Chemnitz University of Technology, Germany for their comments, for their encouragement and for their interest in my studies.
- I would like to thank, every one who have helped me on the path towards the dissertation, my colleagues and all staff members in the Chair of “Modeling and Simulation” Faculty of Computer Science - Chemnitz University of Technology – for valuable discussions and their help.
- Finally, I would like to thank the Egyptian Ministry of Higher Education, the Egyptian Government, and Faculty of Science – Tanta University for the financial support during my study in Germany.

To all of you, Thank you very much.

Moustafa El-Ashry

Dedication

To The Soul of My Father

To my Mother, Brother and Sisters

To my lovely wife and my nice children

Moustafa

Contents

1 Introduction.....	1
1.1 Overview of the Thesis.....	4
2 Problem Definition and Literature Review.....	6
2.1 Literature Review.....	6
2.2 Fleet Size Specification.....	10
2.3 Classification of Empty TU Distribution Models.....	14
2.4 Classification of Fleet Sizing Models.....	18
3 Continuous Time Deterministic Models.....	21
3.1 One-to-One Case.....	21
3.2 One-to-Many Case.....	26
3.3 Queueing Models for the One-to-Many Case.....	30
4 Single-Period Deterministic Models.....	45
4.1 Introduction.....	45
4.2 Description of a Basic Model and Decision Problem.....	45
4.3 A Single-Period Deterministic-Demand Model.....	48
4.3.1 Solutions for the Allocation Problem without Renting Possibilities.....	49
4.3.2 Solutions for the Allocation Problem with Renting Possibilities.....	56
5 Multiple-Period Models.....	71
5.1 Introduction.....	71
5.2 Description of a Basic Model.....	72
5.3 Genetic Algorithm.....	74
5.3.1 Basic Description of the Genetic Algorithm.....	74
5.3.2 Genetic Representation.....	77

5.4	Crossover Operation.....	80
5.4.1	Crossover Mechanisms.....	80
5.4.2	Designed Crossover Operator.....	81
5.5	Mutation Operation.....	83
5.5.1	Principle of Mutation.....	83
5.6	Genetic algorithm implementation.....	85
5.7	General HAS models - a Simulation Optimization Approach.....	89
5.7.1	Simulation Optimization.....	89
5.7.2	Fundamentals.....	90
5.7.3	Calculation Assessment Optimization System (CAOS).....	91
5.7.4	HAS Simulator.....	92
5.7.4.1	Model Elements.....	92
5.7.4.2	Control Policies.....	93
5.7.4.3	Events-Activity Diagrams.....	94
5.7.4.4	Assessment.....	99
5.8	Examples.....	99
6	Summary and Conclusions.....	103
Appendix A:	Marginal Analysis.....	107
Appendix B:	Simulation Optimization Results.....	109
Appendix C:	The results data of the GA program for example 5.1 (4 Customers and 4 Periods).....	125
Appendix D:	The results data of the GA program for example 5.2 (15 Customers and 15 Periods).....	133
References.....		147

List of Figures

2.1	Hub-and-Spoke Networks.....	8
2.2	Empty Freight TU Distribution Model Categories.....	14
2.3	Classification of Fleet Sizing Models.....	19
3.1	Scheme for One-to-One System.....	22
3.2	Typical behavior for the gain function $g(K)$	25
3.3	Scheme for One-to-Many System.....	27
5.1	Flow chart of the genetic Algorithm.....	76
5.2	Scheme of Simulation Optimization.....	90
5.3	Activity Diagram Event “New Demand”.....	95
5.4	Activity Diagram “Serve Enqueued Demand”.....	96
5.5	Activity Diagram “Serve Enqueued Orders”.....	97
5.6	Event “TU Delivery”.....	98
5.7	Event “TU Return”.....	98
5.8	Simulation Optimization Results for the Four-Spokes-Single-Hub System, $N(100,10)$ -Distributions.....	101

List of Tables

3.1	Arrival and Service Rates for Example 3.3.1.....	34
3.2	Numerical Results for Example 3.3.1.....	35
3.3	Numerical Results for location 1.....	39
3.4	Numerical Results for location 2.....	40
3.5	Numerical Results for location 3.....	40
3.6	Numerical Results for location 4.....	40
3.7	Arrival and Service Rates for Example 3.3.3.....	43
3.8	Numerical Results for Example 3.3.3.....	43
4.1	Data for Example 4.3.1.....	61
4.2	Data for Example 4.3.2.....	67
5.1	Cost Information for example 5.1.....	78
5.2	Distribution Demand Requirements for Example 5.1.....	79
5.3	Genetic Representation and Solution Interpretation for the Sample Solution.....	79
5.4	Cost Information for Example 5.2.....	86
5.5	Distribution Demand Requirements for Example 5.2.....	86
5.6	Optimal solution n_i^* for example 2 with $P(\text{Mut}) = 0.01$	87
5.7	Results of the Simulation Optimization Approach for 1 Hub and 4 Spokes.....	102
5.8	Results of the Simulation Optimization Approach for 1 Hub and 50 Spokes.....	102

Chapter 1

Introduction

Transportation is one of the most vital services in modern society. It makes most of the other functions of society possible, such as manufacturing and construction, food and agriculture, energy supply and distribution, safety and security, access to medical care, and tourism and recreation. The future of the nation and the world depends critically on transportation systems that are reliable, efficient, safe and environmentally sustainable. Real transportation systems are so large and complex that in order to build the science of transportation systems it will be necessary to work on fundamental research issues in many areas, including:

- Modelling
- Optimization, especially robust planning, on-line learning and control.
- Decomposition, especially how to decompose complex systems to facilitate optimization and decentralized control.
- Economics and game theory.
- Simulation and development of virtual laboratory.

Transportation systems frequently contain fleets of transportation units (TU's), which circulate on networks, carrying people or goods, depending on customer requests. The term TU refers to reusable shipping containers and factory equipment such as fork lifts or automatic guided vehicles (AGVs), as well as the more traditional types of TU's such as cars, trucks, buses, railcars or airplanes and material handling equipment.

The capacity of a transportation system is directly related to the number of available TU's. Owners and operators of transportation systems invest in TU's in order to provide the capacity to meet demands. Determining the optimal number of TU's for a particular system requires a tradeoff between the ownership costs of the TU's and the potential costs or penalties associated with not meeting some demands. The fleet of TU's which they are available for service at any time (and their locations) depends upon the TU redistribution strategy. This provides motivation to avoid operating a system with too few TU's. On the other hand, these TU's often represent a substantial capital investment either because the individual TU's are expensive or because operation of the system requires large numbers

of relatively inexpensive TU's. A transportation unit standing idle is not earning any return on this investment. As the fleet size increases above the capacity required to serve demand, the percentage of time that TU's spend idle increases. This provides motivation to avoid unnecessarily large fleets. An important element is the determination of optimal (or near optimal) fleet size for a very general class of transportation systems.

Generally, serving a demand results in the relocation of the TU when the demands are not balanced and the TU's are reusable, it is necessary to redistribute the TU's to locations where they can be used again. TU's that are being repositioned are not available to serve demand. Therefore, empty TU distribution strategies have an obvious impact on fleet size. When TU's are plentiful, it makes sense to redistribute them in a way that will minimize the direct transportation cost of empty TU movements, but when they are in short supply, TU's may be redistributed so as to serve the most profitable demands, perhaps at the expense of additional empty TU movements. This relationship between fleet sizing and empty TU distribution decisions has been discussed by Mendiratta (1981). There is a substantial history of research on empty TU's distribution problems, for example Baker (1977), much of it motivated by issues in utilization of empty freight TU's. More recently, similar problems encountered in rental TU agencies, trucking operations and container freight systems have also been studied, see Wu et al. (1999). The numbers of available TU's of various types are assumed to be specified as data, and the models attempt to find the most efficient routing for those TU's. Such formulations can achieve benefits associated with reduced operating costs, but do not address the longer-term decisions of investment in the TU fleet.

In this dissertation we are interested in solutions for the so-called fleet-sizing-and-allocation problem (FSAP). Fleet sizing and allocation problem consists of two interdependent parts. The fleet sizing problem is to determine the number of TU's that optimally balances service requirements against the cost of purchasing and maintaining the TU. Allocation problem is the process of repositioning TU to serve future transportation demand. These two problems are highly related to each other. A fleet unit in our sense is a reusable transportation unit (TU) for the realization of a given kind of transportation service. Transport operators try to offer low cost and high quality transport services given these inherent and institutional restrictions imposed by infrastructure. To offer cheap, reliable, fast, flexible and high accessible transport services transport operators apply

different spatial configurations of transport service networks. One of these typical networks is the hub-and-spoke service network. To make the FSAP a little bit more tractable we concentrate on logistic systems with a special hub-and-spoke structure. Such a structure can be profitable only if there exists concentration of freight volume on some service links in the network. Therefore hub-and-spoke networks are very actual and can be found e.g. in air freight distribution, two-echelon inventory systems, container distribution systems and others. Such networks have been developed with intention to improve competitiveness, efficiency and effectiveness of their operations. The main motivation for such network structure lies in savings in transportation cost due to concentration of volume at the hubs.

Hubs are facilities that serve as switching in transportation and telecommunication networks. Hubs may be large facilities such as airports or postal sorting centers, or small switching devices in telecommunications networks. In the previous studies there are many different design and operational characteristics of these networks have been studied:

1. The optimal hub location problem,
 2. Optimal routing of freights through the service network,
 3. Performance analysis of hub-and-spoke network,
- and so on.

The hub location problem consists locating a single hub location when a certain number of terminal points are given. Each of these points has a weight (cost) associated with them. The process for locating a single hub based on the amount of traffic and cost is quite simple. The approach to solving this problem is the same as used in solve a centroid problem, this means identifying the point where all weights are balanced. If we locate a hub in this point the transportation costs will be minimized, in this case the weights will be determined as a function of the truckloads-per-period of time to each branch (spoke). Optimal routing systems are essential to assure efficient distribution. Therefore, the researchers study the optimal routing to enhance efficiency and reduce costs. Every single hub has to go through the network regardless of its destination. It could be bound for the other side of the addressed to the neighbor, the hub will have to go through the distribution system, which has a hub-and-spoke structure.

It is significant to distinguish two crucial characteristics of most transportation systems: they are dynamic because demands on the system change over time, and there is uncertainty both in the system performance (e.g. travel times) and in forecasting the demands on the system in the future these two characteristics play a key role in the interaction between TU utilization and fleet sizing decisions and are the cornerstones of models to be developed.

1.1 Overview of the Thesis

In this thesis we will discuss the fleet sizing and allocation problem (FSAP). The FSAP is one of the most interesting and hard to solve logistic problems. The FSAP consists in the definition of the most appropriate number of TU's to be maintained by an operator/carrier. In general, the problem is focused on the efficient matching between supply of transportation capacity and demand for transportation services. The thesis is set up as follows.

Chapter 2 consists of a literature review and the specification of fleet sizing divided by Beamon and Chen (1998) into three categories:

- i. Simulation Techniques,
- ii. Analytical Techniques, and
- iii. Hybrid Techniques (Simulation and Analytical).

We also introduce the classification of empty TU distribution models and classification of fleet sizing models.

In chapter 3 we begin with a very simple fleet sizing of one-to-one case that can be solved by inspection. The purpose of this example is to focus attention on several key issues in fleet sizing. The notation and concepts introduced in this case are then extended in order to determine the optimal fleet size, which maximize the profitability and minimize the total cost. We will also introduce another example for fleet sizing of one-to-many case. In this case, we consider the problem of determining the fleet size for a single TU type used to transport the items produced at the origin to many destinations. Items are produced in a deterministic production cycle, but TU travel times are stochastic. Finally in this chapter we apply queueing theory methods to solve the allocation problem in case of stochastic demand in the spokes.

In chapter 4 we will concentrate on some aspects of how to use transportation resources in an optimal way, i.e., we will discuss solution approaches to the problem of the optimal size and allocation of transportation resources. In the logistic literature we can find manifold models and solution methods dealing with that problem. To optimize a logistic system leads as a rule to very complex decision problems. The FSAP consists of two interdependent parts – to define the optimal size of a transport fleet and to reallocate empty fleet units among the locations of a logistic network. We also concentrate on the fleet-sizing-and-allocation problem for single hub networks. Generally, in a hub-and-spoke transportation network a centralized planner has to find freight routes, frequency of service, type of TU's to be used, and transportation volumes. We will study this problem for two cases:

- (1) Renting of additional TU's from outside the system is not possible, and
- (2) Renting of additional TU's from outside the system is possible.

In chapter 5 we introduce a genetic algorithm (GA) approach for the fleet sizing allocation problem (FSAP) and some definitions for the GA. Since the multi-period, deterministic demand problem is NP-hard we suggest to use Genetic Algorithms to solve our problem HAS. This approach is suited to handle multiple and nonlinear objective functions as well as side constraints. We present the developed genetic representation and use a randomized version of the heuristic to generate the initial random population. We design suitable crossover and mutation operators for the GA improvement phase. We will also discuss the principle of simulation optimization and describe a simulator for hub-and-spoke systems and give some examples show the applicability of our simulation optimization approach.

Finally, Chapter 6 contains a summary and conclusions.

Chapter 2

Problem Definition and Literature Review

2.1 Literature Review

The transportation is concerning a large number of complex operations for the freight transportation by means of TU's of different types and several modes, through networks of a complicated structure. There is a large body of data explaining how to design and operate freight logistics, transportation and distribution systems. The previous investigations have addressed strategic or tactical planning issues as well as operational activities. Consistently, Kasilingam's (1998) introduced a book that has been clearly explaining the transportation planning models. The methods of distribution systems planning have been analyzed by Geoffrion and Powers (1980) and Dejax (1986). Then, Waldinger et al. (1991) developed a model for the German Federal Railways (DB). Also, Powell and Carvalho (1996) developed a logistics queuing network for distribution of empty freight TU's. Additionally, some of the earliest models for freight transportation have been reviewed by Friesz et al. (1983), Harker (1985) and Crainic (1987).

The operational models for the ship routing and related scheduling of TU's have been surveyed by Ronen (1983), Ronen (1993) and Christiansen et al. (2004). Blazewicz et al. (1991,1994) investigated the simultaneous scheduling and routing of jobs among identical parallel machines in an AGV-served flexible manufacturing system. The authors applied their results as a loop layout with makespan objective. Then, Hall et al. (2001) studied the effectiveness of three widely used AGV dispatching policies for the objective of minimizing the cycle time. Recently, Grunow et al. (2004) introduced a priority-rule based dispatching procedure for a container terminal where automated guided vehicles (AGVs) with multiple-load capability are used as container transporters. Rana and Vickson (1991) discussed the optimal routing for a fleet of containerships operating on a trade route, to maximize the liner shipping company's profit. They formulate the problem as a mixed integer non-linear programming model and solve it by using Lagrangean relaxation techniques. Fagerholt (1999) studied the problem of determining the optimal fleet and the corresponding weekly liner routes and he solved it by employing a set partitioning approach as a multi-trip vehicle routing problem. Bendall and Stent (2001) proposed a

model of determining the optimal fleet configuration and associated fleet deployment plan in a containership hub and spoke application. Crino et al. (2004) described a deterministic methodology that provided vehicle routing and scheduling of multi-modal theater transportation assets at the individual asset operational level to provide economically efficient time definite delivery of cargo to customers for generalized theater distribution problems.

The fleet sizing and allocation problem (FSAP) is one of the most interesting and hard to solve logistic problems. To make the FSAP a little bit more tractable we concentrate on logistic systems with a special hub-and-spoke (HAS) structure. The general efficacy of hub-and-spoke networks in truckload trucking has been determined using the results of initial testing designs by Taylor et al. (1994). The authors added that there is a limited success by the appropriate use of the configured and managed networks. There are several investigations studied the hub location problem O'Kelly et al. (1995), O'Kelly & Btyan (1998), Campbell (1996), Skorin-Kapov et al. (1996) and Pirkul & Schilling (1998). Additionally, they suggested that all movement should engage one or more hubs between the source and the destination. Campbell et al. (2002) argued that the effect of pricing and competition in hub and spoke network design has received insufficient attention in the literature. Adler (2005) developed a model framework to provide information on the most adaptable and profitable hub and spoke networks available under competition and applied it to Western Europe.

The previous studies classified the Hub-and-Spoke network into four types depending on whether hub(s) should be visited between source and destination as illustrated in Figure (2.1). In the first type, TU's movements among destinations are completely independent on a hub (Figure 2.1-(a)). However, in the second type, TU's movements are dependent only on one hub; TU's can go to a single hub from which it moves to its destination (Figure 2.1-(b)). In the third type, TU's are dependent on two hubs; TU's has to visit the first hub followed by the second hub from which it moves to the final destination (Figure 2.1-(c)). Finally, in the fourth type, some TU's go to their destination directly without visiting any hub whereas the others can go through one or two hubs before reaching the final destination (Figure 2.1-(d)).

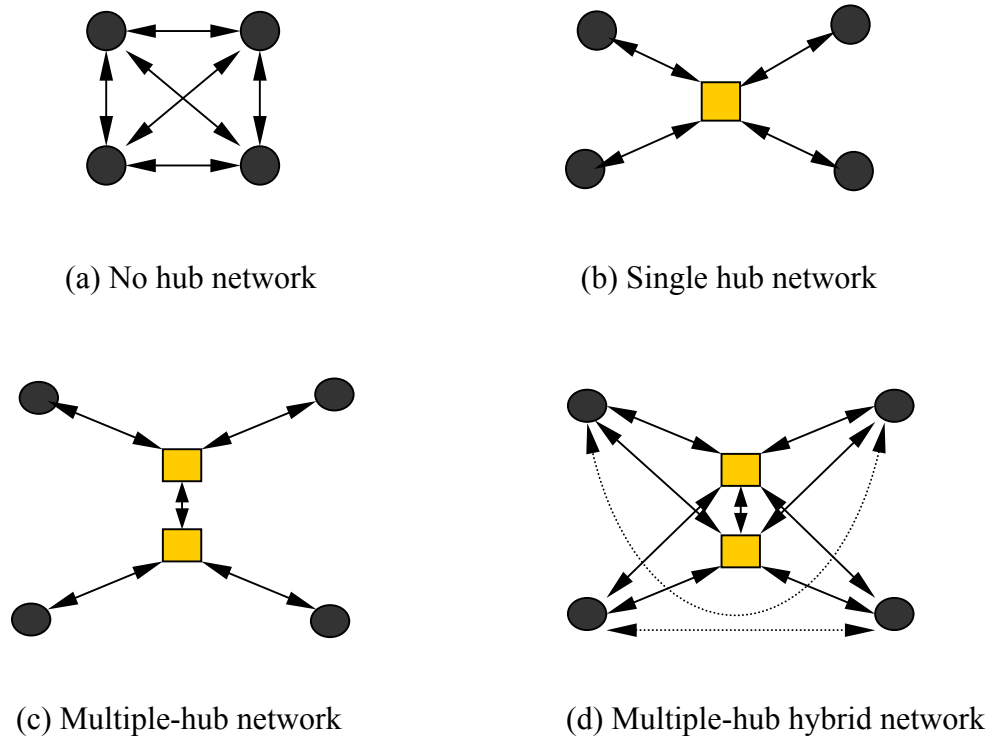


Figure 2.1: Hub-and-spoke networks

Most of the previous studies focused on the problem in the context of air freight network but there are a limited attempts to study the problems in railroad network. One of the earliest researchers in railroad network design problem is Assad (1980), who determines the type of trains, the number of trains of each type, and service routes in order to minimize the total operating cost. Crainic et al. (1984) and Crainic and Rousseau (1986) generalize the model introduced by Assad (1980) by incorporating more decision elements into the model such as service routes, service frequency, and composition of trains. Extending the previous studies, Keaton (1989, 1992) generalize the problem considered in Crainic et al. (1984) and Crainic and Rousseau (1986) by using a decomposition-based Lagrangian relaxation approach to find a lower bound for decomposed problems. The hub locations and the routing are determined so as to minimize the overall transportation cost. Campbell (1994) considered an uncapacitated hub location model where both the transportation cost and the fixed cost of establishing hubs are incorporated, and the number of hubs is an decision variable. Newman and Yano (2000) is one of the few studies that solve the hub location problems in railroad network in order to minimize operating costs.

A long time usually two control problems are considered for a service system: The first, how to decide whether to accept or to reject an arriving client (e.g. Köchel 1997), The second, how many servers (resource units) to install, see Shanthikumar et al. (1987). Practical needs of modern transportation network lead to a third control problem: How to organize transshipments of resource units between the nodes of the network. For instance, today's car rental firms offer the possibility to return a hired unit to an arbitrary location of the firm. As a consequence, a significant imbalance between the inflow of hired units and demand arises for most of the locations. To overcome these imbalances the natural flow of units has to be corrected by reallocations. In scientific literature on design and control of transportation networks, the combination of the second and third problem is known as the fleet sizing and allocation problem (FSAP). Solving the FSAP requires answering two closely connected questions:

- 1- How many units should a TU fleet contain (the fleet sizing problem)?
- 2- How to redistribute empty TU's that are not used in a given location among the locations of the network (the empty TU reallocation problem)?

Köchel et al. (2003) developed a queuing network model for movement of units through the locations without any control and given some answers to the second and third control problems under a given cost and gain structure. They also studied Genetic Algorithms in combination with a simulation model to seek optimal fleet size and repositioning policy by maximizing the gain in the steady state. Koo et al. (2004) studied a two-phase fleet sizing and TU routing procedure. The objective of the procedure is to provide a multiple TU routing to complete all the transportation requirements with the minimum fleet size. Phase one uses an optimization model to produce a lower bound on the required fleet size, and phase two applies a tabu search based heuristic to generate TU routing along with an appropriate fleet size. The optimal design and/or the optimal control of dynamic stochastic systems are an actual and interesting problem for theoreticians as well as practitioners. At present logistics is one of the most important application areas.

There are many different formulations and models to optimize logistic systems. In our work we will concentrate on some aspects of how to use transportation resources in an optimal way, i.e., we will discuss solution approaches to the problem of the optimal size and allocation of transportation resources. In the logistic literature we can find manifold models and solution methods dealing with that problem. To optimize a logistic system leads as a rule to very complex decision problems. Therefore corresponding models suffer

from various simplification assumptions. For instance, most models assume deterministic known demand for transportation services. In this case linear and non-linear network programming models are applied (cp. Wu et al. 1999). More sophisticated models and solution methods assume stochastic demand. Here models from inventory and queueing theory are appropriate (Koenigsberg & Lam 1976, Du & Hall 1997). However, in all cases no closed-form solutions are available. The majority of papers are dealing with algorithms for approximate solutions in the discrete-time case with known demand and infinite transportation capacity. The stochastic models decompose the problem with respect to time periods and assess the impact of the current decisions on the future through value functions. However, because practical fleet management models involve large numbers of decision variables and possible load realizations, standard stochastic optimization methods are not feasible for computing the value functions. Therefore, most of the stochastic fleet management models revolve around the idea of approximating the value function in a tractable manner. For stochastic fleet management models see Godfrey and Powell [2002 (a, b)], Powell et al. (2004), Powell and Topaloglu (2005) and Topaloglu and Powell (2006).

2.2 Fleet Size Specification

There are many papers, many classifications, in the area of fleet size specification. One of them we want to use here is from Beamon and Chen (1998). The authors divided the research in the area of fleet size specification into three broad categories according to the solution approach as follows:

- i. Simulation Techniques,
- ii. Analytical Techniques, and
- iii. Hybrid Techniques (Simulation and Analytical).

Simulation Techniques

The logistic problems are very complicated. To solve realistic problems the most appropriate solution technique seems to be simulation. For large problems, simulation methods offer the advantage of tractability. Furthermore, simulation enables the planner to take a more comprehensive view of the problem domain, making use of heuristics to solve additional aspects of the problem that could not be considered using optimization approaches. There are a lot of papers deals with this approach. For example, in the field of

manufacturing systems we mention the papers by Newton (1985) and Ceric (1990). However, simulation is not an optimization tool. In the last decade comes up the idea to combine simulation and optimisation and apply the simulation optimization approach. The first paper, which applies that approach to some logistic problems seem to be Köchel (2003).

Simulation Optimization (SO) is suited for solving such optimization problems, which cannot be solved by conventional approaches. Importantly, Köchel et al. (2003) discussed the basic principle of simulation optimization and to show the applicability of that approach. They also introduced the main ideas of SO and as a concrete realization discussed the combination of simulation and Genetic Algorithms. The fleet sizing and allocation problem is one of the main decision problems in actual logistic systems. Generally, there are many different formulations and models to optimize logistic systems. Recently, El-Ashry et al. (2006) introduced some structure for the investigated system and optimized corresponding systems by using simulation optimisation.

Analytical Techniques

There are many papers using analytical techniques for different type of models such as: linear/non-linear optimization models, inventory models, queueing models and so on. The linear/non-linear Optimization models is designed to compute an optimized solution that either maximizes or minimizes a given objective of a model while, at the same time, satisfying a set of constraints that may be defined in the model. It is widely used in the field of operations research. For example Chien et al. (1989) formulated the integrated problem of allocating and delivering as a mixed integer program to generate both good upper bounds and heuristic solutions by using Lagrangian-based procedure.

Inventory models like “Economic Order Quantity” (EOQ) is a model that defines the optimal quantity to order that minimizes total variable costs required to order and hold inventory. Furthermore, Du and Hall (1997) developed an approach derived from existing inventory theory techniques to determine the minimum fleet size, subject to meeting the maximum allowed long-run stock-out probability. Also, the authors developed a stochastic process model considering that the arrival process of loads at locations is independent, stationary Poisson processes. Then, the analytical results from these stochastic models are compared to the results obtained from Monte Carlo simulations.

Queueing models have been applied in a number of technical areas, one of them discussed by Johnson and Brandeau (1993). The authors used the M/G/c queueing system to design the model of the TU pool, but they used the optimization techniques to solve the problem, which performed a unique objective in fleet size specification. Furthermore, Lei et al. (1993), performed the analytical fleet sizing study, to design a heuristic procedure, derived from the optimization techniques, for determining the production schedules, which minimize fleet size requirements. Additionally, one of the earliest papers, used the analytical techniques in fleet sizing to introduce an optimization model for determining the minimum fleet size requirements, discussed by Maxwell and Muckstadt (1982).

Hybrid Techniques (Simulation and Analytical)

If we combine the simulation and analytical techniques we get hybrid techniques. The hybrid techniques used at least one of analytical techniques and simulation. Shanthikumar and Sargent (1983) classified the hybrid simulation and analytic models into the following four categories:

- 1- A model whose behavior over time is obtained by alternating between using independent simulation and analytic models.
- 2- A model where a simulation model and an analytic model operate in parallel over time with interactions through their solution procedures.
- 3- A model where a simulation models is used in a subordinate way for an analytic model of the total system.
- 4- A model where a simulation model is used as an overall model of the total system and requires analytic solutions as input parameters from the analytic models.

In the previous study by Egbelu (1993) determined the optimal unit load sizes and TU fleet size with the objective of minimizing total manufacturing cost. The hybrid solution approach used mixed integer programming, numerical search, simulation, and statistical analysis. Another a two-stage hybrid approach has been developed for designing material handling systems by Mahadevan and Narendran (1994). The first stage is an analytical model, which derived from the results obtained by Mahadevan and Narendran (1993) to estimate the system fleet size requirements, and the second stage is a simulation model to estimate the effects of TU failures and dispatching rules on overall material handling system performance. Kasilingam and Gopal (1996) developed an iterative hybrid

simulation-analytical approach to determine the fleet size corresponding to the minimum sum of idle-time costs of TU's and machines as well as the waiting times for parts. Regardless of the type of the solution approaches used, each approach seeks to determine either the minimum or optimal number of required TU's to obtain a set of system parameters, with respect to one or more objectives.

There are many difficult problems, with significant impacts upon the operational and economic performance of the system, related to the distribution and movement of empty freight TU's in all model and intermodal transportation systems. The empty traveling in fleet management is often done before, after and sometimes during loaded trips. The efficient management of TU's can be improved after the detection of infrastructures and TU's of the operating agency. The transportation units management can be classified into two kinds according to loading and emptying. Consistently, the first kind is the loaded-TU movements to meet demands, and the second one is the movement of empty TU's after discharging to demand points. Since, the loaded TU movement only generates revenue, the enhancement of freight transport efficiency can be achieved by reducing the empty TU movement. The planning of empty TU distribution used the optimization models, which minimize empty TU movement in order to meet demands and other operation requirements. There is a growing body of investigations about the application of empty TU management models to railroad operation. The investigations in this area can be divided into two groups. The first one studied the refinements of model structures and algorithmic efficiency for empty TU distribution models. The second group investigated the real railroad operation utilizing available model structures and algorithms. It has been stated that the successful applications of the modeling activities to real railroad operators lead to substantial reduction in the operating cost. For example, Gohring et al. (1993) and Holmberg et al. (1994). Most recent research on empty vehicle redistribution problem utilized nonlinear network programming (see e.g. Beaujon and Turnquist (1991)), multistage dynamic networks (see e.g. Cheung and Powell (1996)), logistics queuing network (see e.g. Powell and Carvalho (1998)), inventory and queuing theory (see e.g. Du and Hall (1997)), simulation-based genetic algorithms (GAs) (see e.g. Köchel et al. (2003)). Accordingly, the groupings for the empty TU flow problems have been performed as in Figure 2.2.

2.3 Classification of Empty TU Distribution Models

The numerous and often very diverse research works which address the empty freight TU movements issue can be classified by several criteria such as their main subject or as a sub problem of more general transportation or logistic planning problems. Also, traditionally, physical, or methodological criteria have been used to classify the empty freight TU movements. Suh and Lee (2001) classified the Empty TU distribution model into two groups as in Figure 2.2. The first is policy models that cover medium-to-long term strategy oriented planning problems. The second is operation models that concentrate on shorter-term problems.

1. Policy Model

- (a) Service Network Design Problem
- (b) The Estimation of Demand
- (c) Fleet Sizing Models
- (d) Prediction of Intercity Freight Flows
- (e) Logistics System Design

2. Operation Model

- (a) Inventory Management of Empty TU
- (b) Empty TU Allocation
 - Rail Carrier
 - Deterministic Approaches
 - Stochastic Approaches
 - Hybrid Approaches
 - Empty TU Transportation
 - Rail-Multicarrier
- (c) Combined Empty and Loaded TU Allocation
 - The Rail Case
 - The Motor Carrier case
 - Backhauling
 - Multimode Distribution Systems

Figure 2.2: Empty Freight TU Distribution Model Categories

The problems are usually defined on a network over which loaded and empty movements take place. At some nodes of the network, representing terminals, depots, demands and supplies of empty TU's are specified. The purpose of operational models is the efficient management of a given fleet of TU's: i.e., to decrease the cost of empty travel while satisfying the demand adequately. Also the models addressed either the management of the inventory of empty TU's at terminals of the network or the allocation / dispatching of empty TU's to certain origin-destination (O / D) pairs to satisfy the demand. Kraft (1994) also proposed four different groupings of rail modeling: tactical operating plan-development, empty equipment distribution, train-dispatching and advanced train control systems, and mechanical component reliability. The author used two groups for the empty equipment distribution problem: stochastic and deterministic model formulation.

In the previous studies, there are many papers, which studied the empty TU distribution problem, but much of them are limited to deterministic situations (see e.g. Crainic 2000). Recently, stochastic programming methods have been used to optimize the TU flows. Crainic et al., (1993) considered the empty TU reposition problem in stochastic environments. They focused on inland transportation of empty TU between ports, depots and individual customers. Cheung and Chen (1998) developed a stochastic model for a sea-borne empty TU allocation problem where owned and leased TU's are considered to meet the total transportation demand. Imai and Rivera (2001) deal with fleet size planning for refrigerated TU's where they determine the necessary number of TU's required to meet predicted future transportation demand. Recently, Choong et al., (2002) developed an integer programming formulation for empty TU relocation with use of both long and short-term leased TU's. However, the treatment of the short-term leased TU in their study is not appropriate, since the cost of the short-term leased TU's is independent of the lease length. Li et al., (2004) studied the empty container allocation in a port with the aim to reduce redundant empty TU's. They consider the problem as a non-standard inventory problem with simultaneous positive and negative demand under a general holding cost function.

The differences between transportation problems of the industrial firm and freight carrier may be exist at the operational level. They are relative to cost and service objectives, fleet size, number of terminals and clients, demand forecasting, repetitiveness of routes and shipments. In addition, the fundamental structure of the models and solution

techniques developed for a given problem size are similar even if they applied differently. Similarly, policy defining questions may be addressed in a different context for the industrial firm and the freight carrier using similar modelling methodologies and algorithms. The criteria used to identify the methodology can be classified as follow:

- 1- Modeling assumptions, which can subdivided as follow:
 - a- deterministic or stochastic.
 - b- time domain.
- 2- Modeling approach, which can subdivided as follow:
 - a- algebraic formulation for subsequent optimization.
 - b- analytic stochastic models such as queueing models.
 - c- simulation models.
- 3- Solution techniques, which can subdivided as follow:
 - a- mathematical programming optimization.
 - b- network algorithms.
 - c- stochastic optimization.
 - d- simulation.

The integration of TU fleet sizing decisions with optimization of TU utilization investigated by Sarmiento and Nagi (1999). By having direct impact on the level of investment in capital resources, the potential benefits from improved utilization of TU's is much larger than would be indicated simply from accounting for reduced operating costs. To model the interaction of TU utilization and fleet sizing decisions and recognize two crucial characteristics of most transportation systems: they are dynamic because demands on the system change over time, and there is uncertainty both in system performance and forecasting the demands on the system in the future. The first attempts at modeling this problem assumed that the demands and loaded flows of TU's are known, deterministic and independent of time.

Furthermore, Beaujon and Turnquist (1991) focused on development of a model to aid in making decisions on fleet sizing in situations where demand fluctuates over time (including both deterministic and stochastic changes), and TU travel times are uncertain, leading to uncertainty regarding when TU's will be available to meet demands. The model is designed to answer several questions that are of interest in the design and operation of vehicle systems:

- 1- How many TU's should be used in the fleet?
- 2- Where should TU pools be located?
- 3- How large should these pools be at any given time?
- 4- At any given time and location, how should available TU's be allocated to loaded movements, empty movements and TU pools?

There is also interaction between inventory decisions and TU routing decisions in the above mentioned model. To decide how many TU's in this system be maintained over a certain time period at each location is traditionally known as fleet sizing problem. Wu et al. (1999) mentioned that the desired size of fleet can be obtained by ways of buying, selling or leasing. Determining the optimal size of a fleet involves decision from three different levels of hierarchies. These three different levels of hierarchies are:

- 1- The strategic decision which defines the expected level of customer satisfaction, determines capital budgets, capacity, model and maximum in-service age of vehicles as well as where to locate vehicle depots;
- 2- The tactical decisions of TU procurement, disposal and storage, which are generally determined in accordance with a capital budgeting plan;
- 3- The operational decisions including the assignment of a TU to customer's request and empty TU repositioning strategy, defining the utilization of a TU over its lifetime or planning horizon.

The above mentioned three different levels have been extensively investigated separately and approached from different ways. The Motivation via the principles of engineering economies, replacement analysis is the examination of cash flows and economic lives of defender and challengers as well as determination of the replacement schedule which optimizes a particular measure of economy. The replacement schedule with either finite or infinite time horizon can be decided according to the following steps:

- 1- whether to keep the defender or replace it immediately with the current challenger,
- 2- which of future challengers to replace with.

Traditional replacement analysis is age-based tactical decision-making, in other words, no asset utilization is concerned explicitly. In the area of transportation logistics, typically, TU assignment or TU allocation combined with inventory and routing decisions are solved to determine optimal size of a fleet.

2.4 Classification of Fleet Sizing Models

Fleet sizing models has been used to deal with the demand for loaded TU's. However, it was observed that the demand for movements between locations was often unbalanced. Therefore, empty TU redistribution strategy has become of special interest. In the meantime, it has been suggested that transportation demand and TU travel time are practically not only dynamic but also uncertain. Thus, the cumulative demand on one time at one location will probably exceed the total available TU's. Also, the previous studies assumed that inventory pool act as a buffer against the imbalance in the TU flow. Du and Hall (1997) examined different spatial patterns of TU movements, such as one origin and one destination, one origin and many destinations (or vice versa), or central-terminal network. The previous studies demonstrated two important characteristics of a fleet sizing problems as follow:

- 1- the spatial traffic pattern served by fleet.
- 2- the size of individual shipments relative to the capacity of a single TU.

The spatial pattern of movements can extend from one origin and one destination to very complex “many-to-many” traffic patterns. Although the main distinction useful in fleet planning is between partial TU loads and full loads, the individual shipments can change a small fraction of a TU's capacity up to multiple full TU's. Figure (2.3) shows a simple classification scheme for fleet sizing problems. This classification explains several types of fleet sizing problems. Daganzo (1999) studied the problem using the transport of full TU loads in a “one-to-many” pattern. This “one-to-many” pattern is considered as TU's movement from one origin to many destinations, and vice versa. Therefore, any TU in the fleet may be transported to any destination. Thus the system is not just the collection of several simple transport (“one-to-one”) systems. The previous studies performed by Koenigsberg and Lam (1976) investigated a problem involving a single origin and destination with TU's moving between them. Their analysis is based on development of a cyclic queuing model, supposing exponential terminal and transit times. Furthermore, Campbell (1993) studied the one-to-many distribution situation, which except the size of the TU's that deliver to destinations is limited. Figure (2.3) classified the literature on the fleet sizing, which includes work on many related problems as follows:

		Traffic Pattern		
		One-to-one	One-to-many	Many-to-many
Shipment Size	Full TU Loads	Cyclic Queueing Models	Fleet Allocation Models	
	Partial TU Loads	Dispatching Models	TU routing Models	Scheduled Operation (Airlines, Transit)

Figure 2.3: Classification of Fleet Sizing Models

Either the Vehicle Routing Problem (VRP) is a complex combinatorial optimization problem, which can be seen as a merge of two well-known problems: the Traveling Salesman Problem (TSP) and the Bin Packing Problem (BPP). It can be described as follows: given a fleet of TU's with uniform capacity, a single depot, and several customer demands, find the set of routes with overall minimum route cost which service all the demands. Most approaches to the VRP depend on heuristics and give approximate solutions to the problem (e.g. heuristic based (Kindervater and Savelsbergh, 1997), constraint programming (Shaw, 1998), and colony optimization (Gambardella et al., 1999)). Afterwards, Daganzo (1999) illustrated the situations, which involve “many-to-many” movements in relation to full TU loads. The author stated that these movements are deterministic, scheduled, and applicable in the airlines and transit operations.

However, stochastic and/or unscheduled operations present more of a problem. Vu Tung and Pinnoi (2000) proposed a flexible routing policy for TU's which is different from the company's fixed routing policy in that for each route, the number and the sequence of nodes a TU visits can be different. The suggested policy is formulated into a mixed integer program with the objective of minimizing the total operating costs. This formulation can be used to derive exact solutions for applications similar to our case. Hall, et al. (2001) considered design and operational issues that arise in repetitive manufacturing systems served by (AGVs) in loop layouts with unidirectional material flow. Such systems are in widespread industrial use, and play an important role in modern manufacturing environments. The objective considered is the minimization of AGV fleet size, given the minimum steady state cycle time required to produce a minimal job set (or equivalently, given the maximum throughput rate).

Also study whether the decomposition of a large AGV-served flow shop loop into several smaller loops improves productivity the original loop and the decomposed design are compared with respect to the minimum cycle time needed for the repetitive manufacture of a minimal job set. When there are three or more machines in the loop, finding the optimal cycle time is an intractable problem. Also discussed a joint sequencing issue that arises in decomposed systems with limited buffers between the loops, and analyzed the tractability of all the relevant joint sequencing problems. The general line of investigation represented by these various models represents a focus on FSAP. The size of the available fleet is specified exogenously. However, one of the important basic questions in such systems is the determination of the appropriate number of TU to have. Owning or leasing a fleet of TU's is generally quite costly, so it is natural to try to minimize the size of the required fleet.

Chapter 3

Continuous Time Deterministic Models

The purpose of this chapter is to present a more aspect explanation of the fleet sizing and allocation problem (FSAP) and to indicate the contribution made by this research in the context of related research by others. In this chapter we are going to go through a very simple fleet sizing of one-to-one case, which is to be solved by inspection. This example will cause us to focus attention on several key issues in fleet sizing. Afterwards, the notations and concepts introduced in this case are extended in order to determine the optimal fleet size, which maximize the profitability and minimize the total cost. Subsequently, we generalized the idea of one-to-one case for one-to-many case. Finally, we apply queueing theory methods to solve the allocation problem in case of stochastic demand in the spokes.

3.1 One-to-One Case

In this case “one-to-one” suppose that there is a single origin and a single destination. The single origin may be a production center and the single destination may be consumer. Let d represent the demand per time unit for transportation product between origin and destination measured in units of TU loads. Assumed that the capacity Q of individual TU is small relative to the total demand, so that TU’s are dispatched fully loaded. Demand d induces loaded TU flows from origin to destination, which will be represented as X i.e. X represent the minimum number of TU’s per time unit, necessary for demand satisfaction such that:

$$X = \lfloor d/Q \rfloor + 1$$

Generally, there are several different types of TU’s available to serve demands, but in this example and through this dissertation it will be assumed that all of the TU’s are of the same type that means we use a single type of transportation unit (i.e. have the same capacity, ownership cost and operating costs). Traveling a loaded TU from origin to destination incurs a direct transportation cost, which will be represented as C_l and requires time T_{12} .

The empty movements from destination to origin are denoted Y and incur a cost C_e and travel time T_{21} . Overall for each cycle, the travel time is T and the cost is C such that: $T = T_{12} + T_{21}$ and $C = C_l + C_e$. Figure (3.1) illustrates this system.

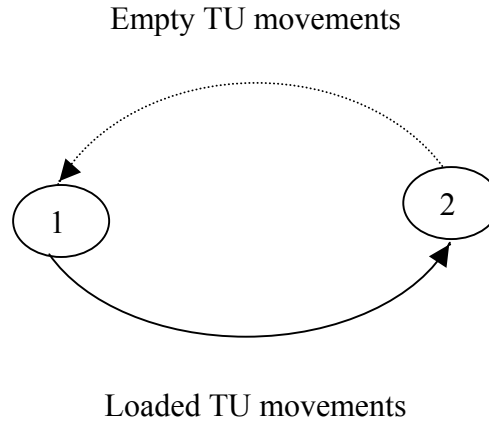


Figure 3.1: Scheme for one-to-one system

In general, the objective of a business enterprise would be to maximize profit, which means to maximize profit and minimize the total cost. In many potential applications, the components of the revenues and costs may be measured indirectly and may be quite incomplete as measures of the overall profit of an entire business enterprise. However, something like revenues are generated by serving demands, while TU ownership, TU movements and failure to serve demand generate costs and the objective is to maximize the difference. The objective will therefore be referred to as maximizing profitability. This terminology is further justified by the fact that is on long-run planning decisions. Such decisions could be expected to position the enterprise for future profitability but may be only crude approximations to actual profits, which will be determined by short-run operating decisions as well.

In the considered one-to-one case, there is only one feasible TU allocation strategy and so it is possible to express the profitability as a function of fleet size alone. Let $K \in I = \{0, 1, 2, \dots\}$ be the number of TU's in the fleet. TU's must complete a roundtrip for each unit of demand, which they satisfy.

Consequently the capacity Q_s of the system is given by:

$$Q_s = Q K / T = A K \quad (3.1)$$

where $A = Q / T$ represent the average amount of product shipped by a single TU per time unit. The amount of demand, which is served, is the minimum of the capacity and the total demand. If the revenue per unit of demand is R then for $R(K)$, the average revenue per time unit for fleet size K , it holds:

$$R(K) = \begin{cases} R A K & ; \text{if } K.A \leq d \\ R d & ; \text{if } K.A > d \end{cases} \quad (3.2)$$

Not serving demand results in a cost due to loss of customer good will, backordering costs (e.g. temporary storage of goods awaiting shipment) or expedited shipments. In calculating the revenues, it was assumed that demand in excess of capacity is lost. Therefore, the proper interpretation of the shortage cost in this example is loss of customer good will. In many cases, including manufacturing systems, the demand must be satisfied somehow. In such cases, the revenue is fixed, but there may be a substantial penalty associated with missed or delayed shipments. To be consistent, and avoid double counting, it is assumed in this example that the penalty for not serving demand, P , represents whatever loss is incurred in excess of the loss of revenue. The total stockout cost $C_s(K)$ per time unit is:

$$C_s(K) = \begin{cases} P(d - A K) & ; \text{if } K.A \leq d \\ 0 & ; \text{if } K.A > d \end{cases} \quad (3.3)$$

Transportation units are only dispatched in response to demand and therefore the direct transportation cost $C_t(K)$ per time unit is:

$$C_t(K) = \begin{cases} C A K & ; \text{if } K.A \leq d \\ C d & ; \text{if } K.A > d \end{cases} \quad (3.4)$$

Finally, if the cost per TU per time unit is V then the TU ownership cost, $C_v(K)$ per time unit is:

$$C_v(K) = VK \quad (3.5)$$

The gain function per time unit, $g(K)$ or profitability is given by total revenue minus total cost; i.e.,

$$g(K) = R(K) - C(K), \quad (3.6)$$

where the total cost per time unit $C(K)$ is equal to

$$C(K) = C_s(K) + C_i(K) + C_v(K)$$

$$= \begin{cases} K [V + A(C - P)] + Pd & ; K.A \leq d \\ KV + Cd & ; K.A > d \end{cases} \quad (3.7)$$

From equation (3.7) we can see that $C(K)$ is a piecewise-linear function. By substituting from equations (3.2) and (3.7) in equation (3.6) we can rewrite equation (3.6) as follows:

$$g(K) = R(K) - C(K)$$

$$= \begin{cases} K [A(R + P - C) - V] - Pd & ; K.A \leq d \\ -KV + d(R - C) & ; K.A > d \end{cases} \quad (3.8)$$

Also, from equation (3.8) we can see that $g(K)$ is a piecewise-linear function. Now, the purpose is how to determine the optimal fleet size K^* in order to maximize the gain function $g(K)$ (i.e. $\max_{K \in I} g(K)$).

Generally, from economic condition we can see that the necessary and sufficient condition for $K^* > 0$ is:

$$g(0) < g(1) \Leftrightarrow A(R + P) > A.C + V \quad (3.9)$$

That means the cost savings per time unit for a single TU must be greater than the cost per time unit for transport by a single TU. To determine the optimal fleet size K^* must be satisfy the following condition:

$$g(K^*) \geq g(K) \quad \text{for all } K \in I \quad (3.10)$$

From equation (3.8), it is obvious to see that $g(K)$ is concave function.

To get an algorithm to search the optimal fleet size K^* we consider the figure (3.2).

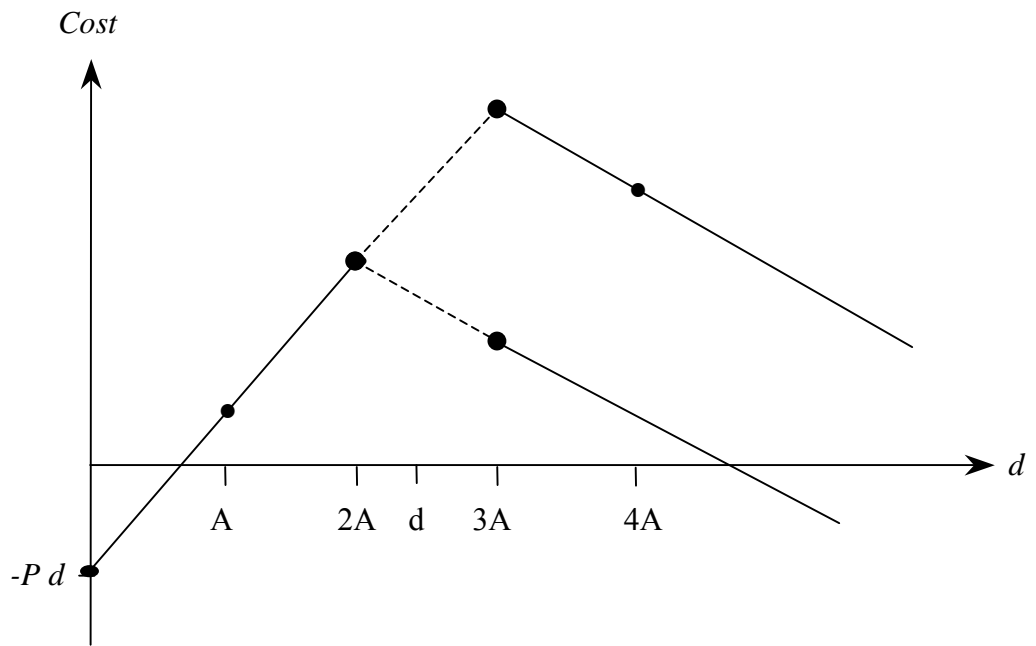


Figure 3.2: Typical behavior for the gain function $g(K)$

Let us assume that for the demand d holds

$$n A \leq d \leq (n+1) A \quad \text{for a given } n \in I .$$

If $d = n A$ then from equation (3.8) follows that:

$$g(n+1) - g(n) = -V < 0 .$$

If $d = (n+1) A$ and by using condition (3.9) we get:

$$g(n+1) - g(n) = A(R + P - C) - V > 0$$

Only in case $n A < d < (n+1)A$ we have to compare the values of $g(n+1)$ and $g(n)$. In depending on the value of the difference $g(n+1) - g(n)$ we get the answer $K^* = n+1$

or $K^* = n$. Thus the following simple algorithm for definition of the optimal fleet size K^* is:

$$\begin{aligned} \text{Define } n^* &= \min \{ n \geq 0 : n \cdot A \leq d \leq (n+1) \cdot A \} \\ &= \min \{ n \geq 0 : n \leq d/A \leq (n+1) \} \end{aligned}$$

IF $g(n^*) \geq g(n^*+1)$
 THEN $K^* = n^*$
 ELSE $K^* = n^* + 1$.

Figure (3.2) illustrate the algorithm in the special case when $n = 2$ i.e., $2A \leq d \leq 3A$.

Taken together, we can conclude that the simplicity of this example illustrates several key ideas:

1. The profitability of the system can be expressed in terms components (revenue, stockout cost, TU ownership cost, and TU movement cost) each of which is a function of fleet size.
2. As long as the marginal value (incremental increase in revenues and decreases in stockout and operating costs) of an additional TU is greater than the marginal cost (incremental increase in fleet ownership cost) it pays to increase the size of the fleet.
3. At some point, the marginal value of TU begins to decrease and eventually drops below the marginal cost at which point additional TU's reduce profitability.
4. The point at which marginal value equals marginal cost is dependent on demand and the cost parameters.

3.2 One-to-Many Case

In this case “one-to-many” we consider the problem of determining the fleet size for a single type of TU used to transport the items, which produced at a single origin to many destinations. The single origin may be a warehouse, manufacturing plant, or distribution center and the destinations may be retail outlets, other manufacturing plants, or other distribution centers. The origin stores the products, which can be ordered by the destinations, which have to serve the demand for these products. The origin else can order product from producers or produce the products itself. Empty TU's are loaded at the origin. Afterwards, the loaded TU's are moved directly to destinations. When the items are

assembled, TU's are emptied, returned directly to the origin, stored in a bank of empty TU's awaiting reloading, and the cycle is repeated. Figure (3.3) shows this system.

The previous studies investigated that the number of TU's needed in the system depends on many things:

- a- The nature of the items.
- b- Ownership costs of TU's.
- c- Transportation cost of the items.
- d- Transportation cost of empty TU's.
- e- Travel times between the origin and destinations.
- f- Shipment schedule and lot sizes of empty and loaded TU's.

Daganzo (1999) illustrated a corresponding model for fleet sizing under the following assumptions:

- 1- the items are homogeneous (i.e. have similar transportation and inventory costs).
- 2- the transport lot size is fixed and small compared to the number of items.
- 3- production of the items is in an arbitrary, but deterministic.
- 4- the items are demanded and produced at a constant rate.
- 5- the items are distributed with identical TU's.
- 6- the items are always enough at the origin.
- 7- travel times of TU's are deterministic.

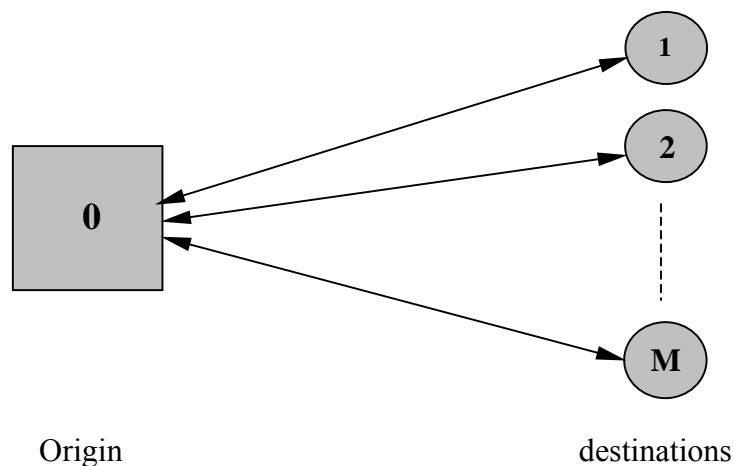


Figure 3.3: Scheme for One-to-Many System

Daganzo investigated the impact of travel time uncertainty, in order to carry out the investigation the last assumption was relaxed. The problem is completely deterministic when the travel time is certain and the production cycle is also fixed. Accordingly, the

obvious solution of the fleet sizing problem require enough TU's in order to accommodate the specified production and transport schedules. In contrast, when TU travel time is uncertain, there is a probability that the transport will not arrive in the definite schedule time, causing a shortage. Therefore, additional TU inventories must be maintained in order to increase the TU fleet size. In conclusion, the optimal fleet size depends mainly on the relative costs of extra TU's in relation to the cost of running out. Consequently, it is not easy to determine how many exactly TU's should be in a given fleet. In the previous research there are many ways, which are applicable to general types of transportation equipment. We will finish the present subchapter with some formulas for a deterministic model for the one-to-many case. Formulas determine the number of transportation units, which are else in the origin, the destinations, or in-transit.

Transportation units at the Origin

The items are produced or stored at the origin at rate R (items / time unit). It is used at different destinations, with usage rates λ_i (items / time unit), $i=1,\dots,M$. Consequently, the total items usages are:

$$\Lambda = \sum_{i=1}^M \lambda_i \quad (3.11)$$

As we said before in the assumptions that enough items will be always available in the origin, which means $R \geq \Lambda$ so that there is a feasible production schedule. Assume that at time t a total items Q^* are produced or stored at the origin, which is sufficient for L time units usage at the destinations such that $t < L$. Thus Q^* is equal to

$$Q^* = L\Lambda \quad (3.12)$$

We have to assume that the empty TU's arrive at the origin continuously and the transit times of TU's between origin and destinations are know with certainty, so that the arrival rate of empty TU's is predictable. Taken together, if production are constant, we will not need any TU bank in the origin because each empty TU arrived will be filled immediately and sent out again. Therefore, we can calculate the present number of TU's easily according to the following: at the time t , the origin have Q^* items. If the number of items per TU is Q , the number of TU's loading is Q^*/Q . The number $\Lambda t/Q$ has been shipped out, so that the number present in the origin is:

$$S = Q^*/Q - \Lambda t/Q. \quad (3.13)$$

From equation (3.12) we can be rewritten (3.13) as follows:

$$S = \Lambda(L-t)/Q. \quad (3.14)$$

We can see that as $t \rightarrow L$ lead to $S \rightarrow 0$ as described above. At the other extreme, if production were instantaneous, $t \rightarrow 0$ and $S \rightarrow \Lambda L/Q$. That is, we need enough TU's in the origin at all times to hold one complete production cycle of items.

Transportation units at the destinations

The destination expend the items at a steady rate instead of using it in production cycles. Consequently, the number of TU in these destinations is the same as in the origin under the extreme of the continuous production, zero. For purposes of the model, let us define a variable η_i as the number of time units usage of a given item carried in inventory at destination i to provide a buffer against internal variability in the origin. Then the total number of TU's at the destination i are $\lambda_i \eta_i / Q$.

Transportation units In-Transit

The crucial part of the TU usage cycle is travel times. The travel time is the total time from TU leaving the origin until reaching the destination and the time required for loading or unloading and repairing TU's before they can be reused. Accordingly, if the transit time from the origin to destination i is T_i time units, then the total number of TU's in-transit between the origin and destinations are $2 \lambda_i T_i / Q$.

Now we can see that the total fleet size K of TU's needed in the system is:

$$\begin{aligned} K &= S + \sum_{i=1}^M \lambda_i (2T_i + \eta_i) / Q \\ &= \Lambda(L-t)/Q + \sum_{i=1}^M \lambda_i (2T_i + \eta_i) / Q \end{aligned} \quad (3.15)$$

In conclusion, equation (3.15) calculates the total number of transportation units, which is needed in this model under the assumptions that the production process at the origin can be adjusted without a penalty to meet the scheduled transport quantities and the items are distributed with identical TU's.

In the present section we assumed a fixed number of own TU's. The fleet sizing problem however looks for an optimal number of fleet units. This makes sense only for the multi-period situation. To formulate corresponding optimization problems we have to adapt the cost structure. For instance we have to introduce some cost for holding a TU as well as for an own TU not used during a period. Another problem is the assumption of deterministic known demand over the planning horizon. It must be expected however that the demand is stochastic. For such a situation queueing theory places as our disposal some approaches. We will briefly discuss that topic in the following section

3.3 Queueing Models for the One-to-Many Case

In most realistic problems we have to deal with randomness. Random influences can be related to travel times of the TU's, demand of TU's by the spokes, and so on. We consider the continuous time case and some queueing models. In addition to the transit times if demand is also stochastic then queueing theory is an appropriate modelling technique (see e.g. Koenigsberg and Lam (1976)). Since analytical solutions are possible only under some simplifying assumptions we define the following basic model for the single hub model with M spokes (cp. El-Ashry et al. (2006)):

1. The hub has an ample amount of a single product and a fleet of K identical TU's.
2. Spoke i generates a demand for a single TU in accordance with a Poisson process with parameter $\lambda_i > 0$, $i = 1, 2, \dots, M$.
3. The time for a trip from the hub to spoke i , for unloading and the return trip to the hub is an exponentially distributed random variable (r.v.) with parameter $\mu_i > 0$.
4. All random variables are independent.
5. Transportation orders will be served in accordance with first-come-first-served (FCFS) policy.
6. If all TU's are on the trip arriving transportation orders will be queued.
7. Following cost and gain parts are considered:
 - $c > 0$ – fixed cost per time unit for one TU,
 - $w > 0$ – waiting cost per time unit and waiting order,
 - $c_s > 0$ – cost per time unit for a used TU.
 - $R > 0$ – revenue for a transportation product unit

Now, we will explain the equivalences and differences to the one-to-one model. From section 3.1 we can see that the owner cost is equivalent to the fixed cost, i.e., $c = V$ but the differences are transportation cost, stockout cost and revenue. The differences arise because in the model from section 3.1 each one of them is calculated according to the product unit whereas here the calculation is per TU, i.e.,

$$c_s = C \cdot Q, \quad w = P \cdot Q \quad \text{and} \quad r = R \cdot Q.$$

Remark 3.3.1 We remark that the defined model is a $M/M/K/\infty$ queueing system with parameters λ and μ , where $\lambda = \sum_{i=1}^M \lambda_i$ and $E(\text{service time}) = 1/\mu = \sum_{i=1}^M \frac{\lambda_i}{\lambda} \times \frac{1}{\mu_i}$.

In this section we consider a random demand. Thus the necessary and sufficient condition for the existence of a stationary regime the minimum number of TU is $K > a$, where $a = \lambda/\mu$.

Now we can derive the gain function $g(K)$, which denotes the expected total gain per time unit in the steady-state regime, as:

$$g(K) = r \cdot E[\text{number of served clients}] - \{c_s \cdot E[\text{number of busy servers}] + c \cdot K + w \cdot E[\text{number of waiting clients}]\}, \text{ i.e.,}$$

$$g(K) = r \cdot E[\text{number of served clients}] - C(K),$$

where $C(K) = c_s \cdot E[\text{number of busy servers}] + c \cdot K + w \cdot E[\text{number of waiting clients}]$.

We need some important performance measure for $M/M/K/\infty$ queueing system. From queueing theory we have following formulas for the different performance measures under the existence of the steady-state regime (see e.g. Trivedi 1982):

(i) The average number of busy servers is equal to

$$E[\text{number of busy servers}] = a = \lambda/\mu.$$

(ii) The average number of served clients per time unit is equal to

$$E [\text{number of served clients}] = \lambda.$$

(iii) The average number $W(K)$ of waiting clients is given by

$$\begin{aligned} W(K) &= E [\text{number of waiting clients}] \\ &= p_0(K) \times a^K / K! \times K \times a / (K - a)^2, \end{aligned} \quad (3.16)$$

where $p_0(K)$ the probability that the system is empty is given as

$$p_0(K) = 1 / \left\{ \sum_{k=0}^K (a^k / k!) + a^{K+1} / [K!(K - a)] \right\} \quad (3.17)$$

(iv) For the steady-state probabilities $p_k(K)$ that k jobs respectively to clients are in the system it holds

$$p_k(K) = p_0(K) \times \begin{cases} a^k / k! , & k \leq K; \\ a^k / (K! K^{k-K}), & k \geq K; \end{cases} \quad k = 0, 1, 2, \dots \quad (3.18)$$

(v) For the average waiting time WT of a client holds

$$WT(K) = \lambda \cdot W(K) = \frac{a^K}{\mu \cdot (K-1)! (K-a)} \times \frac{1}{(K-a) \sum_{k=0}^{K-1} a^k / k! + a^K / (K-1)!}. \quad (3.19)$$

(vi) The average number of clients in the system can be computed as

$$L(K) = W(K) + a.$$

Thus $g(K)$ is equal to

$$\begin{aligned} g(K) &= r \cdot \lambda - C(K) \\ &= r \cdot \lambda - [c_s \cdot a + c \cdot K + w \cdot W(K)] \cdot \end{aligned} \quad (3.20)$$

We are looking for such a number K^* of TU's, which maximizes the average long-run gain per time unit. The corresponding queueing optimization problem (QOP-I) is stated as:

$$\left. \begin{array}{l} \text{Maximize } g(K) \\ \text{s. t.} \\ K > \lambda/\mu; \\ K \in \mathbf{N}. \end{array} \right\} \quad (\text{QOP-I})$$

From equation (3.20) we can see that the maximization of the gain function $g(K)$ is equivalent to the minimization of the total cost function $C(K)$, *i.e.*,

$$\max_{K \in \mathbf{N}} g(K) \Leftrightarrow \min_{K \in \mathbf{N}} C(K)$$

Consequently, instead of solving the optimization problem (QOP-I) we can also solve the optimization problem (QOP-II) for the existence of steady-state regime $K > a$.

$$\left. \begin{array}{l} \text{Minimize } C(K) \\ \text{s. t.} \\ K > \lambda/\mu; \\ K \in \mathbf{N}. \end{array} \right\} \quad (\text{QOP-II})$$

To get a solution algorithm for (QOP-II) we can follow the argumentation in El-Ashry et al. (2006). We remember that the criterion function for (QOP-II) is

$$C(K) = c_s \cdot a + c \cdot K + w \cdot W(K). \quad (3.21)$$

If we could prove that $C(K)$ is integer-convex with respect to K , the design of an optimisation algorithm is straightforward. Since $c \cdot K$ is a linear function we have to consider $W(K)$ only. We notice that from equation (3.19) follows that performance measure $WT(K)$ inherits all properties from $W(K)$.

Dyer and Proll (1977) proved the following

Theorem 3.3.1:

In the $M/M/K/\infty$ system is the performance measure $W(\cdot)$ a decreasing integer-convex functions of K for $K \geq a$.

With Theorem 3.3.1 the criterion function $C(K)$ is integer-convex with respect to the number K of servers. Now the validity of the following optimisation algorithms lies on hand (where we use the notion $\lfloor a \rfloor$ for the greatest integer not exceeding a).

Algorithm Optimal Number of Servers (ONS)

1. Initialisation:

$$K := \lfloor a \rfloor + 1; C_1 := C(K); C_0 := MAX.$$

2. WHILE ($C_1 < C_0$) DO

BEGIN

$$K := K+1;$$

$$C_0 := C_1;$$

$$C_1 := C(K)$$

END.

3. RETURN $K^* = K-1$ and $C^* = C_0$.

To demonstrate the algorithm we consider the following example

Example 3.3.1:

Let $M = 4$, $c = 100$ € /day, $w = 500$ € /day, $c_s = 50$ € /day and $r = 1000$ € /day. The arrival and service rates are given in Table (3.1). From the data in Table (3.1) and the formulas in Remark (3.3.1) we calculate $\lambda = 5 \text{ day}^{-1}$, $\mu = 5 / 6 \text{ day}^{-1}$. Since $a = \lambda/\mu = 6$, we need at least 7 servers respectively TU's. Table (3.2) contains the results of the numerical computations.

N	1	2	3	4
$\lambda_n [\text{day}^{-1}]$	0.8	1.2	0.6	2.4
$\mu_n [\text{day}^{-1}]$	0.5	1.0	0.3	2.0

Table 3.1: Arrival and Service Rates for Example 3.3.1

K	$p_0(K)$	$W(K)$	$C(K)$ [€/day]	$g(K)$ [€/day]
7	0.00157878	3.682978	2 841.489	2 158.5110
8	0.00214238	1.070945	1 635.4725	3 364.5275
9	0.00235231	0.391962	1 395.9810	3 604.0190
10	0.00243174	0.151949	1 375.9745	3 624.0255
11	0.00246166	0.059066	1 429.5330	3 570.4670
12	0.00247273	0.022474	1 511.2370	3 488.7630
13	0.00247670	0.008269	1 604.1345	3 395.8655
14	0.00247808	0.002924	1 701.4620	3 298.5380

Table 3.2: Numerical Results for Example 3.3.1

From the results in table 3.2 we can see the following:

1. The probability $p_0(K)$ that the system is empty is an increasing function of the server number K . From equation (3.17) we deduce

$$\lim_{K \rightarrow \infty} p_0(K) = e^{-a} = e^{-6} = 0.00247875.$$

2. The average number $W(K)$ of waiting transportation orders in the steady-state regime is, as stated in Theorem 3.3.1, a convex function of the number K of TU's or servers.
3. The average cost per time unit in the steady-state regime if there are K servers, $C(K)$, is a convex function of the number K of TU's or servers.
4. The average gain per time unit in the steady-state regime if there are K servers, $g(K)$, is a concave function of the number K of TU's or servers.
5. The optimal number of TU's or servers is equal to $K^* = 10$. For the given values of the cost parameters and rates underestimation of K^* is more dangerous than overestimation.
6. From table 3.2 we can see that the total cost for the optimal fleet size K^* is 1375.9745 and the corresponding gain for the optimal fleet size K^* is 3624.0255.
7. With increasing K function $g(K)$ becomes a linear function with grade $-c = -100$ €/day. This follows from the fact that for $K > a$ the expected number of served clients is a constant λ and that $W(K)$ goes to zero for $K \rightarrow \infty$.

Obviously the considered model and consequently (QOP-II) are very simple. Several generalizations are possible.

A) *With respect to the distributions.*

In case of arbitrary distribution functions for the inter-arrival and service times we get a $G/GI/K/\infty$ system. For such systems Weber (1980) proved

Theorem 3.3.2:

For the $G/GI/K/\infty$ system the performance measures $W(\cdot)$ and $WT(\cdot)$ are decreasing integer-convex functions of K for $K \geq a$.

The problem however is that for $G/GI/K/\infty$ systems we have for W and other performance measures only approximate formulas.

B) *With respect to the distribution laws of the retailers.*

We assume now that $A_i(\cdot)$ and $B_i(\cdot)$ denote the distribution function of the generation time for transportation orders in location i and the service time by the centre, respectively. We assume only that the first moments $m_1(A_i)$ and $m_1(B_i)$ are finite for all locations. If $\lambda_i = 1/m_1(A_i)$ denotes the arrival intensity of transportation orders from i then we get a $G/GI/K/\infty$ system by setting

$$\lambda = \sum_{i=1}^M \lambda_i ,$$

$$A(t) = \sum_{i=1}^M \lambda_i / \lambda \cdot A_i(t), t \geq 0 ,$$

and

$$B(t) = \sum_{i=1}^M \lambda_i / \lambda \cdot B_i(t), t \geq 0.$$

C) *With respect to the capacity of the order queue.*

We can assume that no backorders or only a finite number T of backorders is possible. Then we get the lost-case models $M/M/K/0$ or $G/GI/K/0$ respectively the finite models $M/M/K/T$ or $G/GI/K/T$.

D) *With respect to the optimization criterion.*

We can assume other criterion functions as well as other constraints. Thus instead of criterion function (3.20) we can consider the criterion: “minimum of expected waiting time”. If the company earns some money for each realised transportation then it makes sense to take the criterion “expected profit”, where we are looking for the maximum. In addition to the constraints in (QOP-II) more constraints are thinkable, as for instance: lower bound for expected waiting time; upper bound for $P(w. t. > T_w)$ with given upper bound T_w for the waiting time; bounds on the resources of the system (maximum number of waiting places and so on); or bounds on the fleet size.

Problem (QOP-II) is related to the fleet-sizing problem for a special continuous time model. In the model behind (QOP-II) we have an implicit allocation of TU’s because we assume FIFO service discipline. Another possibility is briefly considered in the following model, where we are looking for an optimal permanent allocation of a given number K of transportation units to retailers. To give a mathematical correct answer we compare the optimal solution for the FIFO discipline with the optimal solution for the permanent allocation discipline. For the latter we use a model similar to El-Ashry et al. (2006) and the there applied Marginal Analysis (see Appendix A).

Let:
$$C(n) = \sum_{i=1}^M C_i(n_i)$$

denote the total cost in the system for allocation vector $\mathbf{n} = (n_1, \dots, n_M)$.

Then we formulate (QOP-III) as follows:

$$\begin{array}{l} \text{s.t.} \\ \text{Minimize } C(n) \\ \sum_{i=1}^M n_i \leq K; \\ n_i \geq \lfloor \lambda_i / \mu_i \rfloor + 1; \\ n_i \in \mathbf{N}, i=1, \dots, M. \end{array} \quad \left. \vphantom{\begin{array}{l} \text{s.t.} \\ \text{Minimize } C(n) \\ \sum_{i=1}^M n_i \leq K; \\ n_i \geq \lfloor \lambda_i / \mu_i \rfloor + 1; \\ n_i \in \mathbf{N}, i=1, \dots, M. \end{array}} \right\} \quad \text{(QOP-III)}$$

This is a non-linear integer optimization problem with convex functions $C_i(\cdot)$. We can apply Marginal Analysis and define the following algorithm.

Algorithm Optimal Allocation of a Fleet (OAF)

1. Initial step:

For $i = 1$ to M DO $n_i^{(0)} = \lfloor \lambda_i / \mu_i \rfloor + 1$.

sum := $n_1^{(0)} + \dots + n_M^{(0)}$.

THEN IF sum > K

THEN “(QOP-III) has no solution. Stop”.

2. Iteration:

$k := 1$;

WHILE ($k \leq K - \text{sum}$) DO

BEGIN

$n^{(k)} := n^{(k-1)} + e_i$, where $e_i = (0, 0, \dots, 1, 0, \dots, 0)$

and i that index, which maximizes $C_i(n_i^{(k-1)}) - C_i(n_i^{(k-1)} + 1)$.

$k := k + 1$

END;

3. Output: optimal allocation $n^* = (n_1^{(k-1)}, \dots, n_M^{(k-1)})$.

The extension of (QOP-III) can be generalized to the consideration of the allocation cost as follows:

$$\begin{array}{l}
 \text{s.t.} \\
 \text{Minimize } C(n) \\
 \sum_{i=1}^M c_i \cdot n_i \leq C; \\
 n_i \geq \lfloor \lambda_i / \mu_i \rfloor + 1; \\
 n_i \in \mathbf{N}, i=1, \dots, M.
 \end{array}
 \left. \vphantom{\begin{array}{l} \text{s.t.} \\ \text{Minimize } C(n) \\ \sum_{i=1}^M c_i \cdot n_i \leq C; \\ n_i \geq \lfloor \lambda_i / \mu_i \rfloor + 1; \\ n_i \in \mathbf{N}, i=1, \dots, M. \end{array}} \right\} \text{(QOP-IV)}$$

where c_i denotes the cost for the allocation of a single server to location i .

For (QOP-IV) we have simply to maximize in step 3 of the algorithm Marginal analysis the fraction

$$[C_i(n_i^{(k-1)}) - C_i(n_i^{(k-1)} + 1)] / c_i.$$

Example 3.3.2:

Let $M = 4$, $c = 100 \text{ € /day}$, $w = 500 \text{ € /day}$, $c_s = 50 \text{ € /day}$. The arrival and service rates are given in Table (3.1). From the data in Table (3.1) we calculate $a_1 = 1.6$, $a_2 = 1.2$, $a_3 = 2$ and $a_4 = 1.2$. If we have one service system for all clients then the condition for existence of the steady-state regime $K > a = \lambda/\mu$. Now we have $M = 4$ isolated service systems with a corresponding condition $n_i > a_i = \lambda_i/\mu_i$, where n_i is the minimum necessary number to be allocated to location i , $i = 1, \dots, 4$. From this condition we obtain that the minimum number of TU's for each location i , $i = 1, \dots, 4$ is $n_1 = 2$, $n_2 = 2$, $n_3 = 3$ and $n_4 = 2$ i.e. the minimum number of TU's for all locations is 9. To compare the solutions of (QOP-III) with that of (QOP-II) we fix the number of TU's as $K = 10$. This means that a single remaining TU must be allocated.

The question now: To which location should the remaining TU be allocated? To answer this question we use algorithm (OAF). We need the functions $C_i(\cdot)$, $i = 1, \dots, 4$. Tables (3.3) to (3.6) contain the results of the numerical computations for each location i , $i = 1, \dots, 4$ by using the following formulas:

$$p_{0,i}(n_i) = 1 / \left\{ \sum_{k=0}^{n_i} (a_i^k / k!) + a_i^{n_i+1} / [n_i!(n_i - a_i)] \right\}$$

$$W_i(n_i) = p_{0,i} \cdot a_i^{n_i} / n_i! \cdot n_i \cdot a_i / (n_i - a_i)^2$$

$$C_i(n_i) = c_s \cdot a_i + c \cdot n_i + w \cdot W_i(n_i).$$

Location 1:

n_i	$P_{0,1}(n_i)$	$W_1(n_i)$	$C_1(n_i)$
2	0.11074197	2.83499443	1697.4972
3	0.18716578	0.31291063	536.4553

Table 3.3: Numerical Results for location1.

Location 2:

n_i	$P_{0,2}(n_i)$	$W_2(n_i)$	$C_2(n_i)$
2	0.25	0.675	597.5
3	0.27472527	0.08791209	403.956

Table 3.4: Numerical Results for location 2.

Location 3:

n_i	$P_{0,3}(n_i)$	$W_3(n_i)$	$C_3(n_i)$
3	0.11111111	0.88888888	844.44444
4	0.10344828	0.13793104	568.96552

Table 3.5: Numerical Results for location 3.

Location 4:

n_i	$P_{0,4}(n_i)$	$W_4(n_i)$	$C_4(n_i)$
2	0.25	0.675	597.5
3	0.27472527	0.08791209	403.956

Table 3.6: Numerical Results for location 4.

To obtain the answer we must calculate the difference $C_i(n_i) - C_i(n_i + 1)$ for each location i , $i = 1, \dots, 4$. After that, the remaining TU will be allocated to the location with maximum difference.

From the results of tables (3.3) to (3.6) we can get the following:

$$C_1(2) - C_1(3) = 1161.0419,$$

$$C_2(2) - C_2(3) = 193.544,$$

$$C_3(3) - C_3(4) = 275.47892,$$

$$C_4(2) - C_4(3) = 193.544.$$

That means the maximum difference occurs at location one, i.e. the remaining TU must be allocated at location one. Thus the optimal allocation of 10 TU's is $n^* = (3, 2, 3, 2)$ with total average cost of $C(n^*) = 536.4553 + 597.5 + 844.44444 + 597.5 = 2575.89974$. This is about 53,42 % higher than in example 3.3.1. Intuitively this result is clear. In example 3.3.1 we had a single service system with common waiting queue, whereas here we have 4 isolated systems. The result shows that pooling resources decreases cost.

Now we will apply the other model $M/M/K/0$. In the present model we have the same assumptions in the model $M/M/K/\infty$ but the different here there is no waiting cost and the other cost is rejected cost:

$$c_r > 0 \text{ --cost per time unit for rejected order of a single TU.}$$

From queueing theory we can define the useful functions as follows:

Poisson ratio-function is:
$$R(K, a) = 1 - \frac{e^{-a} \cdot \frac{a^K}{K!}}{e^{-a} \sum_{k=0}^K \frac{a^k}{k!}}$$

Erlang-B function is:
$$E_B(K, a) = 1 - R(K, a)$$

Now we can derive the gain function $g(K)$, which denotes the expected total gain per time unit in the steady-state regime, as:

$$g(K) = r \cdot E[\text{number of accepted clients}] - \{c_s \cdot E[\text{number of busy servers}] + c \cdot K + c_r \cdot E[\text{number of rejected clients}]\}.$$

We consider now steady-state performance measure for $M/M/K/0$ queueing system (see e.g. Gelenbe and Pujolle 1998):

(i) The traffic intensity is defined as $a = \lambda/\mu$.

(ii) The average number of arriving clients per time unit is equal to

$$E[\text{number of arriving clients}] = \lambda.$$

(iii) The average number of accepted clients per time unit is equal to

$$E[\text{number of accepted clients}] = \lambda_A = \lambda \cdot [1 - E_B(K, a)].$$

(iv) The average number of rejected clients per time unit is equal to

$$E[\text{number of rejected clients}] = \lambda_R = \lambda \cdot E_B(K, a).$$

(v) The average number S of busy servers = the average number L of clients in the system

$$E [\text{number of busy servers}] = S = L = \lambda_A / \mu = a \cdot [1 - E_B (K, a)].$$

(vi) For the steady-state probabilities $p_k (K)$ that k jobs respectively to clients are in the system it holds

$$p_k (K) = \begin{cases} p_0(K) \cdot a^k / k! & \text{if } k \leq K; \\ 0 & \text{otherwise} \end{cases} \quad (3.22)$$

where $p_0(K)$ the probability that the system is empty is given as

$$p_0(K) = \left\{ \sum_{k=0}^K (a^k / k!) \right\}^{-1} \quad (3.23)$$

Thus $g (K)$ is equal to

$$g (K) = r \cdot \lambda_A - \{ c_s \cdot E [\text{number of busy servers}] + c \cdot K + c_r \cdot E [\text{number of rejected clients}] \}$$

$$= r \cdot (\lambda - \lambda_R) - \{ c_s \cdot E [\text{number of busy servers}] + c \cdot K + c_r \cdot E [\text{number of rejected clients}] \}$$

$$= r \cdot \lambda - \{ r \cdot \lambda_R + c_s \cdot S + c \cdot K + c_r \cdot \lambda_R \}, \text{ i.e.,}$$

$$g (K) = r \cdot \lambda - C(K), \quad (3.24)$$

where $C (K) = \lambda_R (r + c_r) + c_s \cdot S + c \cdot K$, i.e.,

$$C (K) = E_B (K, a) [\lambda (r + c_r) - a \cdot c_s] + a \cdot c_s + c \cdot K. \quad (3.25)$$

Messerli (1972) and Harel (1990) proved the convexity of the Erlang-B function with respect to the nonnegative integer variable K . Since $c \cdot K$ is a linear function then the criterion function $C (K)$ is integer-convex with respect to the number K of servers.

From equation (3.24) we can see that the maximization of the gain function $g(K)$ is equivalent to the minimization of the total cost function $C(K)$, i.e.,

$$\max_{K \in N} g(K) \Leftrightarrow \min_{K \in N} C(K)$$

To solve this problem we can use the same procedure in the $M/M/K/\infty$ system.

Now we will give a numerical example to demonstrate the algorithm in this system.

Example 3.3.3:

Let $M = 4$, $c = 100$ €/day, $c_r = 500$ €/day, $c_s = 50$ €/day and $r = 1000$ €/day. The arrival and service rates are given in Table (3.7). From the data in Table (3.7) and the formulas in Remark (3.3.1) we calculate $\lambda = 5$ day⁻¹, $\mu = 5 / 6$ day⁻¹. Since $a = \lambda/\mu = 6$, we need at least 7 servers respectively TU's. Table (3.8) contains the results of the numerical computations.

N	1	2	3	4
λ_n [day ⁻¹]	0.8	1.2	0.6	2.4
μ_n [day ⁻¹]	0.5	1.0	0.3	2.0

Table 3.7: Arrival and Service Rates for Example 3.3.3

K	$p_0(K)$	$E_B(K, a)$	$\lambda_R(K)$	$S(K)$	$C(K)$ [€/day]	$g(K)$ [€/day]
7	0.00333175	0.18505491	0.92527455	4.8897	2332.3968	2667.6032
8	0.00292569	0.12187589	0.60937945	5.2687	1977.5042	3022.4958
9	0.00270584	0.07514504	0.37572520	5.5491	1741.0428	3258.9572
10	0.00258910	0.04314180	0.21570900	5.7411	1610.6185	3389.3815
11	0.00252958	0.02299093	0.11495465	5.8621	1565.5370	3434.4630
12	0.00250083	0.01136481	0.05682405	5.9318	1581.8261	3418.1739
13	0.00248778	0.00521793	0.02608965	5.9687	1637.5695	3362.4305
14	0.00248223	0.00223127	0.01115635	5.9866	1716.0645	3283.9355

Table 3.8: Numerical Results for Example 3.3.3

From the results in table 3.8 we can see the following:

1. The average cost per time unit in the steady-state regime if there are K servers, $C(K)$, is a convex function of the number K of TU's or servers.
2. The average gain per time unit in the steady-state regime if there are K servers, $g(K)$, is a concave function of the number K of TU's or servers.
3. The optimal number of TU's or servers is equal to $K^* = 11$.
4. From table 3.8 we can see that the total cost for the optimal fleet size K^* is 1565.5370 and the corresponding gain for the optimal fleet size K^* is 3434.4630.

Taken together, we can conclude that the total cost for the optimal fleet size K^* in example 3.3.3 is higher than the total cost for the optimal fleet size K^* in example 3.3.1, i.e., the system with waiting queue is better than the system without waiting queue.

Chapter 4

Single-Period Deterministic Models

4.1 Introduction

We concentrate now on the fleet-sizing-and-allocation problem for single hub networks. We formulate the problem as a non-linear integer programming problem, where the objective function represents the sum of transportation, inventory, and shortage cost. Our presentation which has been discussed by El-Ashry et al. (2006) is organized as follows. In Section 4.2 we define a rather general decision problem. Afterwards in Section 4.3 we investigate a simple model with deterministic demand, fixed number and fixed capacity of TU's. We consider two cases:

- (1) Renting of additional TU's from outside the system is not allowed, and
- (2) Renting of additional TU's from outside the system is allowed.

For that simple model we can prove some interesting results on the optimal allocation.

4.2 Description of a Basic Model and Decision Problem

In the following we imagine the hub and spoke system (HAS system) as a supply network for consumable products or services. Fundamental modelling assumptions are related to the

Time flow, which can be continuous or discrete, whereby the elements of the model can be stationary or non-stationary;

Planning horizon – finite or infinite, rolling or fixed;

Available information – full, partial (adaptive model), no information;

Decision/control possibilities – ordering policies, policies for transportations between the warehouse and the retailers and between the retailers, number of TU's, leasing of additional TU's from outside or letting of own TU's, handling of transportation orders (priorities), and satisfaction of the demand;

Goal function – long-run average cost, the total discounted cost, average waiting times, fill rates or service rates, and so on. Thereby the goal function may include multiple criterions or a single criterion.

With respect to the elements of a model we assume the following:

1. There exists a central warehouse, the hub, and M retailers, the spokes. The central warehouse stores the products, which can be ordered by the retailers, which have to serve the demand for these products. The warehouse else can order product from producers or produce the products itself.
2. The central warehouse is described by
 - Number of products,
 - Storage capacities (for all products or for each single product),
 - Ordering policy,
 - Number of own TU's,
 - Allocation policy of own TU's and leased TU's,
 - Cost and gain structure.
3. For the retailers we have to define
 - The demand process, which can be stochastic or deterministic,
 - The acceptance-rejection rule for arriving demand,
 - The ordering policy,
 - The storage capacities,
 - The pooling scheme for lateral transports between the retailers,
 - The cost and gain structure.

The retailers may be identical or different with respect to these characteristics.

4. The TU's we can divide into classes. Such a class is characterised by
 - Transportation times (deterministic or random, identical or different),
 - Transportation capacities (measured in product units),
 - Transportation cost (fixed cost, volume and time proportional cost, other functions),
 - Standstill cost for depreciations and so on.

Each TU moves from the warehouse to a retailer and back. On the return way a TU can transport for instance packing material and/or empty TU's.

5. Lead times can exist for the supply of products to the warehouse, for transportation of product to and between retailers. These lead times may be zero, deterministic or random.

6. The cost and gain structure includes
 - Ordering cost,
 - Fixed cost per time unit and TU for own TU's,
 - Rental cost per time unit and rented TU,
 - Cost for a transportation per time unit and TU,
 - Waiting cost/shortage cost per time unit for transportation orders and demand,
 - Holding cost for stored product in the warehouse and the retailers,
 - Profit from sold product.

We finish the description of model elements with some remarks on the pooling scheme and the acceptance-rejection rule (AR-rule). The AR-rule handles the arriving demand at retailers. In the *backordering* case all demand is accepted, whereas in the *lost-sales* case arriving demand, which cannot be satisfied by available inventory at a retailer, is rejected and lost. In the *intermediate* case exists a waiting queue with finite capacity for waiting demand. To increase the quality of service for the whole system the retailers and their inventories on hand may be pooled. In case of shortage of product at one retailer and available product at another one lateral transportations between retailers of the same pool are an alternative. The pooling scheme should describe as well the pooling of retailers into groups as the pooling of the inventories at the retailers. The realization of lateral transportations requires additional resources. If the lead times for transports from the warehouse to retailers are high compared with those between retailers then pooling may be decrease cost and increase the quality of service. However, pooling complicates the problem to control the whole system.

Now we can formulate a general decision respectively optimization problem as the problem to define for a HAS system

- a. The number of own TU's,
- b. The number of rented TU's,
- c. The ordering of the central warehouse,

- d. The release of transportation orders by the retailers, and
- e. The allocation of TU's to transportation orders

In such a way that given performance criteria will be optimized.

From the above description of model elements and the formulated general optimization problem it is obvious, that in dependence of the concrete assumptions to the elements of a model we get a great variety of different models and problems. Most of them are not analytically tractable. In the following sections we will consider some models, which we have investigated and for which we have at least some algorithms for getting a solution. We will move from the simplest case to more realistic models, i.e., from a single-period model with deterministic demand to a queueing model with continuous time.

4.3 A single-Period Deterministic-Demand Model

We consider the HAS system with a single-period planning horizon, deterministic demand, full information, and total cost criterion. To be concrete we assume the following:

1. There are a single warehouse and M retailers.
2. The warehouse has an ample amount of a single product and K TU's.
3. The demand at retailer i during the period is known and equal to d_i , $i=1, \dots, M$. Each retailer has zero inventory and can order only full TU's. The cost structure comprises for retailer i
 - p_i - shortage cost per unit not satisfied demand,
 - h_i - holding cost per unit not sold product.
4. The K TU's belong to a single class, which is characterized by negligible transportation times, transportation capacity Q , and transportation cost c_i per TU going to retailer i .
5. The system can rent additional TU's from outside with the same characteristics as the own TU's, but with $c_{r,i}$ as transportation cost for a rented TU going to retailer i . It is natural to assume $c_i < c_{r,i}$ for all i .
6. We have no lead times.

The problem is to allocate the K TU's to the retailers such that the total cost will be minimized. In the following we investigate solutions for that problem with and without the possibility to rent TU's from outside. We start with the case that no TU's can be rented.

4.3.1 Solutions for the Allocation Problem without Renting Possibilities

To formalize the corresponding optimisation problem we define

- $n_i \in \mathbf{N} = \{0, 1, 2, \dots\}$ the number of TU's allocated to retailer i ,
- $n = (n_1, n_2, \dots, n_M) \in \mathbf{N}^M$ the allocation-vector,
- $k_i(n_i)$ the single-period cost for retailer i when n_i TU's are allocated,
- $k(n)$ the total cost for the period under allocation-vector n .

It holds that

$$k_i(n_i) = c_i \cdot n_i + \begin{cases} p_i \cdot (d_i - n_i \cdot Q), & n_i \cdot Q \leq d_i; \\ h_i \cdot (n_i \cdot Q - d_i), & n_i \cdot Q \geq d_i, \end{cases} \quad (4.3.1)$$

and

$$k(n) = \sum_{i=1}^M k_i(n_i) = \sum_{i=1}^M \{c_i n_i + p_i (d_i - n_i Q)^+ + h_i (n_i Q - d_i)^+\}, \quad (4.3.2)$$

$$n = (n_1, n_2, \dots, n_M)$$

where $(x)^+ = \max(x, 0)$ for any real x . With these definitions the optimization problem to be solved can be reformulated as optimization problem (OP-I):

For given $K \in \mathbf{N}$ find an allocation-vector $n^{(I)}$, which minimizes the total cost, i.e.,

$$\begin{array}{l} \text{Minimize } k(n) \\ \text{s.t.} \\ \sum_{i=1}^M n_i \leq K; \\ n_i \in \mathbf{N}, i = 1, \dots, M. \end{array} \quad \left. \vphantom{\begin{array}{l} \text{Minimize } k(n) \\ \text{s.t.} \\ \sum_{i=1}^M n_i \leq K; \\ n_i \in \mathbf{N}, i = 1, \dots, M. \end{array}} \right\} \quad (\text{OP-I})$$

Obviously $k_i(\cdot)$ is a piecewise linear, integer-convex function of n_i for $i=1, \dots, M$, and, consequently, $k(\cdot)$ is integer-convex in all its arguments. This convexity property considerably simplifies (OP-I).

Let $n^* = (n_1^*, n_2^*, \dots, n_M^*) \in \mathbf{N}^M$ denote that allocation-vector, which solves (OP-I) for $K = \infty$, i.e., we do not take into consideration that the central warehouse has only K TU's. Since goal function $k(\cdot)$ is additive with respect to its arguments, n_i^* minimizes function $k_i(\cdot)$ for each i . To calculate n_i^* we use the first-order differences $\Delta k_i(\cdot)$, which are defined as

$$\Delta k_i(n_i) := k_i(n_i+1) - k_i(n_i), \quad n_i \geq 0. \quad (4.3.3)$$

From convexity follows that n_i^* fulfils the two inequalities

$$k_i(n_i^*) \leq k_i(n_i^*+1) \quad \text{and} \quad k_i(n_i^*) \leq k_i(n_i^*-1)$$

or, equivalently,

$$\Delta k_i(n_i^*) \geq 0 \quad \text{and} \quad \Delta k_i(n_i^*-1) \leq 0.$$

In other words, it holds that

$$n_i^* = \operatorname{argmin}\{n: \Delta k_i(n) \geq 0\} = \operatorname{argmax}\{n: \Delta k_i(n-1) \leq 0\}. \quad (4.3.4)$$

The characterization of n_i^* by (4.3.4) is too general for use. With (4.3.1) we get from (4.3.3) that

$$\Delta k_i(n_i) = c_i + \begin{cases} -p_i \cdot Q, & (n_i + 1) \cdot Q \leq d_i; \\ h_i \cdot Q - (h_i + p_i) \cdot (d_i - n_i \cdot Q), & n_i \cdot Q \leq d_i < (n_i + 1) \cdot Q \\ h_i \cdot Q, & d_i < n_i \cdot Q. \end{cases} \quad (4.3.5)$$

Now we have to distinguish between four cases.

Case 1: $(n_i^*+1) \cdot Q \leq d_i$.

From the optimality conditions for n_i^* and (4.3.5) follows

$$\Delta k_i(n_i^*) = c_i - p_i \cdot Q \geq 0 \geq \Delta k_i(n_i^*-1) = c_i - p_i \cdot Q,$$

i.e.,

$$\Delta k_i(n_i^*) = \Delta k_i(n_i^*-1) = c_i - p_i \cdot Q = 0.$$

This is equivalent to $c_i = p_i \cdot Q$, i.e., the transportation costs for product quantity Q are equal to the shortage costs for the same quantity. In other words, to order product generates the same cost as not to order. Thus we have a trivial solution $n_i^* = 0$. To exclude that triviality we introduce

Assumption (ET) – *Efficiency of Transportation*:

$$c_i < p_i \cdot Q \quad \text{for } i = 1, \dots, M.$$

One consequence of assumption (ET) is that in case 1 we get the contradiction

$$c_i - p_i \cdot Q > 0 \geq c_i - p_i \cdot Q,$$

i.e., $n_i^* = 0$ is not optimal.

Case 2: $n_i^* \cdot Q \leq d_i < (n_i^*+1) \cdot Q$.

Here we get, taking into account assumption (ET), that

$$\Delta k_i(n_i^*) = c_i + h_i \cdot Q - (h_i + p_i) \cdot (d_i - n_i^* \cdot Q) \geq 0 > c_i - p_i \cdot Q = \Delta k_i(n_i^*-1).$$

Rearranging terms this is equivalent to the optimality condition

$$(n_i^*+1) - \frac{d_i}{Q} \geq \frac{p_i - c_i / Q}{h_i + p_i}$$

or

$$n_i^* = \operatorname{argmin} \left\{ n: (n+1) - \frac{d_i}{Q} \geq \frac{p_i - c_i / Q}{h_i + p_i} \right\}.$$

Case 3: $(n_i^*-1) \cdot Q \leq d_i < n_i^* \cdot Q$.

In the same way as for case 2 we get the optimality condition

$$n_i^* - \frac{d_i}{Q} \leq \frac{p_i - c_i / Q}{h_i + p_i}$$

or $n_i^* = \operatorname{argmax}\{n: n - \frac{d_i}{Q} \leq \frac{p_i - c_i / Q}{h_i + p_i}\}$.

Case 4: $d_i \leq (n_i^*-1) \cdot Q$.

With assumption (ET) and optimality condition (4.3.4) we get

$$0 > c_i - p_i \cdot Q = \Delta k_i(n_i^*) \geq 0.$$

But this is a contradiction.

We have shown that only the cases 2 and 3 are relevant and that the optimality condition is

$$\begin{aligned} n_i^* &= \operatorname{argmin}\{n: (n+1) - \frac{d_i}{Q} \geq \frac{p_i - c_i / Q}{h_i + p_i}\} \\ &= \operatorname{argmax}\{n: n - \frac{d_i}{Q} \leq \frac{p_i - c_i / Q}{h_i + p_i}\}. \end{aligned} \quad (4.3.6)$$

Remark 4.3.1

The optimality condition (4.3.6) has an important intuitive interpretation. The optimal number of ordered quantities Q is at least equal to the integral part n' of the demand divided by that quantity. If there remains some unsatisfied demand the ordering of an additional TU depends on the amount of unsatisfied demand $d_i - n' \cdot Q$ and the cost factors. Thus the minimization of function $k_i(\cdot)$ represents the classical inventory problem with the restriction to order sizes equal to multiples of a given quantity Q .

Let us return now to problem (OP-I). It is obvious that

$$n^{(l)} \equiv n^* \quad \text{if} \quad \sum_{i=1}^M n_i^* \leq K.$$

Otherwise we suggest the following procedure to solve (OP-I):

1. Calculate n^* using (4.3.6).
2. REPEAT
 - (i) Chose a retailer i .
 - (ii) Decrease n_i^* by one.
- UNTIL $\sum_{i=1}^M n_i^* \leq K$.

For the applicability of that procedure we have to answer two questions:

1. How to chose the retailer i in Step 2(i) ?
2. Will the search process stop on an optimal solution ?

The convexity of the goal function allows applying Marginal Analysis (MA) (see Fox 1966). MA will answer both questions. The given algorithm (MA) and some important properties are given in the appendix A. The formulation there is for the maximization of a concave and strictly increasing function f .

We remark that maximization, concavity, and to be strictly increasing for a function f is equivalent to minimization, convexity, and to be strictly decreasing, respectively, of function $(-f)$. Now we put $(-f) = k$ with k from (4.3.3). Since $k_i(\cdot)$ is an integer-convex function of n_i and since from the definition of n_i^* follows that $k_i(\cdot)$ is strictly decreasing for $n_i = 0, \dots, n_i^*, i=1, \dots, M$, algorithm Marginal Analysis (MA) generates an optimal solution for (OP-I) by Property 3 in the Appendix A. The general algorithm (MA) substantiates to

Algorithm Marginal analysis for (OP-I)

{to solve (OP-I) in case $\sum_{i=1}^M n_i^* > K$ }

Input : $K; n_i^*, i=1, \dots, M$.

1. Initial solution $n^{(0)} = (n_1^{(0)}, \dots, n_M^{(0)})$ with $n_i^{(0)} = n_i^*, i=1, \dots, M$.

2. $r := 1$.

3. $n^{(r)} = n^{(r-1)} - e_i$, where $e_i = (0, 0, \dots, 1, 0, \dots, 0)$

and i that index, which minimizes $\Delta k_i(n_i^{(r-1)}) := k_i(n_i^{(r-1)} - 1) - k_i(n_i^{(r-1)})$, $n_i^{(r-1)} > 0$.

4. Stop if $r = K - \sum_{i=1}^M n_i^*$. Otherwise $r := r+1$ and go to 3.

Output : Allocation vector n .

In the above given formulation algorithm (MA) is applied “in reverse”. From Property 3 it follows that we stop at the optimal solution. Another approach to solve (OP-I) in case $\sum_{i=1}^M n_i^* > K$ is dynamic programming, but sometimes with a considerable higher numerical effort. A stolid application of Marginal Analysis naturally solves (OP-I). But, using the property that $k_i(\cdot)$ is piecewise linear, it is easy to speed up the solution process in case $\sum_{i=1}^M n_i^* > K$. For that we sort all retailers such that

$$-\Delta k_1(0) \geq -\Delta k_2(0) \geq \dots \geq -\Delta k_M(0). \quad (4.3.7)$$

Since $\Delta k_i(n_i) = c_i - p_i \cdot Q$ for $(n_{i+1}) Q \leq d_i$ the inequalities (4.3.7) are equivalent to

$$p_1 \cdot Q - c_1 \geq p_2 \cdot Q - c_2 \geq \dots \geq p_M \cdot Q - c_M. \quad (4.3.8)$$

From the inequalities (4.3.8) follows that to retailer 1 will be allocated TU's until

$$n_1 \cdot Q \leq d_1 \text{ and } n_1 \leq K.$$

Thus

$$n'_1 := \operatorname{argmax}\{n: n \cdot Q \leq d_1 \text{ and } n \leq K\}$$

denotes a first amount of TU's allocated to retailer 1 (cp. Remark 4.3.1).

If $n'_1 = K$ then the process stops. Otherwise, in (4.3.8) we have to replace

$$p_1 \cdot Q - c_1 \text{ by } -\Delta k_1(n'_1) = (d_1 - n'_1 \cdot Q) \cdot (h_1 + p_1) - c_1 - h_1 \cdot Q$$

and to find for retailer 1 the new place in (4.3.7) respectively (4.3.8).

If $-\Delta k_1(n'_1) \geq p_2 \cdot Q - c_2$

then retailer 1 gets a last additional TU, because of

$$-\Delta k_1(n'_1+1) = -(c_1 + h_1 \cdot Q) < p_M \cdot Q - c_M.$$

Otherwise the process continues with retailer 2 and so on. If in this allocation procedure the retailer to be considered next is an already considered retailer (with a first amount n'_i of allocated TU's) that retailer gets an additional single TU and is no more considered. We describe that allocation procedure by the following algorithm.

Algorithm *Optimal Allocation Procedure (OAP-I)*

{to solve (OP-I) in case $\sum_{i=1}^M n_i^* > K$ }

Input: $K; Q; d_i, i=1, \dots, M.$

1. Preliminaries: $n'_i = \lfloor d_i/Q \rfloor, i=1, \dots, M.$ ¹

2. Initialization:

$$n_i := 0, i=1, \dots, M; nsum := 0.$$

3. Sorting:

Define a permutation $\mathbf{i} = (i_1, \dots, i_M)$ of integers 1 to M such that

$$p_{i_1} \cdot Q - c_{i_1} \geq p_{i_2} \cdot Q - c_{i_2} \geq \dots \geq p_{i_M} \cdot Q - c_{i_M}.$$

4. Iteration:

WHILE $nsum < K$ DO

a) Allocation-step:

$$\text{IF } n_{i_1} = 0 \text{ AND } n'_{i_1} > 0$$

$$\text{THEN } n_{i_1} := \min(n'_{i_1}, K - nsum) \quad \text{and} \quad nsum := nsum + n_{i_1}$$

$$\text{ELSE } n_{i_1} := n_{i_1} + 1 \quad \text{and} \quad nsum := nsum + 1.$$

b) Resorting-step:

Calculate $-\Delta k_{i_1}(n_{i_1})$ and find the new position for retailer i_1 in the permuted sequence of retailers.

Output : Optimal allocation vector $n.$

¹ $\lfloor x \rfloor$ denotes the greatest integer not exceeding x for any real $x.$

We remark that in case $\sum_{i=1}^M n_i^* > K$ algorithm OAP-I as well as algorithm Marginal Analysis for (OP-I) stops before all retailers are satisfied, i.e., there are some retailers without any delivery, independent of their demand values. Such discrepancies between the demand and the total transporting capacity can lead to very high costs (cp. the examples). To prevent this we next consider a model, where additional TU's can be rented from outside the system.

4.3.2 Solutions for the Allocation Problem with Renting Possibilities

We assume now that the central warehouse can rent additional TU's from outside the system with the same characteristics as the own TU's, but with $c_{r,i}$ as transportation cost for a rented TU going to retailer i . It is natural to assume $c_i \leq c_{r,i}$ for all i . Furthermore, to rent a TU will be profitable only if it leads to decreasing costs. A sufficient condition for that is to complete assumption (ET) by assumption (ER) – *Efficiency of Renting*:

$$c_{r,i} < p_i \cdot Q \text{ for } i = 1, \dots, M.$$

Let $r = (r_1, r_2, \dots, r_M) \in \mathbf{N}^M$ denote the allocation-vector for rented TU's and $k(n, r)$ the total single-period cost under the two allocation-vectors n and r . In analogy to (4.3.1) and (4.3.2) it holds for $n, r \in \mathbf{N}^M$ that

$$k_i(n_i, r_i) = c_i \cdot n_i + c_{r,i} \cdot r_i + \begin{cases} p_i \cdot (d_i - (n_i + r_i) \cdot Q), & (n_i + r_i) \cdot Q \leq d_i; \\ h_i \cdot ((n_i + r_i) \cdot Q - d_i), & (n_i + r_i) \cdot Q \geq d_i, \end{cases} \quad (4.3.9)$$

and

$$\begin{aligned} k(n, r) &= \sum_{i=1}^M k_i(n_i, r_i) \\ &= \sum_{i=1}^M \left\{ c_i n_i + c_{r,i} \cdot r_i + p_i (d_i - (n_i + r_i) Q)^+ + h_i ((n_i + r_i) Q - d_i)^+ \right\} \end{aligned} \quad (4.3.10)$$

Again $k_i(\cdot, \cdot)$ is a piecewise linear, integer-convex function of n_i and r_i for $i=1, \dots, N$, $k(\cdot, \cdot)$ is integer-convex in all its arguments, and we have a second *optimisation problem* (OP-II):

For given $K \in \mathbf{N}$ find allocation-vectors $n^{(II)}$ and r^* , which minimize the total cost, i.e.,

$$\left. \begin{array}{l} \text{Minimize } k(n, r) \\ \text{s.t.} \\ \sum_{i=1}^M n_i \leq K ; \\ n_i, r_i \in \mathbf{N}, i = 1, \dots, M. \end{array} \right\} \quad (\text{OP-II})$$

From the assumption $c_i < c_{r,i}$ for all i it naturally follows that before any TU will be rented all own TU's must be allocated. Thus (OP-II) makes sense only if $\sum_{i=1}^M n_i^* > K$, which we will assume in the following.

At first we introduce for each i the differences

$$\Delta_n k_i(n_i, r_i) := k_i(n_i+1, r_i) - k_i(n_i, r_i)$$

and

$$\Delta_r k_i(n_i, r_i) := k_i(n_i, r_i+1) - k_i(n_i, r_i).$$

From (4.3.9) follows for all i that

$$\Delta_n k_i(n_i, r_i) = c_i + \begin{cases} -p_i \cdot Q, & (n_i + r_i + 1) \cdot Q < d_i; \\ h_i \cdot Q - (h_i + p_i) \cdot [d_i - (n_i + r_i) \cdot Q], & (n_i + r_i) \cdot Q \leq d_i < (n_i + r_i + 1) \cdot Q; \\ h_i \cdot Q, & d_i < (n_i + r_i) \cdot Q \end{cases} \quad (4.3.10)$$

and

$$\Delta_r k_i(n_i, r_i) = \Delta_n k_i(n_i, r_i) + c_{r,i} - c_i.$$

At first sight we can not apply Marginal Analysis to solve (OP-II) because of the functions $k_i(\cdot, \cdot)$ depend now on two non-negative integer variables. But it is possible to transform these functions in such a way that they will depend on a single variable only. To realise that we argue as follows. Obviously, for the optimal solution $(n^{(II)}, r^*)$ of (OP-II) holds $n_i^{(II)} + r_i^* \leq n_i^*$, $i = 1, \dots, M$, where n_i^* is from (4.3.6).

In case of $K = 0$ it holds $n^{(II)} = (0, \dots, 0)$ and $r^* \leq n^*$. Since the number of rented TU's is not limited we can start our procedure to solve (OP-II) with $r = n^*$.

The corresponding cost for retailer i are

$$k_i(0, n_i^*) = c_{r,i} \cdot n_i^* + p_i (d_i - n_i^* \cdot Q)^+ + h_i (n_i^* \cdot Q - d_i)^+$$

If we now replace n_i rented TU's by own TU's we get cost of

$$k_i(n_i, n_i^* - n_i) = c_i \cdot n_i + c_{r,i} \cdot n_i^* + p_i (d_i - n_i^* \cdot Q)^+ + h_i (n_i^* \cdot Q - d_i)^+$$

or

$$k_i(n_i, n_i^* - n_i) = n_i \cdot (c_i - c_{r,i}) + k_i(0, n_i^*) < k_i(0, n_i^*).$$

Consequently, since $k_i(0, n_i^*)$ is a constant and $c_i < c_{r,i}$ the cost saving

$$k_i(0, n_i^*) - k_i(n_i, n_i^* - n_i) = n_i \cdot (c_{r,i} - c_i)$$

replacing n_i rented TU's by own TU's is a linear, strictly increasing function of n_i , $n_i \leq n_i^*$. Thus to maximize the gain from replacing rented TU's by the available K own TU's we can directly apply Marginal Analysis if we put

$$f_i(n_i) = n_i \cdot (c_{r,i} - c_i), \quad i = 1, \dots, M.$$

Again the solution process can be sped up using the linearity property of $f_i(n_i)$. For this we number the retailers such that

$$c_{r,1} - c_1 \geq c_{r,2} - c_2 \geq \dots \geq c_{r,M} - c_M > 0.$$

Then it is obvious that for retailer 1 the number of replaced rented TU's by own TU's is equal to $\min\{n_1^*, K\}$. If $K > n_1^*$ then retailer 2 replaces $\min\{n_2^*, K - n_1^*\}$ rented TU's. Continuing that replacement process until all K TU's are allocated the process stops at a

retailer $j \leq M$. Now we must consider two cases – at retailer j all rented TU's can be replaced and not all rented TU's can be replaced.

In the first case we have

$$n = (n_1^*, \dots, n_j^*, 0, \dots, 0) \quad \text{and} \quad r = (0, \dots, 0, n_{j+1}^*, \dots, n_M^*)$$

as up-to now calculated solution of (OP-II). To finish the solution process it remains to calculate for retailers $i = j+1, \dots, M$ the optimal number r_i^* of rented TU's. It is obvious that the optimality condition for each i is similar to (4.3.6), i.e., r_i^* can be calculated from

$$\begin{aligned} r_i^* &= \operatorname{argmin} \left\{ r : (r+1) \geq \frac{d_i}{Q} + \frac{p_i - c_{r,i}/Q}{h_i + p_i} \right\} \\ &= \operatorname{argmax} \left\{ r : r \leq \frac{d_i}{Q} + \frac{p_i - c_{r,i}/Q}{h_i + p_i} \right\}. \end{aligned} \quad (4.3.11)$$

We remark that from (4.3.6) and (4.3.11) follows that

$$r_i^* \leq n_i^* \quad \text{for } i=1, \dots, M.$$

In the second case retailer j cannot replace all rented TU's by own TU's. Thus the up-to now calculated solution is

$$n = (n_1^*, \dots, n_{j-1}^*, n_j, 0, \dots, 0)$$

and

$$r = (0, \dots, 0, r_j, n_{j+1}^*, \dots, n_M^*)$$

with

$$0 < n_j < n_j^* \quad \text{and} \quad r_j = n_j^* - n_j > 0.$$

For a retailer j with own and rented TU's, whereby the number n_j of own TU's is fixed, the optimal number $r_j(n_j)$ of rented TU's has also to fulfill the optimality condition (4.3.11), i.e.,

$$\begin{aligned} r_j(n_j) &= \operatorname{argmax} \left\{ r : r + n_j \leq \frac{d_j}{Q} + \frac{p_j - c_{r,j}/Q}{h_j + p_j} \right\} \\ &= \operatorname{argmax} \left\{ r : r \leq \frac{d_j - n_j \cdot Q}{Q} + \frac{p_j - c_{r,j}/Q}{h_j + p_j} \right\}. \end{aligned} \quad (4.3.12)$$

Again it remains to calculate for retailers $i = j+1, \dots, M$ from (4.3.11) the optimal numbers r_i^* . Summarizing these considerations we get an algorithm similar to algorithm (OAP-I) for (OP-I).

Algorithm *Optimal Allocation Procedure OAP-II*

{to solve (OP-II) in case $\sum_{i=1}^M n_i^* > K$ }

Input: $K; Q; d_i, i=1, \dots, M$.

1. Preliminaries :

Calculate n_i^* from (4.3.6), $i=1, \dots, M$.

2. Initialization:

$n_i := 0, r_i := n_i^*, i=1, \dots, M; nsum := 0$.

3. Sorting:

Define a permutation (i_1, \dots, i_M) of integers 1 to M such that

$$c_{r,i_1} - c_{i_1} \geq c_{r,i_2} - c_{i_2} \geq \dots \geq c_{r,i_M} - c_{i_M}.$$

4. Iteration:

WHILE $nsum < K$ DO

a) Allocation-step:

$$n_{i_1} := \min(n_{i_1}^*, K - nsum); r_{i_1} := r_{i_1} - n_{i_1}; nsum := nsum + n_{i_1}.$$

b) Resorting-step:

Delete i_1 from the permutation and renumber the remaining elements.

5. Final step:

For $i := 1$ to M DO

IF $r_i > 0$ THEN $r_i := r_i^{(II)}$, where $r_i^{(II)}$ is calculated from (4.3.11)

if $n_i = 0$ and from (4.3.12) otherwise.

Output : Optimal allocation vector $n^{(II)}$ and $r^{(II)}$.

We want to remark that from the algorithms OAP-I and OAP-II follows that the two optimisation problems (OP-I) and (OP-II) can have very different solutions. Nevertheless they have a common structural property – there exists at most a single retailer (retailer j in OAP-II) with two different transportation modes (if we assume for (OP-I) that no transport is a special mode). Of course it is no problem, having the optimal allocation vectors, to calculate the values of the corresponding goal functions.

Let us finally consider a simple numerical example.

Example 4.3.1

We assume $M = 10$ and $Q = 10$. The cost parameters and two demand vectors are given in Table (4.1). Table (4.1) contains also the values for $p_i Q - c_i$ and $c_{r,i} - c_i$ with the corresponding order places in i . The last two rows contain the optimal TU-values, calculated from (4.3.6) respectively (4.3.11). We apply now OAP-I and OAP-II to the two demand vectors.

i	1	2	3	4	5	6	7	8	9	10
c_i	1	2	3	4	5	6	5	4	3	2
$c_{r,i}$	10	10	10	10	10	10	10	10	10	10
h_i	1	1	1	1	1	1	1	1	1	1
p_i	3	4	5	6	7	8	9	8	7	6
$d_i^{(1)}$	12	24	36	48	60	72	60	48	36	24
$d_i^{(2)}$	100	150	80	30	5	20	200	60	40	90
$p_i Q - c_i$	29	38	47	56	65	74	85	76	67	58
i	10	9	8	7	5	3	1	2	4	6
$c_{r,i} - c_i$	9	8	7	6	5	4	5	6	7	8
i	1	2	4	6	8	10	9	7	5	3
n_i^*	1	3	4	5	6	8	6	5	4	3
r_i^*	1	3	4	5	6	7	6	5	4	3

Table 4.1: Data for Example 4.3.1

(i) Demand vector $d^{(1)}$.

Let us assume $K = 25$. First we solve (OP-I) applying OAP-I.

1. Preliminaries: For the present $d^{(1)}$ we get

$$n' = (1, 2, 3, 4, 6, 7, 6, 4, 3, 2).$$

2. Initialisation: $n_i := 0, i=1, \dots, M; nsum := 0$.

3. Sorting: $i = (7, 8, 6, 9, 5, 10, 4, 3, 2, 1)$.

4. Iteration:

(1) $nsum = 0 < 25 \rightarrow$ a) $n_7 = 0$ and $n'_7 > 0$ gives $n_7 = 6, nsum = 6$.

b) $\Delta k_7(6) = c_7 + h_7 Q > 0$
gives $i = (8, 6, 9, 5, 10, 4, 3, 2, 1, 7)$.

(2) $nsum = 6 < 25 \rightarrow$ a) $n_8 = 0$ and $n'_8 > 0$ gives $n_8 = 4, nsum = 10$.

b) $\Delta k_8(4) = c_8 + h_8 Q - (h_8 + p_8)(d_8 - 4Q) = -58$
gives $i = (6, 9, 5, 8, 10, 4, 3, 2, 1, 7)$.

(3) $nsum = 10 < 25 \rightarrow$ a) $n_6 = 0$ and $n'_6 > 0$ gives $n_6 = 7, nsum = 17$.

b) $\Delta k_6(7) = c_6 + h_6 Q - (h_6 + p_6)(d_6 - 7Q) = -2$
gives $i = (9, 5, 8, 10, 4, 3, 2, 1, 6, 7)$.

(4) $nsum = 17 < 25 \rightarrow$ a) $n_9 = 0$ and $n'_9 > 0$ gives $n_9 = 3, nsum = 20$.

b) $\Delta k_9(3) = c_9 + h_9 Q - (h_9 + p_9)(d_9 - 3Q) = -35$
gives $i = (5, 8, 10, 4, 3, 2, 9, 1, 6, 7)$.

(5) $nsum = 20 < 25 \rightarrow$ a) $n_5 = 0$ and $n'_5 > 0$ gives $n_5 = 5, nsum = 25$.

b) $\Delta k_5(5) = c_5 - p_5 Q = -65$
gives $i = (8, 5, 10, 4, 3, 2, 9, 1, 6, 7)$.

Output: $n^{(1)} = (0, 0, 0, 0, 5, 7, 6, 4, 3, 0)$

with $k(n^{(1)}, d^{(1)}) = 1\ 058$.

Now we allow renting of TU's, i.e., we apply OAP-II.

1. Preliminaries: From (4.3.6) we get
 $n^* = (1, 3, 4, 5, 6, 8, 6, 5, 4, 3)$.
2. Initialization: $n_i := 0, r_i := n_i^*, i=1, \dots, M; nsum := 0$.
3. Sorting: $i = (1, 2, 10, 3, 9, 4, 8, 5, 7, 6)$.
4. Iteration:
 - (1) $nsum = 0 < 25 \rightarrow$
 - a) $n_1 = 1, r_1 = 0, nsum = 1$.
 - b) $i = (2, 10, 3, 9, 4, 8, 5, 7, 6)$.
 - (2) $nsum = 1 < 25 \rightarrow$
 - a) $n_2 = 3, r_2 = 0, nsum = 4$.
 - b) $i = (10, 3, 9, 4, 8, 5, 7, 6)$.
 - (3) $nsum = 4 < 25 \rightarrow$
 - a) $n_{10} = 3, r_{10} = 0, nsum = 7$.
 - b) $i = (3, 9, 4, 8, 5, 7, 6)$.
 - (4) $nsum = 7 < 25 \rightarrow$
 - a) $n_3 = 4, r_3 = 0, nsum = 11$.
 - b) $i = (9, 4, 8, 5, 7, 6)$.
 - (5) $nsum = 11 < 25 \rightarrow$
 - a) $n_9 = 4, r_9 = 0, nsum = 15$.
 - b) $i = (4, 8, 5, 7, 6)$.
 - (6) $nsum = 15 < 25 \rightarrow$
 - a) $n_4 = 5, r_4 = 0, nsum = 20$.
 - b) $i = (8, 5, 7, 6)$.
 - (7) $nsum = 20 < 25 \rightarrow$
 - a) $n_8 = 5, r_8 = 0, nsum = 25$.
 - b) $i = (5, 7, 6)$.

5. Final step:

$$\begin{aligned}
 n &= (1, 3, 4, 5, 0, 0, 0, 5, 4, 3) \\
 r &= (0, 0, 0, 0, 6, 8, 6, 0, 0, 0) \\
 r^* &= (1, 3, 4, 5, 6, 7, 6, 5, 4, 3) \\
 n^{(II)} &= (1, 3, 4, 5, 0, 0, 0, 5, 4, 3) \text{ and} \\
 r^{(II)} &= (0, 0, 0, 0, 6, 7, 6, 0, 0, 0).
 \end{aligned}$$

Output: $n^{(II)}, r^{(II)}$ and $k(n^{(II)}, r^{(II)}, d^{(1)}) = 313$,
 which is 29,58 % of $k(n^{(I)}, d^{(1)})$.

(ii) Demand vector $d^{(2)}$.

Let us assume now $K = 50$. Again we solve first (OP-I) applying OAP-I.

1. Preliminaries: For $d^{(2)}$ we get
 $n' = (10, 15, 8, 3, 0, 2, 20, 6, 4, 9)$.
2. Initialisation: $n_i := 0, i=1, \dots, M; nsum := 0$.
3. Sorting: $i = (7, 8, 6, 9, 5, 10, 4, 3, 2, 1)$.
4. Iteration:
 - (1) $nsum = 0 < 50$ → a) $n_7 = 0$ and $n'_7 > 0$ gives $n_7 = 20, nsum = 20$.
b) $\Delta k_7(20) = c_7 + h_7 Q = 15$
gives $i = (8, 6, 9, 5, 10, 4, 3, 2, 1, 7)$.
 - (2) $nsum = 20 < 50$ → a) $n_8 = 0$ and $n'_8 > 0$ gives $n_8 = 6, nsum = 26$.
b) $\Delta k_8(6) = c_8 + h_8 Q = 14$
gives $i = (6, 9, 5, 10, 4, 3, 2, 1, 8, 7)$.
 - (3) $nsum = 26 < 50$ → a) $n_6 = 0$ and $n'_6 > 0$ gives $n_6 = 2, nsum = 28$.
b) $\Delta k_6(2) = c_6 + h_6 Q = 16$
gives $i = (9, 5, 10, 4, 3, 2, 1, 8, 7, 6)$.
 - (4) $nsum = 28 < 50$ → a) $n_9 = 0$ and $n'_9 > 0$ gives $n_9 = 4, nsum = 32$.
b) $\Delta k_9(4) = c_9 + h_9 Q = 13$
gives $i = (5, 10, 4, 3, 2, 1, 9, 8, 7, 6)$.
 - (5) $nsum = 32 < 50$ → a) $n_5 = 0$ and $n'_5 = 0$ gives $n_5 = 1, nsum = 33$.
b) $\Delta k_5(1) = c_5 + h_5 Q = 15$
gives $i = (10, 4, 3, 2, 1, 9, 8, 7, 5, 6)$.
 - (6) $nsum = 33 < 50$ → a) $n_{10} = 0$ and $n'_{10} > 0$ gives $n_{10} = 9, nsum = 42$.
b) $\Delta k_{10}(9) = c_{10} + h_{10} Q = 12$
gives $i = (4, 3, 2, 1, 10, 9, 8, 7, 5, 6)$.

$$(7) \text{ } nsum = 42 < 50 \quad \rightarrow \quad \text{a) } n_4 = 0 \text{ and } n'_4 > 0 \text{ gives } n_4 = 3, \text{ } nsum = 45.$$

$$\text{b) } \Delta k_4(3) = c_4 + h_4 Q = 14$$

$$\text{gives } i = (3, 2, 1, 10, 9, 8, 4, 7, 5, 6).$$

$$(8) \text{ } nsum = 45 < 50 \quad \rightarrow \quad \text{a) } n_3 = 0 \text{ and } n'_3 > 0 \text{ gives } n_3 = 5, \text{ } nsum = 50.$$

$$\text{b) } \Delta k_3(5) = c_3 - p_3 Q = -47$$

$$\text{gives } i = (3, 2, 1, 10, 9, 8, 4, 7, 5, 6).$$

Output: $n^{(1)} = (0, 0, 5, 3, 0, 2, 20, 6, 4, 9)$
with $k(n^{(1)}, d^{(2)}) = 1\ 278$.

The application of OAP-II gives following results.

1. Preliminaries: From (4.3.6) we get
 $n^* = (10, 15, 8, 3, 1, 2, 20, 6, 4, 9)$.
2. Initialisation: $n_i := 0, r_i := n_i^*, i=1, \dots, M; nsum := 0$.
3. Sorting: $i = (1, 2, 10, 3, 9, 4, 8, 5, 7, 6)$.
4. Iteration:
 - (1) $nsum = 0 < 50 \quad \rightarrow \quad \text{a) } n_1 = 10, r_1 = 0, nsum = 10.$
 $\text{b) } i = (2, 10, 3, 9, 4, 8, 5, 7, 6).$
 - (2) $nsum = 10 < 50 \quad \rightarrow \quad \text{a) } n_2 = 15, r_2 = 0, nsum = 25.$
 $\text{b) } i = (10, 3, 9, 4, 8, 5, 7, 6).$
 - (3) $nsum = 25 < 50 \quad \rightarrow \quad \text{a) } n_{10} = 9, r_{10} = 0, nsum = 34.$
 $\text{b) } i = (3, 9, 4, 8, 5, 7, 6).$
 - (4) $nsum = 34 < 50 \quad \rightarrow \quad \text{a) } n_3 = 8, r_3 = 0, nsum = 42.$
 $\text{b) } i = (9, 4, 8, 5, 7, 6).$
 - (5) $nsum = 42 < 50 \quad \rightarrow \quad \text{a) } n_9 = 4, r_9 = 0, nsum = 46.$
 $\text{b) } i = (4, 8, 5, 7, 6).$
 - (6) $nsum = 46 < 50 \quad \rightarrow \quad \text{a) } n_4 = 3, r_4 = 0, nsum = 49.$
 $\text{b) } i = (8, 5, 7, 6).$
 - (7) $nsum = 49 < 50 \quad \rightarrow \quad \text{a) } n_8 = 1, r_8 = 5, nsum = 50.$
 $\text{b) } i = (5, 7, 6).$

5. Final step:

$$n = (10, 15, 8, 3, 0, 0, 0, 1, 4, 9)$$

$$r = (0, 0, 0, 0, 1, 2, 20, 5, 0, 0)$$

$$r^* = (10, 15, 8, 3, 1, 2, 20, 6, 4, 9)$$

$$n^{(II)} = (10, 15, 8, 3, 0, 0, 0, 1, 4, 9) \text{ and}$$

$$r^{(II)} = (0, 0, 0, 0, 1, 2, 20, 5, 0, 0).$$

Output :

$$n^{(II)}, r^{(II)} \text{ and } k(n^{(II)}, r^{(II)}, d^{(2)}) = 395,$$

which is 30,91% of $k(n^{(I)}, d^{(2)})$.

The results for Example 4.3.1 show that the solutions of (OP-I) and (OP-II) in general are very different. That fact has some consequences for practice – if there exists the possibility to rent TU's the optimal solution differs considerably (dependent on the demand realisation) as well as in cost as in the allocation vector from the solution without renting. Furthermore, from the algorithm OAP-II follows that the solution for (OP-II) possesses an interesting property: There exists at most a single retailer with two different TU-types. The demand in all other retailers will be satisfied else by own TU's or by rented TU's. This in some sense facilitates the organization of the transportation in reality.

Example 4.3.2

In this example we assume that $M = 10$ and $Q = 20$. The cost parameters in this example = 1.5 the cost parameters in the example (4.3.1) because the capacity of the TU's in this example is double in the previous example but the rented TU's still have the same capacity and the same cost in the previous example and demand vector are given in Table (4.2). Table (4.2) contains also the values for $p_i Q - c_i$ and $c_{r,i} - c_i$ with the corresponding order places in i . The last two rows contain the optimal TU-values, calculated from (4.3.6) respectively (4.3.11). We apply now OAP-I and OAP-II to the demand vector.

i	1	2	3	4	5	6	7	8	9	10
c_i	1.5	3	4.5	6	7.5	9	7.5	6	4.5	3
$c_{r,i}$	10	10	10	10	10	10	10	10	10	10
h_i	1	1	1	1	1	1	1	1	1	1
p_i	3	4	5	6	7	8	9	8	7	6
d_i	100	150	80	30	5	20	200	60	40	90
$p_i Q - c_i$	58.5	77	95.5	114	132.5	151	172.5	154	135.5	117
i	10	9	8	7	5	3	1	2	4	6
$c_{r,i} - c_i$	8.5	7	5.5	4	2.5	1	2.5	4	5.5	7
i	1	2	4	6	8	10	9	7	5	3
n_i^*	5	8	4	2	1	1	10	3	2	5
r_i^*	5	8	4	2	1	1	10	3	2	5

Table 4.2: Data for Example 4.3.2

(i) Demand vector d .

Let us assume now $K = 30$. First we solve (OP-I) applying OAP-I.

1. Preliminaries: For d we get
 $n' = (5, 7, 4, 1, 0, 1, 10, 3, 2, 4)$.
2. Initialisation: $n_i := 0, i=1, \dots, M; nsum := 0$.
3. Sorting: $i = (7, 8, 6, 9, 5, 10, 4, 3, 2, 1)$.
4. Iteration:
 - (1) $nsum = 0 < 30 \rightarrow$
 - a) $n_7 = 0$ and $n'_7 > 0$ gives $n_7 = 10, nsum = 10$.
 - b) $\Delta k_7(10) = c_7 + h_7 Q = 27.5$
gives $i = (8, 6, 9, 5, 10, 4, 3, 2, 1, 7)$.
 - (2) $nsum = 10 < 30 \rightarrow$
 - a) $n_8 = 0$ and $n'_8 > 0$ gives $n_8 = 3, nsum = 13$.
 - b) $\Delta k_8(3) = c_8 + h_8 Q = 26$
gives $i = (6, 9, 5, 10, 4, 3, 2, 1, 8, 7)$.

- (3) $nsum = 13 < 30$ → a) $n_6 = 0$ and $n'_6 > 0$ gives $n_6 = 1$, $nsum = 14$.
b) $\Delta k_6(1) = c_6 + h_6 Q = 29$
gives $i = (9, 5, 10, 4, 3, 2, 1, 8, 7, 6)$.
- (4) $nsum = 14 < 30$ → a) $n_9 = 0$ and $n'_9 > 0$ gives $n_9 = 2$, $nsum = 16$.
b) $\Delta k_9(2) = c_9 + h_9 Q = 24.5$
gives $i = (5, 10, 4, 3, 2, 1, 9, 8, 7, 6)$.
- (5) $nsum = 16 < 30$ → a) $n_5 = 0$ and $n'_5 = 0$ gives $n_5 = 1$, $nsum = 17$.
b) $\Delta k_5(0) = c_5 + h_5 Q = 27.5$
gives $i = (10, 4, 3, 2, 1, 9, 8, 7, 5, 6)$.
- (6) $nsum = 17 < 30$ → a) $n_{10} = 0$ and $n'_{10} > 0$ gives $n_{10} = 4$, $nsum = 21$.
b) $\Delta k_{10}(4) = c_{10} + h_{10} Q - (h_{10} + p_{10}) \cdot (d_{10} - n_{10} \cdot Q) = -47$
gives $i = (4, 3, 2, 1, 10, 9, 8, 7, 5, 6)$.
- (7) $nsum = 21 < 30$ → a) $n_4 = 0$ and $n'_4 > 0$ gives $n_4 = 1$, $nsum = 22$.
b) $\Delta k_4(1) = c_4 + h_4 Q - (h_4 + p_4) \cdot (d_4 - n_4 \cdot Q) = -44$
gives $i = (3, 2, 1, 10, 9, 8, 4, 7, 5, 6)$.
- (8) $nsum = 22 < 30$ → a) $n_3 = 0$ and $n'_3 > 0$ gives $n_3 = 4$, $nsum = 26$.
b) $\Delta k_3(4) = c_3 + h_3 Q = 24.5$
gives $i = (2, 1, 10, 9, 8, 4, 7, 5, 6, 3)$.
- (9) $nsum = 26 < 30$ → a) $n_2 = 0$ and $n'_2 > 0$ gives $n_2 = 4$, $nsum = 30$.
b) $\Delta k_3(4) = c_2 + h_2 Q - (h_2 + p_2) \cdot (d_2 - n_2 \cdot Q) = -27$
gives $i = (2, 1, 10, 9, 8, 4, 7, 5, 6, 3)$.

Output:

$$n^{(1)} = (0, 4, 4, 1, 0, 1, 10, 3, 2, 4)$$

$$\text{with } k(n^{(1)}, d) = 894.$$

The application of OAP-II gives following results.

1. Preliminaries: From (4.3.6) we get
 $n^* = (5, 8, 4, 2, 1, 1, 10, 3, 2, 5)$.
2. Initialisation: $n_i := 0, r_i := n_i^*, i=1, \dots, M; nsum := 0$.
3. Sorting: $i = (1, 2, 10, 3, 9, 4, 8, 5, 7, 6)$.
4. Iteration:
 - (1) $nsum = 0 < 30 \rightarrow$
 - a) $n_1 = 5, r_1 = 0, nsum = 5$.
 - b) $i = (2, 10, 3, 9, 4, 8, 5, 7, 6)$.
 - (2) $nsum = 5 < 30 \rightarrow$
 - a) $n_2 = 8, r_2 = 0, nsum = 13$.
 - b) $i = (10, 3, 9, 4, 8, 5, 7, 6)$.
 - (3) $nsum = 13 < 30 \rightarrow$
 - a) $n_{10} = 5, r_{10} = 0, nsum = 18$.
 - b) $i = (3, 9, 4, 8, 5, 7, 6)$.
 - (4) $nsum = 18 < 30 \rightarrow$
 - a) $n_3 = 4, r_3 = 0, nsum = 22$.
 - b) $i = (9, 4, 8, 5, 7, 6)$.
 - (5) $nsum = 22 < 30 \rightarrow$
 - a) $n_9 = 2, r_9 = 0, nsum = 24$.
 - b) $i = (4, 8, 5, 7, 6)$.
 - (6) $nsum = 24 < 30 \rightarrow$
 - a) $n_4 = 2, r_4 = 0, nsum = 26$.
 - b) $i = (8, 5, 7, 6)$.
 - (7) $nsum = 26 < 50 \rightarrow$
 - a) $n_8 = 3, r_8 = 0, nsum = 29$.
 - b) $i = (5, 7, 6)$.
 - (8) $nsum = 29 < 30 \rightarrow$
 - a) $n_5 = 1, r_5 = 0, nsum = 30$.
 - b) $i = (7, 6)$.

5. Final step:

$$n = (5, 8, 4, 2, 1, 0, 0, 3, 2, 5).$$

$$r = (0, 0, 0, 0, 0, 1, 10, 0, 0, 0)$$

$$r^* = (5, 8, 4, 2, 1, 1, 10, 3, 2, 5)$$

$$n^{(II)} = (5, 8, 4, 2, 1, 0, 0, 3, 2, 5) \text{ and}$$

$$r^{(II)} = (0, 0, 0, 0, 0, 1, 10, 0, 0, 0)$$

Output : $n^{(II)}, r^{(II)}$ and $k(n^{(II)}, r^{(II)}, d) = 266$,

which is 29,75 % of $k(n^{(I)}, d)$.

The results for Example 4.3.2 show that the solutions of (OP-I) and (OP-II) in general are also very different like in the example (4.3.1). In this example the solution of (OP-I) which is 69,95 % of the solution (OP-I) in the example (4.3.1) and the solution of (OP-II) which is 67,34 % of the solution (OP-II) in the example (4.3.1) because the capacity is double in the example (4.3.2). Taken together, we can conclude that when the capacity of TU is increased, the total cost is reduced.

Chapter 5

Multiple-Period Models

5.1 Introduction.

In this chapter we will introduce a genetic algorithm (GA) approach for the fleet sizing allocation problems (FSAP). Since the multi-period, deterministic demand problem is NP-hard we suggest to use Genetic Algorithms to solve our problem HAS. This approach is suited to handle multiple and nonlinear objective functions as well as side constraints. Genetic algorithms are inspired by Darwin's theory of evolution. A genetic search uses the mechanics of natural selection and natural genetics to evolve a population into a near optimal solution. We present the developed genetic representation and use a randomized version to generate the initial random population. Goldberg (1989) stated that Genetic algorithm is a randomized search technique that is based on the natural selection process. The author added that the generations of new solutions at starting from an initial set of solutions could be obtained by applying genetic operators (crossover and mutation). Furthermore, the previous studies by Gen and Cheng, (1997), and Coley (1999) stated that the GAs have been successfully implemented to a wide range of combinatorial optimization problems.

A GA is heuristic, which means it estimates a solution, but genetic algorithms are different from other heuristic methods in several ways. The most important difference is that a GA works on a population of possible solutions, while other heuristic methods use a single solution in their iterations. Another difference is that the genetic algorithms are stochastic, not deterministic. In a few cases, a single period planning problem has been discussed by Federgruen and Zipkin (1984) and Chien *et al.* (1989). In contrast, there are several studies in the multi-period problem stated that the decisions are conducted for the specific number of planning periods, or reduced the problem to a single period problem by considering the effect of the long term decisions on the short term ones, for example Trudeau and Dror (1992), Viswanathan and Mathur (1997), and Herer and Levy (1997). Furthermore, Abdelmaguid and Dessouky (2006) investigated that the inventory distribution problem (IDP) considers multiple planning periods, both inventory and transportation costs as well as the situation in which backorders are allowed acceptable.

Thus, backorder decisions are accepted only when there is insufficient TU capacity to deliver to a customer or there is a transportation cost saving that is higher than the gained backorder cost by a customer.

This subchapter of GA is organized as follows. In section 5.2 we describe the formal problem definition of a basic model. Next we introduce the basic idea of genetic algorithm in section 5.3. Afterwards we introduce the Basic Description of Genetic algorithm in section 5.3.1. Thereafter, the GA representation is illustrated in section 5.3.2. Then, we illustrate the designs of the crossover and mutation operators in sections 5.4 and 5.5. Finally, we describe the GA implementation in section 5.6.

5.2 Description of a Basic Model

The FSAP for HAS systems was formulated in chapter 4 as a non-linear integer programming problem (NLIP). Since a non-linear integer programming problem in multi-period is very hard to solve we suggest to use Genetic Algorithms. We study a distribution system consisting of a single origin may be a warehouse, denoted 0, and M dispersed customers. Each customer i faces a different demand d_{it} per time period t , maintains its own inventory up to capacity C_i , and incurs inventory holding cost of h_i per period per unit and a backorder penalty of P_i per period per unit on the end of period inventory position. We assume that the warehouse has sufficient items that can cover all customers' demands throughout the planning horizon. The planning horizon considers T periods. The amount of delivery to customer i in period t , $q_{it} = n_{it} \cdot Q$, is to be decided. Where n_{it} refer to the number of TU's for a customer i in period t and Q the capacity of TU. We assume that K transportation units are available for each period t .

Let $k(n) = \sum_{t=1}^T \sum_{i=1}^M k_i(n_{it})$ denote the total cost for decision

$$n = \begin{pmatrix} n_{11} & n_{12} & \cdot & \cdot & \cdot & n_{1T} \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ n_{M1} & n_{M2} & \cdot & \cdot & \cdot & n_{MT} \end{pmatrix}$$

The costs for period t in location i under decision n_{it} are represented in the form

$$k_i(n_{it}) = c_{it} \cdot n_{it} + h_i (I_{it})^+ + P_i (I_{it})^-$$

where I_{it} is the inventory at end of period t . For given I_{i0} we can calculate I_{it} by the inventory balance equation

$$I_{it} = I_{i,t-1} + n_{it} \cdot Q - d_{it} \quad i = 1, \dots, M \text{ and } t = 1, \dots, T \quad (5.3.1)$$

The total planned delivery amounts for the customers in a given period are restricted by the total TU capacity $Q^{total} = K \cdot Q$.

Now we can formulate the follow nonlinear integer programming model, [NLIP] as follows:

$$\text{Min} \sum_{t=1}^T \sum_{i=1}^M k_i(n_{it}) \quad (5.3.2)$$

Subject to:

$$I_{it} = I_{i,t-1} + n_{it} \cdot Q - d_{it} \quad i = 1, \dots, M \text{ and } t = 1, \dots, T \quad (5.3.3)$$

$$\sum_{i=1}^M n_{it} \leq K \quad t = 1, \dots, T \quad (5.3.4)$$

$$n_{it} \geq 0 \quad \text{integer} \quad (5.3.5)$$

$$I_{it} \leq C_i \quad i = 1, \dots, M \text{ and } t = 1, \dots, T \quad (5.3.6)$$

The objective function (5.3.2) includes transportation costs and inventory holding and shortage costs on the end inventory positions. Constraints (5.3.3) are the inventory balance equations for the customers. Constraints (5.3.4) limit of the total TU's used. Constraints (5.3.5) number of TU's in period t are integer. Constraints (5.3.6) limit the inventory level of the customers to the corresponding storage capacity.

5.3 Genetic Algorithm

If we are solving a problem, we are usually looking for some solutions, which will be the best among others (optimal solution). The space of all feasible solutions (the set of solutions among which the desired solution resides) is called search space. Thus each point in the search space represents one feasible solution. Each feasible solution can be marked by its value or fitness for the problem. Looking for a solution of the problem is then equal to looking for some feasible solution with extreme value (minimum or maximum) in the search space. At times the search space may be well defined, but usually we know only a few points in the search space. The problem is that the search can be very complicated. We may not know where to look for a solution or where to start. There are many methods one can use for finding a suitable solution, but these methods do not necessarily provide the best solution. Some of these methods are hill climbing, tabu search, simulated annealing and the genetic algorithm. The solutions found by these methods are often considered as good solutions, because it is not often possible to prove what the optimum is. We concentrate on genetic algorithm. Next we consider its basic elements.

5.3.1 Basic Description of the Genetic Algorithm

The basic concepts of a genetic algorithm are individual, population, population size, fitness, elitism, parents, children and genetic operators. We will briefly describe each one of the elements of GA as follows: A single solution is called individual, while a set of individual forms a population. Number of individual in a population is called the population size. The capability of the individuals to solve the problem is quantified by their fitness, which represents the value of the performance of a solution. The elitism consists in individuating the best solutions of a population (elite) according to their fitness, and in letting them join the next population directly without any modification. The parents are individuals, which can generate a new individual of the next generation (children). The generation of new individual or childrens is realized by genetic operators.

The most important of genetic operators are selection, crossover and mutation. The selection operator choose parent individual for production of child individual. The selection is based on the fitness value of parents i.e. selection according to fitness. The

crossover is the method for combining those selected individuals into new individuals i.e. crossover to produce new individuals (offspring). Finally, the mutation simply adds some noise to an individual child. GA begins with a set of solutions called population. Solutions from one population are taken and used to form a new population. Solutions which are that selected to form new solutions are selected according to their fitness. Individuals with higher fitness have higher probability of generate offspring in next generation. This is repeated until some stopping condition is satisfied. There are many ways to describe general genetic algorithms.

One possibility is the description by some kind of pseudo-code as follows (see <http://www.cs.unibo.it/~babaoglu/courses/cas/resources/tutorials/ga/>) :

1. **[Start]** Generate random population of P individuals.
2. **[Fitness]** Evaluate the fitness function of each individual in the population.
3. **[New population]** Create a new population by repeating following steps until the new population is complete.
 - a. **[Selection]** Select two parents from the current population according to their fitness.
 - b. **[Crossover]** With crossover probability crossover the parents to form new individual (children).
 - c. **[Mutation]** With a mutation probability mutate new individual.
 - d. **[Accepting]** Place new individual in the new population.
4. **[Replace]** Replace the current population with the new population for a further run of the algorithm.
5. **[Test]** If the end condition is satisfied then, **stop**, and return the best solution found up to now else go to step **2**.

Each iteration of this process is called a generation. The entire set of generation is called a run. At the end of a run there are often one or more highly fit individuals in the population. Since randomness plays a large role in each run, two runs with different random number seeds with generally produce different detailed behaviors.

Another way is flow chart as in figure 5.1

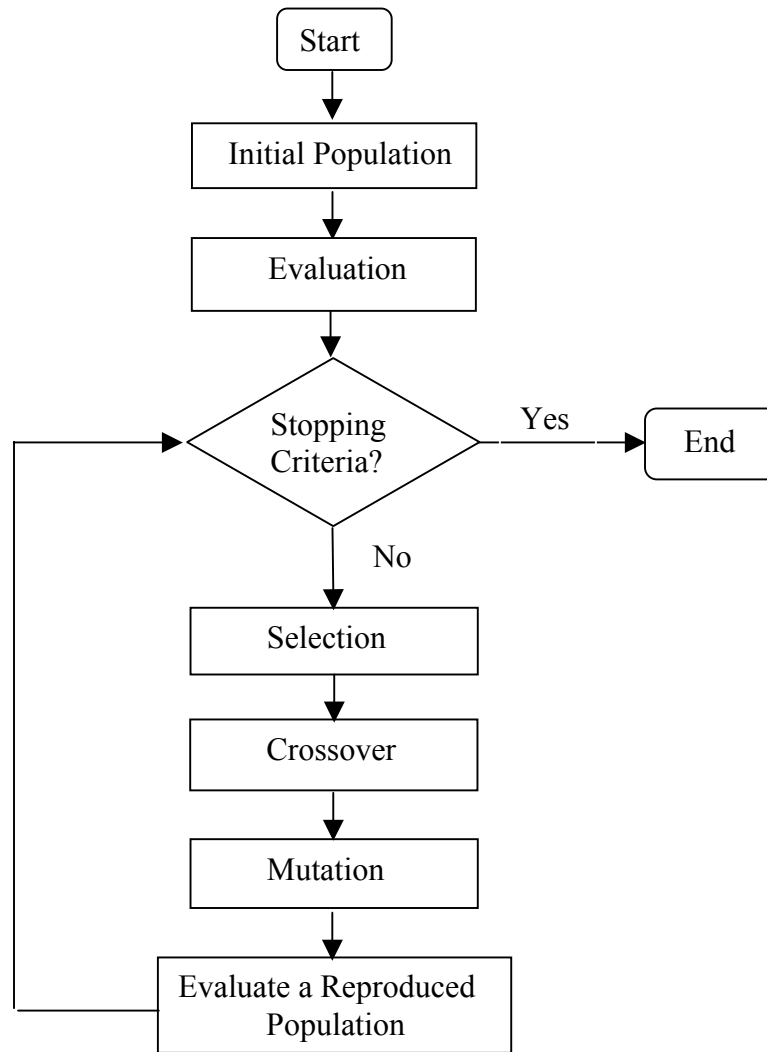


Figure 5.1: Flow chart of the genetic Algorithm

In all cases we have the following steps:

Initial step: Generate and evaluate an initial population.

Loop : WHILE a stopping criteria is not fulfilled generate a new population and evaluate the fitness of its solutions.

Output : Return the best solutions and their fitness.

By various parameters and different setting for the initial step and the loop we can design a general variety of genetic algorithms. The first question to ask is how to create individuals and what type of encoding to choose. We then address Crossover and Mutation, the two basic operators of GA. The next question is how to select parents for crossover. This can be done in many ways, but the main idea is to select the better parents in the hope that the better parents will produce better offspring.

5.3.2 Genetic Representation

Genetic algorithm is an iterative procedure that consists of a constant-size population of individuals. As refer to figure 5.1 the standard genetic algorithm proceeds as follows: First of all, we have to choose the population size P . If the population size is too small, the genetic algorithm will converge too quickly to find the optimal solution but if the population size is too large, the computation cost may be prohibitive. Afterwards, the initial population of individuals is generated randomly or heuristically. Randomly, covering the entire range of possible solutions. Heuristically, which means it estimates a solution, but GA works on a population of possible solutions. In every evaluation step the individuals in the current population are evaluated according to some predefined quality criterion, referred to as the fitness, or fitness function. However, it is important to distinguish between the evaluation function and the fitness function. While evaluation functions provide a measure of an individual's performance, fitness functions provide a measure of an individual's reproduction opportunities.

To form a new population, individuals are selected according to their fitness. Selection plays an important role in driving the search towards better individuals and in maintaining a high genotypic diversity in the population. Selection alone cannot introduce any new individuals into the population. These are generated by genetically inspired operators, of which the most well known are crossover and mutation. Crossover is performed with some probability between two selected individuals, called parents, by exchanging parts of their individuals to form two new individuals, called offspring. The mutation operator is introduced to prevent premature convergence to local optima by randomly sampling new points in the search space. The new population generated with these operators replaces the old population. Stopping criteria, there are many ways to terminate the run of GA. The termination condition may be maximal number of generations or the attainment of an acceptable fitness level. A critical point when applying GAs to an optimization problem is to find a suitable solution that transforms feasible solutions into representations amenable to a GA search.

One critical point of a GA is the genetic representation of individuals (see Fleming and Zalzala (1997)). In the classical form of a GA is representation as a bit-string. We use here values of the decision variables. More exactly, we presented the delivery schedule in the

form of a two-dimensional matrix n in which each cell contains the number n_{it} of TU's to be delivered to customer i in a given period t . Furthermore, each row in the matrix represents a specific customer but the columns represent the planning periods from 1 to T . In addition, we considered the delivery amounts as integers in order to simplify the genetic search process. This condition which is assumed without loss of generality in this study does not prevent the GA from reaching the optimal solutions as well as the customer demand values and the customer storage and TU capacities to be integers. The proposed GA satisfies the necessary conditions for successful GA implementation by minimizing the complete expressive. Consistently, our nonlinear programming formulation as well as the delivery number of TU's are the key decision variables whose values can be easily used to determine other interesting variable. Therefore, the sole use of the delivery number of TU's in the representation suffices satisfies the condition of being minimal. Meanwhile, the representation is capable of representing every possible solution in the search space including the optimal ones.

Example 5.1

This example demonstrate the proposed of GA representation, we assume that $M = 4$, $Q = 20$ units, 10 TU's are available to serve the customers in every period for a 4-period planning horizon, storage capacity for each customer is 50 units, and inventory holding and shortage costs given in table (5.1). At the beginning of the planning period, all customers have zero inventory positions. The demand requirements for every period in the planning horizon are given in table (5.2).

Costs	Customer			
	1	2	3	4
Unit holding cost (€/unit/period)	0.07	0.07	0.11	0.09
Unit shortage cost (€/unit/period)	2.5	3.1	2.2	2.8

Table 5.1: Cost Information for Example 5.1

	Period (t)			
	1	2	3	4
Demand (d_{1t})	10	20	15	25
Demand (d_{2t})	30	20	10	35
Demand (d_{3t})	25	35	20	30
Demand (d_{4t})	45	20	30	25

Table 5.2: Distribution Demand Requirements for Example 5.1

		Period			
		1	2	3	4
Customer	1	2	0	0	2
	2	0	3	0	3
	3	3	0	2	1
	4	3	0	3	0

(a) Genetic Representation of a Sample Solution

		Period			
		1	2	3	4
Customer	1	30	10	-5	10
	2	-30	10	0	25
	3	35	0	20	10
	4	15	-5	25	0

(b) Resultant end of Period Inventory Positions

Table 5.3: Genetic Representation and Solution Interpretation for the Sample Solution

The genetic representation of this example takes the form of a two dimensional matrix with four rows and four columns. Each cell in the matrix defines the scheduled number of TU's for the corresponding customer (given in the row) and the corresponding period (given in the column). Table (5.3) describes the sample solution for this sample problem. Based on the scheduled number of TU's, the inventory position variables I_{it} of each customer can be easily determined from equation (5.3.1) as shown in Table ((5.3) b).

5.4 Crossover Operation

Crossover operators are an essential part of GAs as they help in inheriting better characteristics from the fittest solutions among generations. Crossover is a genetic operator that combines two individuals (parents) to produce a new individual (offspring). The idea behind crossover is that the new individual may be better than both of the parents if it takes the best characteristics from each of the parents, i.e. the purpose of the crossover operation is to create new individuals having greater performance than their parents. Crossover occurs during evolution according to a user-definable crossover probability.

5.4.1 Crossover Mechanisms

There are many different kinds of crossover operator, but the general idea of all of them is to exchange genetic items between two strings. The power of the GA is mostly due to crossover. It is the most important operator to the GA. Diversity is indispensable to evolution. The population's diversity is obtained and maintained by crossover, which allows the GA to find better solutions in the search space. Now, shows an example of how this could happen in the case of the sample problem presented earlier.

Assume that two matrices $n_1 = (n_{it}^1)$ and $n_2 = (n_{it}^2)$ are selected as parents for crossover operation.

		Period			
		1	2	3	4
Customer	1	3	3	0	0
	2	1	1	2	0
	3	2	2	4	0
	4	3	0	3	0
Remaining vehicles no.		1	4	1	10

Parent n_1

		Period			
		1	2	3	4
Customer	1	2	4	0	0
	2	2	0	0	0
	3	3	3	0	2
	4	2	3	0	2
Remaining vehicles no.		1	0	10	6

Parent n_2

5.4.2 Designed Crossover Operator

The crossover is performed in three steps (see Gen and Cheng 1997):

1. Create two temporary matrices $n = (n_{it})$ and $r = (r_{it})$ as follows:

$$n_{it} = \lfloor (n_{it}^1 + n_{it}^2) / 2 \rfloor \quad \text{and} \quad r_{it} = (n_{it}^1 + n_{it}^2) \bmod 2$$

Matrix n keeps rounded average values from both parents, and matrix r keeps track of whether any rounding is necessary.

2. Divide matrix r into two matrices $r^1 = (r_{it}^1)$ and $r^2 = (r_{it}^2)$ such that

$$r = r^1 + r^2$$

It is easy to see that there are too many possible ways to divide r into r^1 and r^2 while satisfying above condition.

3. Then we produce two offspring of n_1' and n_2' as follows:

$$n_1' = n + r^1 \quad \text{and} \quad n_2' = n + r^2$$

		Period			
		1	2	3	4
Customer	1	2	3	0	0
	2	1	0	1	0
	3	2	2	2	1
	4	2	1	1	1

Matrix n

		Period			
		1	2	3	4
Customer	1	1	1	0	0
	2	1	1	0	0
	3	1	1	0	0
	4	1	1	1	0

Matrix r

		Period			
		1	2	3	4
Customer	1	1	0	0	0
	2	0	1	0	0
	3	1	0	0	0
	4	1	0	1	0

Matrix r^1

		Period			
		1	2	3	4
Customer	1	0	1	0	0
	2	1	0	0	0
	3	0	1	0	0
	4	0	1	0	0

Matrix r^2

		Period			
		1	2	3	4
Customer	1	3	3	0	0
	2	1	1	1	0
	3	3	2	2	1
	4	3	1	2	1
Remaining vehicles number		0	3	5	8

Offspring n_1

		Period			
		1	2	3	4
Customer	1	2	4	0	0
	2	2	0	1	0
	3	2	3	2	1
	4	2	2	1	1
Remaining vehicles number		2	1	6	8

Offspring n_2

5.5 Mutation Operation

After selection and crossover, we now have a new population full of individuals. Some are directly copied, and others are produced by crossover. In order to ensure that the individuals are not all exactly the same, we allow for a small chance of mutation. Additionally, the designed crossover operator is not sufficient to investigate such solution alternatives. Therefore, the mutation operator is specially designed to investigate it. Mutation is a genetic operator used to maintain genetic diversity from one generation of the population of individuals to the next. Mutation operators are applied to each child solution resulting from the crossover operation. They help the GA to reach further solutions in the search space. The idea of the mutation operation is to randomly mutate the individual solution and hence produce a new solution that is not very far from the original one.

5.5.1 Principle of Mutation:

Having in mind the principle that each individual solution should be mutated with a small probability we consider the following four variants for mutation in our problem (5.3.2)-(5.3.6).

Variant I:

Let n be a given solution. Each solution $n = (n_{it}), i=1, \dots, M; t=1, \dots, T$ will be mutated as follows. For each period $t \in T$ each customer $i \in M$ will be mutated with probability P_{Mut} .

The mutation runs as follows:

Let $RV(M)$ denote a random variable, which takes values -1 and $+1$ with equal probability 0.5. i.e.

$$RV(M) = \begin{pmatrix} -1 & +1 \\ 1/2 & 1/2 \end{pmatrix}$$

Then we set:

$$n_{it} = [n_{it} + RV(M)]^+ \quad \text{if } RV(M) = -1$$

and

$$n_{it} = \min(n_{it} + RV(M); K_{Rem}) \quad \text{if } RV(M) = +1$$

Where K_{Rem} refer to remaining vehicles number. If we look at the matrix size of the solution n , it follows, that, if we mutate every element of the matrix n with probability P_{Mut} , the probability that the solution n is mutated is growing fast to 1 with increasing matrix dimensions. But behind mutation stays the idea that a single individual will be mutated with small probability. That is why the probability P_{Mut} must be very small. However, for variant I holds, for example, if $P_{Mut} = 0.005$, $M=15$ and $T=15$, the probability, that solution n will be mutated is 0.676. This is too high and we expect that variant I will show a bad performance. Therefore another variant must be used in this case.

Variant II:

We look at each solution n in the whole population and decide with probability $P_{Mut'}$ to mutate it. If the answer is “yes” we mutate it as in variant I. Note, that if $P_{Mut'} = 1$ – that means we select every solution for mutation – variant II is same as variant I. Here $P_{Mut'}$ can be equal 0.1, 0.01, 0.005 i.e. as for usual GA.

Variant III:

Let us have N_p individual solutions in the population. Then we choose one solution out of the population and mutate it. Each solution is equally likely to be selected for mutation. That means the probability to mutate a solution $P_{Mut'} = 1 / N_p$. The mutation is done as in variant I.

Let $\text{Random}(N_p)$ be a function that generates a random integer number in range $[1; N_p]$, i.e. $\text{Random}(N_p) \rightarrow [1; N_p]$ and let the probability for each number to be selected is $1/N_p$, i.e. numbers are equally distributed. Then we select one individual to mutate it as follows:
Individual nr. := $\text{random}(N_p)$

Variant IV:

We consider each individual solution and choose it for mutation as in variant II with given probability $P_{Mut'}$. The mutation is performed as follows:

For each $t \in T$ we randomly choose one customer $i \in M$ and mutate him by the following equation:

$$n_{it} = \begin{cases} \max(n_{it} + RV(M); 0) & \text{if } RV(M) = -1 \\ \min(n_{it} + RV(M); K_{Rem}) & \text{if } RV(M) = +1 \end{cases}$$

That mean for each period we move one vehicle from a customer to the remaining/unused vehicles or vice versa to obtain the optimal solution i.e. to obtain the minimum cost.

5.6 Genetic algorithm implementation

In our GA implementation, we use a simple GA search structure with elite preservation. The algorithm starts by generating the initial population using the randomized version. The size of this population remains constant throughout the application of the algorithm. Then the improvement phase of the GA follows by applying the designed crossover and mutation operators for a randomly selected pair of solutions from the current population. To move from the current population to a new one, the selection process followed by the crossover and the mutation operations is repeated a number of times equal half the population size. The creation of a new population is repeated a number of times called the number of generations. In order not to lose the best solutions found throughout the generations due to the randomized selection mechanism, a set of the best solutions found are reserved in what is referred to as the elite set. This elite set has a fixed size and used to feed the starting population of solutions in every generation. In our experimentation, we used the following parameters. Number of generations: 1000, population size: 100, crossover probability: 0.5, and mutation probability: 0.01 and 0.005.

Now we will apply all variants to tested two examples, first for 4 customers 4 periods (see example 5.1) and second for 15 customers 15 periods (see example 5.2), with mutation probability 0.01 and 0.005. The results data of the GA program see Appendices C and D.

Example 5.2

In this example we consider another demonstrate example for the GA representation. We assume that $M = 15$, $Q = 20$ units, 25 TU's are available to serve the customers in every period for a 15 period planning horizon, storage capacity for each customer is 50 units, and inventory holding and shortage costs given in table (5.4). At the beginning of the planning period, all customers have zero inventory positions. The demand requirements for every period in the planning horizon are given in table (5.5). From the

results data of the GA program we got the optimal solution n^* for each variants and table (5.6) show for example the optimal solution n_i^* for variant I with $P(\text{Mut}) = 0.01$.

Cost	Customer														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Holding cost (€/U/P)	0.07	0.06	0.11	0.09	0.07	0.08	0.12	0.09	0.11	0.07	0.06	0.09	0.13	0.10	0.07
Shortage cost (€/U/P)	2.5	3.1	2.2	2.8	2.5	2.3	2.1	1.5	2.6	2.5	3.2	2.9	2.2	2.3	2.5

Table 5.4 Cost Information for example 5.2

Period (t)	d_{1t}	d_{2t}	d_{3t}	d_{4t}	d_{5t}	d_{6t}	d_{7t}	d_{8t}	d_{9t}	d_{10t}	d_{11t}	d_{12t}	d_{13t}	d_{14t}	d_{15t}
1	10	30	35	25	30	35	50	35	30	45	30	35	45	30	50
2	30	40	10	30	40	15	45	35	25	20	25	20	40	50	30
3	25	35	30	40	35	50	00	40	35	35	30	40	20	20	35
4	40	30	20	20	45	00	40	30	20	25	50	20	20	30	20
5	15	30	40	40	50	40	20	40	30	30	20	10	45	15	40
6	25	20	45	35	25	10	30	20	45	45	40	00	50	45	30
7	45	10	35	45	10	30	25	50	00	00	30	35	45	40	55
8	50	15	40	30	30	15	35	20	55	50	20	25	10	20	35
9	30	35	30	20	50	45	10	50	30	30	15	10	35	05	10
10	30	45	40	35	20	30	00	25	40	15	00	45	20	45	00
11	50	10	20	00	10	10	50	25	20	10	15	00	45	10	30
12	25	50	10	30	00	30	30	30	00	00	35	35	00	35	25
13	10	30	35	50	30	30	30	10	50	30	00	20	40	30	15
14	40	25	30	20	25	20	40	00	10	35	40	30	20	50	35
15	10	20	10	10	30	10	10	20	30	25	10	35	15	00	40

Table 5.5 Distribution Demand Requirements for example 5.2

Customer	Period														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	1	0	1	2	3	2	1	4	1	3	1	2	1
2	0	2	3	0	1	1	2	1	0	1	3	2	1	1	3
3	1	0	1	2	1	3	2	2	1	0	3	1	2	1	1
4	1	1	1	1	1	2	2	3	2	1	1	1	2	1	1
5	1	1	3	2	1	1	2	1	0	4	0	0	0	0	5
6	0	1	2	1	1	1	1	0	3	3	0	2	0	2	1
7	0	2	1	1	3	0	2	0	0	2	4	0	3	1	1
8	2	1	1	2	3	0	0	4	0	2	3	1	1	0	1
9	0	2	2	1	0	4	0	2	2	1	0	2	3	0	2
10	1	2	2	1	1	1	0	3	1	2	0	1	1	2	1
11	0	2	1	0	0	4	2	1	3	0	0	1	0	3	1
12	0	1	1	3	1	0	1	1	1	2	1	0	2	2	2
13	2	2	1	1	2	2	0	3	1	0	4	0	2	1	0
14	1	0	3	1	1	3	1	1	0	0	2	3	1	2	2
15	2	2	0	2	0	1	2	1	5	0	0	1	2	1	3
Remaining Vehicles number	14	6	2	7	8	0	5	0	5	3	3	7	4	6	0

Table 5.6 Optimal solution n_i^* for example 2 with $P(\text{Mut}) = 0.01$

From the results data of the GA program for illustrative examples (see Appendices C and D) we can conclude:

The mutation probability must be low, to introduce some difference into the solution, because the designed crossover operator is actually some kind of averaging operation. If the two crossover solutions are quite equal, there is little effect. For example in variant IV, the mutation operation modifies only a little bit of the solution. That means little difference is introduced into the population. The variant III also only changes one solution of the population, that seems to be not enough, although the newly generated solution is potentially more different to the previous solution, as all elements n_{it} are possibly to be mutated.

The variants I and II achieve lot better results than variant III and IV, as more difference is introduced into the population. This results in better sampling of the solution space, so more possibly good solutions can be discovered. If a solution gets worse after the mutation operation, it is likely to be discarded in the next operation of the algorithm, as good solutions have a higher chance to survive the next iterations. The mutation probability

should not be too high, as, if we are already near the optimum; we have to advance toward it in smaller steps. With big steps we could advance too fast, and “jump over” the optimum.

From the result data of the GA program for illustrative example with 4 customers and 4 periods (see Appendix C) we can see that the data for variant I with $P_{Mut} = 0.01$ hit at about 30 iterations a minimum, but then the solution is degenerated by too many mutations, resulting that we go away from the minimum. On the other hand, a lower probability P_{Mut} results in slower advancing towards the optimum. For variant I, with $P_{Mut} = 0.01$ it took about 30 iterations to come near to the optimum, with $P_{Mut} = 0.005$, it took 10 iterations. But from the result data of the GA program for illustrative example with 15 customers and 15 periods (see Appendix D) we can see that the data for variant I with $P_{Mut} = 0.01$ it took about 500 iterations to come near to the optimum, with $P_{Mut} = 0.005$, it took about 350 iterations. The difference between variant I and II is, that in variant II not every solution is subject to mutation. In the test runs variant I was compared to variant II. The results show, that variant II takes less iterations, than variant I, and the speed to reach near the optimum seems to be faster. A possible explanation is, that if not all solutions are subject to mutation, the good ones are not modified, and better solutions are found via crossover instead of mutation.

In general, we can conclude:

- If we mutate, we should generate a solution that is not so similar to the previous one, especially in the beginning of the algorithm. This is to explore the solution space. If the new solution is not good, it will be discarded in the next algorithm step. If we mutate not enough, we can get stuck at a local optimum, as the crossover operation works same averaging.
- If we mutate too many solutions, there is a higher chance, that already good solutions are mutated and possibly degenerated. If we mutate too few solutions, there is not enough difference introduced in the population, for the crossover operation to work efficiently.

So variant II seems to be the best, as it does include variant I with $P_{Mut} = 1$ and variant III with $P_{Mut} = 1/N_p$. Variant II is better than variant IV, because the mutation operation generates a solution that is more different than the original one. This seems to be good in conjunction with the “averaging” crossover-operation.

5.7 General HAS Models - a Simulation Optimization Approach

We briefly discuss the principle of simulation optimization. For more realistic models and solutions the broadest applicable approach is simulation. However, simulation is not an optimization tool. But we can combine simulation and optimisation and apply the simulation optimization approach. Fu (1994) gives a good overview on simulation optimisation. The first paper, which applies that approach to some logistic problems seem to be Köchel et al. (2003). We describe also a simulator for hub-and-spoke systems and its combination with an optimization tool. Some examples show the applicability of our simulation optimization approach (see, El-Ashry et al. (2006)).

5.7.1 Simulation Optimization

Solving an optimization problem (OP) by analytical approaches the underlying system has to be reduced to an idealized model. However real life distribution network are highly complex. Thus an idealization will reduce the correctness of the results of the optimization process. Another way to optimize a given system is to simulate the behavior of the investigated system under different configurations. By assessing the behavior of the system we can get informations about it's goodness under the given configuration. New configurations can be produced, which may be better or lead to a better configuration.

This approach is called the simulation optimization approach. The main improvement by using the simulation optimization approach is, there is no need to simplify the real problem to an idealized model. Actually it is possible to rebuild most of the real life system by simulation software. Furthermore simulation opens new chances in gaining system relevant data, since all implemented system details are traceable. Another improvement of simulation is that it is possible to deal with huge and even heterogeneous models. But depending on the complexity such models will not be solvable by deterministic optimization algorithms. We suggest applying simulation optimization, where a simulator is coupled with an optimizer. For the optimizer we will investigate some heuristic based algorithms like genetic algorithm, tabu search and combinations of optimization algorithms.

In the following we will investigate the application of simulation optimization to more general HAS models.

5.7.2 Fundamentals

The simulation optimization approach can be divided into two parts, the simulation and the optimization. While simulation is used to assess a given solution, optimization will provide new solutions. To solve a given OP we use the optimization cycle shown in figure (5.2). By using knowledge about considered solutions an optimization algorithm creates a new solutions for the decision parameters of the model, which has to be solved. Next the simulator is configured by the given data for the current solution(s) and he is started. After stopping the simulation the collected statistical data are assessed to gain information about the goodness of the considered solution(s). The optimization software then will add the new solution to the knowledge base and compare it to the other solutions. To stop the optimization cycle a stopping condition is assigned. Such conditions may be a given number of solutions, an improvement rate etc.

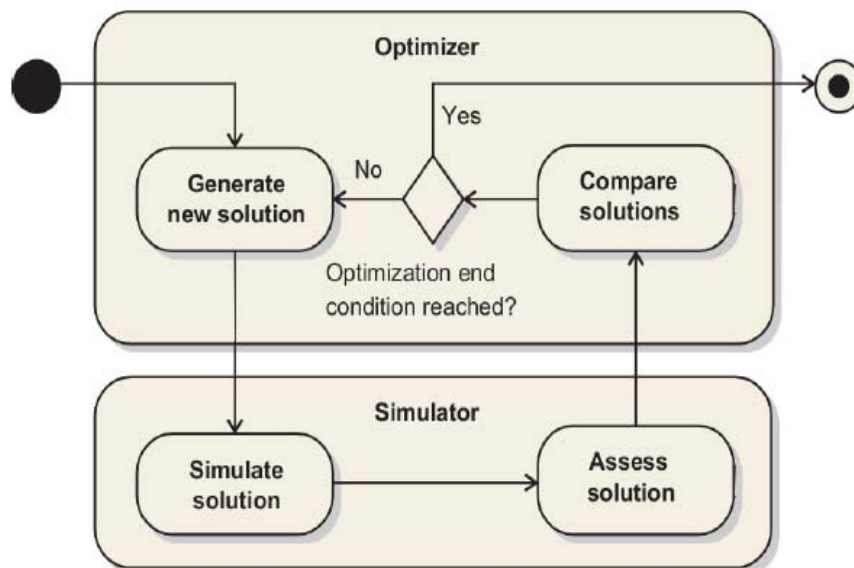


Figure 5.2: Scheme of Simulation Optimization

5.7.3 Calculation Assessment Optimization System (CAOS)

To realize the simulation optimization approach we will use “Calculation Assessment Optimization System” (CAOS), a software production of the professorships “Modeling and Simulation” at the Chemnitz University of Technology (see Kämpf 2004 & 2005).

The idea of CAOS is to provide a software system, which separates the optimization process from the optimization problem. This means, it is only necessary to have information about the problem but not about the optimizing algorithm(s). To solve an optimization problem the user of CAOS has to build up a model of the system to which the problem is related. Furthermore he has to define the decision parameters and their domain. To solve the given optimization problem an optimizer sets up the model with some special values for the decision parameters, and the simulation with these settings is started. As the result of the simulation we get at least one assessment value, on which the optimization decisions are based. CAOS also allows multi criterion optimization; this means it can also optimize models, which provide more than one assessment value. All assessment values are based on simulation data. The whole environment of CAOS consists of:

- Optimizer prototypes and some implementations, e.g., tabu search, genetic algorithms, etc., and also combinations of different optimization algorithms,
- An analysis tool for tracing the optimization process,
- Model declaration language for building up general models,
- Prototypes of simulators.

The CAOS software is extensible by implementing new optimization algorithms and by defining types of optimization problems. Important is another feature, that any model is defined by its general structure. CAOS provides a construction kit for any kind of system with the same structure.

In case of the HAS structure this means that only the model elements hub and spoke must be declared. The number of spokes, the properties of the instances of model elements can be configured without reengineering the hub and spoke model. Once built up a model of a system this design of CAOS brings the advantage of short developing time, when creating a special realization of a system.

In the following we describe the structure and behavior of the general HAS model

5.7.4 HAS Simulator

As written above we build up a construction kit for HAS structured simulators. At present it is now possible to create HAS models with:

- a single hub,
- a nearly infinite number of homogeneous or heterogeneous spokes,

To realize various structures the simulation software contains the following model elements.

5.7.4.1 Model Elements

Product Currently there is only one product to be distributed. (Different products will differ in index i . Here we neglect this index because there is only one product.) All products are measured in item units a .

Inventory The amount of any inventory is measured in item units. While the capacity of storage in a spoke is assumed to be finite the hub has an infinite amount of the product to be distributed. For ordering reasons the inventory of a spoke is analyzed by the inventory position r^{future} , which is the inventory level r^{now} plus deliveries from open orders o minus unsatisfied demand d is the inventory repository.

$$r^{\text{future}} = r^{\text{now}} + o - d$$

Distance matrix Distances between all the locations (hub and spokes) are measured in x distance units.

Transportation units It is possible to build a model with a number of TU-classes. The classes differ with respect to the following adjustable values:

- Transportation capacity Q (measured in item a),
- Speed v ,
- Transportation costs per product unit and distance unit $c_{a,x}^{\text{full}}$,
- Transportation costs per empty capacity unit and distance unit $c_{a,x}^{\text{penalty}}$
($c_{a,x}^{\text{penalty}}$ maybe less, equal or higher than $c_{a,x}^{\text{full}}$),
- Fixed costs for loading and unloading c^{loading} and finally
- Rental costs for leasing a TU c^{rental} .

Hub (central warehouse) Each model, which can be created, has a single hub. Furthermore the hub has a TU pool with different TU classes. The number of TU's in each class is a decision variable and can be optimized. In case an order of a retailer can not be satisfied, because of there is no TU in the TU pool, the order will be back ordered in a waiting queue. For this queue exist different serving policies (see 5.7.4.2, note that EDF is not suitable for this queue since there exist no waiting times).

Spoke (retailers) A model of a HAS structure can have a nearly infinite² number M of spokes but minimum one spoke is required.

Each spoke can have different, adjustable values for the:

- Customers arriving process with rate λ_k (deterministic or stochastic)
- Customers demand d_k (deterministic or stochastic)
- Customers waiting time $t_k^{waiting}$ (deterministic or stochastic)
- Customers serving policy
- Initial inventory level r
- Ordering policy (see 5.7.4.2)

Furthermore a customer may wait if his demand cannot be satisfied at once, for waiting customers exists a waiting queue.

Initially each spoke has a given amount of the product (maybe 0) in it's inventory. The spoke will release an order to the hub in accordance with its ordering policy. For the case a waiting customer leaves the queue because of he has reached his waiting time limit we assume a loss for not satisfied demand $c_{k,a}^{shortage}$. Finally, there are holding costs $c_{k,a}^{holding}$ for the product in the storage.

5.7.4.2 Control Policies

To control the product through the HAS distribution network, we use some ordering and service policies.

The order policies are used by the spokes to refill their inventories. Following policy types are implemented:

(s,S) If the level of the observed inventory r is lower than bound s , an order up to level S is released.

² As much as the given hardware resources allow.

(s,nQ) If the level of the observed inventory r is lower than bound s , an order of n times the amount of Q , the lot size, is released.

Service policies are used by the hub and the spokes to satisfy waiting customers. In case of the hub the customers are the spokes. The hub does not provide the policy EDF, while the spokes do not have waiting times for their orders.

FIFO/FCFS (First In First Out / First Come First Serve) The waiting elements of the queue will be served in order they entered the queue.

LIFO/LCFS (Last In First Out / Last Come First Serve) The waiting elements of the queue will be served in reverse order they entered the queue.

EDF (Earliest Deadline First) The waiting elements of the queue will be served by their remaining waiting time, that means the element with the lowest waiting time will be served first (even if it's below zero!). Elements with the same left waiting time are served by FIFO.

SAN (Smallest Amount Next) The waiting elements of the queue will be served by the amount of their demand. The elements with the smallest amount will be served first.

BAN (Biggest Amount Next) The waiting elements of the queue will be served by the amount of their demand. The elements with the biggest amount will be served first.

Random The waiting elements of the queue will be served in random order.

5.7.4.3 Events - Activity Diagrams

The sequence of events describes the evolution of the system in time. The event types, which are relevant for the considered hub-and-spoke system, are the arrival of a client with defined demand (new demand), the arrival of a TU with product at a spoke (TU delivery), and the return of a TU to the hub (TU return). In the activity diagram below we show, which activities are related with various events.

New demand This event causes all the dynamics in the hub and spoke structure. If it is occurred it will involve complex reactions and may initiate new events of the event types TU delivery and TU return.

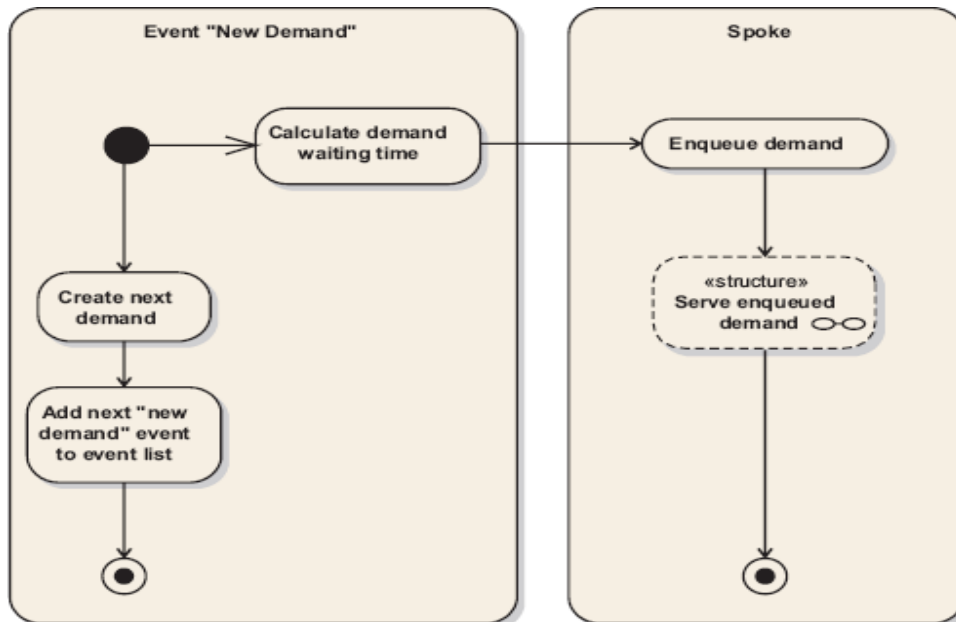


Figure 5.3: Activity Diagram: Event “New Demand”

After simulating demand and waiting time (these may be stochastic) the demand order is queued in the customers waiting list of the current spoke. To satisfy this demand the activities shown in figure 5.4 have to be executed. The event “new demand” will always generate the time moment for the following event “new demand”. Thus the time evolvement of the model is kept running.

Figure 5.4 describes how the demand of any customer can be satisfied (if possible) and also what will happen if it cannot be satisfied. Necessary an order of products has to be placed. To deal with this the activities of figure 5.5 must be realized.

Looking closer at the diagram not only tasks for running the simulation are executed, but also gain and cost are added to a cost manager. The cost manager will provide the assessment of the current HAS configuration. We will consider this part later.

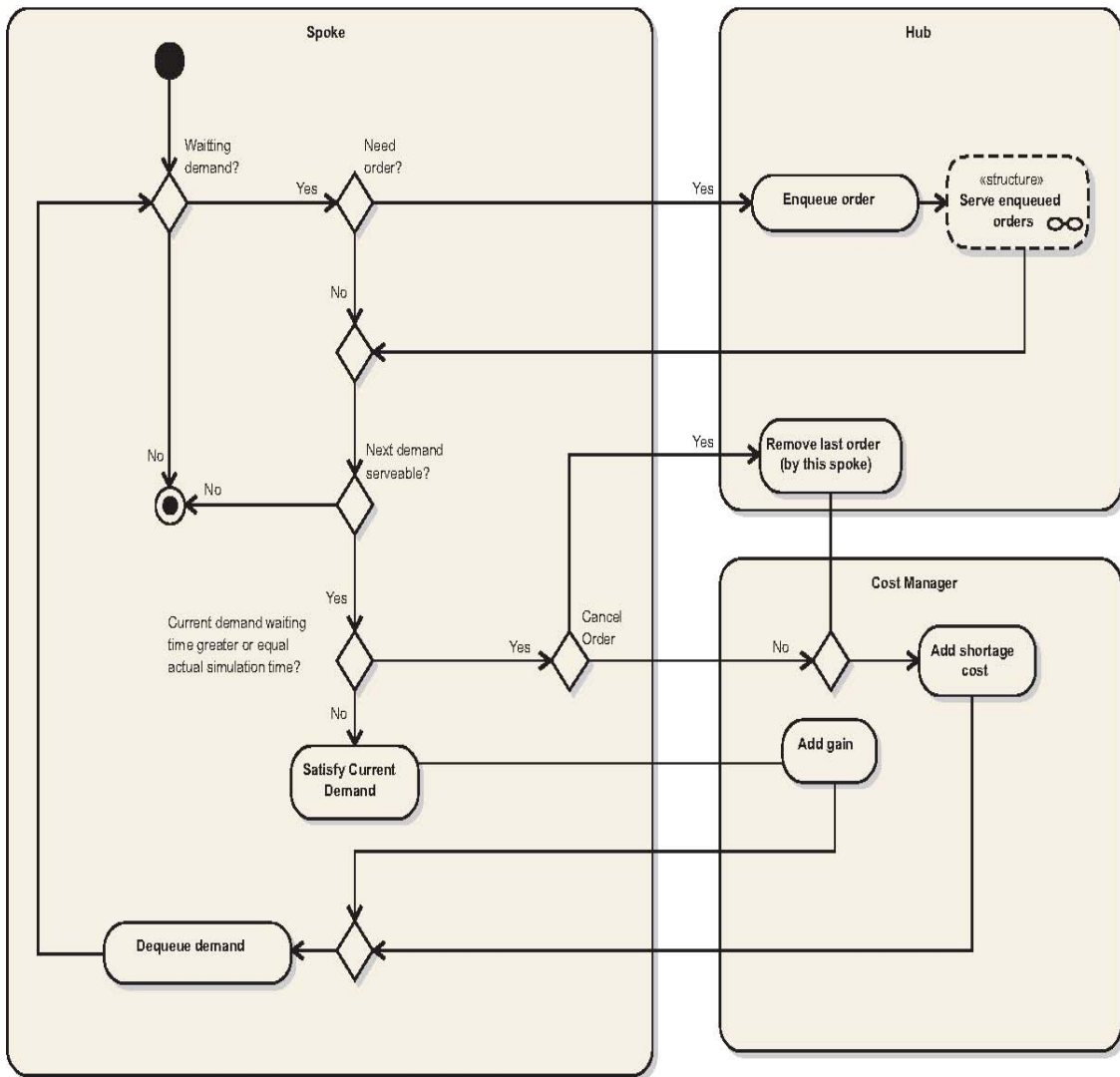


Figure 5.4: Activity Diagram: Serve Enqueued Demand

When an order has been placed in the order queue of the spoke, transshipment decisions are necessarily to prove and execute (if possible). As shown in figure 5.5 the transshipment is realized by the event types TU delivery and TU return. We will continue with these event types.

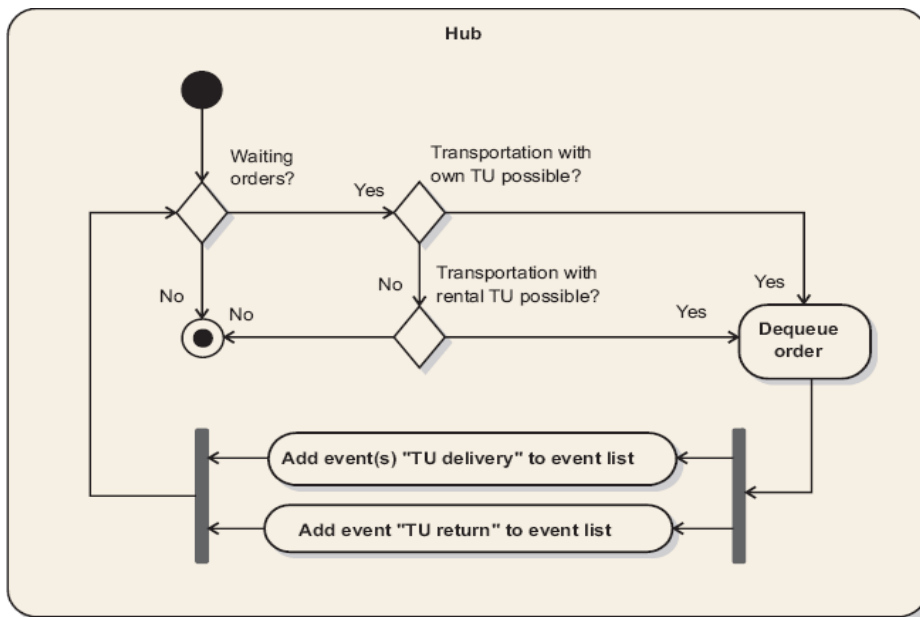


Figure 5.5: Activity Diagram: Serve Enqueued Orders

TU delivery This event will occur if a TU arrives a spoke. It generates the activities shown in figure 5.6. Using one of the available service policies the hub selects the spoke fitting to this policy. According to the order(s) of the spoke the repository will be filled. If the spoke has more one open order and the amount of the first order would not use the whole transportation capacity of a TU, then this capacity will be used for further orders of the same spoke because of after the arrival of new product waiting customers can be served. To realize this the activities in figure 5.4 will be executed.

Delivering products generates costs, which are added to the cost manager. The costs for loading and unloading are fixed costs, whereas the costs for transportation are proportional to the transported amount of products and the transport distance. The distance matrix gives the transport distance.

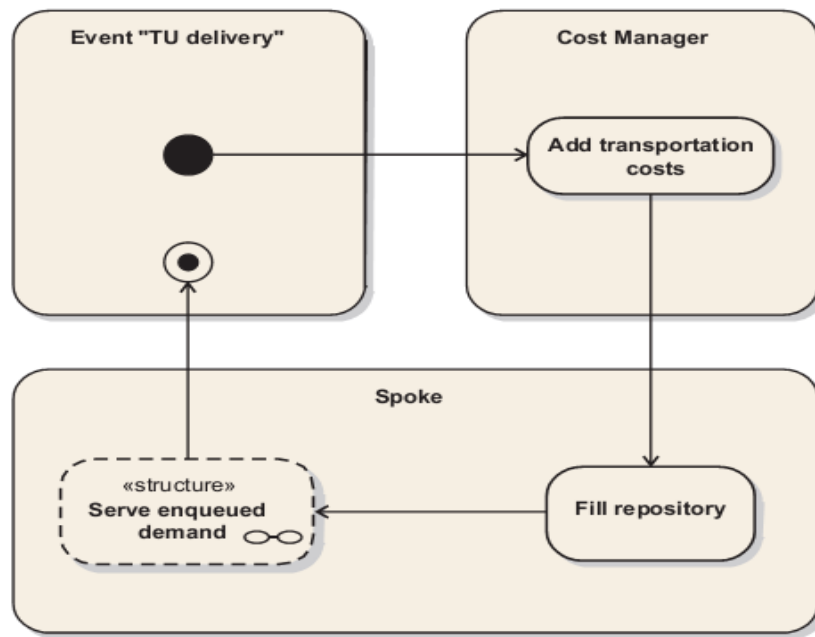


Figure 5.6: Activity Diagram: Event “TU Delivery”

Refilling the inventory implies that customers may be served now. Therefore the activities shown in activity diagram 5.4 have to be executed.

TU return The “TU return” event describes what has to be done, when a TU returns to the hub (see figure 5.7). Obviously this includes the reactivation of the hub, so that transshipment decisions can be placed.

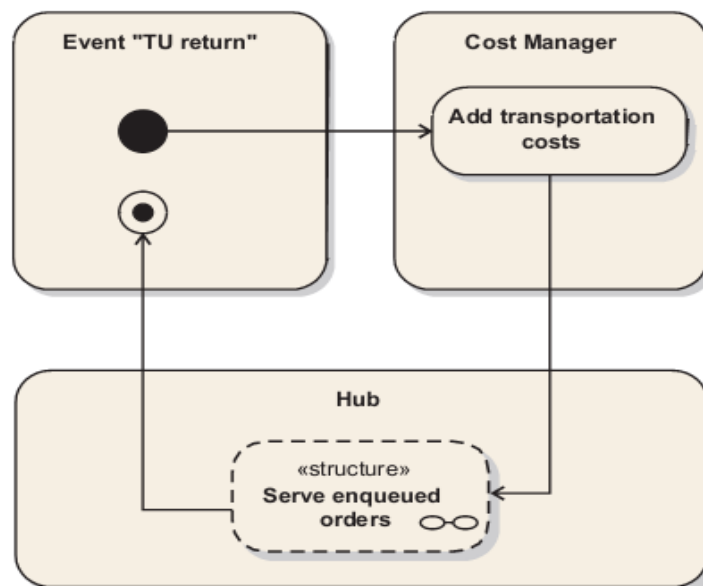


Figure 5.7: Event “TU Return”

5.7.4.4 Assessment

As shown before there are several cost, which are added to a cost manager. Additionally to the up to now costs there are some other costs, which are collected over the simulation time. First of all there are holding costs $cost_{holding}$, which accrue for each spoke for positive inventory. The costs are added at time moments when the inventory level changes. These costs are assumed to be proportional to storage time and the according amount of products. Furthermore costs for holding TU's are added to the cost manager. These live time costs $cost_{TUlifetime}$ accrue on the time a TU exists in the HAS. TU live time costs are collected for each TU.

Since the assessment of current version of simulator is based on a single criterion, all cost are combined to a total cost value $cost_{total}$, so that:

$$cost_{total} = cost_{shortage} - gain + cost_{transporter} \\ + cost_{transportation} + cost_{TU\ live\ time} + cost_{holding}$$

The value of $cost_{total}$ provides information about the goodness of the current configuration of the HAS, comparing it to other configurations the configuration can be assessed. Obviously a higher value indicates a worse configuration and a HAS which shall pay has to have a negative value.

5.10 Examples

In the present section we report on the simulation optimization of two classes of hub-and-spoke system – a single hub with four respectively fifty spokes. In all examples we assume the same demand process for all spokes. In detail we assume exponentially or normally distributed interarrival times of clients with exponentially respectively normally distributed demand per client. Similar distributions are assumed for the waiting time limits. Time is measured in time units, which may be one hour, one day, and so on. The other model parameters are as follows:

- | | |
|----------------------------------|----------------------------------|
| Distances between hub and spokes | - 50 distance units; |
| TU capacity | - 100 item units; |
| Lot size Q | - 10 item units; |
| TU speed | - 1 distance unit per time unit; |

Transportation cost $c_{a,x}^{penalty}$	- 0.1 per item unit and distance unit;
Transportation cost $c_{a,x}^{penalty}$	- 2.0 per capacity unit and distance unit;
Fixed loading / unloading cost $c^{loading}$	- 100;
Service policy of the hub	- FIFO;
Service policy of the spokes	- FIFO;
Ordering policy of the spokes	- (s, nQ)
Initial inventory	- zero
Gain g_k	- 12 per sold item unit;
Shortage cost $c_{k,a}^{shortage}$	- 6 per item unit;
Holding cost $c_{k,a}^{holding}$	- 0.12 per item unit and time unit.

The decision variables are the number T of TU's and for the spokes the order levels s and the number n of lot size. For the search for an optimal solution we restrict the possible values for T to the set 1, 2,..., 10, for s to the set -120, -110, -100,..., 120, and for n to the set 1, 2,..., 20. We applied four optimizers – a Genetic algorithm, Tabu search, and two hybrid algorithms, which apply the Genetic algorithm and Tabu search in a parallel and serial manner. The performance of a solution is estimated by the total cost, which accrue as the result of a single simulation run over 100 000 time units with a transition phase of 1000 time units.

The graph in figure (5.8) shows for the single-hub-four-spokes system with N(100,10)-distributions, how the optimization process evolves in time under the hybrid parallel algorithm. Similar pictures we have for all variants of the applied optimizers and all other examples (see Appendix C). Table 5.7 summarizes the results for all four optimizers for the single-hub-four-spokes system. There are also given the results when the normal distributions are replaced by exponential distributions with parameter 0.1.

Example: 1 Hub x 4 Spokes homogeneous structure - Normal Distribution

Optimization: Hybrid Parallel (N [100;10])

Solution number = 1747

Number of TU = 2

Total Costs = -23.953

Spoke	Order Level	Order Amount
1	10	130
2	0	60
3	-10	200
4	0	60

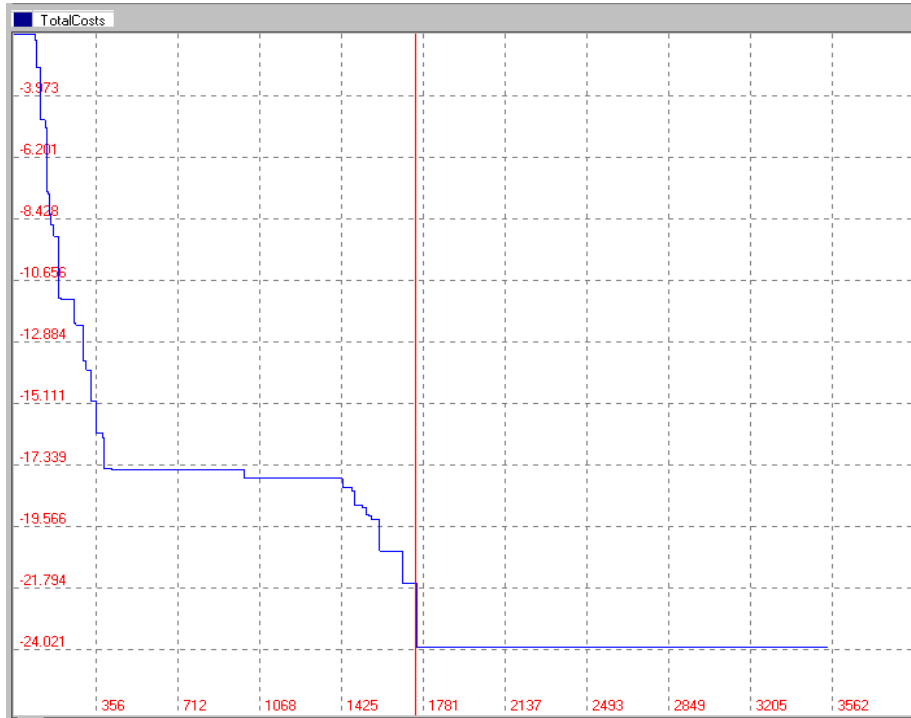


Figure 5.8: Simulation Optimization Results for the Four-Spokes-Single-Hub System, $N(100,10)$ -Distributions

In the single-hub-fifty-spokes system the decision variable T can vary between 1 and 50. Table 5.8 consists the results, but without the policy parameters (see Appendix B for more information).

Considering the results in Table 5.7 and Table 5.8 we can formulate some conclusions.

1. The two hybrid optimizers and Tabu search outperform the Genetic algorithm. This was not to expect and gives a hint to the fact that the parameters of Genetic algorithm may be bad adjusted to the considered problem. It is surprising that Tabu search works well.
2. The hybrid serial optimizer performs best, excluding the problem with fifty spokes and exponential distribution.

3. For the HAS with four spokes the optimization process finds the best solution after maximum 2500 considered solutions. For the much more complex problems with fifty spokes that maximal number increases up to about 18 000. Also the convergence to a good solution is considerably slower.
4. Very different solutions lead to similar total cost expectations. This can be a hint to the fact that our problem has many local minima with small cost differences.

<i>N</i> (100;10) distribution						
	T	<i>(s, n)</i>				<i>c</i> ^{total cost}
Hybrid parallel	2	(10, 13)	(0, 6)	(-10, 20)	(0, 6)	-23.95
Hybrid serial	9	(0, 6)	(0, 6)	(0, 6)	(0, 6)	-29.77
Genetic algorithm	4	(-30, 18)	(70, 7)	(0, 7)	(0, 11)	-15.00
Tabu search	10	(0, 6)	(0, 6)	(0, 6)	(0, 6)	-29.77
<i>EXP</i> (0.1) distribution						
Hybrid parallel	6	(50, 9)	(40, 11)	(40, 9)	(40, 10)	-6.84
Hybrid serial	9	(40, 11)	(40, 11)	(40, 9)	(40, 10)	-6.86
Genetic algorithm	5	(20, 13)	(20, 13)	(20, 13)	(20, 13)	-4.39
Tabu search	10	(50, 9)	(50, 10)	(50, 8)	(50, 11)	-6.79

Table 5.7: Results of the Simulation Optimization Approach for 1 Hub and 4 Spokes

	<i>N</i> (100;10)		<i>EXP</i> (0.1)	
	<i>T</i>	<i>c</i> ^{total cost}	<i>T</i>	<i>c</i> ^{total cost}
Hybrid parallel	21	-260.66	50	-82.40
Hybrid serial	26	-395.75	41	-86.09
Genetic algorithm	18	-161.40	23	-57.18
Tabu search	23	-337.95	42	-85.82

Table 5.8: Results of the Simulation Optimization Approach for 1 Hub and 50 Spokes

To summarize we can say that the suggested simulation optimization approach works well, but we need more empirical material to improve the here-applied optimizers and to prove the above given conclusions.

Chapter 6

Summary and conclusion

The fundamental premise upon which this dissertation is based is that fleet sizing and allocation problems are interdependent, and together have a significant effect on the long-run profitability of a transportation system. This relationship has not been adequately addressed in previous research in order to realistically model transportation systems, it is important to recognize crucial characteristics of most transportation systems; they are dynamic because demands on the system change over time, and there is uncertainty both in system performance (e.g., travel time), and in forecasting the demands on the system in the future. The objective of this research has been to develop a model to aid decisions on fleet sizing in situations in which demand fluctuates over time (including both deterministic and stochastic changes) and TU travel times over network are uncertain, leading to uncertainty regarding when specific TU's will be available for use to meet specific demands.

Chapters 1 and 2 The FSAP has been a widely discussed topic in the literature and the specification of fleet sizing divided by previous research into three categories:

- (i) Simulation techniques,
- (ii) Analytical techniques, and
- (iii) Hybrid techniques (Simulation and Analytical).

We also introduced the classification of empty TU distribution models and Classification of fleet sizing models.

In chapter 3 we introduced a simple case “one-to-one case” problems to illustrates the following basic principles:

1. The profitability of the system can be expressed in terms components (revenue, stockout cost, TU ownership cost, and TU movement cost) each of which is a function of fleet size.
2. As long as the marginal value (incremental increase in revenues and decreases in stockout and operating costs) of an additional TU is greater than the marginal cost (incremental increase in fleet ownership cost) it pays to increase the size of the fleet.

3. At some point, the marginal value of TU begins to decrease (in this case quite suddenly) and eventually drops below the marginal cost at which point additional TU's reduce profitability.
4. The point at which marginal value equals marginal cost is dependent on demand and the cost parameters.
5. TU routing decisions affect profitability both directly through the cost of moving the TU's and indirectly through their impact on the required fleet size.
6. The recognition of stochastic and dynamic elements of the transportation systems has proven to be very important.

In this chapter we also introduced another example for fleet sizing of one-to-many case. In this case, we consider the problem of determining the fleet size for a single TU type used to transport the items produced at the origin to many destinations. Items are produced in a deterministic production cycle, but TU travel times are stochastic. Finally in this chapter we also applied queueing theory methods to solve the allocation problem in case of stochastic demand in the spokes.

In chapter 4 we discussed solution approaches to the problem of the optimal size and allocation of transportation resources. We concentrate on the fleet-sizing-and-allocation problem for single hub networks. Generally, in a hub-and-spoke transportation network a centralized planner has to find freight routes, frequency of service, type of TU's to be used, and transportation volumes. We have studied this problem with two cases:

1. Solutions for the allocation problem without renting possibilities, and
2. Solutions for the allocation problem with renting possibilities.

In chapter 5 we introduced a genetic algorithm (GA) approach for the fleet sizing and allocation problem (FSAP) and some definitions for the GA. We present the developed genetic representation and use a randomized version to generate the initial random population. We designed suitable crossover and mutation operators for the GA improvement phase. We also discussed the principle of simulation optimization and describe a simulator for hub-and-spoke systems and give some examples show the applicability of our simulation optimization approach.

In conclusion, the contributions of this research are as follows:

1. We concentrated on the fleet-sizing-and-allocation problem for single hub networks and discussed solution approaches to the problem of the optimal size and allocation of transportation resources.
2. We studied a HAS problem in a single-period and deterministic-demand for two cases are mentioned above. For each case, based on Marginal Analysis, we developed a simple algorithm, which gives us the cost-minimal allocation. Examples show that the optimal solutions for both two cases above can be very different with respect to the optimal allocation as well as the corresponding cost values.
3. We also studied a HAS problem with continuous time and stochastic demand. To solve this problem, based on Marginal Analysis, we applied queueing theory methods.
4. An approximate solution for the multi-period, deterministic demand model can be determined by applying the algorithm for the single-period, deterministic demand in a successive way. Since the multi-period, deterministic demand problem is NP-hard we suggest to use Genetic Algorithms. Some building elements for these are described.
5. For the most general situation (e.g., infinite planning horizon with continuous time, stochastic demand, stochastic transportation times) we suggest to use simulation optimization. To realize the simulation optimization approach we could use the software tool CAOS. We used CAOS for two classes of hub-and-spoke system: i. A single hub with four spokes, and ii. A single hub with fifty spokes.
6. In the case of a single hub with four spokes the optimization process finds the best solution after maximum 2500 considered solutions. For the much more complex problems with fifty spokes that maximal number increases up to about 18 000. Also the convergence to a good solution is considerably slower.

The conclusions and future studies:

In this dissertation we are studied the solutions for the fleet-sizing-and-allocation problem (FSAP) for a single hub and spokes structure (HAS) with a single type of transportation units. Also, we considered the transportation units moving directly from the hub to spoke and return to the hub. In the future study we can apply this work in many cases:

- 1- The solutions of FSAP for a single hub and spokes structure with different types of transportation units.
- 2- The solutions of FSAP for multiple-hub and spokes, or multiple-hub hybrid and spokes structure with a single type of transportation units or different type of transportation units.
- 3- We can also study this problem with transshipment (peddling case), which involves dispatching transportation units that deliver the items to more than one destination per load.
- 4- To use as optimization tool not only GA, but also other soft-methods like tabu search or simulated annealing.

Appendix A: Marginal Analysis

We consider here the Marginal Analysis (MA) and some important properties. The formulation is for the maximization of a concave and strictly increasing function f .

FOX 1966 investigated for the problem

$$\left. \begin{array}{l} \text{Maximise } f(n) = \sum_{i=1}^M f_i(n_i) \\ \text{s.t.} \\ C(n) := \sum_{i=1}^M c_i \cdot n_i \leq C; \\ n_i \in \mathbf{N}, i = 1, \dots, M. \end{array} \right\}$$

where $C > 0$ and $c_i > 0, i=1, 2, \dots, M$, the following algorithm:

Algorithm „Marginal analysis“ (MA):

1. **Initialisation:**
 $r := 0; n^r := (0, 0, \dots, 0); C^r := 0.$
2. **Search:**
 WHILE $C^r < C$ DO
 BEGIN
 2.1 $r := r + 1.$
 2.2 Define index j , which maximises

$$\frac{f_j(n_j^{r-1} + 1) - f_j(n_j^{r-1})}{c_j}.$$

 2.3 $n^r := n^{r-1} + e_j.$
 2.4 $C^r := C^{r-1} + c_j.$
 END.
- 3 **Output:** n^{r-1} and $C^{r-1}.$

To describe properties of algorithm (MA) we call an allocation-vector n *undominated* if for all $n' \in \mathbf{N}^M$ from $f(n') \geq f(n)$ follows $C(n') \geq C(n)$. Obviously undominated vectors generate the Pareto-front of solutions. The main results of Fox (1966) are the following ones.

Property 1

If all $f_i(\cdot)$ are integer-concave and strictly increasing, $i=1, 2, \dots, M$, then algorithm (MA) leads to undominated vectors n .

Property 2

Let all $f_i(\cdot)$ be integer-concave and strictly increasing, $i=1, 2, \dots, M$. Let n^1, n^2, \dots, n^m denote the sequence of allocation-vectors generated by algorithm (MA), and let n^* denote the optimal allocation-vector. Then it holds:

- (I) $f(n^{m-1}) \leq f(n^*) < f(n^m)$.
- (II) $C(n^{m-1}) \leq C(n^*) < C(n^m)$.
- (III) $0 < C(n^m) - C(n^{m-1}) \leq \max_j c_j$

Property 3

If, in addition to the conditions stated in Property 2, $c_1 = c_2 = c_M = c > 0$, then algorithm (MA) generates an optimal solution.

Remarks:

- i. If Z denotes the set of all optimal allocations then n^{m-1} must not be an element of Z . However, (I) to (III) give some information on the quality of n^{m-1} .
- ii. For $c_1 = c_2 = c_N = c > 0$ and C as an integer multiple of c the optimal value of the criterion $f(n^*(C))$ is concave and strictly increasing with respect to C .
- iii. For a linear criterion $f(n) = f_1 n_1 + \dots + f_N n_N$ we have a version of the Knapsack-Problem.
- iv. Let the constraints be non-linear, i.e., $C(n) = \sum_{i=1}^N c_i(n_i)$ with $c_i(\cdot)$ convex and strictly increasing for $i=1, 2, \dots, M$. If in algorithm „Marginal analysis“ Step 2.2 is replaced by

$$\frac{f_j(n_j^{k-1} + 1) - f_j(n_j^{k-1})}{c_j(n_j^{k-1} + 1) - c_j(n_j^{k-1})},$$

then Property 1 holds also.

Appendix B: Simulation Optimization Results

This appendix contains the simulation optimization results for HAS with four and fifty spokes.

I - 1 Hub x 4 Spokes homogeneous structure - Exponential Distribution

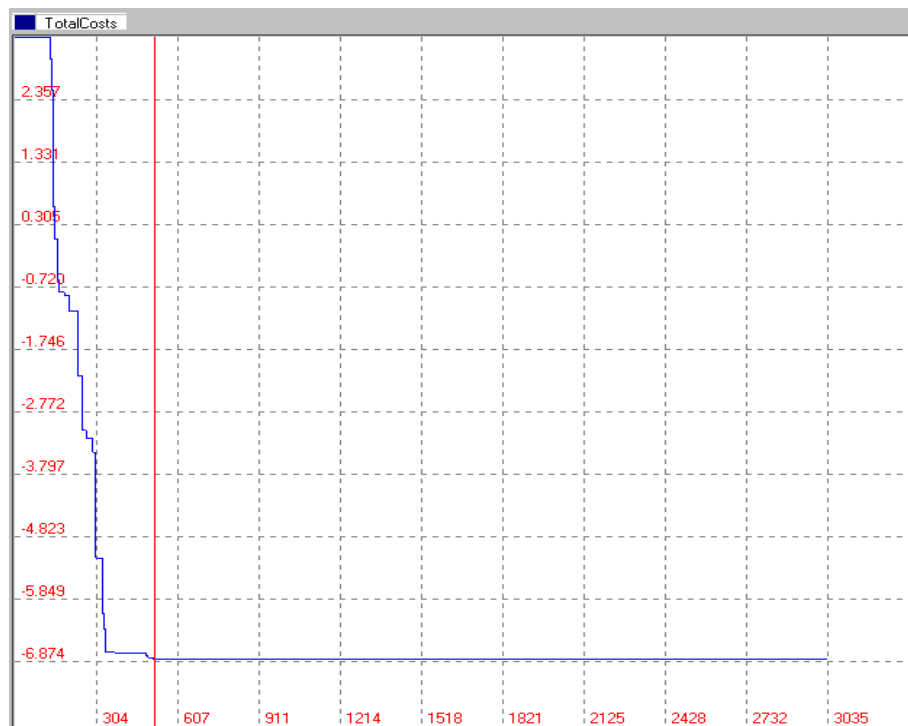
Optimization: Hybrid Parallel (Exp [0.1])

Solution number = 565

Number of TU = 6

Total Costs = -6.843

Spoke	Order Level	Order Amount
1	50	90
2	40	110
3	40	90
4	40	100



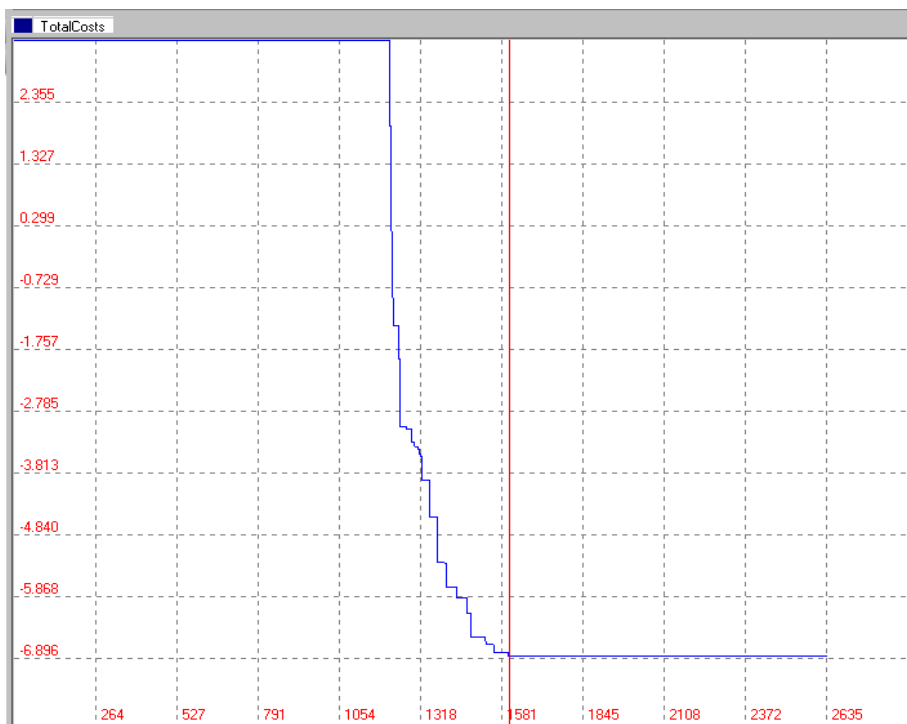
Optimization: Hybrid Seriell (Exp [0.1])

Solution number = 1586

Number of TU = 9

Total Costs = -6.864

Spoke	Order Level	Order Amount
1	40	110
2	40	110
3	40	90
4	40	100



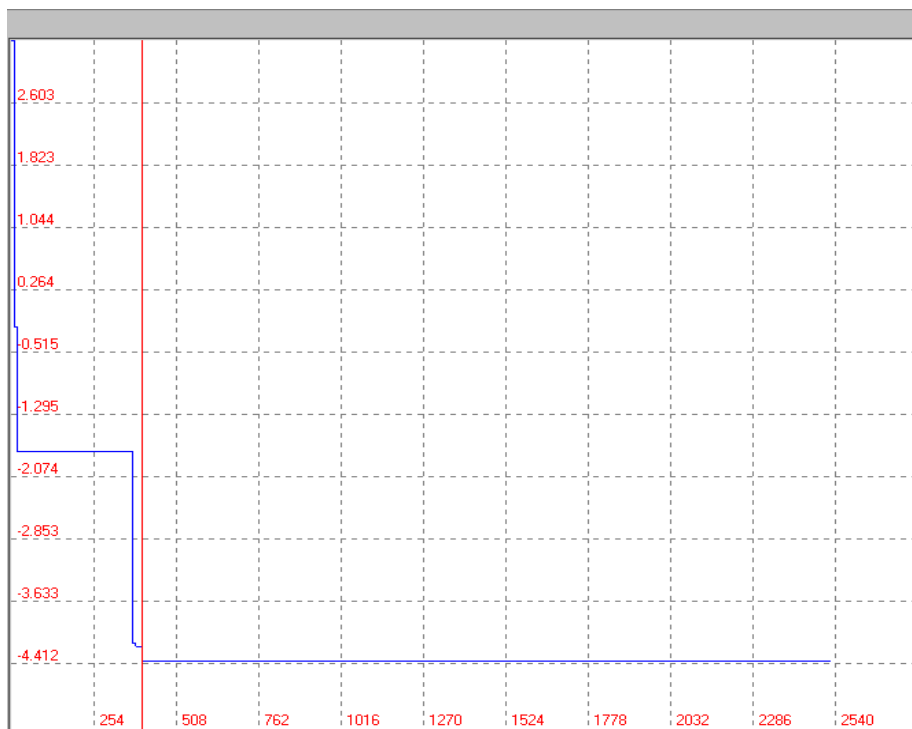
Optimization: GA Generation (Exp [0.1])

Solution number = 401

Number of TU = 5

Total Costs = -4.388

Spoke	Order Level	Order Amount
1	20	130
2	20	130
3	20	130
4	20	130



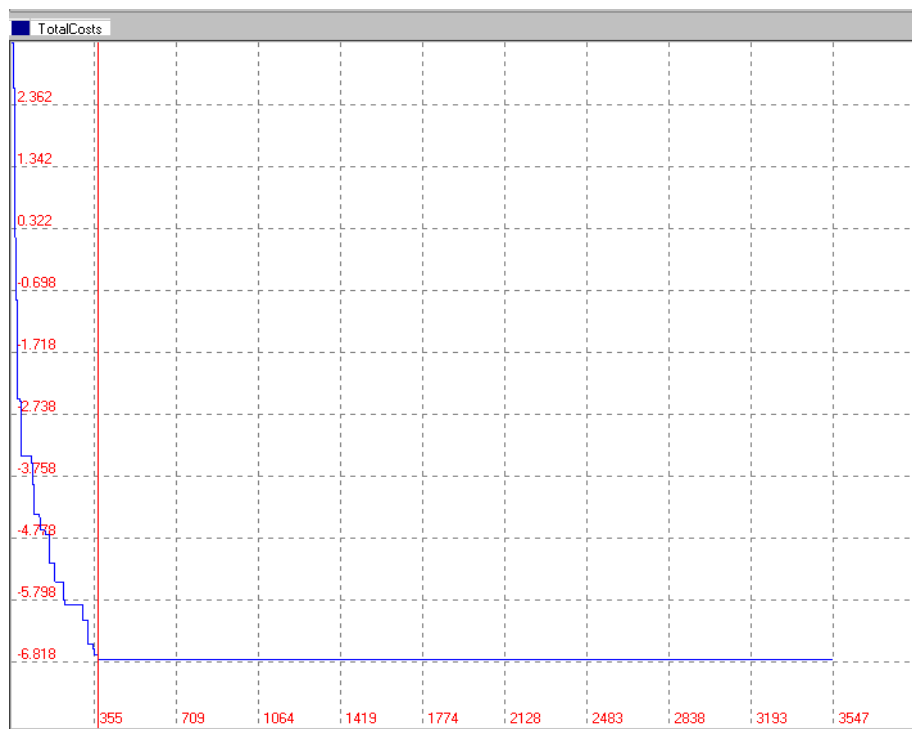
Optimization: Tabu Search (Exp [0.1])

Solution number = 370

Number of TU = 10

Total Costs = -6.786

Spoke	Order Level	Order Amount
1	50	90
2	50	100
3	50	80
4	50	110



II- 1 Hub x 4 Spokes homogeneous structure - Normal Distribution

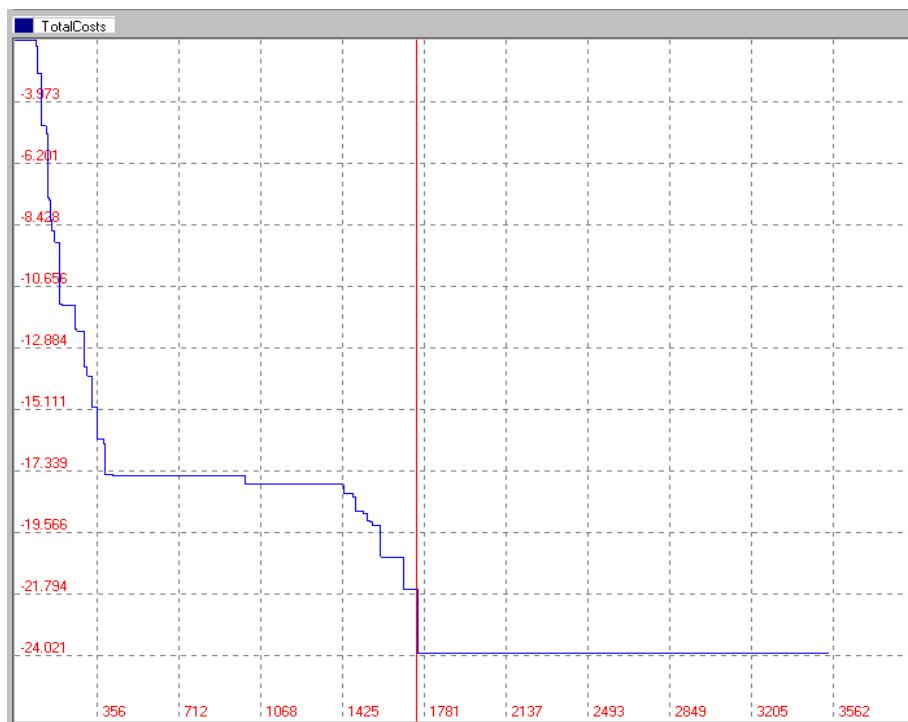
Optimization: Hybrid Parallel (N [100;10])

Solution number = 1747

Number of TU = 2

Total Costs = -23.953

Spoke	Order Level	Order Amount
1	10	130
2	0	60
3	-10	200
4	0	60



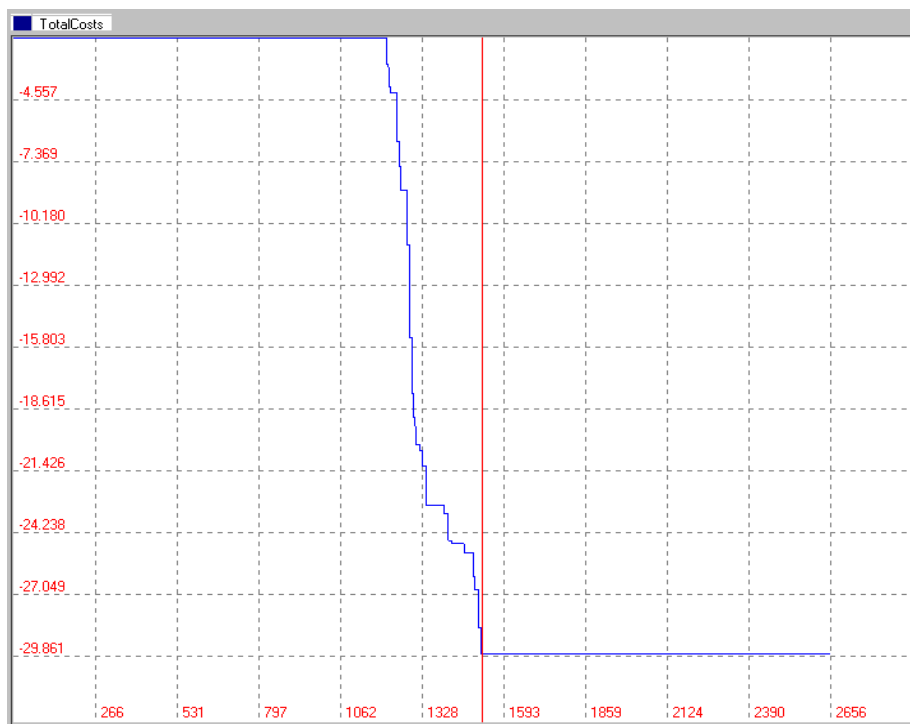
Optimization: Hybrid Seriell (N [100;10])

Solution number = 1508

Number of TU = 9

Total Costs = -29.774

Spoke	Order Level	Order Amount
1	0	60
2	0	60
3	0	60
4	0	60



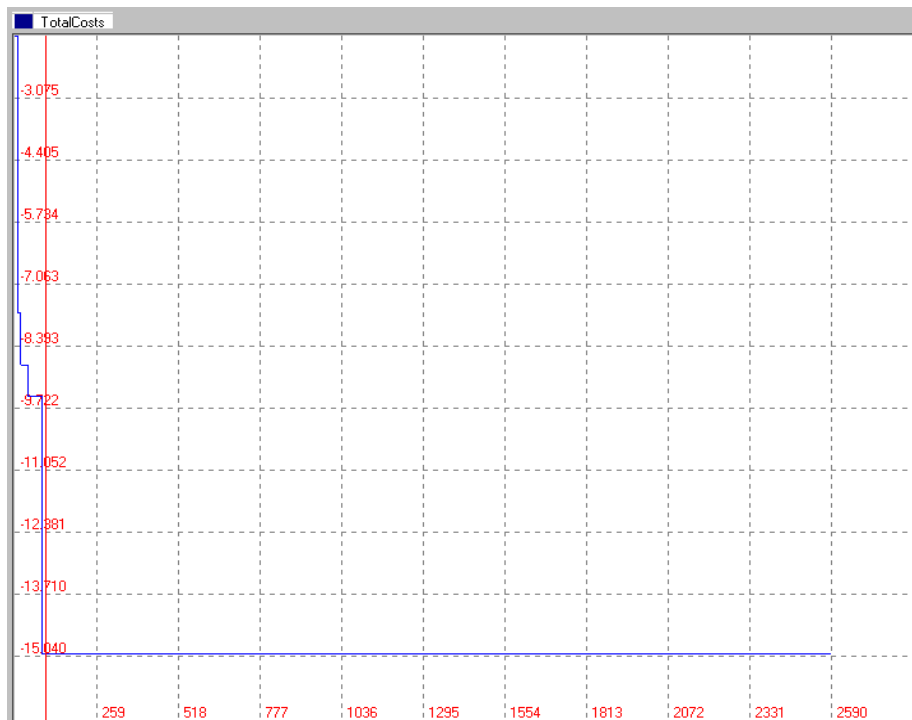
Optimization: GA Generation (N [100;10])

Solution number = 85

Number of TU = 4

Total Costs = -14.999

Spoke	Order Level	Order Amount
1	-30	180
2	70	70
3	0	70
4	0	110



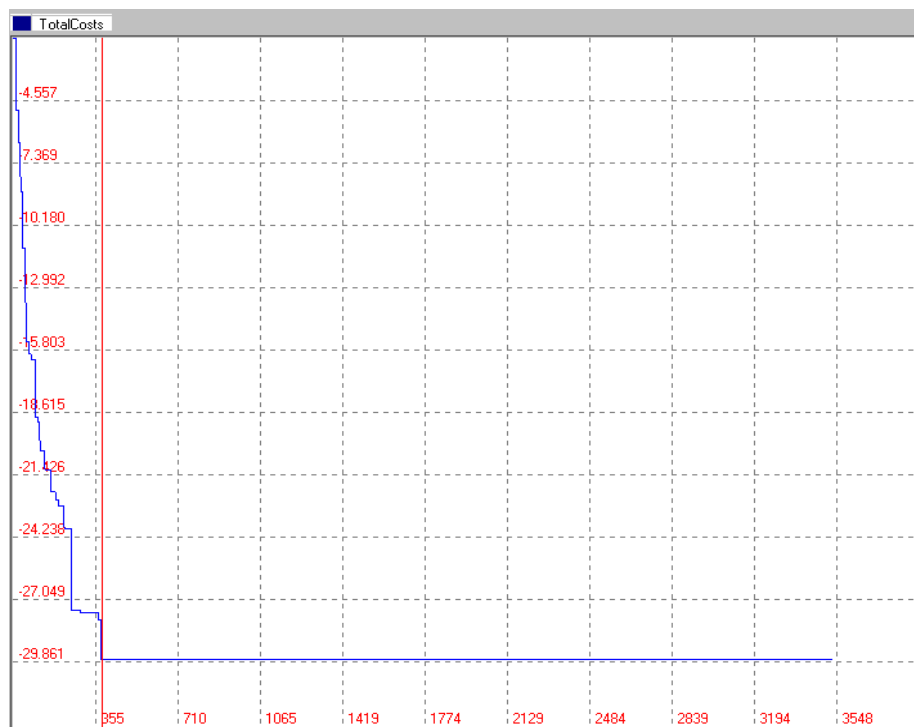
Optimization: Tabu Search (N [100;10])

Solution number = 378

Number of TU = 10

Total Costs = -29.774

Spoke	Order Level	Order Amount
1	0	60
2	0	60
3	0	60
4	0	60



III- 1 Hub x 50 Spokes homogeneous structure - Exponential Distribution

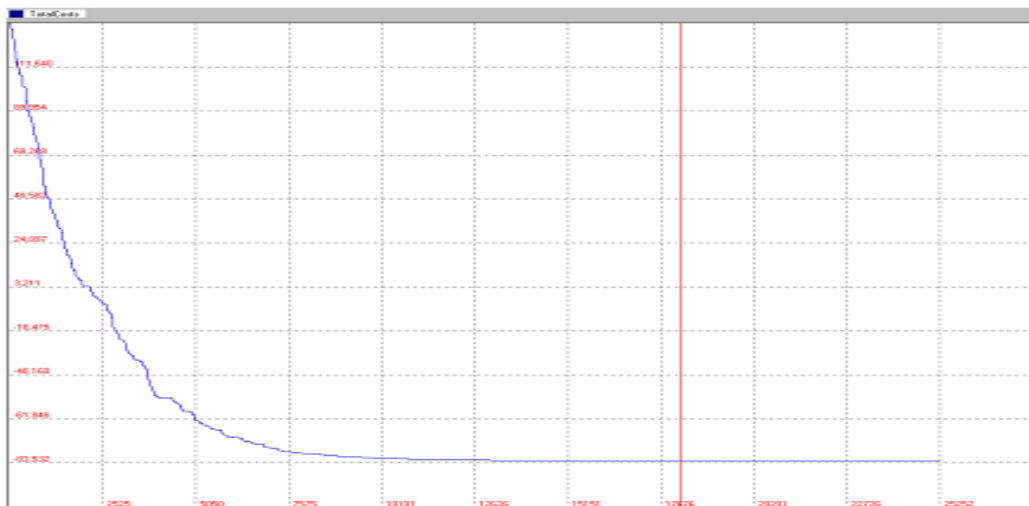
Optimization: Hybrid Parallel (E[0.1])

Solution number = 18017

Number of TU = 50

Total Costs = -82.404

Spoke	Order Level	Order Amount	Spoke	Order Level	Order Amount
1	50	100	26	50	100
2	50	90	27	50	110
3	40	110	28	30	110
4	40	100	29	40	90
5	50	90	30	40	100
6	50	90	31	40	110
7	50	90	32	50	110
8	50	100	33	50	100
9	50	100	34	50	90
10	40	120	35	50	100
11	40	110	36	50	90
12	40	100	37	50	80
13	50	100	38	50	100
14	40	100	39	50	90
15	50	120	40	30	120
16	50	100	41	40	90
17	40	100	42	40	100
18	40	110	43	40	110
19	50	110	44	50	90
20	40	100	45	40	90
21	40	110	46	50	80
22	50	100	47	50	100
23	50	90	48	40	110
24	40	100	49	40	110
25	40	110	50	50	100



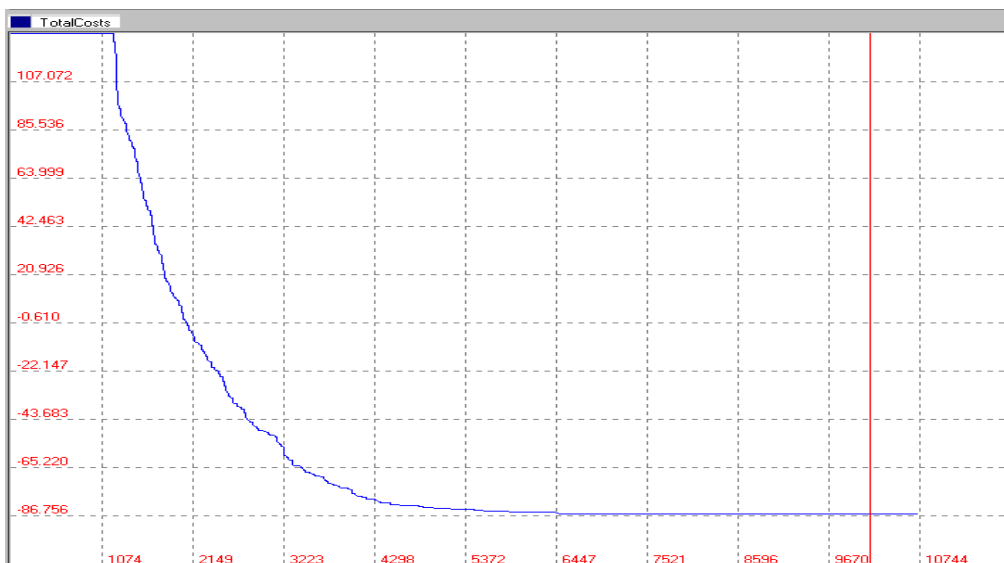
Optimization: Hybrid seriell (E[0.1])

Solution number = 10077

Number of TU = 41

Total Costs = -86.094

Spoke	Order Level	Order Amount	Spoke	Order Level	Order Amount
1	50	100	26	40	110
2	30	110	27	30	130
3	30	110	28	50	90
4	50	100	29	40	120
5	50	110	30	50	110
6	50	100	31	40	90
7	40	100	32	50	90
8	40	90	33	40	110
9	50	100	34	40	100
10	50	100	35	30	100
11	50	90	36	40	110
12	50	90	37	50	110
13	50	100	38	50	90
14	50	100	39	40	110
15	40	110	40	40	110
16	30	100	41	30	110
17	50	110	42	40	100
18	50	110	43	40	100
19	50	90	44	50	90
20	40	90	45	40	100
21	50	100	46	30	110
22	40	110	47	40	110
23	60	100	48	50	90
24	40	100	49	40	100
25	50	90	50	40	120



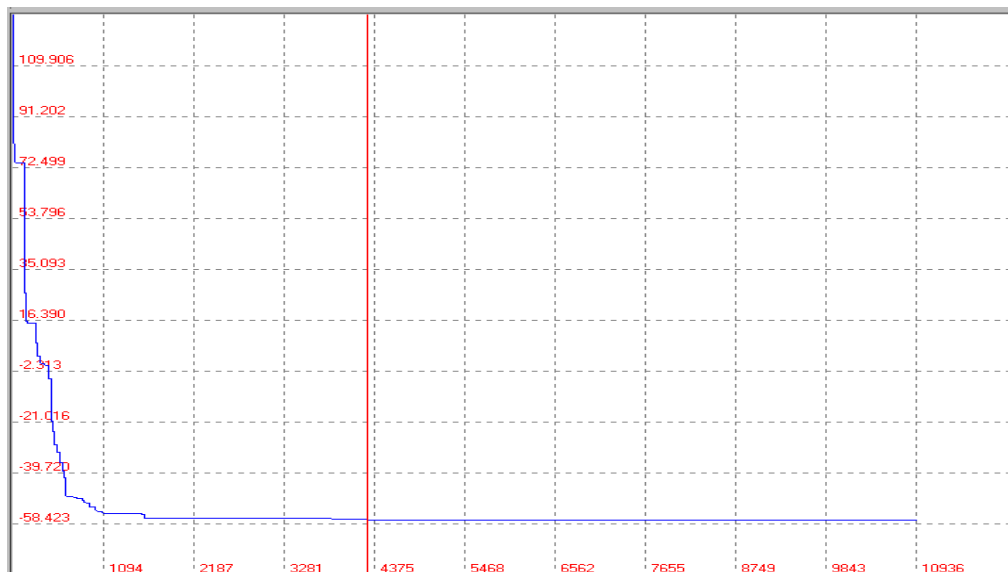
Optimization: GA Generation (E [0.1])

Solution number = 4252

Number of TU = 23

Total Costs = -57.176

Spoke	Order Level	Order Amount	Spoke	Order Level	Order Amount
1	30	140	26	20	130
2	30	140	27	20	130
3	30	130	28	30	140
4	30	130	29	30	140
5	30	130	30	30	140
6	30	130	31	30	140
7	30	130	32	30	130
8	30	140	33	30	130
9	30	140	34	30	130
10	30	140	35	30	130
11	30	130	36	30	130
12	30	130	37	30	130
13	30	130	38	30	140
14	30	130	39	30	140
15	30	130	40	30	140
16	30	140	41	30	140
17	30	140	42	40	140
18	30	140	43	40	140
19	30	130	44	40	140
20	30	130	45	50	140
21	30	140	46	40	140
22	30	140	47	40	140
23	30	140	48	40	140
24	30	140	49	30	140
25	30	130	50	30	140



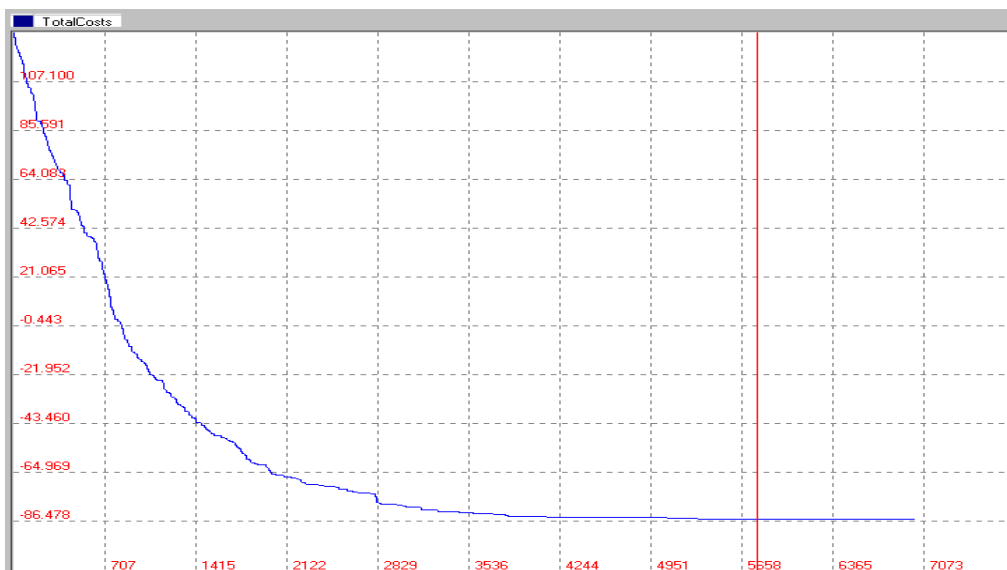
Optimization: Tabu search (E[0.1])

Solution number = 5764

Number of TU = 42

Total Costs = -85.816

Spoke	Order Level	Order Amount	Spoke	Order Level	Order Amount
1	50	100	26	40	110
2	50	100	27	30	130
3	50	100	28	50	90
4	40	90	29	40	120
5	40	90	30	50	110
6	50	100	31	40	90
7	40	100	32	50	90
8	40	90	33	40	110
9	50	100	34	50	90
10	50	100	35	40	110
11	50	90	36	30	100
12	40	100	37	40	110
13	50	100	38	40	110
14	30	110	39	40	110
15	40	110	40	40	110
16	30	100	41	40	110
17	40	100	42	40	100
18	50	110	43	40	90
19	50	90	44	50	100
20	40	110	45	40	110
21	50	100	46	50	100
22	40	110	47	40	110
23	60	100	48	50	90
24	40	100	49	40	100
25	50	90	50	40	120



IV- 1 Hub x 50 Spokes homogeneous structure – Normal Distribution

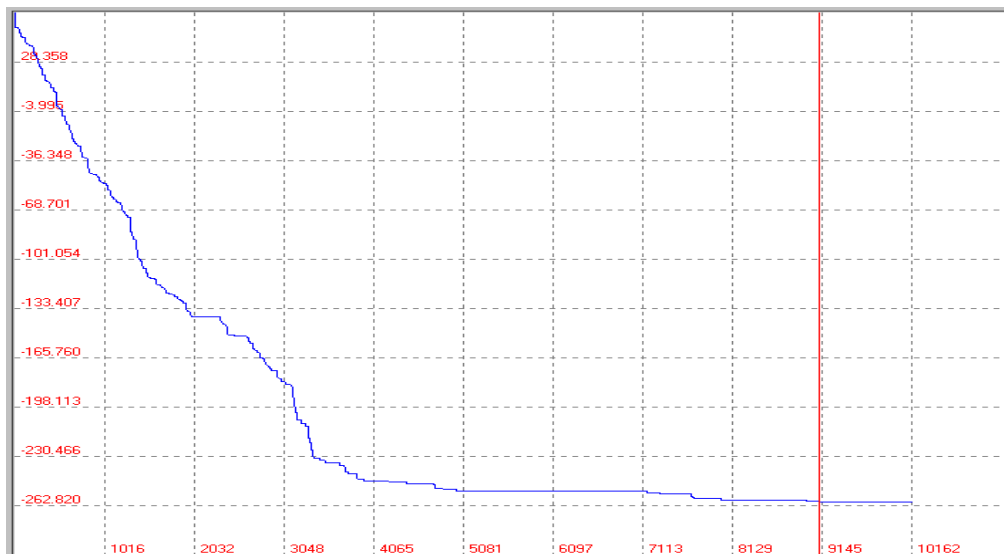
Optimization: Hybrid Parallel (N [100;10])

Solution number = 9011

Number of TU = 21

Total Costs = -260.663

Spoke	Order Level	Order Amount	Spoke	Order Level	Order Amount
1	-10	180	26	-10	220
2	0	170	27	60	120
3	20	60	28	0	60
4	0	60	29	0	120
5	10	100	30	10	190
6	-10	150	31	-50	100
7	0	60	32	30	170
8	-10	100	33	10	130
9	10	100	34	0	60
10	0	60	35	80	60
11	0	110	36	110	130
12	40	80	37	-10	110
13	0	60	38	-10	200
14	0	60	39	0	60
15	100	60	40	0	100
16	0	140	41	0	190
17	0	170	42	0	60
18	-10	100	43	10	60
19	-20	100	44	0	100
20	-20	200	45	80	80
21	60	70	46	0	60
22	10	160	47	0	150
23	-30	100	48	0	60
24	20	120	49	110	100
25	0	160	50	0	130



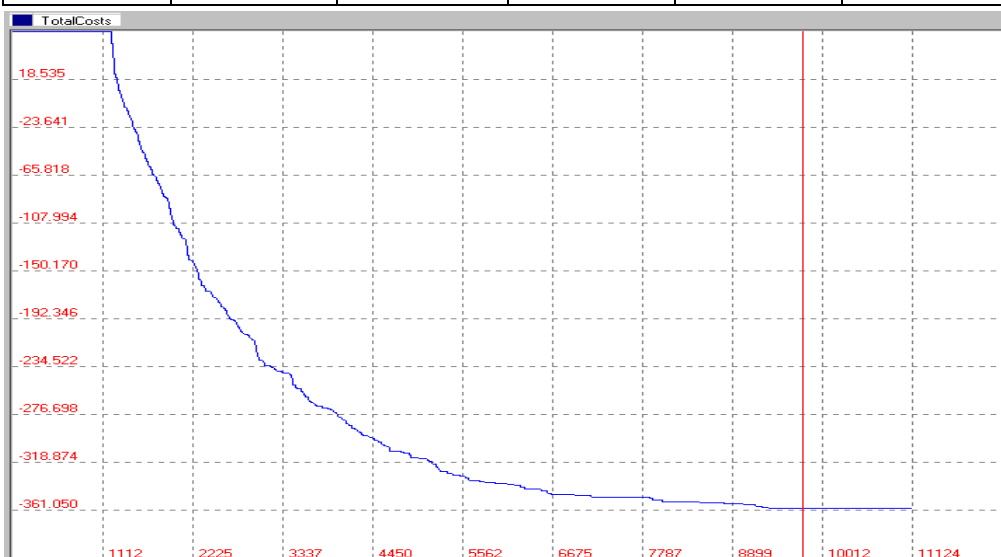
Optimization: Hybrid Seriell (N [100;10])

Solution number = 9656

Number of TU = 26

Total Costs = -359.753

Spoke	Order Level	Order Amount	Spoke	Order Level	Order Amount
1	0	60	26	-10	100
2	0	60	27	0	60
3	0	60	28	0	60
4	0	60	29	0	60
5	0	60	30	0	60
6	0	60	31	0	60
7	0	60	32	-20	100
8	0	60	33	0	60
9	0	60	34	0	60
10	0	60	35	0	60
11	0	60	36	0	60
12	0	60	37	0	130
13	0	60	38	0	60
14	0	60	39	0	60
15	0	60	40	0	60
16	0	60	41	0	60
17	0	60	42	0	60
18	0	60	43	0	60
19	0	60	44	0	60
20	-10	100	45	0	60
21	0	60	46	0	60
22	0	60	47	0	60
23	0	60	48	0	60
24	0	130	49	0	60
25	0	60	50	-10	100



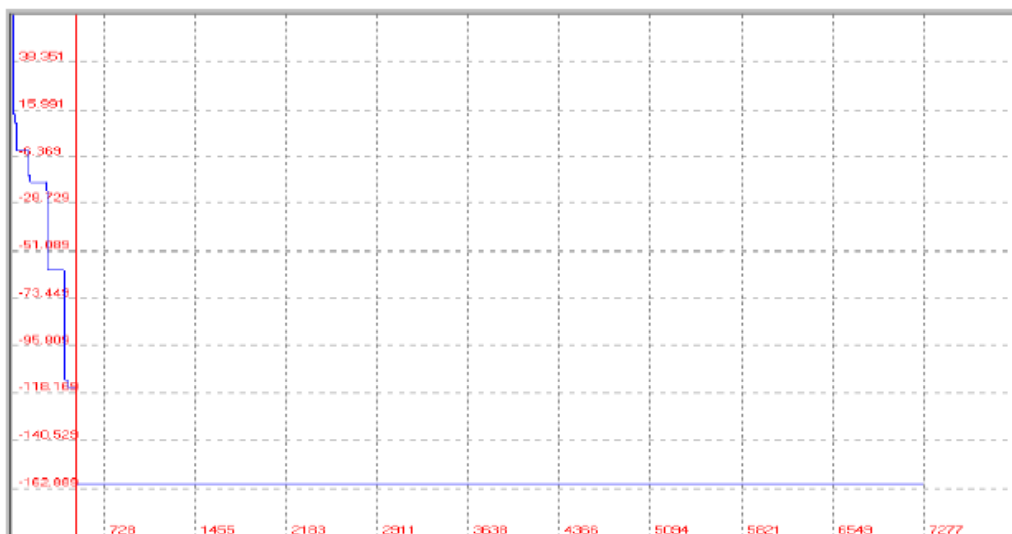
Optimization: GA Generation (N [100;10])

Solution number = 498

Number of TU = 18

Total Costs = -161.399

Spoke	Order Level	Order Amount	Spoke	Order Level	Order Amount
1	10	130	26	-10	110
2	10	160	27	-20	120
3	30	160	28	50	150
4	50	170	29	120	120
5	60	110	30	50	90
6	50	90	31	20	140
7	50	100	32	0	180
8	20	130	33	70	130
9	0	180	34	70	130
10	60	120	35	60	160
11	-10	120	36	-30	130
12	10	180	37	10	170
13	90	160	38	30	180
14	20	160	39	70	180
15	0	200	40	30	190
16	10	190	41	60	190
17	40	130	42	70	120
18	-20	110	43	80	110
19	90	150	44	50	150
20	110	150	45	30	150
21	40	160	46	30	130
22	50	180	47	70	120
23	50	130	48	100	100
24	40	140	49	90	100
25	40	130	50	90	140



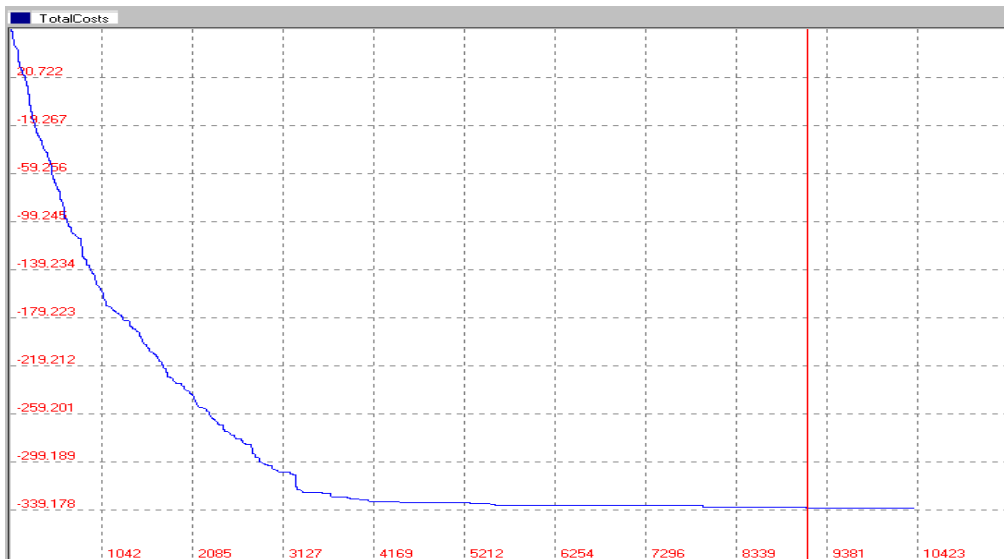
Optimization: Tabu search (N [100;10])

Solution number = 9083

Number of TU = 23

Total Costs = -337.948

Spoke	Order Level	Order Amount	Spoke	Order Level	Order Amount
1	0	60	26	0	60
2	0	60	27	0	60
3	0	60	28	0	60
4	0	60	29	0	110
5	0	60	30	0	60
6	0	60	31	0	60
7	0	60	32	0	60
8	0	60	33	0	60
9	0	100	34	0	60
10	0	60	35	0	130
11	-10	100	36	0	130
12	100	60	37	0	130
13	0	60	38	0	130
14	0	60	39	0	60
15	0	60	40	0	60
16	0	60	41	-20	200
17	0	60	42	0	120
18	0	170	43	10	130
19	10	160	44	0	60
20	0	150	45	0	60
21	0	60	46	0	130
22	0	130	47	0	60
23	0	60	48	10	120
24	0	60	49	10	120
25	0	200	50	0	60



Appendix C: The results data of the GA program for example 5.1 (4 Customers and 4 Periods).

This appendix contains all the result data for 4 customers, 4 periods and four variants of mutation with mutation probability 0.01 and 0.005.

Customer	Period			
	1	2	3	4
1	0	1	0	1
2	1	2	1	2
3	1	2	1	1
4	1	2	2	2
Remaining Vehicles no.	7	3	6	4

n_I^* with P(Mut) = 0.01

Customer	Period			
	1	2	3	4
1	0	1	0	1
2	1	2	1	2
3	1	2	1	1
4	1	2	2	2
Remaining Vehicles no.	7	3	6	4

n_{II}^* with P(Mut) = 0.01

Customer	Period			
	1	2	3	4
1	1	0	0	1
2	1	2	1	2
3	2	1	1	1
4	1	2	2	2
Remaining Vehicles no.	5	5	6	4

n_{III}^* with P(Mut) = 0.01

Customer	Period			
	1	2	3	4
1	0	1	0	1
2	1	1	2	2
3	1	2	1	1
4	1	2	2	2
Remaining Vehicles no.	7	4	5	4

n_{IV}^* with P(Mut) = 0.01

Customer	Period			
	1	2	3	4
1	0	1	0	1
2	1	2	1	2
3	1	2	1	1
4	1	2	2	2
Remaining Vehicles no.	7	3	6	4

n_I^* with P(Mut) = 0.005

Customer	Period			
	1	2	3	4
1	0	1	0	1
2	1	2	1	2
3	1	2	1	1
4	1	2	2	2
Remaining Vehicles no.	7	3	6	4

n_{II}^* with P(Mut) = 0.005

Customer	Period			
	1	2	3	4
1	1	0	0	1
2	1	1	2	2
3	1	2	1	1
4	1	2	2	2
Remaining Vehicles no.	6	5	5	4

n_{III}^* with P(Mut) = 0.005

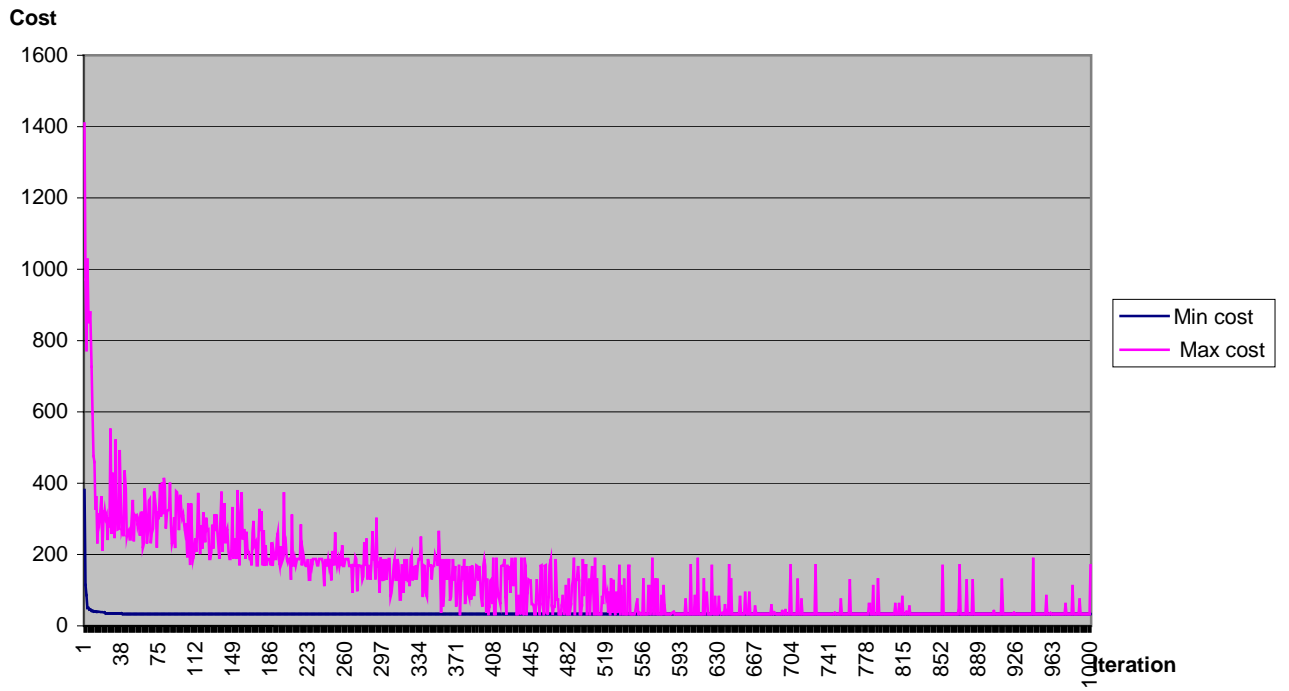
Customer	Period			
	1	2	3	4
1	0	1	0	1
2	1	2	1	2
3	1	2	1	1
4	1	2	2	2
Remaining Vehicles no.	7	3	6	4

n_{IV}^* with P(Mut) = 0.005

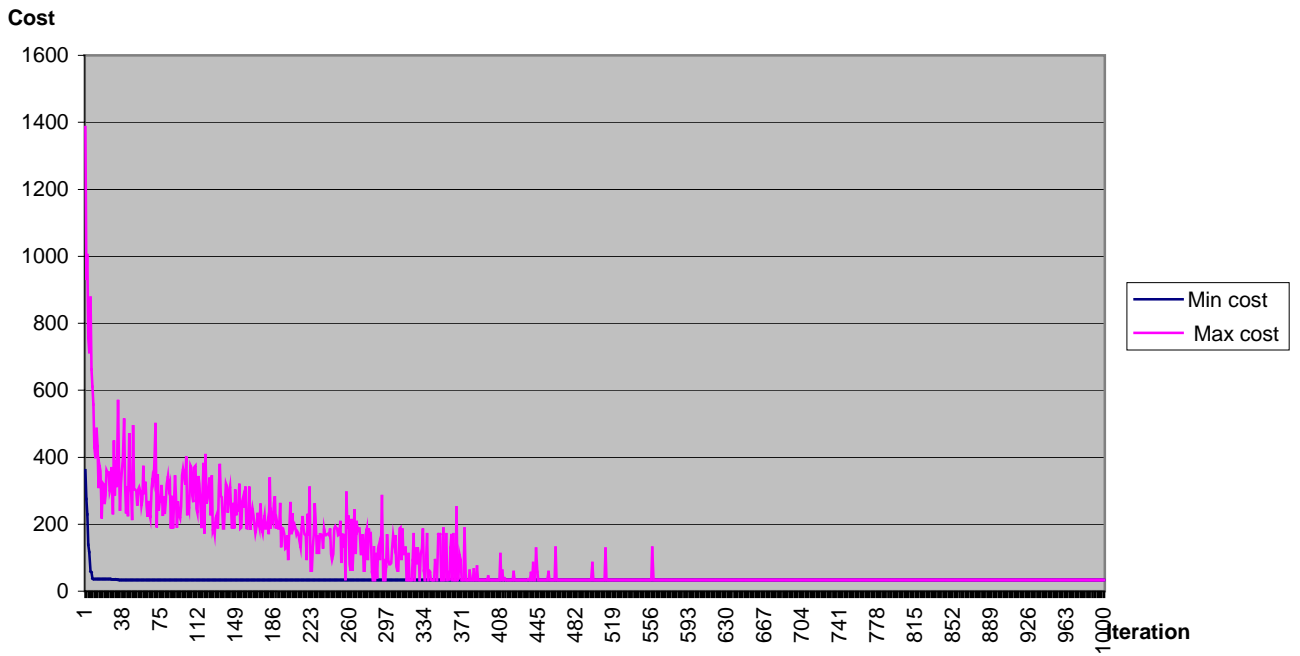
Iteration	Variant I		Variant II		Variant III		Variant IV	
	Min. Cost	Max. Cost	Min. Cost	Max. Cost	Min. Cost	Max. Cost	Min. Cost	Max. Cost
50	33.53	237.36	33.53	303.01	142.26	142.26	34.93	232.96
100	33.53	285.66	33.53	402.3	142.26	182.3	34.93	312.43
150	33.53	246.4	33.53	245.16	73.5	73.5	34.93	93.9
200	33.53	216.26	33.53	92.5	73.1	73.1	34.93	93.9
250	33.53	262.68	33.53	169.81	73.1	73.1	34.93	62.8
300	33.53	187.43	33.53	80.56	73.1	73.1	34.93	169.11
350	33.53	187.43	33.53	33.53	73.1	73.1	34.93	34.93
400	33.53	36.73	33.53	33.53	73.1	73.1	34.93	34.93
450	33.53	127.26	33.53	33.53	73.1	73.1	34.93	34.93
500	33.53	33.53	33.53	33.53	73.1	73.1	34.93	34.93
550	33.53	73.57	33.53	33.53	73.1	73.1	34.93	34.93
600	33.53	33.53	33.53	33.53	73.1	73.1	34.93	34.93
650	33.53	33.53	33.53	33.53	73.1	73.1	34.93	34.93
700	33.53	33.53	33.53	33.53	73.1	73.1	34.93	34.93
750	33.53	33.53	33.53	33.53	73.1	73.1	34.93	34.93
800	33.53	33.53	33.53	33.53	73.1	73.1	34.93	34.93
850	33.53	33.53	33.53	33.53	73.1	73.1	34.93	34.93
900	33.53	33.53	33.53	33.53	73.1	73.1	34.93	34.93
950	33.53	33.53	33.53	33.53	73.1	73.1	34.93	34.93
1000	33.53	169.81	33.53	33.53	73.1	73.1	34.93	34.93

Cost for 4 Customers, 4 Periods with $P(\text{Mut}) = 0.01$

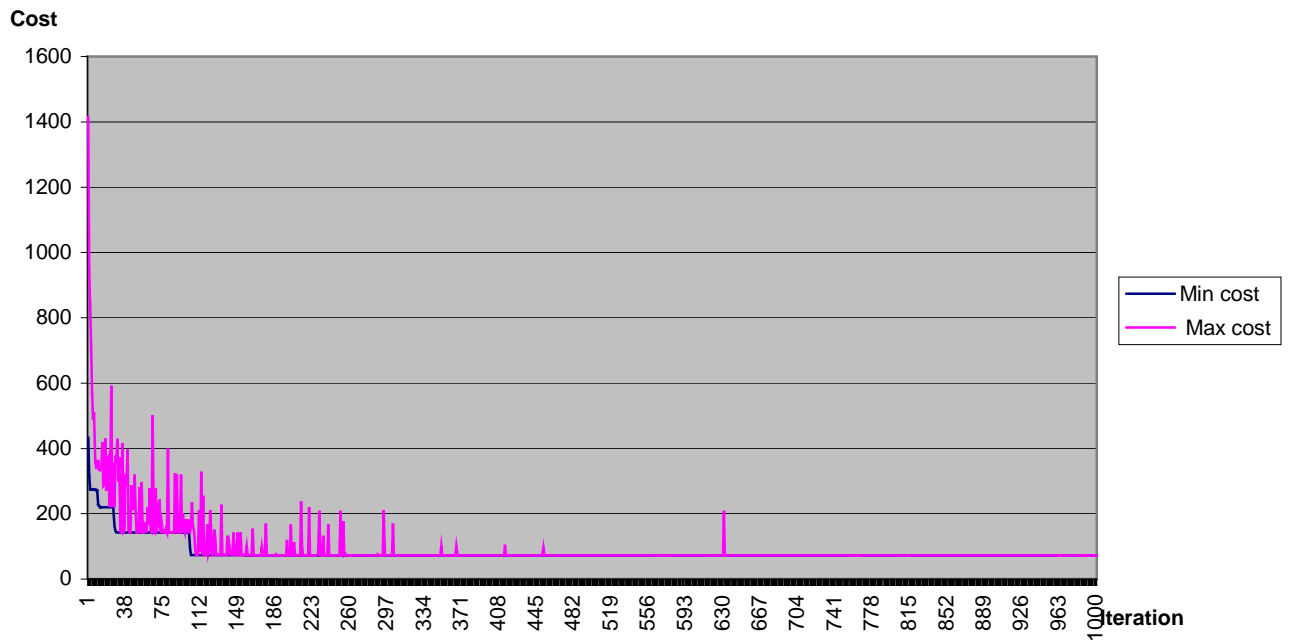
V1, P(Mut)=0.01



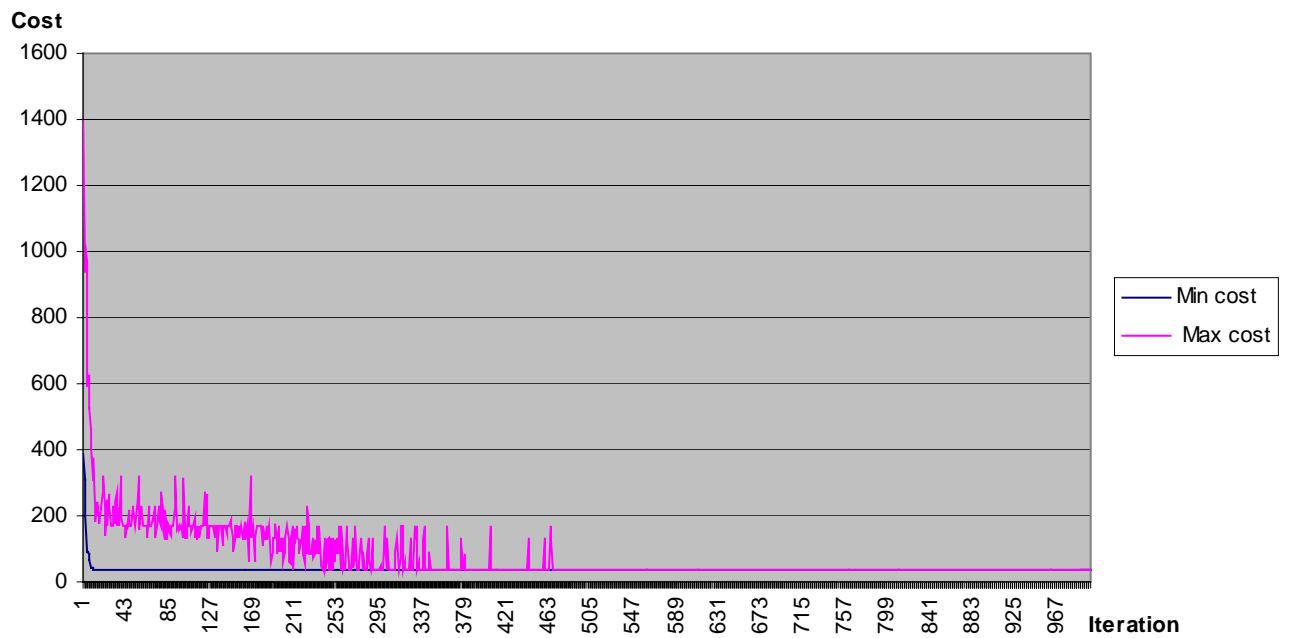
V2, P(Mut)=0.01



V3, P(Mut)=0.01



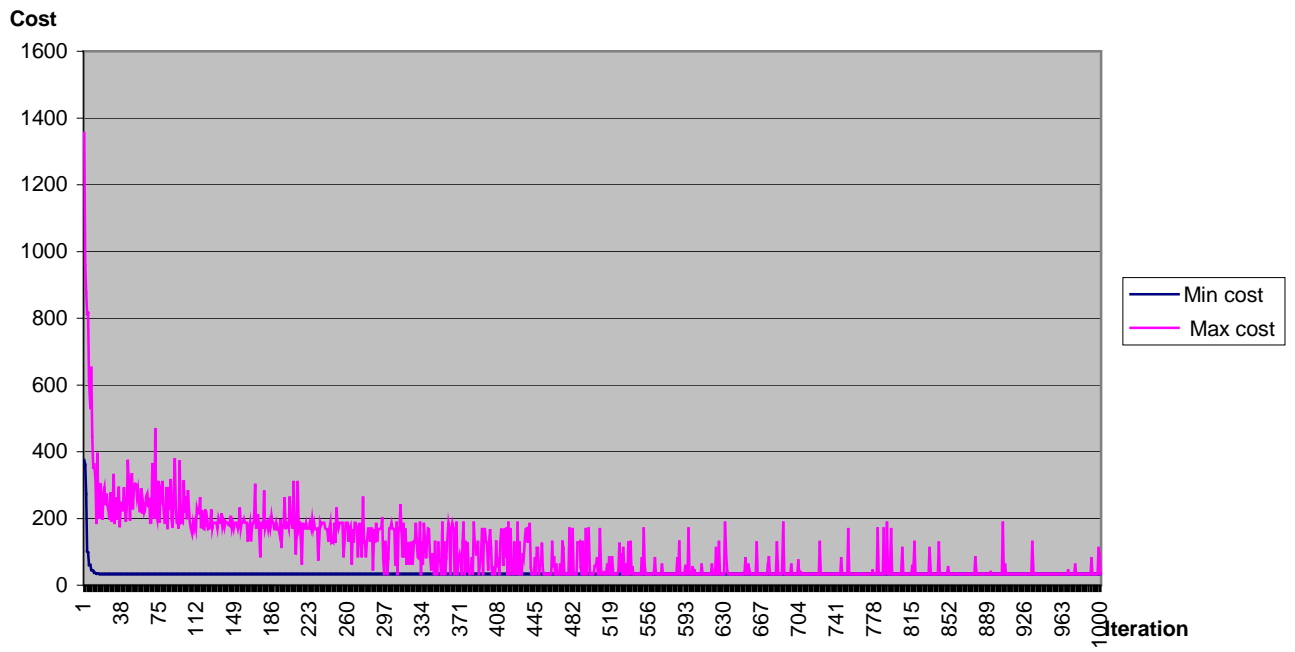
V4, P(Mut)=0.01



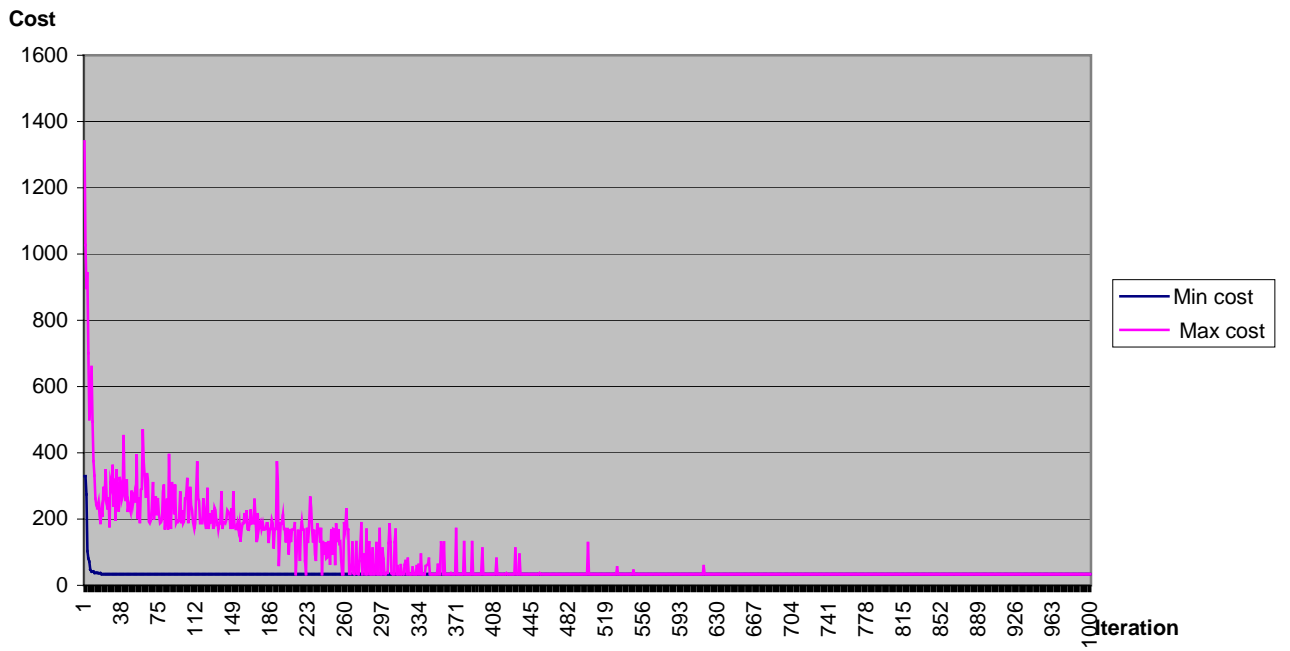
Iteration	Variant I		Variant II		Variant III		Variant IV	
	Min. Cost	Max. Cost	Min. Cost	Max. Cost	Min. Cost	Max. Cost	Min. Cost	Max. Cost
50	33.53	286.48	33.53	265.96	119.11	119.11	33.53	169.81
100	33.53	217.62	33.53	194.02	91.24	115.45	33.53	167.71
150	33.53	187.43	33.53	187.43	91.24	91.24	33.53	130.56
200	33.53	187.43	33.53	169.81	91.24	91.24	33.53	169.81
250	33.53	187.43	33.53	60.92	72.08	72.08	33.53	33.53
300	33.53	80.56	33.53	33.53	72.08	72.08	33.53	33.53
350	33.53	80.56	33.53	33.53	72.08	72.08	33.53	33.53
400	33.53	167.71	33.53	33.53	72.08	72.08	33.53	33.53
450	33.53	61.4	33.53	33.53	72.08	72.08	33.53	33.53
500	33.53	33.53	33.53	33.53	72.08	72.08	33.53	33.53
550	33.53	33.53	33.53	33.53	72.08	72.08	33.53	33.53
600	33.53	33.53	33.53	33.53	72.08	72.08	33.53	33.53
650	33.53	39.13	33.53	33.53	72.08	72.08	33.53	33.53
700	33.53	33.53	33.53	33.53	72.08	72.08	33.53	33.53
750	33.53	33.53	33.53	33.53	72.08	72.08	33.53	33.53
800	33.53	33.53	33.53	33.53	72.08	72.08	33.53	33.53
850	33.53	54.05	33.53	33.53	72.08	72.08	33.53	33.53
900	33.53	33.53	33.53	33.53	72.08	72.08	33.53	33.53
950	33.53	33.53	33.53	33.53	72.08	72.08	33.53	33.53
1000	33.53	33.53	33.53	33.53	72.08	72.08	33.53	33.53

Cost for 4 Customers, 4 Periods with $P(\text{Mut}) = 0.005$

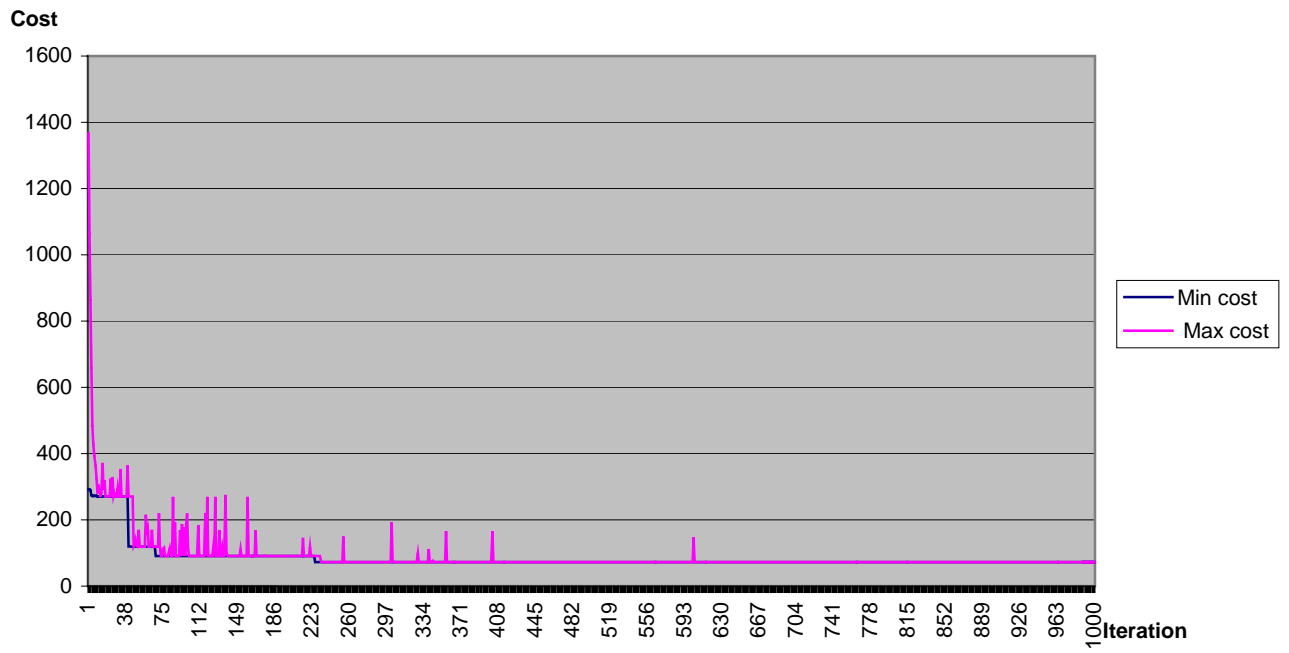
V1, P(Mut)=0.005



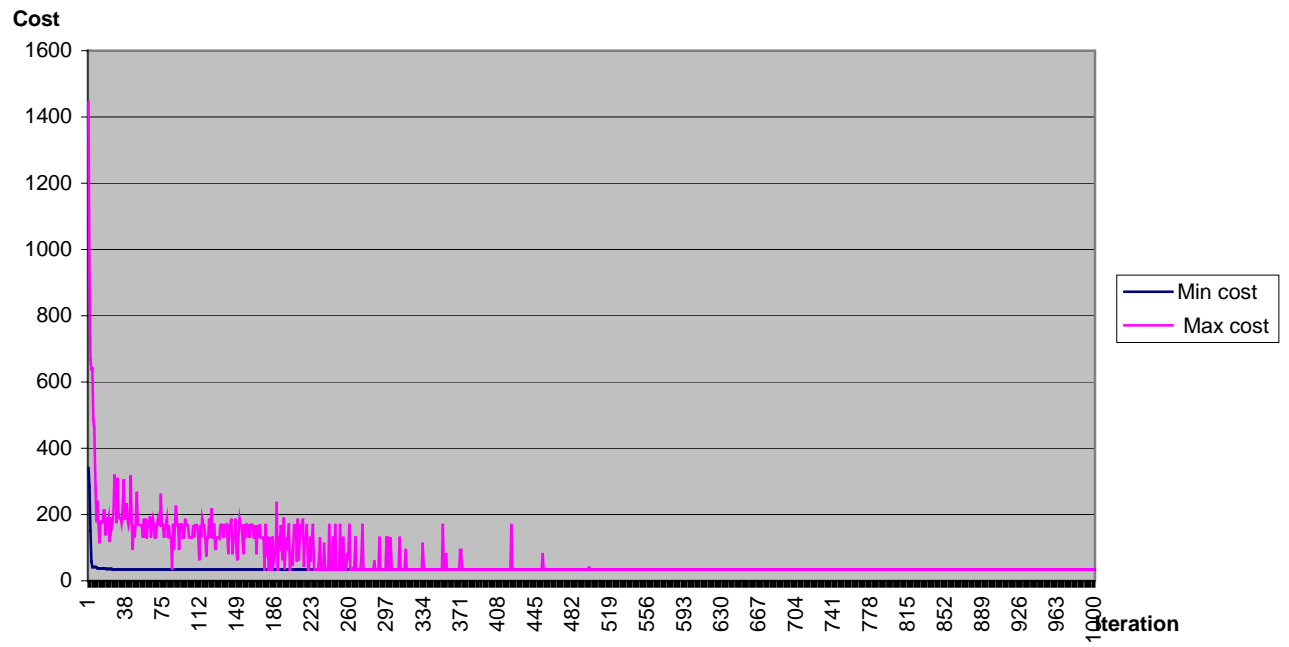
V2, P(Mut)=0.005



V3, P(Mut)=0.005



V4, P(Mut)=0.005



Appendix D: The results data of the GA program for example 5.2 (15 Customers and 15 Periods).

This appendix contains all the result data for 15 customers, 15 periods and four variants of mutation with mutation probability 0.01 and 0.005.

Customer	Period														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	1	0	1	2	3	2	1	4	1	3	1	2	1
2	0	2	3	0	1	1	2	1	0	1	3	2	1	1	3
3	1	0	1	2	1	3	2	2	1	0	3	1	2	1	1
4	1	1	1	1	1	2	2	3	2	1	1	1	2	1	1
5	1	1	3	2	1	1	2	1	0	4	0	0	0	0	5
6	0	1	2	1	1	1	1	0	3	3	0	2	0	2	1
7	0	2	1	1	3	0	2	0	0	2	4	0	3	1	1
8	2	1	1	2	3	0	0	4	0	2	3	1	1	0	1
9	0	2	2	1	0	4	0	2	2	1	0	2	3	0	2
10	1	2	2	1	1	1	0	3	1	2	0	1	1	2	1
11	0	2	1	0	0	4	2	1	3	0	0	1	0	3	1
12	0	1	1	3	1	0	1	1	1	2	1	0	2	2	2
13	2	2	1	1	2	2	0	3	1	0	4	0	2	1	0
14	1	0	3	1	1	3	1	1	0	0	2	3	1	2	2
15	2	2	0	2	0	1	2	1	5	0	0	1	2	1	3
Remaining Vehicles number	14	6	2	7	8	0	5	0	5	3	3	7	4	6	0

$$n_i^* \text{ with } P(\text{Mut}) = 0.01$$

Customer	Period														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	2	2	0	2	1	1	2	1	4	0	3	0	3
2	1	1	1	0	3	2	1	0	3	2	0	2	3	1	1
3	1	1	1	1	2	2	1	1	4	0	1	1	0	2	3
4	1	0	2	2	2	1	1	1	3	2	0	2	0	2	2
5	1	2	0	2	4	1	1	1	0	2	1	0	2	2	2
6	1	0	2	1	1	0	2	2	2	0	3	1	1	1	1
7	0	1	1	2	2	1	3	1	1	0	2	1	2	1	2
8	0	0	4	0	2	1	1	4	1	4	0	1	2	0	1
9	1	1	0	2	0	3	0	3	1	3	1	1	1	2	2
10	1	0	0	3	1	4	0	3	0	1	1	0	2	2	0
11	0	1	1	1	2	1	0	1	2	0	4	1	1	2	1
12	0	2	0	0	3	0	3	0	0	2	2	1	1	2	2
13	1	3	1	1	1	2	3	1	1	0	0	0	5	2	1
14	1	3	1	1	1	0	0	3	1	3	1	1	2	2	1
15	2	0	1	1	0	1	5	2	1	2	1	2	0	1	3
Remaining Vehicles number	14	9	8	6	1	4	3	1	3	3	4	11	0	3	0

n_{II}^* with P(Mut) = 0.01

Customer	Period														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	0	3	4	1	1	3	1	0	3	1
2	0	3	1	3	1	1	0	0	1	3	1	1	2	2	2
3	1	1	3	1	0	1	1	2	1	1	4	1	1	2	1
4	0	3	1	2	0	1	4	1	2	0	1	2	0	2	1
5	2	2	1	2	3	0	1	2	1	1	1	1	1	1	1
6	2	0	1	1	0	1	3	1	1	1	1	2	1	1	1
7	1	1	4	1	1	0	0	1	2	1	2	1	1	1	3
8	4	0	1	1	1	2	4	0	1	0	1	3	1	0	1
9	1	0	1	1	1	1	3	2	3	1	2	0	1	2	2
10	0	2	1	3	0	1	2	1	1	2	1	1	1	1	2
11	1	1	0	1	4	1	1	0	1	1	1	1	1	0	1
12	1	1	1	2	1	0	0	2	1	0	1	1	2	2	1
13	1	0	1	1	1	1	0	5	2	1	3	1	3	1	1
14	1	4	1	1	1	0	0	0	3	0	0	3	4	1	1
15	1	1	1	0	2	1	0	0	1	0	3	3	1	6	2
Remaining Vehicles number	8	5	6	4	8	14	3	4	3	12	0	3	5	0	4

n_{III}^* with $P(\text{Mut}) = 0.01$

Customer	Period														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	2	0	1	1	2	3	1	3	1	3	1	1	2	1
2	0	3	0	1	1	2	1	2	1	1	3	2	2	0	2
3	1	0	1	2	2	1	1	1	2	1	4	1	2	2	0
4	1	0	3	0	1	2	3	2	2	0	2	1	1	2	1
5	0	3	0	4	2	1	1	1	2	1	2	0	1	2	1
6	1	1	1	2	0	2	1	1	1	1	1	3	1	1	1
7	2	0	3	1	0	2	1	1	2	1	1	1	2	3	0
8	0	1	5	1	1	1	0	4	2	0	3	1	1	0	1
9	1	0	1	2	1	1	1	1	1	1	2	3	3	1	2
10	1	2	0	2	2	3	0	1	3	0	1	0	1	2	1
11	1	1	2	1	0	3	3	1	1	0	0	2	0	2	1
12	0	1	1	0	3	1	1	1	1	3	0	1	1	2	2
13	3	1	1	1	1	0	1	1	1	6	0	1	1	3	1
14	1	3	1	1	1	0	3	1	1	2	1	1	2	2	1
15	1	0	1	1	3	1	3	2	2	1	1	1	1	1	3
Remaining Vehicles number	12	7	5	5	6	3	2	4	0	6	1	6	5	0	7

n_{IV}^* with P(Mut) = 0.01

Customer	Period														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	2	2	1	1	2	0	4	1	1	2	2	2	1
2	0	3	2	1	1	1	1	0	3	0	1	3	1	3	1
3	1	1	1	1	1	2	2	1	2	2	3	0	2	2	0
4	1	0	2	1	3	2	1	2	2	1	0	2	1	1	2
5	1	2	2	0	1	2	0	1	2	5	1	0	1	2	1
6	0	2	1	0	0	2	0	2	1	3	3	1	2	1	0
7	2	1	1	0	2	2	1	1	1	1	1	3	2	1	1
8	0	2	4	1	2	1	1	0	1	1	3	4	0	0	1
9	1	1	2	0	2	0	3	1	2	3	1	0	2	1	2
10	2	1	0	2	1	0	2	2	4	0	0	1	0	0	4
11	1	1	0	0	3	4	1	1	0	0	2	1	1	1	2
12	2	0	1	1	1	0	1	1	2	0	3	1	1	2	2
13	0	1	1	4	2	2	3	0	1	2	2	0	1	2	1
14	1	0	3	1	2	0	4	1	0	3	0	1	1	3	1
15	0	0	3	3	1	0	3	2	0	1	3	1	1	2	2
Remaining Vehicles number	13	9	0	8	2	6	0	10	0	2	1	5	7	2	4

n_i^* with $P(\text{Mut}) = 0.005$

Customer	Period														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	1	3	0	1	1	3	3	1	3	0	2	1	2
2	1	1	0	4	2	1	0	1	2	2	1	0	4	1	1
3	0	2	1	0	3	3	1	2	1	0	2	0	1	4	1
4	0	1	2	1	3	1	0	2	0	5	1	1	1	3	0
5	1	1	3	1	2	3	0	1	1	1	1	0	1	3	2
6	0	2	1	1	2	0	1	2	2	1	1	2	1	1	1
7	0	4	1	1	1	1	1	1	2	0	1	1	2	1	3
8	1	1	3	0	4	0	1	0	4	2	2	0	2	0	1
9	1	0	0	2	1	1	2	3	1	3	0	2	2	1	2
10	2	1	0	3	1	2	0	2	3	0	0	0	0	1	4
11	1	1	0	2	3	1	1	2	1	1	0	1	1	1	2
12	1	1	1	1	1	0	2	1	0	0	2	3	1	2	2
13	1	0	3	2	1	4	1	0	0	1	4	1	1	1	2
14	1	2	1	2	0	3	1	2	1	1	0	0	4	2	1
15	2	1	2	0	1	2	0	2	0	0	3	4	1	2	1
Remaining Vehicles number	13	6	6	2	0	2	13	1	4	7	4	10	1	1	0

n_H^* with $P(\text{Mut}) = 0.005$

Customer	Period														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	0	1	1	0	2	3	2	1	3	1	1	1	1	3
2	2	1	1	2	1	0	0	1	1	0	3	1	3	3	1
3	0	2	1	1	1	2	1	3	1	1	1	1	4	1	1
4	1	1	0	1	0	1	1	7	1	2	1	0	1	1	0
5	1	5	1	0	3	1	0	1	3	2	0	0	0	1	3
6	2	1	1	1	1	1	1	1	2	1	1	1	1	2	1
7	3	0	1	1	1	1	1	1	3	1	1	1	2	1	2
8	1	1	3	3	1	1	1	1	2	1	3	1	0	1	1
9	3	1	0	1	1	1	2	1	1	2	2	0	2	1	1
10	0	1	1	4	1	3	0	0	3	1	0	0	1	1	2
11	3	1	1	3	1	2	1	1	0	0	0	1	1	1	1
12	1	3	0	1	0	1	1	1	2	2	0	1	1	1	3
13	1	3	1	1	1	4	0	0	2	2	1	2	1	1	2
14	0	0	3	2	3	2	1	0	1	1	1	2	4	2	0
15	0	0	2	0	1	3	2	0	0	3	4	2	1	2	2
Remaining Vehicles number	5	5	8	3	9	0	10	5	2	3	6	11	2	5	2

n_{III}^* with $P(\text{Mut}) = 0.005$

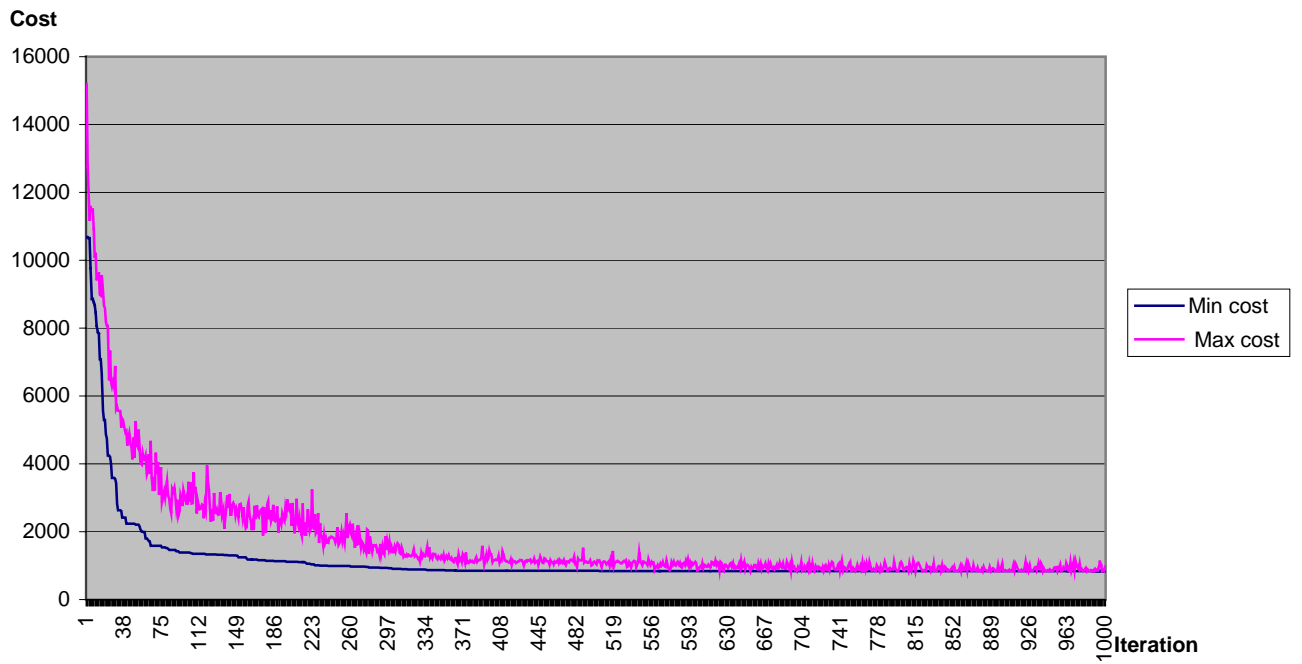
Customer	Period														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	2	1	1	2	0	3	2	2	0	4	1	1	2	1
2	0	1	3	1	1	1	1	2	1	1	1	1	5	1	1
3	2	0	1	1	1	1	3	3	1	2	1	1	1	2	1
4	1	1	1	1	2	3	1	1	1	2	1	1	4	0	1
5	1	2	1	3	2	2	0	2	0	2	1	1	1	2	1
6	1	1	1	1	3	0	2	0	2	1	2	1	2	1	0
7	2	0	2	0	3	1	1	1	2	1	2	1	1	2	1
8	1	1	4	0	1	1	1	4	1	1	3	1	1	0	1
9	1	1	2	0	3	1	1	1	2	1	3	0	1	1	3
10	1	1	1	1	3	2	1	1	2	1	1	0	0	1	3
11	1	0	1	2	3	1	1	1	1	1	1	2	0	2	1
12	1	1	1	1	1	1	1	1	1	2	1	0	1	3	2
13	0	0	1	1	0	1	2	1	2	8	0	1	3	1	1
14	0	0	1	1	0	1	6	1	0	2	3	2	1	3	0
15	0	0	0	6	0	0	1	1	7	0	1	0	2	1	2
Remaining Vehicles number	13	14	4	5	0	9	0	3	0	0	0	12	1	3	6

n_{IV}^* with P(Mut) = 0.005

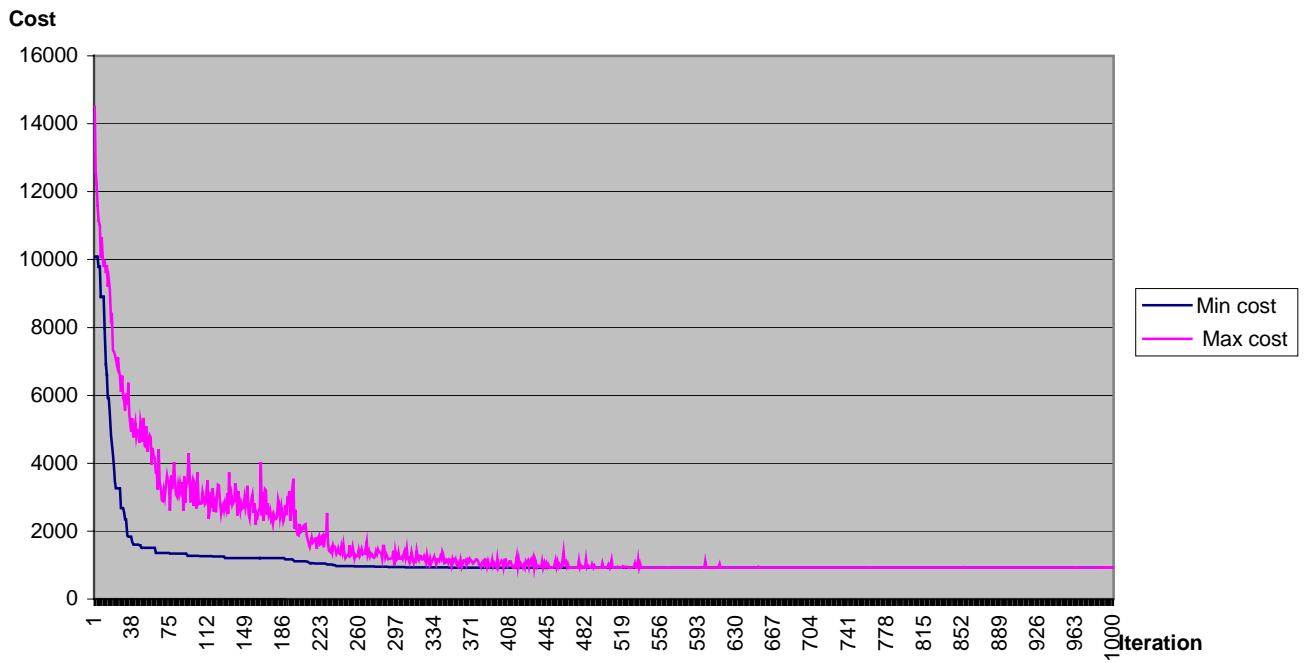
Iteration	Variant I		Variant II		Variant III		Variant IV	
	Min. Cost	Max. Cost	Min. Cost	Max. Cost	Min. Cost	Max. Cost	Min. Cost	Max. Cost
50	2205.7	4578.3	1507.5	4513.25	8435.45	9825.3	1108.5	1558.6
100	1385.55	2803.25	1269.75	3313.4	3827.25	4151.15	950.95	1414.85
150	1249.2	2408.3	1204.8	2897.95	2608.8	3055.55	908.3	1326.4
200	1110.75	2613.6	1108.05	1896.5	2075.15	2369.55	903.3	1253.3
250	986.05	1789.65	967.95	1272.65	1920	2109.25	903.1	1002.1
300	927.3	1560.5	943.75	1188.85	1595.7	1664.8	903.1	941.25
350	863.9	1215.1	926.25	1134.5	1586.9	1593.1	903.1	903.9
400	849	1155.75	926.25	1124.7	1568.4	1759.15	903.1	1085.65
450	849	1213.75	926.25	926.25	1567.4	1695.35	903.1	903.1
500	849	1086.1	926.25	926.25	1553.7	1564.9	903.1	903.1
550	839.2	1064.75	926.25	926.25	1553.7	1553.7	903.1	903.1
600	839.2	861.6	926.25	1109.65	1552.5	1552.5	903.1	903.1
650	839.2	1008.8	926.25	926.25	1552.5	1552.5	903.1	903.1
700	839.2	896.85	926.25	926.25	1552.5	1552.5	903.1	903.1
750	839.2	1111.9	926.25	926.25	1552.5	1552.5	903.1	903.1
800	839.2	1112.2	926.25	926.25	1552.5	1552.5	903.1	903.1
850	839.2	951.9	926.25	926.25	1552.5	1552.5	903.1	903.1
900	839.2	839.2	926.25	926.25	1552.5	1552.5	903.1	903.1
950	839.2	936.25	926.25	926.25	1552.5	1552.5	903.1	903.1
1000	839.2	963.1	926.25	926.25	1552.5	1552.5	903.1	903.1

Cost for 15 Customer, 15 Periods with $P(\text{Mut}) = 0.01$

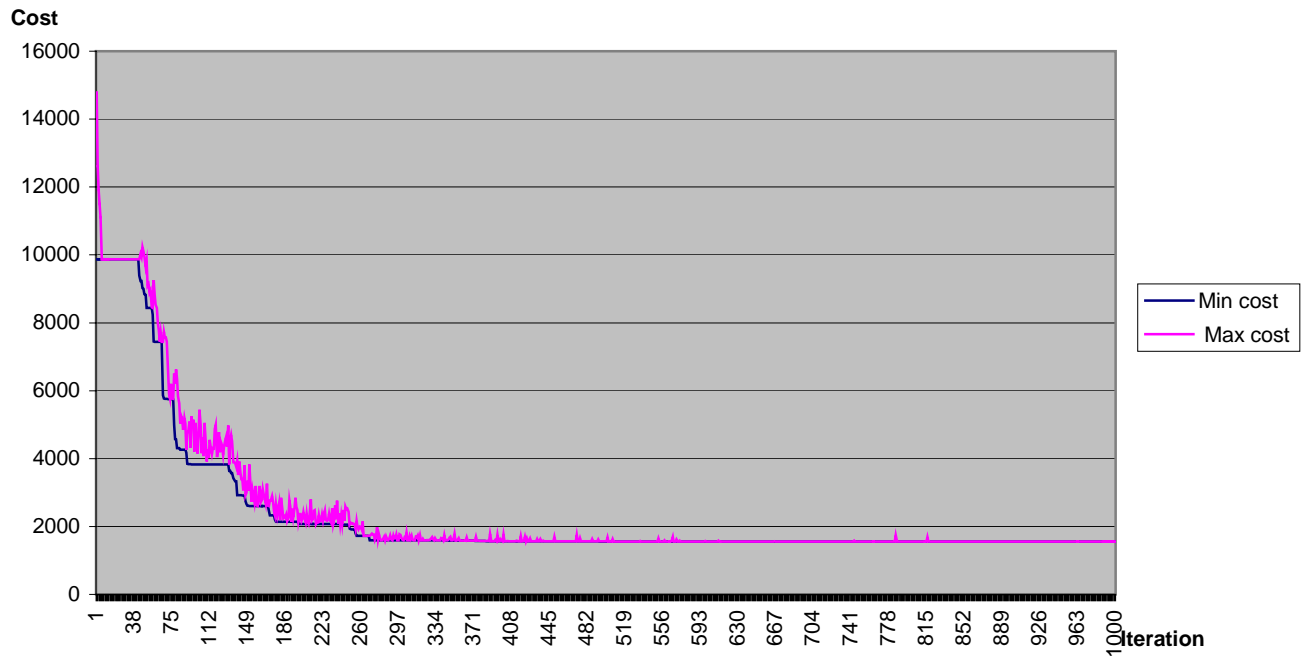
V1, P(Mut)=0.01



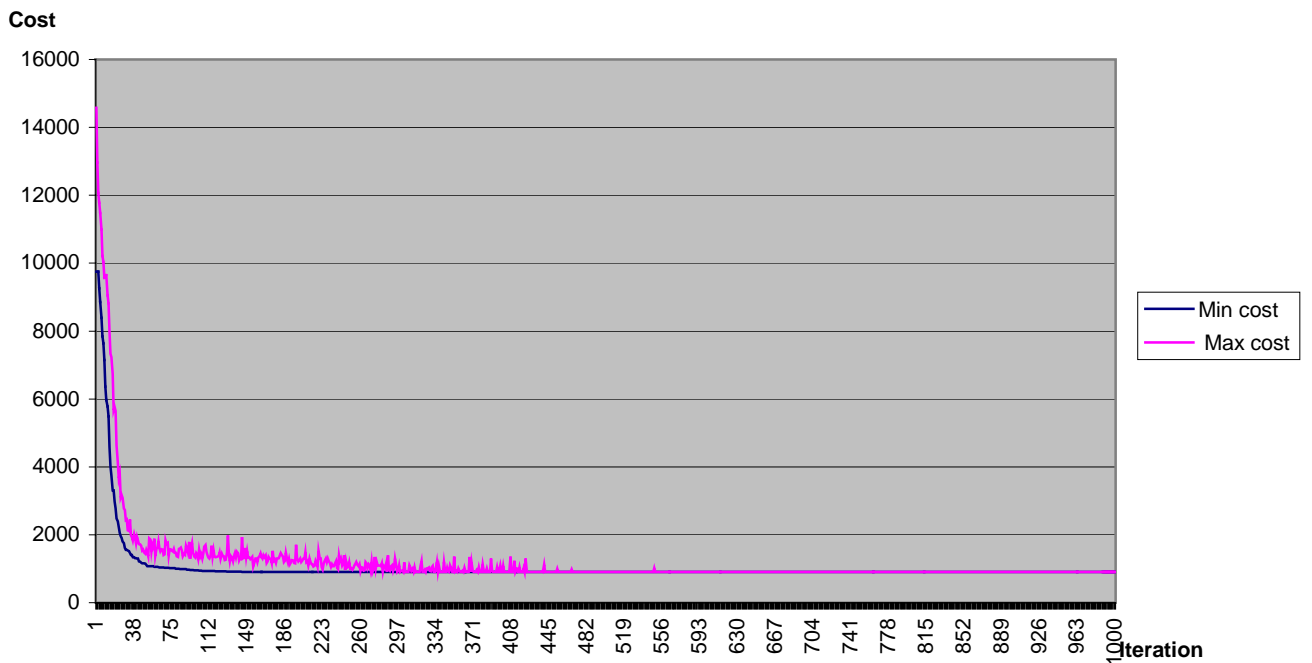
V2, P(Mut)=0.01



V3, P(Mut)=0.01



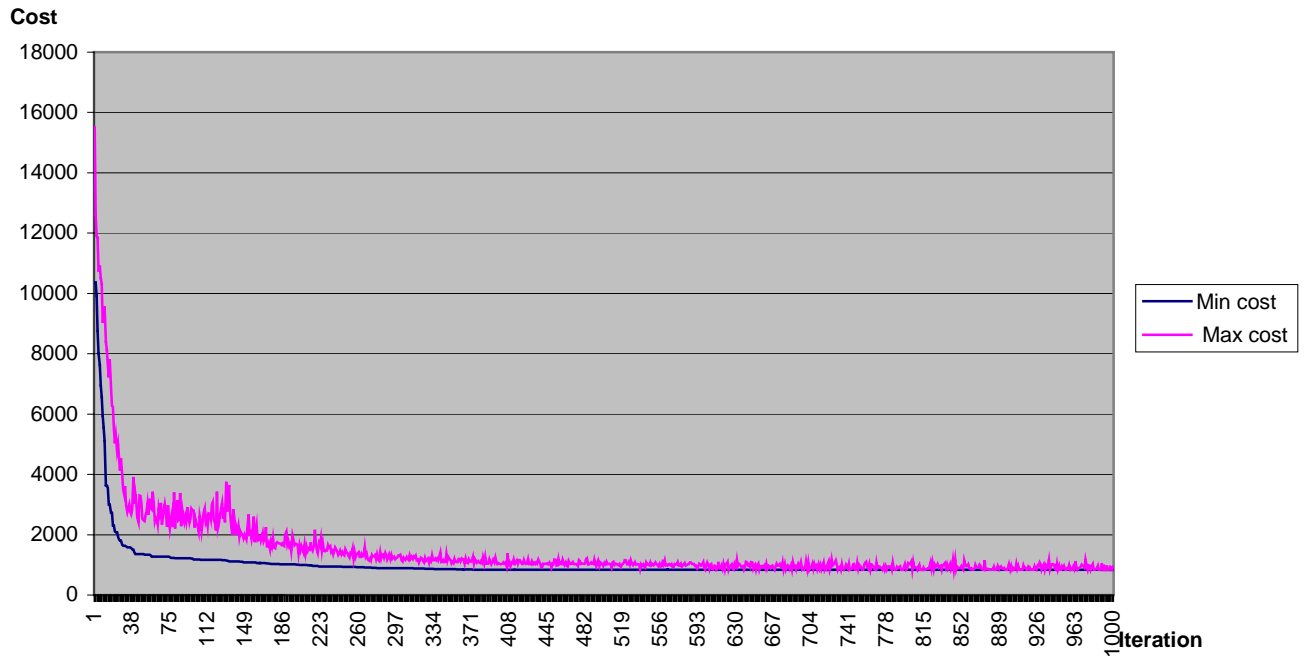
V4, P(Mut)=0.01



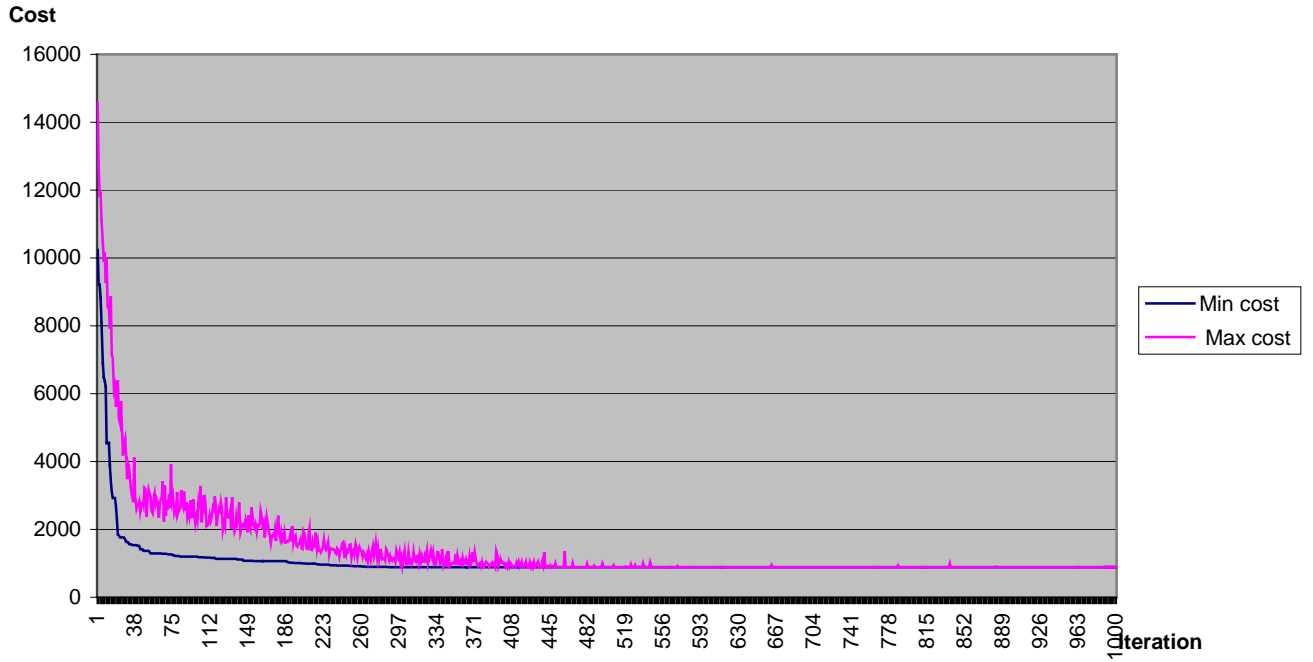
Iteration	Variant I		Variant II		Variant III		Variant IV	
	Min. Cost	Max. Cost	Min. Cost	Max. Cost	Min. Cost	Max. Cost	Min. Cost	Max. Cost
50	1341	2451.1	1363.9	2906.6	6215.15	6913.15	1784.65	2506.95
100	1172.95	2513.65	1177.55	2796.4	3341.2	4654.6	1119	1724.75
150	1086.2	1842.55	1072.6	2060.75	2986.3	2986.3	1012.9	1368.15
200	1013.35	1660.15	1013.4	1709.55	2672.75	2965.65	976.3	1313.3
250	934.25	1297.2	924.7	1191.75	2546.1	2709.35	955.4	955.4
300	887.35	1320.75	888.35	923.95	2478.85	2478.85	955.4	1125.1
350	856.55	1161.05	883.95	1006.1	2466.9	2467.6	955.4	955.4
400	843.3	1025.1	883.95	1006.9	2461.95	2466.9	955.4	955.4
450	843.3	1035.9	883.95	1000.85	2461.95	2461.95	955.4	955.4
500	843.3	964.2	883.95	883.95	2461.95	2461.95	955.4	955.4
550	843.3	1025.8	883.95	892.75	2419.05	2419.05	955.4	955.4
600	843.3	843.3	883.95	883.95	2412.5	2412.5	955.4	955.4
650	843.3	843.3	883.95	883.95	2412.5	2412.5	955.4	955.4
700	843.3	854.9	883.95	883.95	2412.5	2412.5	955.4	955.4
750	843.3	866.5	883.95	883.95	2412.5	2412.5	955.4	955.4
800	843.3	1039.2	883.95	883.95	2412.5	2412.5	955.4	955.4
850	843.3	891.7	883.95	883.95	2412.5	2412.5	955.4	955.4
900	843.3	965.3	883.95	883.95	2412.5	2412.5	955.4	955.4
950	843.3	952.95	883.95	883.95	2249.5	2249.5	955.4	955.4
1000	843.3	945.35	883.95	883.95	2249.5	2249.5	955.4	955.4

Cost for 15 Customer, 15 Periods with $P(\text{Mut}) = 0.005$

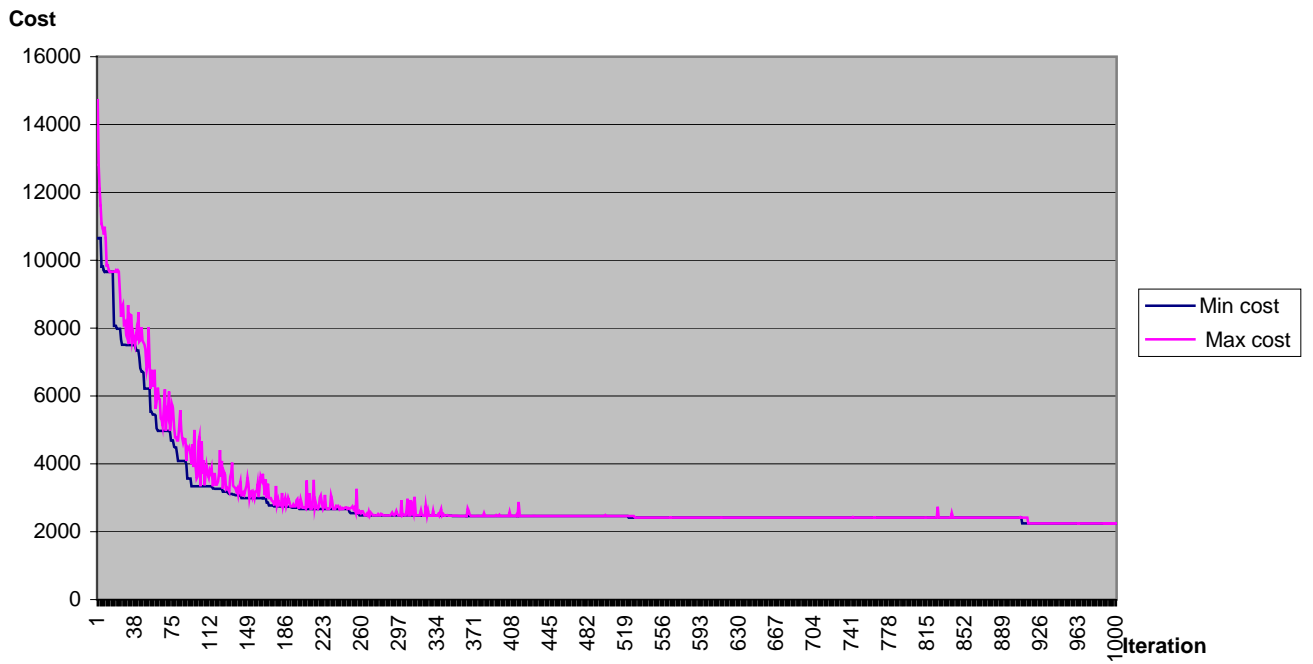
V1, P(Mut)=0.005



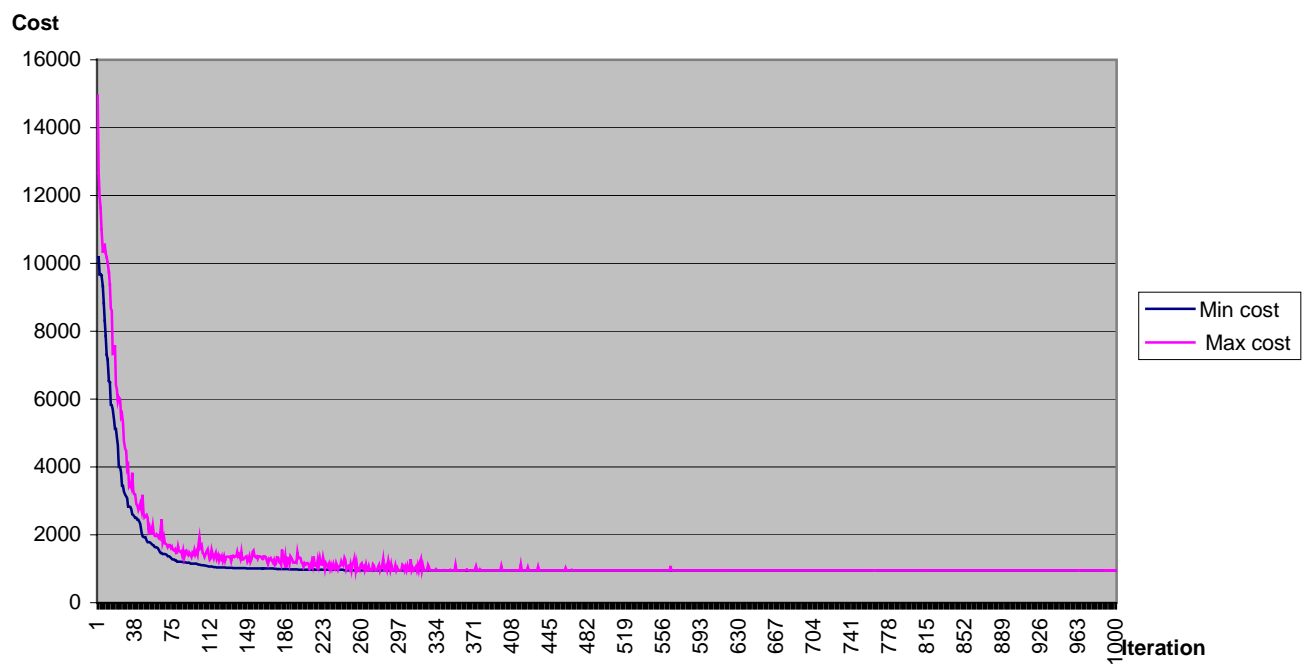
V2, P(Mut)=0.005



V3, P(Mut)=0.005



V4, P(Mut)=0.005



REFERENCES

- [1] Abdemaguid, T.F. and Dessouky, M. M. "Genetic Algorithm Approach to the Integrated Inventory-Distribution Problem," *International Journal of Production Research* , 44, 4445-4464 (2006).
- [2] Adler, N. "The Effect of Competition on The Choice of An Optimal Network in A Liberalized Aviation Market with An Application to Western Europe," *Transportation Science*, 39, 58–72 (2005).
- [3] Assad, A. A. "Modelling of Rail Networks: Toward a Routing / Makeup Model," *Transportation Research-B*, 14B, 101-114 (1980).
- [4] Baker, L. "Overview of Computer Based Models Application to Freight Car Utilization," *Report prepared for U.S. Department of Transportation, NTIS*, Springfield VA, U.S.A. (1977).
- [5] Beamon, B. M. and Chen, V. C. P. "Performability-Based Fleet Sizing in a Material Handling System," *International Journal of Advanced Manufacturing Technology*, 14, No. 6, 441-449 (1998).
- [6] Beaujon, G. J. and Turnquist, M. A. "A Model for Fleet Sizing and Vehicle Allocation," *Transportation Science*, 25, No. 1, 19-45 (1991).
- [7] Bendall, H.B. and Stent, A.F. "A Scheduling Model for a High Speed Containership Service: A Hub and Spoke Short-Sea Application," *International Journal of Maritime Economics* 3, 262–277 (2001).
- [8] Blazewicz, J.; Burkard, R.E.; Finke, G. and Woeginger, G.J. "Vehicle Scheduling in Two-Cycle Flexible Manufacturing Systems," *Mathematical and Computer Modeling*, 20, 19-31 (1994).

- [9] Blazewicz, J.; Eiselt, H.A.; Finke, G.; Laporte, G. and Weglarz, J. "Scheduling Tasks and Vehicles in a Flexible Manufacturing System," *International Journal of Flexible Manufacturing Systems*, 4, 5-16 (1991).
- [10] Campbell, J. "Hub Location and the P-Hub Median Problem," *Operational Research*, 44, 923-935 (1996).
- [11] Campbell, J. "Integer Programming Formulations of Discrete Hub Location Problems," *European Journal of Operational Research* 72, 387-405 (1994).
- [12] Campbell, J. "One-to-Many Distribution with Transshipments: An Analytic Model," *Transportation Science*, 27, No. 4, 330-340 (1993).
- [13] Campbell, J.; Ernst, A. and Krishnamoorthy, M. "Hub location problems," *Facility Location: Application and Theory*, Springer-Verlag, Berlin, 373-408 (2002).
- [14] Ceric, V. "Simulation Study of an Automated Guided-Vehicle System in a Yugoslav Hospital," *Journal of the Operational Research Society*, 41, No. 4, 299-310 (1990).
- [15] Cheung, R.K. and Chen, C. "A Two-Stage Stochastic Network Model and Solution Methods for the Dynamic Empty Container Allocation Problem," *Transportation Science* 32, 142-162 (1998).
- [16] Cheung, R. K. and Powell, W. B. "An Algorithm for Multistage Dynamic Networks with Random Arc Capacities, with an Application to Dynamic Fleet Management," *Oper. Res.*, 44, no. 6, 951-963, (1996).
- [17] Chien, T. W.; Balakrishnan, A. and Wong, R. T. "An Integrated Inventory Allocation and Vehicle Routing Problem," *Transportation Science*, 23, No. 2, 67-76 (1989).
- [18] Choong, S.T.; Cole, M.H. and Kutanoglu, E. "Empty Container Management for Intermodal Transportation Networks," *Transportation Research E* 38, 423-438 (2002).

- [19] Christiansen, M.; Fagerholt, K. and Ronen, D. “Ship Routing and Scheduling: Status and Perspective,” *Transportation Science* 38, no. 1, 1–18 (2004).
- [20] Coley, D. A. “ An Introduction to Genetic Algorithms for Scientists and Engineers,” *World Scientific Publishing Co. Pte. Ltd.* (1999).
- [21] Crainic, T.G. “ Intercity Multimode Freight Transportation: The State-of-the-Art,” *The logistics and Transportation Review*, 23, 189-206 (1987).
- [22] Crainic, T.G. “Service Network Design in Freight Transportation,” *European Journal of Operations Research* 122, 272-288 (2000).
- [23] Crainic, T.; Ferland, J. and Rousseau, J. “A Tactical Planning Model for Rail Freight Transportation,” *Transportation Science*, 18, 165-184 (1984).
- [24] Crainic, T.G.; Gendreau, M. and Dejax, P. “Dynamic and Stochastic-Models for the Allocation of Empty Containers,” *Operations Research* 41, 102–126 (1993).
- [25] Crainic, T. and Rousseau, J. “Multicommodity, Multimode Freight Transportation: A General Modelling and Algorithmic Framework for the Service Network Design Problem,” *Transportation Research-B*, 20B, 225-242 (1986).
- [26] Crino, M. J.; Moore, J. T.; Barnes, J. W and Nanry, C. W. “Solving the Theater Distribution Vehicle Routing and Scheduling Problem Using Group Theoretic Tabu Search,” *Defence Transportation: Algorithms, Models, and Applications For The 21st Century*, pp. 599-616, published by Elsevier Science and Technology Books (2004).
- [27] Daganzo, C. F. “Logistics Systems Analysis,” *Springer*, (1999).
- [28] Dejax, P. J. “Methodology for Facility Location and Distribution Systems Planning,” *Technical Report LEIS No. 86-04*, Ecole Centrale de Paris, France (1986).

- [29] Du, Y. and Hall, R. "Fleet Sizing and Empty Equipment Redistribution for Center-Terminal Transportation Networks," *Management Science*, 43, 145-157 (1997).
- [30] Dyer, M.E. and Proll, L.G. "On the Validity of Marginal Analysis for Allocating Servers in M/M/c Queues, » *Management Science*, 23, No. 9, 1019-1022 (1977).
- [31] Egbelu, P. J. "Concurrent Specification of Unit Load Sizes and Automated Guided Vehicle Fleet Size in Manufacturing System," *International Journal of Production Economics*, 29, No. 1, 49-64 (1993).
- [32] El-Ashry, M. ; Köchel, P. and Schüler, S. "Solutions for the Fleet Sizing and Allocation Problem in Hub-and-Spoke Systems," *Chemnitzer Informatik - Berichte CSR-06-09*, TU Chemnitz, Germany (2006).
- [33] Fagerholt, K. "Optimal Fleet Design in a Ship Routing Problem," *International Transactions in Operational Research* 6, no. 5, 453–464 (1999).
- [34] Federgruen, A. and Zipkin, P. "Combined Vehicle Routing and Inventory Allocation Problem," *Operations Research*, 32, 1019-1037 (1984).
- [35] Fleming, P. J. and Zalzal, A. M. S. "Genetic Algorithms in Engineering Systems," *Institution of Engineering and Technology (IET)*, (1997).
- [36] Fox, B. "Discrete Optimization via Marginal Analysis," *Management Science*, 13, No. 3, 210-216 (1966).
- [37] Friesz, T. L.; Tobin, R. L. and Harker, P. T. " Predictive Intercity Freight Network Models: The State-of-the-Art," *Trans. Res.* 17 A, 409-417(1983).
- [38] Fu, M.C. "Optimization via Simulation: A Review," *Ann. of Oper. Res.*, 42, 199-247 (1994).

- [39] Gambardella, L. M.; Taillard, E. and Agazzi, G. "MACSVRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows," *In D. Corne, M. Dorigo and F. Glover, editors, New Ideas in Optimization. McGraw-Hill, London, UK, 63-76 (1999).*
- [40] Gelenbe, E. and Pujolle, G. "Introduction to Queueing Networks," *John Wiley & Sons Ltd (1998).*
- [41] Gen, M., and Cheng, R. "Genetic Algorithms and Engineering Design," *New York: Wiley (1997).*
- [42] Geoffrion, A. M. and Powers, R. F. "Facility Location Analysis Is Just the Beginning (if You Do It Right)" *Interfaces, 10, 22-30 (1980).*
- [43] Godfrey, G. A. and Powell, W. B. "An Adaptive, Dynamic Programming Algorithm for Stochastic Resource Allocation Problems I: Single-period Travel Times," *Transportation Sci. 36, No. 1, 21-39, (2002 a).*
- [44] Godfrey, G. A. and Powell, W. B. "An Adaptive, Dynamic Programming Algorithm for Stochastic Resource Allocation Problems II: Multi-period Travel Times," *Transportation Sci. 36, No. 1, 40-54, (2002 b).*
- [45] Gohring, K. W.; Spraker, T. W.; Lefstead, P. M. and Harvey, A.E. "Norfolk Southern's Empty Freight Car Distribution System Using Goal Programming," *paper presented at the ORSA/TIMS Joint National Meeting, May (1993).*
- [46] Goldberg, D. E. "Genetic Algorithms in Search, Optimization and Machine Learning," *New York: Addison-Wesley (1989).*
- [47] Grunow, M.; Günther, H. and Lehmann, M. "Dispatching Multi-load AGVs in Highly Automated Seaport Container Terminals," *OR Spectrum, 26, 211- 235 (2004).*

- [48] Hall, N.G.; Sriskandarajah, C. and Ganesharajah, T. “Operational Decisions in AGV-Served Flowshop Loops: Scheduling,” *Annals of Operations Research* 107, 161-188 (2001).
- [49] Hall, N.G.; Sriskandarajah, C. and Ganesharajah, T. “Operational Decisions in AGV-Served Flowshop Loops: Fleet Sizing and Decomposition,” *Annals of Operations Research* 107, 189-209 (2001).
- [50] Harel, A. “Convexity Properties of the Erlang Loss Formula,” *Operations Research* 38, 3, 499–505 (1990).
- [51] Harker, P. T. “The State-of-the-Art in the Predictive Analysis of Freight Transportation Systems,” *Trans. Rev.*, 5, 143-164 (1985).
- [52] Herer, Y. T. and Levy, R. “The Metered Inventory Routing Problem, an Integrative Heuristic Algorithm,” *International Journal of Production Economics*, 51, 69-81 (1997).
- [53] Holmberg, K.; Joborn, M. and Lundgren, J. T. “A Model for Allocation of Empty Freight Cars with Capacity Restrictions and Fixed Costs,” *In Computers in Railway Operations* (1994).
- [54] Imai, A. and Rivera, F. “Strategic Fleet Size Planning for Maritime Refrigerated Containers,” *Maritime Policy and Management* 28, 4, 361–374 (2001).
- [55] Johnson, M. E. and Brandeau, M. L. “An Analytic Model for Design of a Multivehicle Automated Guided Vehicle System,” *Management Science*, 39, No. 12, 1477-1489 (1993).
- [56] Kasilingam, R. G. “Logistics and Transportation: Design and Planning,” *Springer*, (1998).

- [57] Kasilingam, R. G. and Gobal, S. L. "Vehicle Requirements Model for Automated Guided Vehicle Systems," *International Journal of Advanced Manufacturing Technology*, 12, 276-279 (1996).
- [58] Kämpf, M. Projektseite > CAOS > hauptseite. Internet <http://www-user.tu-chemnitz.de/mkae/de/Projects/CAOS/index.php>, Stand 12:44, 27. Sep 2006, November (2004).
- [59] Kämpf, M. Softwaresystem zur Modellierung, Simulation und Optimierung Ökonomischer Problemstellungen. Internet <http://www-user.tu-chemnitz.de/mkae/Publications/WorkingPapers/CSR/CSR.pdf>, Stand 12:44, 27. Sep 2006, April (2005).
- [60] Keaton, M. "Designing Optimal Railroad Operating Plans: Lagrangian Relaxation and Heuristic Approaches," *Transportation Research-B*, 23B, 415-431 (1989).
- [61] Keaton, M. "Designing Railroad Operating Plans: A Dual Adjustment Method for Implementing Lagrangian Relaxation," *Transportation Science*, 26, 263-279 (1992).
- [62] Kindervater, G. A., Savelsbergh, M. W. "Vehicle Routing: Handling Edge Exchanges," *E. H. Aarts, J. K. Lenstra, eds., Local Search in Combinatorial Optimization. John Wiley & Sons, Chichester, UK*, 311–336 (1997).
- [63] Köchel, P. "An Approximate Solution for the Fleet Sizing and Allocation Problem," *Promet-Traffic-Traffico*, 9, 195-200 (1997).
- [64] Köchel, P. "Solving Logistic Problems Through Simulation and Evolution," *Proceedings of the 7th International Symposium on Operational Research SOR'03*, Podèetrtek, Slovenia, Sept 24-26, 9-15 (2003).
- [65] Köchel, P. "Some Considerations on Optimization in Logistic Systems," *Proceedings of the 8th Intern. Symposium on Operational Research SOR'05*, Nova Gorica, Slovenia, Sept. 28-30, 219-226 (2005).

- [66] Köchel, P.; Kunze, S. and Nieländer, U. "Optimal Control of a Distributed Service System with Moving Resources: Application to the Fleet Sizing and Allocation Problem," *International Journal of Production Economics*, 81-82, 443-459 (2003)
- [67] Koenigsberg, E. and Lam, R.C. "Cyclic Queue Models of Fleet Operations," *Oper. Res.*, 24, 516-529 (1976).
- [68] Koo, P. H.; Lee, W. S. and Jang, D. W. "Fleet Sizing and Vehicle Routing for Container Transportation in a Static Environment," *OR Spectrum*, 26, 193-209 (2004).
- [69] Kraft, E. R. "Demand Variability, Service Reliability and Railway Freight Research Needs," *Department of Operations and Information Management*, The Wharton School (1994).
- [70] Lei, L.; Armstrong, R. and Gu, S. "Minimizing the Fleet Size with Dependent Time-Window and Single-Track Constraints," *Operations Research Letters*, 14, 91-98 (1993).
- [71] Li, J.-A.; Liu, K.; Leung, S.C.H. and Lai, K.K. "Empty Container Management in a Port with Long-run Average Criterion," *Mathematical and Computer Modelling* 40, 85-100 (2004).
- [72] Mahadevan, B. and Narendran, T. T. "Estimation of Number of AGVs for an FMS: An Analytical Model," *International Journal of Production Research*, 31, No. 7, 1655-1670 (1993).
- [73] Mahadevan, B. and Narendran, T. T. "A Hybrid Modeling Approach to the Design of an AGV-Based Material Handling System for an FMS," *International Journal of Production Research*, 32, No. 9, 2015-2030 (1994).
- [74] Maxwell, W. L. and Muckstadt, J. A. "Design of Automatic Guided Vehicle Systems," *IIE Transactions*, 14, No. 2, 114-124 (1982).

- [75] Mendiratta, V. B. "A Dynamic Optimization Model of the Empty Car Distribution Process," *Ph.D. dissertation, Department of Civil Engineering, Northwestern University, Evanston, III, (1981).*
- [76] Messerli, E.J. "Proof of a Convexity Property of the Erlang-B Formula," *The Bell System Technical Journal* 51, 951–953 (1972).
- [77] Newman, A. M. and Yano, C. A. "Centralized and Decentralized Train Scheduling for Intermodal Operations," *IIE Transactions*, 32, No. 8, 743-754 (2000).
- [78] Newton, D. "Simulation Model Calculates How Many Automated Guided Vehicles Are Needed," *Industrial Engineering*, 68-78 (1985).
- [79] O'Kelly, M. and Bryan, D. "Hub Location with Flow Economics of Scale," *Transportation Research-B*, 32, 605-616 (1998).
- [80] O'Kelly, M.; Skorin-Kapov, D. and Skorin-Kapov, J. "Lower Bounds for the Hub Location Problem," *Management Science*, 41, 713-721 (1995).
- [81] Pirkul, H. and Schilling, D. "An Efficient Procedure for Designing Single Allocation Hub and Spoke Systems," *Management Science*, 44, S235-S242 (1998).
- [82] Powell, W. B. and Carvalho, T.A. "Dynamic Control of Logistics Queueing Networks for Large-scale Fleet Management," *Transport. Sci.*, vol. 32, 90–109, (1998).
- [83] Powell, W. B. and Carvalho, T.A. "Real-Time Optimization of Containers and Flatcars for Intermodal Operations," *Technical paper SOR-96-05, Department of Civil Engineering and Operations research, Princeton University, (1996).*
- [84] Powell, W. B.; Ruszczyński, A. and Topaloglu, H. "Learning Algorithms for Separable Approximations of Stochastic Optimization Problem," *Mathematics of Operations Research*, 29, no. 4, 814-836 (2004).

- [85] Powell, W. B. and Topaloglu, H. “Fleet Management in Applications of Stochastic Programming,” *Math Programming Society – SIAM Series in Optimization*, Philadelphia, 185-216 (2005).
- [86] Rana, K. and Vickson, R.G. “Routing Container Ships Using Lagrangean Relaxation and Decomposition,” *Transportation Science* 25, no. 3, 201–214 (1991).
- [87] Ronen, D. “Cargo Ships Routing and Scheduling: Survey of Models and Problems,” *European Journal of Operations Research* 12, 119–126 (1983).
- [88] Ronen, D. “Ship Scheduling: The Last Decade,” *European Journal of Operations Research* 71, 325–333 (1993).
- [89] Sarmiento, A. M. and Nagi, R. “A Review of Integrated Analysis of Production-Distribution Systems,” *IIE Transactions*, 31, No. 11, 1061-1074 (1999).
- [90] Shanthikumar, J.G. and Sargent, R.G. “Unifying View of Hybrid Simulation/Analytic Models and Modeling,” *Operations Research*, 31, No. 6, 1030–1052 (1983).
- [91] Shanthikumar, J.G. and Yao, D.D. “Optimal Server Allocation in a System of Multi-Server Stations,” *Management Science*, 33, 1173-1180 (1987).
- [92] Shaw, P. “Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems,” *Proceedings of the Fourth International Conference on Principles and Practice of Constraint Programming (CP '98)*, Springer-Verlag, 417-431 (1998).
- [93] Skorin-Kapov, D.; Skorin-Kapov, J. and O’Kelly, M. “Tight Linear Programming Relaxations of Uncapacitated P-Hub Median Problems,” *European Journal of Operational Research*, 94, 582-593 (1996).
- [94] Suh, S. D. and Lee, S. “Formulation and Application of Empty Freight Management Model to Korean Railway,” *Working Report*, Hanyang University, Korea (2001).

- [95] Taylor, G.D.; Harit, S. and English, J.R. "Hub and Spoke Networks in Truckload Trucking: Configuration and Operational Concerns," *Logistics and Transportation Review* (1994).
- [96] Topaloglu, H. and Powell, W. B. "Dynamic Programming Approximations for Stochastic, Time-staged Integer Multicommodity Flow Problems," *INFORMS J. Comput.* 18, No.1, 31–42 (2006).
- [97] Trivedi, K.S. "Probability & Statistics with Reliability, Queueing and Computer Science Applications," *Prentice-Hall, 1st Edition* (1982).
- [98] Trudeau, P. and Dror, M. "Stochastic Inventory Routing: Route Design with Stockouts and Route Failures," *Transportation Science*, 26, 171-184 (1992).
- [99] Viswanathan, S. and Mathur, K. "Integrating Routing and Inventory Decisions in One Warehouse Multicustomer Multiproduct Distribution Systems," *Management Science*, 43, 294-312 (1997).
- [100] Vu Tung, D. and Pinnoi, A. "Vehicle Routing-Scheduling for Waste Collection in Hanoi," *European Journal of Operational Research*, 125, 449-468 (2000).
- [101] Waldinger, P.W.; Pallman, Schlaepfer, F. and Plesske, H. "Rechnergesteuerte Leerwagenverteilung (LWV)," *Die Bundesbahn, Germany*, No. 4, 389-431 (1991).
- [102] Weber, R.R. "On the Marginal Benefit of Adding Servers to G/GI/m Queues," *Management Science*, 26, No. 9, 946-951 (1980).
- [103] Wu, P.; Hartman, J. C. and Wilson, G. R. "Fleet Sizing in the Truck Rental Industry," *Working Report No. 98W-009*, Lehigh University (1999).

Thesen

1. Transportation is one of the most vital services in modern society. It makes most of the other functions of society possible, such as manufacturing and construction, food and agriculture, energy supply and distribution, safety and security, access to medical care, and tourism and recreation. At present it is more and more necessary to design transportation systems that are reliable, efficient, safe and environmentally sustainable.
2. Real transportation systems are so large and complex that in order to build the science of transportation systems it will be necessary to work on fundamental research issues in many areas, including: Modeling, Optimization and Simulation.
3. We are interested in solutions for the so-called fleet-sizing-and-allocation problem (FSAP). Fleet sizing and allocation problems are one of the most interesting and hard to solve logistic problems. A fleet sizing and allocation problem consists of two interdependent parts. The fleet sizing problem is to determine a number of transportation units that optimally balances service requirements against the cost of purchasing and maintaining the transportation units. The allocation problem is dealing with the repositioning of transportation units to serve future transportation demand. These two problems are highly related to each other.
4. To make the fleet sizing and allocation problem a little bit more tractable we concentrate on logistic systems with a special hub-and-spoke structure. We begin with a brief discussion regarding the feasibility of such systems. We discuss the major configuration issues and operational concerns associated with the use of hub-and-spoke transportation networks.
5. We are going to go through a very simple fleet sizing of one-to-one case. This case will cause us to focus attention on several key issues in fleet sizing. Afterwards are the notations and concepts introduced in this case extended in order to provide us with a framework, which will be used for discussion of related research.

6. A first generalization of the one-to-one system is the one-to-many system (a hub –and–spoke system with transportation only from hub to spokes). As a simple example can serve the continuous time situation where a single origin delivers items to many destinations. For the case that items are produced in a deterministic production cycle and transportation times are stochastic. We consider the problem of determining necessary size of a fleet with single type of transportation units (TU’s) to fulfill a given service rate.
7. We also studied a hub-and-spoke problem with continuous time and stochastic demand. To solve this problem, based on Marginal Analysis, we applied queueing theory methods.
8. The investigation of the fleet-sizing-**and**-allocation problem for hub-and-spoke systems is started for a single-period, deterministic-demand model. In that the model hub has to decide how to use a given number of TU’s to satisfy a known (deterministic) demand in the spokes.
9. We consider two cases:
 - i. Renting of additional TU’s from outside the system is not possible, and
 - ii. Renting of additional TU’s from outside the system is possible.For each case, based on Marginal Analysis, we developed a simple algorithm, which gives us the cost-minimal allocation. Examples show that the optimal solutions for both two cases above can be very different with respect to the optimal allocation as well as the corresponding cost values.
10. An approximate solution for the multi-period, deterministic demand model can be determined by applying the algorithm for the single-period, deterministic demand in a successive way. Since the multi-period, deterministic demand problem is NP-hard we suggest to use Genetic Algorithms. Some building elements for these are described.
11. For the most general situation (e.g., infinite planning horizon with continuous time, stochastic demand, stochastic transportation times) we suggest to use simulation optimization. The simulation optimization approach is divided into two parts, simulation

and optimization. Simulation is used to assess a given solution and optimization will provide new solutions.

12. To realize the simulation optimization approach we could use the software tool “Calculation Assessment Optimization System” (CAOS) developed at the professorship “Modelling and Simulation” at the Chemnitz University of Technology.

13. The idea of CAOS is to provide a software system, which separates the optimization process from the optimization problem. To solve an optimization problem the user of CAOS has to build up a model of the system to which the problem is related. Furthermore he has to define the decision parameters and their domain.

14. Finally, we used CAOS for two classes of hub-and-spoke system:

(i) A single hub with four spokes, and

(ii) A single hub with fifty spokes.

We applied four optimizers – a Genetic Algorithm, Tabu Search, Hybrid Parallel and Hybrid Serial with two distributions (Normal Distribution and Exponential Distribution) for a customer interarrival times and their demand.

15. From the results of the experiment we can see that:

(i) In the case single hub-four spokes, the two hybrid optimizers and Tabu search work well but we can say that the Tabu search is the best for this case.

(ii) In the case single hub-fifty spokes, the hybrid serial optimizer performs best. This suggests that for complex systems hybrid optimizers outperform non-hybrid ones.

(iii) The problem of dimension, i.e., with increase number of spokes the computation time will increase considerably (from 2500 investigated solutions for four spokes up to about 18000 for fifty spokes).

(iv) Finally, we can say that the suggested simulation optimization approach works well.