DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

AN INTEGRATED INVENTORY-ROUTING SYSTEM WITH LIMITED VEHICLE CAPACITY AND STORAGE CONSTRAINT

by
Neşe ERKURT

October, 2009

İZMİR

AN INTEGRATED INVENTORY-ROUTING SYSTEM WITH LIMITED VEHICLE CAPACITY AND STORAGE CONSTRAINT

A Thesis Submitted to the

Graduate School of Natural and Applied Sciences of Dokuz Eylül University In Partial Fulfillment of the Requirements for the Degree of Master of Science in Industrial Engineering

by
Neşe ERKURT

October, 2009

İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **AN INTEGRATED INVENTORY-ROUTING SYSTEM WITH LIMITED VEHICLE CAPACITY AND STORAGE CONSTRAINT** completed by **NEŞE ERKURT** under supervision of **YRD.DOÇ. DR. ÖZCAN KILINÇÇI** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Yrd.Doç.Dr. Özcan KILINÇÇI

Supervisor

Yrd.Doç.Dr. Umay KOÇER

(Jury Member)

Öğr.Gör. Dr. Özgür ESKİ

(Jury Member)

Prof.Dr. Cahit HELVACI
Director
Graduate School of Natural and Applied Science

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to Yrd.Doç.Dr. Özcan KILINÇÇI for serving as my thesis advisor, and providing his advice, nsupport, encouragement during this search. He has patiently read and commented on my thesis drafts

I would also like to thank to my mother, to my father,mto my sister, to my brother, to my mother in-law, to my husband's sister. I have been awed by the substantial support and guidance you have provided through this long and sometimes painful process.

I dedicate this thesis to my husband İhsan. I thank you so much for your help, patience and understanding throughout these stressful and long months.

AN INTEGRATED INVENTORY-ROUTING SYSTEM WITH LIMITED VEHICLE CAPACITY AND STORAGE CONSTRAINT

ABSTRACT

This study is concerned with the inventory routing problem with limited vehicle capacity and storage constraint.. The inventory routing problem attempts to coordinate inventory management and vehicle routing in such a way that the cost is minimized over the long run.

A mathematical model for coordinating inventory and vehicle routing decisions in an inbound commodity collection system composed of a central warehouse and suppliers is presented. A heuristic is developed due to difficulties on solving large problems with LINGO. Computational results which are obtained from the heuristic are compared with solution of LINGO for small instances. Computational tests are performed on a set of randomly generated problem instances. Computational results are obtained in short time for large and complex models with developed heuristic.

Keywords: Inventory Routing Problem, Heuristic for inventory routing, mathematical model

LİMİTLİ ARAÇ KAPASİTESİ VE DEPO ALANI KISITLARI İLE BÜTÜNLEŞİK ENVANTER YÖNETİMİ VE GÜZERGAH BELİRLEME PROBLEMİ

ÖZ

Bu çalışmada limitli araç kapasitesi ve depo alanı kısıtı bulunan bütünleşik envanter yönetimi ve güzergah belirleme problemi incelenmektedir.Tedarik zinciri kapsamında ortaya çıkan bu problemde envanter yönetimi ve güzergah belirleme kararlarının birlikte değerlendirilmesiyle özellikle maliyet avantajı sağlanması planlanmaktadır.

Bir merkezi depo ve tedarikçilerden oluşan bir sistemde envanter yönetimi ve güzergah belirleme kararı için matematiksel model oluşturulmuştur. Büyük problemleri LINGO ile çözmekteki zorluktan dolayı bir höristik geliştirilmiştir. Küçük örnekler için höristik ile elde edilen sonuçlar LINGO'dan elde edilen sonuçlar ile karşılaştırılmıştır. Hesaplamalar rastgele alınan problem örnekleri için yapılmıştır. Höristik ile kısa zamanda büyük ve karmaşık modeller için sonuç elde edilmiştir.

Anahtar sözcükler: Bütünleşik Envanter Yönetim ve Güzergah Belirleme Problemi, Bütünleşik Envanter Yönetimi ve Güzergah Belirleme Problemi için Höristik, Matematiksel Model

CONTENTS

Page

CHAPTER ONE
INTRODUCTION

The integration of production, distribution and inventory management is one of the challenges of today's competitive environment. In the last decade the importance of the relations between internal management and external environment has been widely recognized, and the expression "supply chain management", which emphasizes the view of the company as part of the supply chain, has become of common use Sometimes the expression "coordinated supply chain management" is used to emphasize the coordination among the different components of the supply chain. The availability of data and information tools, which derive from the advances in technology and communication systems, has created the conditions for the coordination inside the supply chain (Bertazzi et al, 2005).

The availability of new information technologies has also led in the last years to the development of new forms of relationships in the supply chain. One of these is the so called Vendor–Managed -Inventory (VMI), in which the supplier monitors the inventory of each retailer and decides the replenishment policy of each retailer. This contrasts with conventional inventory management, in which customers monitor their own inventory levels and place orders when they think that it is the appropriate time to reorder. VMI has several advantages over conventional inventory management. Vendors can usually obtain a uniform utilization of production resources, which leads to reduced production and inventory holding costs. Similarly, vendors can often obtain a uniform utilization of transportation resources, which in turn leads to reduced transportation costs. Furthermore, additional savings in transportation costs may be obtained by increasing the use of low-cost full-truckload shipments and decreasing the use of high-cost less-than-truckload shipments, and by using more efficient routes by coordinating the replenishment at customers close to each other (Kleywegt et al, 2002).

VMI also has advantages for customers. Service levels may increase, measured in terms of reliability of product availability, due to the fact that vendors can use the information that they collect on the inventory levels at the customers to better anticipate future demand, and to proactively smooth peaks in the demand. In addition, customers do not have to devote as many resources to monitoring their inventory levels and placing orders, as long as the vendor is successful in earning and maintaining the trust of the customers.

A first requirement for a successful implementation of VMI is that a vendor is able to obtain relevant and accurate information in a timely and efficient way. One of the reasons for the increased popularity of VMI is the increase in the availability of affordable and reliable equipment to collect and transmit the necessary data between the customers and the vendor. However, access to the relevant information is only one requirement. A vendor should also be able to use the increased amount of information to make good decisions. In fact, it is a very complicated task, as the decision problems involved are very hard (Kleywegt et al, 2002).

In VMI, the supplier monitors the inventory at the customers. This is made possible with modern equipment that can both measure the inventory at the customers and communicate with the supplier’s computer. The rapidly decreasing cost of this technology has probably made a significant contribution to the increasing popularity and success of VMI. The supplier is responsible for maintaining a desirable inventory level at each customer, and decides which customers should be replenished at which times, and with how much product. To make these decisions, the supplier has the benefit of access to a lot of relevant information, such as the current (and possibly past) inventory levels at all the customers, the customers’ demand behavior, the customers’ locations relative to the supplier and relative to each other and the resulting transportation costs, and the capacity and availability of vehicles and drivers for delivery.

The inventory routing problem (IRP) is one of the core problems that has to be solved when implementing the emerging business practice called vendor managed inventory replenishment (VMI). IRP is to determine the delivery quantity for each customer and a set of feasible vehicle routes for the delivery of the quantities in each period, subject to the vehicle capacity constraints and the customers' product requirements and inventory capacity constraints, so that a total inventory and transportation cost is minimized (Yu et al,2008).

The inventory management problem and the vehicle routing problem have frequently been studied for years, whereas coordinated decision making for these problems, i.e., the inventory routing problem (IRP), has been of interest mostly in the last two decades. The main inspiration for studying IRP is the logistics systems, in which either vendor-managed replenishment systems are in use or when the supplier and the multiple retailers represent different echelons in the supply chain of a single firm. As generally cited in the literature, vendor-managed replenishment systems can be seen in the grocery industry, for instance, when the producer (supplier) of the goods on the shelves of the supermarkets (retailers) has the responsibility of monitoring and replenishing the products. In these types of distribution systems, the decision of how much inventory to maintain at the supplier and the retailers is affected by delivery times and amounts for the retailers, which in turn is affected by the capacity of the vehicles used for the deliveries. Thus, simultaneous decision making is important in these systems to obtain significant cost savings. The objective is minimizing the total distribution and inventory costs of the whole supply chain, in an integrated way, over a given planning horizon.

Inventory routing problems are very different from vehicle routing problems. Vehicle routing problems occur when customers place orders and the delivery company, on any given day, assigns the orders for that day to routes for trucks. In inventory routing problems, the delivery company, not the customer, decides how much to deliver to which customers each day. There are no customer orders. Instead, the delivery company operates under the restriction that its customers are not allowed to run out of product.

Another difference is the planning horizon. Vehicle routing problems typically deal with a single day, with the only requirement being that all orders have to be delivered by the end of the day. Inventory routing problems deal with a longer horizon. Each day the delivery company makes decisions about which customers to visit and how much to deliver to each of them, while keeping in mind that decision made today impact what has to be done in the future. The objective is to minimize the total cost over the planning horizon while making sure no customers run out of product. The flexibility to decide when customers receive a delivery and how large these deliveries will be may significantly reduce distribution costs. However, this flexibility also makes it very difficult to determine a good, much less an optimal, cost effective distribution plan. When the choice becomes which of the customers to serve each day and how much to deliver to them, the choices become virtually endless (Campbell et al, 2002).

In this thesis, an inventory routing problem with limited vehicle capacity and storage constraint is considered. The problem consists of coordinated decision making for inventory management and vehicle routing in an inbound commodity collection system composed of a central warehouse and suppliers. Each supplier produces one different item, each of which faces demand from outside retailers. The warehouse replenishes the retailers who meet outside demand for the items. The demands at the retailers are deterministic.

The warehouse uses fleet of vehicles to collect the items from its supplier's. These vehicles have limited capacity. The warehouse has a limited storage area. The associated inventory quantities do not exceed available the storage space. It assumed that there is no shortage or delay at any supplier. The costs of the integrated inventory-routing system include inventory holding cost at the central warehouse, fixed vehicle dispatching costs and vehicle routing costs. Objective in this study is to develop mathematical programming model and a heuristic to coordinate inventory and transportation decisions faced by the warehouse in the system described above. This problem is introduced by Sindhuchao et al. (2005) for single period without storage constraint.

The chapters in this thesis are organized as follows.

Chapter 2 includes the related studies on inventory routing problems in the literature. A scheme to classify the inventory routing problems, based on Baita, Ukovich, Pesenti and Favaretto (1998), is described in Section 2.1. In Section 2.2, the literature review is presented.

In Chapter 3, the mixed integer programming (MIP) model developed for the IRP is presented. Section 3.1 starts with a description of the problem environment and the features of our problem are summarized according to the classification scheme given in Section 2.1. In Section 3.2 the basic assumptions are made. In Section 3.3, the mathematical formulation of the IRP is presented and the constraints of this model are described in detail. The difficulties related with this model are explained afterwards.

Due to difficulties faced with the solution to the mathematical formulation of the IRP, a heuristic is suggested in Chapter 4. The heuristic is developed by modifying the nearest neighbor heuristic, which is using to solve vehicle routing problems and an inventory routing algorithm is developed and is used in modified heuristic to solve IRP.

Numerical experiments are carried out both with the integrated model given in Chapter 3 and heuristic given in Chapter 4. Chapter 5 includes all results obtained with these methods in two sets of randomly generated numerical tests. In Section 5.1, the properties of the test problems are discussed and the datas used in experiments are presented. In section 5.2, the total cost gap between the holding inventory option and without holding inventory and the results obtained for the test instances are presented for each set of experiment

In Chapter 6, main stages and contributions of our study are summarized. The performance of heuristic that developed is identified. The future research is suggested.

CHAPTER TWO
LITERATURE REVIEW FOR THE INVENTORY ROUTING PROBLEM

The foundation for the inventory routing problem is the vehicle routing problem (VRP). The VRP represents a class of problems that seek to develop routes by assigning vehicles to customers. When inventory constraints are added to the VRP, the problem is known as the Inventory Routing Problem. Both inventory management problems and vehicle routing problems have independently been analyzed in numerous studies for years. On the other hand, combined analysis of inventory management and vehicle routing problems, i.e., inventory routing problems, has mostly arisen in the last two decades. The orders in VRP are specified by the customers and the aim of the supplier is to satisfy these customers while minimizing its total distribution cost whereas in the IRP the orders are determined by the supplier, obviously based on some input from the customers whose aim is to minimize the sum of the inventory cost and the distribution cost. In the VRP, the question is to find (i) the delivery routes, whereas in the IRP the question does not include (i) only but also the quantity to deliver to each customer as well as the time for delivery (Moin and Salhi ,2006).

In general, the IRP focused on the route design and inventory management. The route design component determines how to cluster customers on routes based on travel distances and supply or demand volume. A more detailed route design considers vehicle or crew scheduling and determines how the routes can be serviced given limited vehicles or available working hours. Much of the literature focused strictly on inventory management and basic route design. An extension of such research is to examine the routes in detail in order to incorporate scheduling for feasibility purposes.

In this chapter, we describe the classification scheme used for reviewing inventory routing problems studied in the literature and then we present review of the related literature.

2.1 Classification Scheme

To classify the literature on inventory routing problems, a system similar to the one provided in Baita, Ukovich, Pesenti and Favaretto (1998) is used. The elements of classification scheme on inventory routing problems are explained below.

Number of items

- One: The items to be distributed are of one type.
- Many: The items to be distributed are of multiple types.

Decision domain

- Time: The decisions related to inventory and distribution problems are carried out over time periods.
- Frequency: The decisions are executed over delivery frequencies. The studies, whose decision domains are frequency, consider infinite-horizon problems.

Demand

- Deterministic: The demands at the retailers are assumed to be known.
- Stochastic: The demands at the retailers are uncertain.

Time behavior of the demand

- Constant: The demand at each retailer is constant over time.
- Dynamic: The demands at the retailers vary over time.

Note that constant demand over time brings changes in the solution procedure adapted particularly for the studies whose decision domain is frequency. Assuming that demand is constant over time at the retailers makes it possible to determine fixed visit frequencies for the retailers.

Number of vehicles

- Given: The distribution is performed with a given number of vehicles.
- Not constraining: It is assumed that there are enough number of vehicles to make the required deliveries.

Vehicle capacity

- Equal: In case of multiple vehicles, the capacity of each vehicle is the same.
- Different: In case of multiple vehicles, the vehicles have different capacities.
- NA: It is used when single vehicle case is considered.

Note that vehicles with unequal capacities do not generally result in significant changes in the solution methods developed for the studies whose decision domain is time. For the problems, in which the decision domain is frequency, vehicles with unequal capacities give rise to different visit frequency bounds for the regions that are visited with different vehicles.

Stock capacity constraints

- Yes: There are limits on the amount of inventory carried at the retailers.
- No: The amount of inventory carried at the retailers is not restricted.

Supply capacity constraints

- Yes: There exists a limit on the amount of product that is supplied. Limits on the production capacity (i.e., time availability, resource availability, and so on) are considered in this class, as well.
- No: There is no limit on the product supply.

Inventory parameters

- H_1: Inventory holding cost is incurred at the retailers.
- H_2: Inventory holding cost is incurred at the supplier.
- Penalty: Inventory stock out or other penalty costs are taken into account.

- Order: Product ordering costs are considered.
- Revenue: Revenue is earned in proportion to the amount of product distributed.
- Setup: The cost incurred when setting up the facility for production.
- No: Inventory related costs are not taken into account.

Transportation costs

- Fixed: A fixed transportation cost is incurred at each trip.
- Distance: Transportation cost is incurred according to the distance traveled.
- Amount: Transportation cost is incurred in proportion to the amount of product carried or unloaded.

2.2 Literature Review

Due to the characteristics of the problem we studied, the literature examined mostly consists of works, in which retailers with deterministic demands are taken into account and the amount of production at the supplier is not a consideration. The reader is referred to Baita et al. (1998) and Federgruen and Simchi-Levi (1995) for more comprehensive reviews of inventory routing problems.

Solution techniques for the IRP can be classified into two categories namely the theoretical approach, where the derivation of the lower bounds to the problem is sought and a more practical approach, where heuristics are employed to obtain the near-optimal solutions. Most papers that deal with the theoretical approach employ some strategies that allow the IRP to be decomposed into two underlying problems, namely the inventory and the travelling salesman problem (TSP). The inventory problem is solved to determine the replenishment quantities for each customer as well as the replenishment times. Most papers in this category adopt a two-stage solution approach. They either (i) find the routes first and then solve the IRP formulation, which is a simple linear programming-based inventory control problem, or (ii) solve the inventory control problem first (sometimes with approximated transportation cost), aggregate (cluster) the customers with the same replenishment time instants and then construct the routes for each cluster. As the modification of routes entails resolving a new inventory allocation problem and vice versa, most algorithms iterate between obtaining a new set of routes and resolving the inventory problem until a suitable stopping criterion is satisfied.

The literature review section is structured according to two main approaches proposed in the literature for IRP problems. The first approach, considered in Section 2.2.1, operates in the time domain the schedule of shipments is decided, or, with discrete time models, quantities and routes are decided at fixed time intervals. Conversely, in frequency domain: decision variables are replenishment frequencies, or headways between shipments which is considered in Section 2.2.2

2.2.1 Decision Domain: Time

In the time domain approach to IRP problems, operations are scheduled through a (possibly infinite) horizon. In principle, decisions could be taken once and for all, in an a priori attitude, as in the frequency domain approach. However, the more interesting case is when, with a discrete time model, decisions are taken at the beginning of each time slot (e.g. every day or week), knowing the state of the system (i.e. the inventory levels). This is a closed loop approach, in the sense that is a feedback of the decisions taken in one period that affect the decisions of the following period.

Bell et al. (1983) consider a real life problem. Although costs related with inventories are not taken into account in this study as in Campbell et al. (2002), it is assumed that revenue related with the amount of product delivered is earned. A fixed cost and cost related with the amount of product unloaded from the transportation costs. Customer data (i.e., tank capacity, historical product usage), resource data (i.e., truck capacities, product availabilities), cost data, time and distance data, and schedule data of the firm are incorporated into a mixed integer programming (MIP)formulation as problem parameters. To formulate the model, demand forecasts are used to compute minimum and maximum inventory levels. To formulate the problem as a MIP, a set of vehicle routes, each composed of a set of customers to be visited at a trip, is generated by a program, which produces feasible and efficient routes. The least-cost order of the customers on the route is decided by complete enumeration. A subset of routes that are to be driven are selected from the set of routes generated and the starting time of each route, the vehicle to be used at each route, and the delivery amount to each customer on the route are determined with the aim of maximizing the value of deliveries minus the cost of deliveries by using the MIP model. Due to large size of the model, lagrangean relaxation is applied, after which the problem is decomposed into sub problems (one for each vehicle) and an upper bound is obtained for the problem. A heuristic based on the lagrangean relaxation approach is used to find a feasible solution (a lower bound) to the problem.

Dror and Ball (1987) reduce the annual problem to a shorter planning horizon (m-day) problem. The key in doing so is to define short-term costs that reflect the long-term costs. This problem is modeled as MIP, in which the effects of current decisions on later periods are accounted for by penalty or incentive factors. It is assumed that the customers keep inventory in tanks with predetermined sizes and whenever a customer receives a delivery, its tank is filled up. The relationship between the annual distribution cost, the fixed delivery cost, and the amount delivered to the customers are examined and the customers to be visited on a given day are selected according to these costs. The authors formulate a mathematical model that takes into account the relationships between the costs mentioned. Single-customer deterministic, single-customer stochastic, and multiple-customers cases are considered in the paper.

In the single-customer deterministic problem, to reduce the annual problem to a single period problem, a penalty cost for the long term effects of the decisions that are carried out in the short term is utilized and a continuous time deterministic model is formulated. For this purpose, an m-day period, in which all costs incurred are considered explicitly, and the following n-day period, in which the effects of the decisions made for the m-day period are considered, are defined. The changes in the costs over the succeeding n days are investigated depending on whether the customer requires replenishment during m days or not. The increase in the costs over the succeeding n days, if a customer who needs replenishment on day t (t<m) not to stock out is resupplied before day t and the decrease in the costs over n days, if a customer who does not need replenishment in m-day period is replenished within m days, are calculated.

In the single-customer stochastic problem, it is decided whether or not to visit a customer at the beginning of the planning period. In case a customer that is not visited runs out of the stock, a penalty cost is incurred and it is assumed that the customer is automatically replenished. Since the customer runs out of stock, if it is not resupplied, the expected stock out payment, the expected future cost penalty, and a safety stock level are calculated.

In the multiple-customer case, a mathematical model for the inventory routing problem is constructed using a generalized assignment VRP formulation that includes transportation costs and costs and incentives related with early deliveries. Although stochastic demands are used in safety stock calculations, the formulation of the model is based on deterministic demands. In this formulation, the amount of product to be delivered to a customer on a visit is determined by the day of the visit. Therefore, it is not considered as a decision variable. This problem is solved by a modified generalized assignment algorithm and it is seen that the algorithm provides more than 50% increase in the performance over a manual system that is used at the time of the study and more than 25% increase in the performance over another existing system.

In Chien et al. (1989), a series of single period inventory allocation and vehicle routing problems is solved with the aim of maximizing revenues minus costs to obtain an approximate solution to the multi-period inventory routing problem. As a first step for solving this problem, a multi-commodity flow-based mixed integer programming model is formulated. Since it is difficult to solve this problem optimally, a lagrangean relaxation based approach is developed to obtain upper and lower bounds for the problem. By applying lagrangean relaxation to four constraints of the model, two sub problems are obtained. The first sub problem is an inventory allocation problem and the second sub problem is a customer assignment/vehicle utilization problem, which can further be decomposed into customers and vehicles. Sub problems are solved by greedy procedures to identify an upper bound to the original problem. To find a lower bound for the problem, a heuristic composed of two phases is applied. In phase I, an initial set of vehicle routes is obtained using the solutions of the inventory allocation and customer assignment/vehicle utilization sub problems. The first step of Phase II is to check for feasibility. In case the solution is not feasible, a feasible solution is found. Then, it is checked whether the amount supplied to the customers on the routes can be increased. If the customers that are currently on a route are fully supplied, new customers with unsatisfied demands are inserted to the related route.

In the study of Chandra (1993), the decisions related with the inventory policies of the supplier (warehouse) are also taken into account. The aim is to determine replenishment quantities for both the warehouse and the customers together with the delivery routes. The author provides a MIP model, which is decomposed into two sub problems by separating the constraints related with the warehouse and the customer replenishments. The first sub problem is a single facility, incapacitated, multiproduct, multi-period warehouse ordering problem (WOP), which is NP-hard. The second sub problem is the distribution planning problem (DP), which determines delivery routes in every period and amount of product to be delivered to each customer upon delivery. An approximate solution algorithm to the integrated problem is defined and the solutions obtained by this approach are compared to the case in which the two sub problems mentioned above are solved separately and sequentially. The approximation algorithm finds an initial feasible solution to the WOP. Based on the distribution lots obtained from the solution of WOP, DP is solved. Then possible cost reductions are searched, when the distribution patterns are changed. This iterative procedure is applied until further gains are not realized.

In study of Fumero and Vercellis (1999) the production and distribution decisions are isolated by relaxing the constraints that link the two decisions to obtain solvable sub problems. After these constraints are relaxed, it is possible to decompose the problem into four sub problems that are used to make decisions on production, inventory, distribution, and routing, separately. The solution of the sub problems gives a lower bound on the original problem. Using the solutions of the sub problems, a feasible solution (i.e., an upper bound) for the original problem is identified by a heuristic procedure.

In Campbell et al. (2002), a real life inventory routing problem of a firm, which negotiated with its customers about initiating a vendor-managed replenishment policy is studied. Although meeting the demands at the customers on time is required, inventory related costs are not considered when making decisions. The solution approach suggested is composed of two phases. In Phase I, the customers to be visited in each day and the delivery amounts are determined, whereas in Phase-II, the actual delivery routes and schedules for each day are determined. For the first phase, an integer programming model is constructed. Upon this basic model, three variations are considered: (1) handling the stop times at the customers together with the vehicle reloading time at the supplier, (2) the start and end times of customer usage, and (3) the time windows that restrict the delivery times. For solving the integer programming model constructed in phase I, the customers are clustered at the beginning and it is assumed that only the customers that are in the same cluster can be on the same route. After identifying the clusters, the integer programming model is solved by replacing the daily variables with weekly variables for the days after a specific day (k days) and removing integrality for these weekly variables. Thus, the volume of product that will be delivered to each customer in the next k days is determined in phase I. In phase II, vehicle routing problems with time windows are solved to obtain daily vehicle routes and schedules. For the computational experiments, the actual data of two production facilities of the company is used. The solutions obtained are compared with another heuristic, which is composed of the rules that constitute an approximation of the methods used in the industry. It is seen that the two-phase approach out performs the industry approximation approach for both facilities considered on the important statistics.

Bertazzi et al. (2002) consider a periodic-review, multi-product IRP with a given inventory policy. The inventory policy, I, is restricted to a type such that each retailer has a given a minimum and a maximum level of the inventory for each product; and each retailer must be visited before any of its inventories reach the minimum level. Further, every time a retailer is visited, the quantities delivered are such that the prespecified maximum levels are reached for each product. The problem is to determine for each discrete time period the retailers to be visited and the route of the vehicle. The authors suggest a two-step heuristic method for solving the problem. Firstly, a set of delivery times are obtained for the retailers. For this purpose, a cyclic network is formed for each retailer then, for these selected delivery times, the retailers are inserted into the routes by using insertion at cheapest cost method. These steps form Phase-I of the two-phase heuristic developed. In the second phase, iterative improvements are made in the following way. A pair of retailers is removed from the routes and possible improvements are searched by identifying new sets of delivery time instants for these retailers and repeating the cheapest insertion method.

Lee et al. (2003) study the IRP in an automotive part supply chain that comprises several suppliers and an assembly plant. The problem addressed is based on a finite horizon, multi-period, multi-supplier and a single assembly plant partsupply network where a fleet of capacitated identical vehicles transport parts from the suppliers to meet the demand specified by the assembly plant for each period. This problem represents an in-bound logistic problem of type many-to-one network and is equivalent to the one-to-many under certain conditions. The authors propose MIP model to minimize the overall transportation cost and the inventory carrying cost. This MIP model is decomposed into two sub problems, namely the VRP and the inventory control. A heuristic based on simulated annealing is proposed to generate and evaluate alternative sets of vehicle routes while a linear program determine the optimum inventory levels for a given set of routes. Then, a route perturbation routine is implemented to modify a set of vehicle routes based in some information obtained from the optimal solution to the linear program. The modification of routes entails resolving the linear program to get new inventory levels. This scheme is carried out iteratively until a stopping criteria is reached, namely the specified maximum number of iterations. They also observe an important property that the optimal solution is dominated by the transportation cost regardless of the magnitude of the unit inventory carrying cost. This claim is then proven analytically for a simpler version of the above problem based on an infinite planning horizon and stationary demand with a single supplier providing either a single-part type or multiple-part types.

Yu et al. (2008) study an inventory routing problem (IRP) with split delivery and vehicle fleet size constraint. They develop the multiple period IRP with a set of customers, a central depot, and a fleet of homogeneous vehicles with limited capacity, for which in each period the depot has to deliver sufficient units of a product to each customer to completely fulfill its demand, which is deterministically known. No backorder is allowed at each customer but each customer may hold a local inventory used to meet future demands with a holding cost. The delivery to each customer in each period can be performed by multiple vehicles. The objective is to minimize over a given time horizon the total inventory holding and transportation cost, which is the sum of the inventory holding costs of all customers and the transportation costs for all deliveries to the customers.

Abdelmaguid et al. (2008) study the integrated inventory distribution problem which is concerned with multi period inventory holding, backlogging and vehicle routing decisions for a set of customers who receive units of a single item from a depot with infinite supply. Each customer maintains its own inventory up to is capacity and incurs inventory carrying cost per period per unit and backorder penalty on the end of period inventory position. Demand of each customer is small compared to the vehicle capacity and the customers are located closely. Deliveries to customers are made by a capacitated heterogeneous fleet of vehicles starting from the depot at the beginning of each period. They introduce a constructive heuristic for solving this NP hard problem.

2.2.2 Decision Domain:Frequency

In the frequency domain approach to IRP problems, loads, routes, visit frequencies (and possibly phasing) are decided once and for all, in an a priori planning attitude. As a result, periodic operations are proposed. Decision variables are replenishment frequencies, or headways between shipments. Within the field of the frequency domain, two main lines can be distinguished. The first one is identified as the Aggregate Approach, where the basic idea is to find reasonable solutions with as little data as possible, avoiding mathematical programming models requiring detailed data. A second line is referred to as the approach based on Fixed-Partition Policies, which is the decomposition strategy used to assign customers to routes.

A stream of studies dealing with fixed-partition policies are started with Anily and Federgruen (1990). In this study, it is assumed that inventory is not kept at the warehouse and an upper and a lower bound to the long run average transportation and retailer inventory holding costs are determined. The routing schemes are identified by a modified circular partitioning scheme. Following the partitioning of the customers, the customers in each partition are separated into regions. Whenever a customer in a region receives a delivery with a vehicle, all other customers in the related region are also visited by the same vehicle. Under this strategy, partial fulfillment is possible since it is probable to assign a customer to multiple regions.

The Anily and Federgruen (1994) study the same problem, whose generalizes the results of their previous study (Anily and Federgruen ,1990) for the case, in which holding costs at the retailers are not identical. Due to this property, a difference of the proposed solution from that of the previous work is that the partitioning of the retailers to the regions is performed taking the holding costs at the retailers into account. In the experiments, it is seen that the gaps between the lower and upper bounds are less than 10% for all instances and the computational time is at most a few seconds. However, the authors recommends use of the algorithm presented by Anily and Federgruen (1990), if the holding costs at the retailers are identical since it provides better quality policies.

In another study by Anily and Federgruen (1993), the authors consider an extension to their previous work, (Anily and Federgruen, 1990), so that keeping inventory at the depot is allowed, i.e., central inventories are possible. For obtaining a solution to this problem, a similar strategy to the one used in the preceding work is utilized.

Gallego & Simchi-Levi (1990) assume that each retailer faces a constant, retailer-specific, daily demand. The depot holds no inventory. Holding cost and fixed ordering cost are charged only at the retailers. Gallego and Simchi-Levi evaluate the long-run effectiveness of direct delivery and study conditions under which direct delivery is an efficient policy. An upper bound is identified for the case in which direct shipments are carried out by fully loaded trucks.

Chan et al. (1998) characterize the asymptotic effectiveness of the class of fixed partition (FP) policies (i.e., partitioning the retailers into regions and serving each region independently) and the class of so-called Zero-Inventory Ordering (ZIO) policies, under which a retailer is replenished if and only if its inventory is zero. Their analysis is motivated by the observation that the class of FP policies is a subset of the class of ZIO policies. Worst-case studies as well as probabilistic bounds under a variety of probabilistic assumptions are provided. A heuristic algorithm is developed for partitioning the retailers into regions so that each region is assigned a vehicle that visits all retailers in that region at equidistant epochs.

Sindhucho et al. (2005) consider a system that consists of a set of geographically dispersed suppliers that manufacture one or more items and a central warehouse that stocks these items. The warehouse faces a constant and deterministic demand for the items from the outside retailer The warehouse uses a fleet of vehicles to collect items from its suppliers. These vehicles have limited capacity and they are also subject to frequency constraints that limit the number of trips that each truck can make per time unit. There is no shortage or delay at any supplier. They adopt a policy where the set of items is partioned into disjoint groups and each group of items is assigned to a vehicle. The warehouse leads to a joint replenishment of all items in a group using an economic order quantity. They develop a branch-price algorithm that can be used to solve inventory routing problem. This algorithm is based on a column generation approach to the set partitioning formulation of the problem. Then they develop constructive heuristics. The developed heuristics are used to find an initial feasible solution by constructing routes for vehicles either sequentially or simultaneously and then they developed a neighborhood search algorithm that can be used to improve a solution found by constructive heuristics.

Savelsbergh and Song (2006) focus on the inventory routing problem with continuous moves (IRP-CM), which incorporates two important real-life complexities: limited product availabilities at facilities and customers that cannot be served using out-and-back tours. They design delivery tours spanning several days, covering huge geographic areas, and involving product pickups at different facilities. They present an integer multi-commodity flow formulation on a time-expanded network for IRP-CM to capable of solving small to medium size instances and develop a customized solution approach for its solution. The solution approach is capable of producing optimal or near optimal solutions for small to medium size instances in a reasonable amount of time. The objective is to minimize transportation costs over the planning horizon T while trying to ensure that none of the customers experiences a stock out and that none of the plants have to vent product.

Table 2.1 provides our classification of the IRP literature, according to classification scheme that is presented in section 2.1

Table 2.1 Classification IRP literature

No	Author	Year	Objective	Number of Items	Decision Domain	Demand	Time Behavior of Demand	Number of Vehicles	Vehicle Capacity	Stock Capacity	Supply Capacity	Inventory Parameters	Transportation Cost	Solution Method
1	Bell et al.	1983	Max profit	O	T	D	D	G	D	Y	Y	Revenue	Fixed+Amount	Integer Programming
2	Dror and Ball	1987	Min cost	O	T	D	C	G	E	Y	N	H_1	Fixed+Distance	Analytical Results and Heuristics
3	Chien et al.	1989	Max profit	O	T	D	C	G	D	Y	Y	Penalty+ Revenue	Fixed+Amount	Mixed Integer Programming, Heuristic and Upper Bound
4	Anily, Federgruen	1990	Min cost	O	F	D	C	NC	E/D	N	N	H_1	Fixed+Distance	Heuristic, Lower and Upper Bounds
5	Gallego & Simchi-Levi	1990	Min cost	O	F	D	C	NC	E	N	N	H_1+ Order	Fixed+Distance	Heuristic and Lower Bound
6	Anily, Federgruen	1993	Min cost	O	F	D	C	NC	E	N	N	H_1+H_2	Fixed+Distance	Heuristic, Lower and Upper Bounds
7	Chandra	1993	Min cost	M	T	D	D	NC	E	N	N	H_1+H_2+ Order	Fixed+Distance	Mixed Integer Programming
8	Anily	1994	Min cost	O	F	D	C	NC	E	N	N	H_1	Fixed+Distance	Heuristic and Lower Bound
9	Chan et al.	1998	Min cost	O	F	D	C	NC	E	N	N	H_1	Fixed+Distance	Heuristic and Bounds

10	Fumero &Vercellis	1999	Min cost	M	T	D	D	G	E	N	Y	H_1+H_2+ Setup	Fixed+Amount+ Distance	Mixed Integer Programming, Heuristic and Lower Bound
11	Campbell,Clarke, Savelsbergh	2002	Min cost	O	T	D	C	G	E	Y	N	No	Distance	Integer Programming
12	Bertazzi et al.	2002	Min cost	M/ O	T	D	D	G	NA	Y	Y	H_1+H_2	Distance	Heuristic
13	Lee et al.	2003	Min cost	M	T	D	C	G	E	N	N	H_1	Fixed+Distance	Mixed Integer Programming, Heuristic
14	Sindhuchao et al.	2005	Min cost	M	F	D	C	G	E	N	N	H_1	Fixed+Distance	Column Generation, Heuristic,
15	Savelsberg &Song	2006	Min cost	O	F	D	C	G	E	Y	Y	H_1+H_2	Distance	Integer Programming, Heuristic
16	Abdelmaguid et al.	2008	Min cost	O	T	D	C	G	E	Y	N	H_1+ Penalty	Fixed +Distance	Mixed Integer Programming, Heuristic
17	Yu et al.	2008	Min cost	O	T	D	C	G	E	Y	N	H_1	Fixed +Distance+ Amount	Lagrangian relaxation

CHAPTER THREE
MODEL FORMULATION

In this chapter, firstly, the problem environment is introduced, next basic assumptions in the problem are listed, and then an integrated model to formulate the problem is described. A discussion of difficulties faced upon identifying a solution to IRP using the integrated model is given afterwards.

3.1 Problem Environment

The problem in this thesis is like the problem, which is presented by Sindhuchao et al (2005). They consider a system that consists of a set of geographically dispersed suppliers that manufacture one or more non-identical items and these items are stocked in a central warehouse. The items are collected by a fleet of vehicles that are dispatched from central warehouse. The vehicles are capacitated and must satisfy a frequency constraint. The warehouse replenishes the retailers who, in turn meet outside demand for the items.

There are some differences between our problem and the problem of Sindhuchao et al (2005). First difference is the decision domain. Our decision domain is time, whereas their decision domain is frequency. In their problem, a frequency constraint limits the number of trips that each truck can make per time unit. In our problem, the vehicles can make one trip in each period. Second difference is the storage capacity. In our problem, the warehouse has a limited storage area ignored in their problem. The associated inventory quantities should not exceed the available storage area. It is assumed that there is no shortage or delay at any supplier. Third difference is the total cost calculation. The total cost in their problem includes inventory holding costs at the central warehouse, fixed vehicle dispatching costs, vehicle routing costs and fixed ordering cost. In our study, fixed ordering cost is ignored as any inventory policy is not considered. We consider meeting demands with minimum costs in each period. The aim of the both problems is to develop mathematical programming model and heuristic for the system described above.

Sindhuchao et al.(2005) adopt a policy where the set of items is partitioned into disjoint groups and each group of items is assigned to a vehicle. The warehouse faces a constant demand for each item leads to a joint replenishment of all items in a group using economic order quantity policy. The vehicle leaves the warehouse, visits the set of suppliers corresponding to the items in its group, and returns to the warehouse, where the items are unloaded and stored. Every supplier has non–identical items. Items cannot be assigned to more than one group, i.e. the orders cannot be split across multiple vehicles.

In our problem, there is no group for items and vehicles. The warehouse faces a deterministic demand equal to demand of retailers. The vehicle leaves the warehouse, visits the supplier that provides the minimum cost. Then vehicle goes on with another supplier, which satisfies the vehicle capacity constraint and minimum cost goal. When the vehicle capacity is not enough for any orders, the system checks if receiving any inventory has advantage or not. If receiving inventory for next periods is profitable than visiting the same supplier in the next period, the vehicle receives inventory. This supplier will not be visited in the next period. As a result, the empty space in the vehicle will be less, and the total distances that should be visited until the end of planning horizon will be less. The orders can be split across multiple vehicles.

In Figure 3.1, an example is illustrated in which there are five suppliers that the warehouse purchases items from and three retailers that are served by the warehouse. The warehouse has two vehicles, one of which collects items from suppliers S1 and S2 while the other one collects items from suppliers S3, S4, and S5. The warehouse faces demand for the items at the retailers R1, R2, and R3.

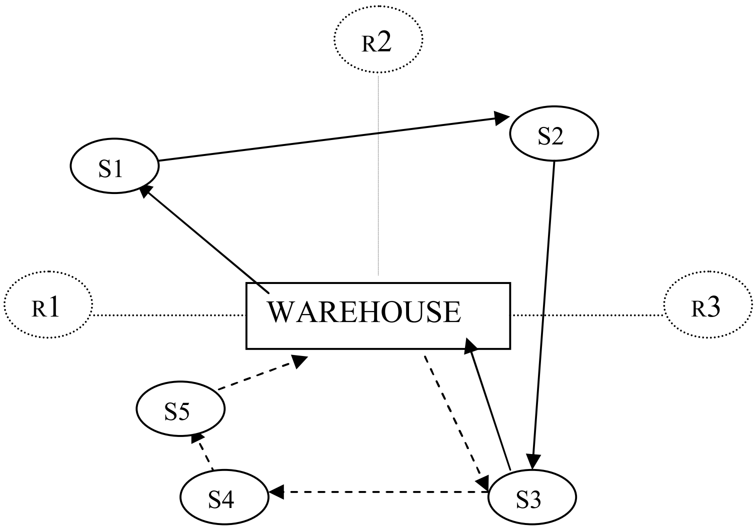

Figure 3.1 Inventory-routing system with multiple suppliers and a centralized warehouse

The features of the problem we studied are summarized in Table 3.1 below according to elements of the classification scheme described in Section 2.1

Table3.1 The characteristics of the problem we studied

Element	Feature
Number of items	More
Decision domain	Time
Demand	Deterministic
Demand time behavior	Dynamic
Number of vehicles	Given
Vehicle capacity	Different
Stock capacity	Yes
Supply capacity	No
Inventory parameters	H_1
Transportation cost	Distance+Fixed

3.2. Basic Assumptions for Our Problem

The basic assumptions in model formulation are as following:

- The warehouse faces deterministic demand of retailers.
- Demand of each retailer is small compared to vehicle capacity.
- The warehouse can keep inventory and covers inventory holding cost per period.
- Inventory holding costs are incurred at the end of each period.
- The warehouse has a limited inventory area for each product.
- A fleet of vehicles is collected the items from suppliers. These vehicles have limited capacity and capacity of each vehicle is equal.
- It is assumed that fleet of vehicles remains unchanged through the planning horizon. Renting additional vehicles is not an option due to technological or economic constraints.
- Suppliers can be visited by more than one vehicle in a period.
- Vehicles must return to warehouse at the end of each period and no further delivery assignments should be made during the same period.
- A distance dependent transportation cost is incurred upon deliveries.
- The sequence of the events for any period is as follows. Items are picked from suppliers according to demand of retailers and on hand inventory. Inventory holding costs are incurred according to on hand inventory at the warehouse. Transportation cost includes fixed usage cost per vehicle, and a variable transportation cost between two distances. Warehouse delivers items to retailers at the end of each period (Costs and details of these shipments are ignored). The order of current period is not considered as inventory at the warehouse.

3.3 Model Formulation

The mathematical formulation is given below.

Sets

S: Set of suppliers, i = {0,1, 2, 3, …, N} The index 0 is used for the warehouse

T: Set of periods in the planning horizon, T = {1, 2, 3, …, T}

V: Set of vehicles V={1 . . .v}

Parameters

h_{it} = unit inventory holding cost for item of supplier i in period t

I_{ikt} = inventory level for item of supplier i in period t for the period k

c_{ij} = a variable transportation cost between supplier i and supplier j

ds_{ij} = distance between supplier i and supplier j

d_{it} =demand for item of supplier i for period t

S_{it} = available inventory area in warehouse for item of supplier i in period t

Q_v = vehicle capacity

f_t = fixed usage cost per vehicle for period t

Decision variables:

x_{ijt}^{v} = vehicle v travels from supplier i to supplier j in period t and carrying item of supplier i equals 1 or 0

y_{ijt}^{v} = the amount of item of supplier i transported from supplier i to supplier j in period t by vehicle v

$$\min\left[\sum_{t=1}^{T}\left[\sum_{j=1}^{N}\sum_{v=1}^{V} f_t x_{0jt}^{v} + \sum_{i=0}^{N}\sum_{\substack{j=0\\ j\neq i}}^{N}\sum_{v=1}^{V} c_{ij} ds_{ij} x_{ijt}^{v} + \sum_{i=1}^{N}(k-t)*\left(h_{it} I_{ikt}\right)\right]\right] \qquad [0]$$

Subject to:

$$\sum_{\substack{j=1\\ j\neq i}}^{N} x_{ijt}^{v} \leq 1 \qquad \begin{array}{l}\forall i = 1,.., N\\ \forall t = 1.....T\\ \forall v = 1....V\end{array} \qquad [1]$$

$$\sum_{\substack{j=0\\ j\neq i}}^{N} x_{ijt}^{v} = 1 \qquad \begin{array}{l}\forall i = 0\\ \forall t = 1.....T\\ \forall v = 1....V\end{array} \qquad [2]$$

$$x_{ijt}^{v} + x_{jit}^{v} \leq 1 \qquad \begin{array}{l}\forall i = 1,.., N\\ \forall j = 1,...N, i \neq j\\ \forall t = 1,.., T\\ \forall v = 1,.., V\end{array} \qquad [3]$$

$$\sum_{\substack{l=0\\ l\neq i}}^{N} x_{ilt}^{v} - \sum_{\substack{k=0\\ k\neq i}}^{N} x_{kit}^{v} = 0 \qquad \begin{array}{l}\forall i = 0,.., N\\ \forall t = 1,.., T\\ \forall v = 1,...V\end{array} \qquad [4]$$

$$y_{ijt}^{v} <= Q_v x_{ijt}^{v} \qquad \begin{array}{l} \forall i = 1,..,N \\ \forall j = 0,...N, i \neq j \\ \forall t = 1,..,T \\ \forall v = 1,..,V \end{array} \qquad [5]$$

$$\sum_{\substack{l=0 \\ l \neq i}}^{N} y_{ilt}^{v} - \sum_{\substack{k=0 \\ k \neq i}}^{N} y_{kit}^{v} \leq 0 \qquad \begin{array}{l} \forall i = 1,..,N \\ \forall t = 1,..,T \\ \forall v = 1,..,V \end{array} \qquad [6]$$

$$\sum_{\substack{t=1 \\ k=t}}^{t-1} I_{ikt} - \sum_{k=t+1}^{T} I_{ikt} + \sum_{v=1}^{v} \left(\sum_{\substack{j=0 \\ j \neq i}}^{N} y_{ijt}^{v} \right) = d_{it} \qquad \begin{array}{l} \forall t = 1,..,T \\ \forall i = 1....N \end{array} \qquad [7]$$

$$\sum_{i=1}^{N} \sum_{\substack{j=0 \\ j \neq i}}^{N} (y_{ijt}^{v}) <= Q_v \qquad \begin{array}{l} \forall v = v....V \\ \forall t = 1,..,T \end{array} \qquad [8]$$

$$\sum_{k=t+1}^{T} \sum_{t=1}^{t} I_{ikt} \leq S_{it} \qquad \begin{array}{l} \forall t = 1,..,T \\ \forall i = 1....N \end{array} \qquad [9]$$

$$I_{ikt} \geq 0 \qquad \begin{array}{l} \forall i = 1,..,N \\ \forall t = 1,..,T \end{array} \qquad [10]$$

$$y_{ijt}^{v} \geq 0 \qquad \begin{array}{l} \forall t = 1,..,T, \forall i = 0...N \forall j = 0,...N, \\ i \neq j, \forall t = 1...T, \forall v = 1...V \end{array} \qquad [11]$$

$$x_{ijt}^{v} = 0 \; or \; 1 \qquad \begin{array}{l} \forall t = 1,..,T, \forall i = 0...N, \forall j = 0,...N \\ i \neq j, \forall t = 1...T, \forall v = 1...V \end{array} \qquad [12]$$

In the above formulation, the objective function [0] minimizes the total of inventory holding cost at the warehouse, fixed dispatching cost and transportation cost.

Constraint [1] makes sure that a vehicle will visit a location no more than one in a period. The vehicle, which is left from supplier i can visit only one of j suppliers or returns to warehouse in period t.

Constraint [2] makes sure that a vehicle will visit only one of the suppliers after leaving the warehouse in period t.

Constraint [3] prevents supplier is visited again in the same period with the same vehicle. If any vehicle visits supplier i and then j, it is impossible to visit j and then i during the same period.

Constraint [4] ensures route continuity, otherwise vehicles cannot return to warehouse. The vehicle which visits supplier i in the first sequence, should go supplier j from supplier i.

Constraint [5] serves for two purposes: the first one is to ensure that the amount transported between two locations will always be zero whenever there is no vehicle moving between these locations, and the second is to ensure that the amount transported is less than or equal to vehicle's capacity.

Constraint [6] is eliminated sub-tours. Every vehicle turns to warehouse after picking up items.

Constraint [7] is inventory balance equations for the items. It makes sure that sum of the inventory at the warehouse and the amount delivered in that period is equal to sum of the demand for the current period and inventory for the next period.

Constraint [8] provides that amount of item distributed to the warehouse in a period does not exceed the vehicle capacity. If there is no vehicle transported between supplier i and supplier j, this means no item is carried from supplier i. Total amount in a vehicle cannot be more than vehicle capacity.

Constraint [9] limits the inventory level of the items to the corresponding storage capacity. The demand of retailer in a given period is not kept in the warehouse; there is only inventory for next period on storage location.

Constraint [10] is nonnegativity and constraints [11] and [12] are restrictions on the related decision variables.

The solution of model will identify the following basic issues, while minimizing the total cost.

- The amount of item supplier i delivered to the warehouse in each period with each vehicle.
- The route followed by the vehicle while picking the items.

The presented MIP is so complex when the number of constraints is considered. The number of first constraint in the model is the product of supplier number, period and number of vehicles. If we investigate this complexity as indexes, the constraint numbers are given in Table 3.2.

Table 3.2 Constraint number according to indexes

Constraint Group	Index	Constraint
[1]	∀i,t,v	i*t*v
[2]	∀ i,t,v	t*v
[3]	∀i,t,v,j	j*t*v
[4]	∀i,t,v	(i+1)*t*v
[5]	∀i,t,v,j	i*j*t*v
[6]	∀i,t,v	i*t*v
[7]	∀t,i	t*i
[8]	∀v,t	v*t
[9]	∀t,i	(t-1)*i

Total constraint number is expressed as $1(i \times j \times t \times v) + 3(i \times t \times v) + 1((i+1) \times t \times v) + 2(t \times v) + 1(t \times i) + 1((t-1) \times i)$. If the constraint numbers are illustrated in a graph, we get an equation graph, which increases as parabolic. Period is considered as constant 3, and number of vehicles is considered as 1/5 of supplier number. After these considerations, number of constraints are given in Table 3.3 and the graph is illustrated in Figure 3.2

Table 3.3 Constraint numbers with change in number of suppliers and vehicles

i	v	Constraint Number
5	1	169
10	2	908
15	3	2667
25	5	11045
50	10	81340
75	15	267135

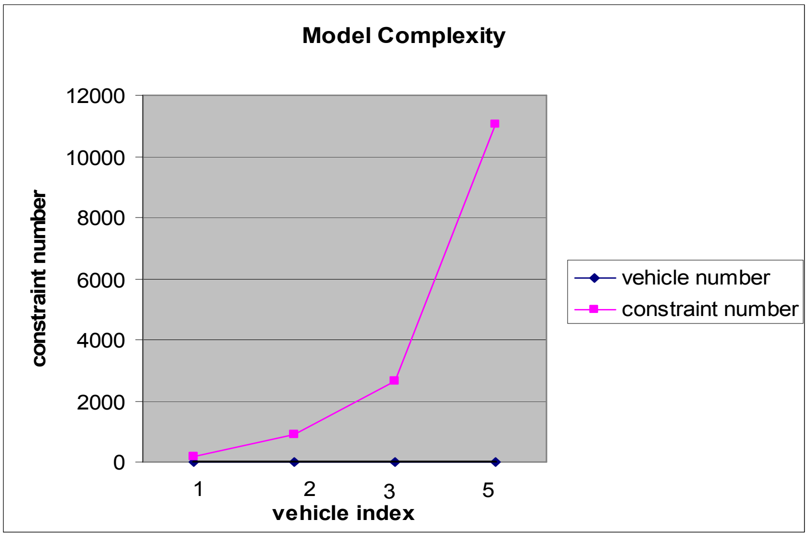

Figure 3.2 Graphical show of model complexity

As seen in Figure 3.2, complexity of problem increases as parabolic. On the other hand, the model given above includes vehicle routing problem (VRP). Thus, even if the inventory related decisions are excluded from the formulation, it is required to identify solutions to VRP to solve our model. The VRP is known to be NP-hard, so it is hard to come up with a polynomial time algorithm to solve it. This implies that model is also difficult to solve. In addition to the routing difficulty, the complexity caused by other decisions included in our model motivates us to develop heuristic procedures to solve the IRP. The developed heuristic for problems with more suppliers and vehicles is presented in the next chapter.

CHAPTER FOUR
HEURISTIC DESIGN

In chapter 3, the complexity of problem is presented. The computational effort is likely to increase rapidly when the number of suppliers, and/or the number of vehicles increase. In this chapter 4, the heuristic which is developed for solving IRP will be discussed. We describe the nearest neighbor heuristic that is using to find solutions to VRP. Then an algorithm for inventory decision is presented. As a conclusion, a modified heuristic to solve IRP is developed.

4.1 Heuristic

The developed heuristic includes two steps: one is to solve VRP and the other one is to solve inventory problem. The nearest neighbor heuristic is used to decide the route. There is an inventory decision on IRP. The nearest heuristic in literature does not provide inventory decision. An algorithm is improved from the algorithm of Abdelgemaid et al. (2008) for the inventory decision. We decide whether to receive the future demand for any supplier in the current delivery by minimizing the total transportation and inventory costs, and satisfying the capacity limits. A modified heuristic will be presented in which the nearest neighbor heuristic and inventory algorithm is used together for solving IRP.

4.1.1 The Nearest Neighbor Heuristic

The nearest neighbor heuristic is used to find optimal route with consideration of vehicle capacity. The heuristic adds customers into a route starting from the warehouse and chooses customer based on the nearest distance to the current location, until either the capacity and storage constraints are violated. The nearest neighbor heuristic starts a route by finding the unrouted customer "closest" to the warehouse. At every subsequent iteration, the heuristic searches for the customer that is "closest" to the last customer added to the route. This search is performed among all the customers who can feasibly be added to the end of the emerging route. A new route is started any time the search fails, unless there are no more customers to schedule (Solomon, 1987).

This heuristic is a well–known heuristic to solve VRP. This heuristic will be used in our heuristic to find route of vehicles for each period. In VRP there is not any inventory option, however in IRP we should consider the inventory option.

4.1.2 Inventory Algorithm

Abdelmaguid et al. (2008) are improved a sub algorithm for inventory decision. They study a distribution system consisting of a depot and geographically dispersed customers. Each customer faces a different demand for a single item per period. Each customer maintains its own inventory up to capacity and incurs an inventory carrying cost per period per unit. The developed algorithm is based on a greedy search that selects the next possible route that has the maximum positive savings. The transportation cost value is estimated from an approximate routing solution for each customer in each period. The algorithm looks into inventory decisions that provide savings in future transportation costs that are higher than inventory carrying costs. They calculate the transportation cost as a function of delivery amount. For every customer in route, the difference between the transportation cost estimate (for customer j in period t for amount q) and inventory holding cost (for the amount q) is calculated. The largest positive value is found. If the vehicle capacity and customer storage limits are satisfied with the amount q then add amount q to delivery amount and the transportation cost is updated. If the vehicle capacity is not available then evaluate the next maximum positive saving. The process is done sequentially from the first period onward and comparison of transportation and inventory costs is done in each period. The future demand in a certain period is to be considered only if the customer's preceding period demand is fulfilled. The customer's storage limits are presented by period limit and savings are controlled until time period of each customer. The algorithm neglects the effect of changes in transportation cost in period t that may result from changing delivery amounts in that period. The algorithm does not guarantee optimality to the solution of this problem however, it can provide good solutions.

4.1.3 Modified Heuristic

In this study, the nearest heuristic algorithm is used to decide vehicle routes, and inventory algorithm is used to decide the delivery amounts in every period. In modified heuristic, suppliers are sorted based on ascending distance. The vehicle selects the nearest supplier to warehouse. The remaining vehicle capacity is calculated. The item in the nearest supplier to current location is received to vehicle until the vehicle capacity constraint is violated. If there are no suppliers that can be added to route, the inventory algorithm is applied. In the developed algorithm, the orders cannot be split across multiple vehicles as opposite of MIP.

In our inventory algorithm, the transportation cost saving when a supplier is removed from the route in the next period is calculated. Abdelmaguid et al. (2008) consider transportation cost saving as a function of delivery amount. In our algorithm, transportation cost saving is considered as a function of distance. The route of a vehicle changes every period. It is assumed that the route will be same in the next period during transportation cost saving calculation. We consider that supplier j will be visited after supplier i in the next period. Vehicle v receives the demand of period k as inventory in period t by consideration of vehicle capacity and storage capacity constraints. The received inventory is held on warehouse until the demand period. The inventory holding cost is calculated by multiplying unit inventory holding cost (h_{it}), the inventory amount that receives in period t for period k from supplier i(I_{ikt}), and the inventory holding period (k-t). The transportation cost saving is calculated by multiplying the unit variable distance cost (c_{ij}), and differences between the distance if the supplier i is visited in period t or not. The cost difference between transportation cost saving and inventory holding cost is calculated for period k and sorts in descending. The demand that provides the maximum positive saving is received in period t. The delivery amount will be total of demand of period t and demand of period k for supplier i. If the difference is 0 or less than 0, delivery of this inventory will cause increase on total cost. As a result of this, the inventory will not be received before the demand period. If the following events are occurred, the same controls are done from period k+1 to the end of the planning horizon.

- If there is no positive saving in period k.
- If the demand which has positive saving does not provide the storage or vehicle constraints in period k.

In Abdelmaguid et al.(2008) algorithm's, the future demand is only received if the preceding demand is fulfilled. In our algorithm, there is no constraint about preceding demand. If the current demand is fulfilled, the demand of any period which minimizes the total cost can be received. The modified nearest neighbor heuristic is shown in Figure 4.1 and described at following. The code of heuristic is given in Appendix B.

Step 0. Initialize an empty route for the next vehicle.
Step 1. Sort suppliers based on distance from warehouse and select the nearest supplier to warehouse.
Step 2. Obtain the nearest supplier location to the previously located supplier. Go on until violate vehicle capacity constraint.
Step 3. If there is no demand for period t which is not violated vehicle capacity constraint, check if there is a demand in next periods which provides minimizing the cost of total system. TR is total distance in a route and AR is the distance if supplier i has not visited in that route. Find the supplier i that has the largest positive value. (TR-AR)*c_{ij}-((k-t)*h_{ik}*d_{ik}). If demand of supplier i for period k does not violate capacity and storage constraints then add d_{ik} (demand of product of supplier i for period t) to i's delivery. If no suitable item is found then if there are available vehicles left run to step 0; otherwise, go to step 4.
Step 4. Increase period t+1 and return to Step 0.

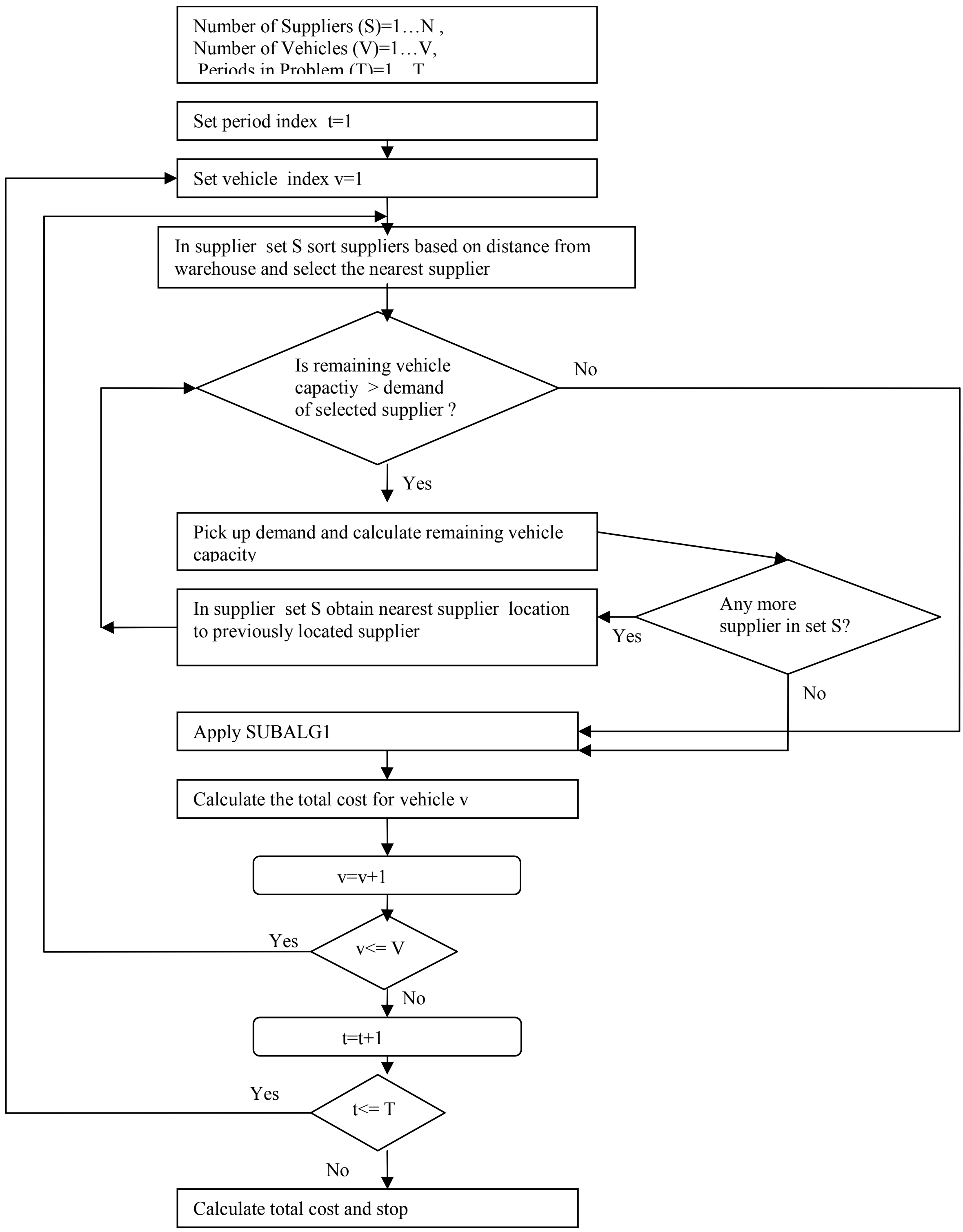

Figure 4.1 Nearest customer heuristic

The algorithm, which is related with inventory decision, is shown in Figure 4.2 .The inventory holding cost is occurred at the warehouse.

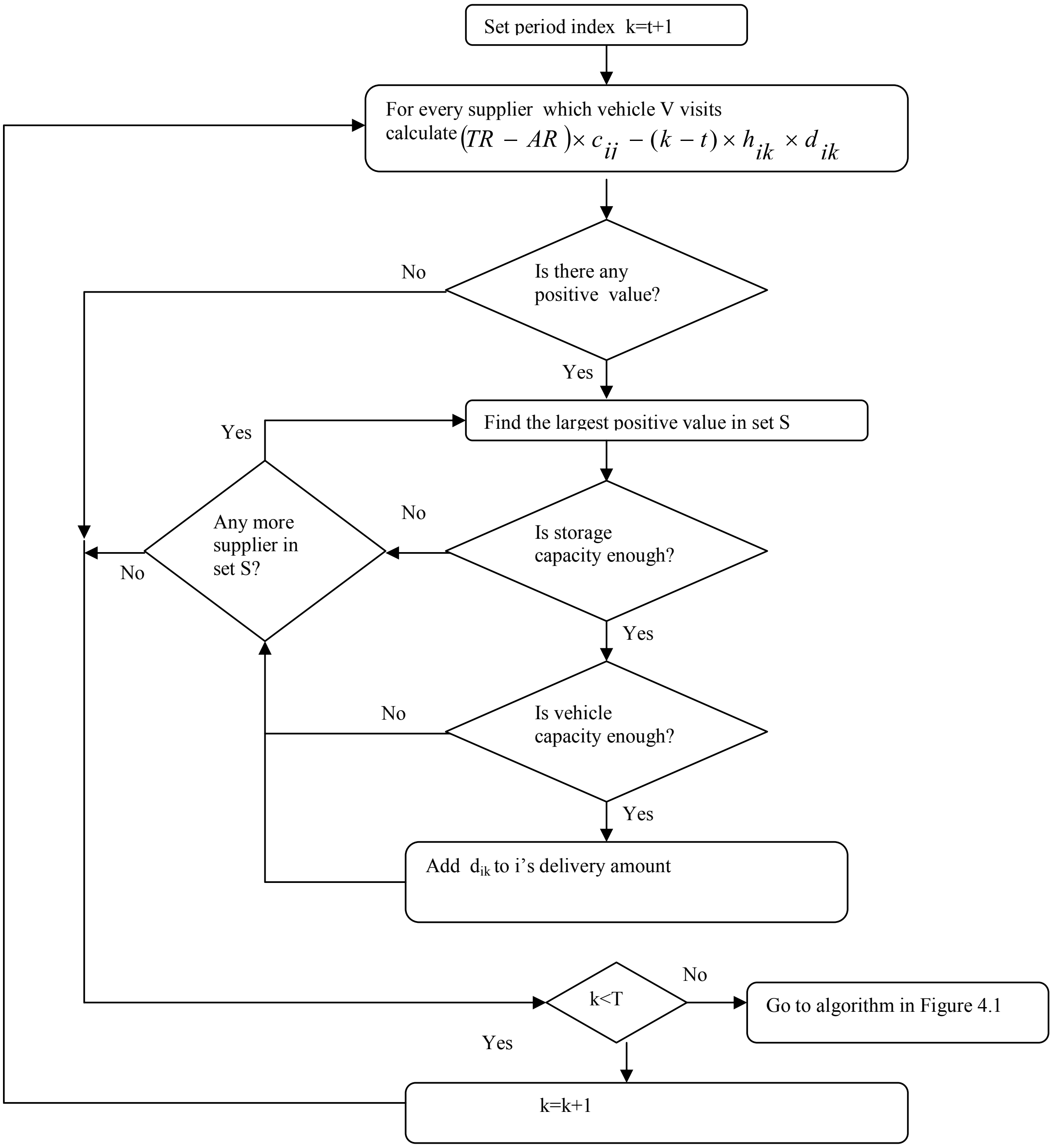

Figure 4.2 SUBALG1

To explain the modified heuristic more clearly, a small example is presented following.

There are 2 suppliers, 2 periods and 1 vehicle on our example. The distances, the demands of retailers for items and the storage capacity for items of suppliers are given in Table 4.1. The vehicle capacity is 300 pcs. The unit inventory holding cost (h)=0.2 €/pcs; the fixed vehicle cost (f_t)=100€ /vehicle and unit transportation cost (c_{ij})=2 € /km.

Table 4.1 Values for small example

	Distances 1	**Distances 2**	**Demand Period 1**	**Demand Period 2**	**Stock Capacity**
0	15	20			
1		30	100	80	50
2			120	60	80

Solution:. The route is started with supplier 1 as it is the nearest supplier to warehouse. The delivery amount is 100 pcs then remaining vehicle capacity is calculated as 300-100 =200 pcs. Then the nearest customer is selected and checked if the remaining vehicle capacity is enough for the demand or not. In our example, it is enough for vehicle 1 visits to supplier 2 and picks up 120 pcs. The remaining vehicle capacity is 200-120=80 pcs. If there is no available supplier's demand for remaining vehicle capacity, the heuristic will continue with inventory decision. We calculate the differences between the transportation cost and inventory holding cost. The results are as following.

The route in period 1 will be 0-1-2 -0 and total distance is 15+30+20=65km. If the vehicle does not visit the supplier 1 in period 2, the route will be 0-2-0. The distance will be 20+20=40km and the difference will be 65-40=25km. As a result, saving cost for supplier 1 in period 2 is 25*2- 80*0.2= 34 €.

If the vehicle does not visit the supplier 2 in period 2, the route will be 0-1-0. The distance will be 15+15=30km and the difference will be 65-30=35km. As a result, saving cost for supplier 2 in period 2 is 35*2- 60*0.2=58 € .

The largest positive saving is 58 €. If the demand satisfies the vehicle and storage constraints, we should receive the demand for item of supplier 2 for period 2 in period 1. The demand for item of supplier 2 for period 2 is 60 pcs. Storage capacity for supplier 2 is 80 pcs and it is more than 60 pcs. The remaining vehicle capacity is 80 pcs and enough for the demand of period 2. Both of the constraints are satisfied so we can delivery the next period demand of retailer 2 as inventory. The supplier 2 will not be visited in period 2. The saving cost for supplier 1 is positive, we should check for constraints. The remaining vehicle capacity is 80-60 =20 pcs however the demand for item of supplier 1 in period 2 is 80 pcs so the vehicle constraint is not satisfied. The total cost for vehicle 1 in period 1 is 100+(15+30+20)*2+ 60*0.2=242 €. There is no demand for vehicle 1 to pick up in period 1. The vehicle visits only supplier 1 in period 2. The cost of vehicle 1 in period 2 is 100+30*2=160 €. The total cost of system is 242+160=402 €.

The total cost of system is calculated by ignoring the inventory decision. The cost of system with holding inventory and without holding inventory is compared. The nearest customer is supplier 1. The demand for item of supplier 1 is 100 pcs. The remaining vehicle capacity is calculated as 300-100 =200 pcs. Then, the nearest customer is selected and checked if the remaining vehicle capacity enough for the demand or not. In our example, vehicle capacity is enough to visit supplier 2. Vehicle visits supplier 1 and then 2 in each period. The total cost of vehicle 1 in period 1 is 100+ (15+30+20)*2= 230 €. The cost of vehicle 1 in period 2 is 100 +(15+30+20)*2=230 €. The total cost of system is 230+230=460 €.

By considering inventory decision, the total cost of system is decreased from 460€ to 402 €. The cost saving is 58 € and the saving rate is 12.6%

CHAPTER FIVE
COMPUTATIONAL RESULTS

In this chapter, we explain our computational experiments, our purposes of performing them, and the environment we work in. In Section 5.1, general scheme of test problems and the values used in experiments are presented. The differences between the cost of system with holding inventory and without holding inventory in warehouse are presented in Section 5.2. The results obtained in each experiment are provided in Section 5.2.

The experiments start with a preliminary run set, in which a distribution system composed of 3-8 suppliers and 2 periods is considered. The purpose of these runs is to obtain optimum solutions for MIP and to examine the quality of the heuristic we developed.

The second part of preliminary experiments has been carried out considering distribution systems involving 3 suppliers and 2-6 periods. The aim of the runs performed with this problem set is to gain an insight into the changes in the performance of heuristic as the period increases.

The last part of experiments (main experiment) performed involves test problems with from 10 to 20 suppliers and from 3 to 10 periods. These runs are carried out to test the performance of heuristic in large-scale problems.

Heuristic is implemented by using Turbo C 2.0 programming language on a PC with a 2,19 GHz AMD Athlon and 1GB of RAM. MIP for optimal solutions is solved by LINGO 9.0 that is unlimited. LINGO is used to solve the model, which is presented in Chapter 3. Command window of LINGO for a model which consists of 5 suppliers, 2 vehicles and 2 periods is presented in Appendix A. Solution report of given model is given in Appendix C.

5.1 Test Problems

The all test problem's parameters are produced randomly. The distances parameters between suppliers and warehouse are given in Table 5.1.

Table 5.1 Distance (km) between suppliers and warehouse

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	35	85	50	45	28	32	60	54
1		80	75	5	35	40	58	60
2			50	35	52	12	42	27
3				20	40	8	45	16
4					75	10	32	24
5						80	14	22
6							18	52
7								11
8								

The constant values, which are used in experiments, are as following;

c_{ij}=2 € /km
h_{it}=0.2 €/pcs
f_t=100 € /vehicle

The retailers inform their demands to warehouse. The demand of retailers by consideration of inventory at the warehouse are picked up. The demands of retailers for each period are given in Table 5.2. These values are used on preliminary experiments I and II.

Table 5.2 Demands of retailers for item of supplier i

Supplier	Period 1(pcs)	Period 2(pcs)	Period 3(pcs)	Period 4(pcs)	Period 5(pcs)	Period 6(pcs)
1	800	1000	1200	950	1100	550
2	600	800	550	1200	850	750
3	850	500	1100	650	1250	950
4	700	700				
5	950	600				
6	500	1100				
7	450	520				
8	900	300				

The warehouse has different stock capacity for different suppliers. The stock capacities for suppliers are given in Table 5.3

Table 5.3 Stock capacity on warehouse for different supplier

Supplier	Stock Capacity (pcs)
1	500
2	800
3	800
4	700
5	600
6	800
7	600
8	800

The vehicle capacity is constant and 2000 pcs for experiment I and II. The number of vehicles in a problem is changed according to dimension of test problems. The vehicle capacities and number of vehicles for experiment III are given in Table 5.4

Table 5.4 Number of vehicles and vehicle capacity for experiment III

Supplier Number	Vehicle Capacity	Number of Vehicles
10	2000	5
12	2500	4
14	3000	5
16	3000	5
18	4000	4
20	4500	4

The values that are given above are used for compare of the heuristic and LINGO solutions. The values that are used on experiment III to test satisfactory of heuristic for large problems are presented at the following tables. The stock capacities on warehouse for 9 suppliers to 20 suppliers are given in Table 5.5.

Table 5.5 Stock capacity in warehouse for items of 9 suppliers to 20 suppliers

Supplier	Stock Capacity (pcs)
9	800
10	800
11	900
12	800
13	900
14	900
15	900
16	800
17	800
18	800
19	800
20	900

The values for 8 suppliers are same with test values of experiment I and II. In table 5.6, the distances between 20 suppliers and warehouse are presented.

Table 5.6 Distance (km) between suppliers and warehouse for large problem

	9	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
0	64	39	28	45	67	75	45	68	81	30	75	36
1	45	45	84	24	72	65	29	48	57	70	36	39
2	45	25	61	62	50	75	17	36	27	64	54	84
3	56	45	49	68	81	30	45	35	61	72	38	48
4	62	35	50	28	50	26	34	36	72	30	51	45
5	81	62	45	75	36	65	56	39	45	75	38	26
6	54	75	52	36	28	25	16	52	60	45	40	35
7	75	45	30	38	54	72	65	58	72	52	25	36
8	65	65	26	56	20	82	45	45	34	60	36	45
9		45	48	38	27	28	85	54	64	75	48	38
10			56	62	36	27	75	45	35	45	82	45
11				84	64	35	54	62	64	82	37	52
12					54	42	38	75	23	38	65	64
13						35	28	16	52	42	45	43
14							65	45	38	45	75	42
15								75	35	28	20	80
16									29	75	45	65
17										36	84	45
18											54	45
19												28
20												

The demands for items of 20 suppliers for 10 periods are presented in Table 5.7.

Table 5.7 Demand of retailers for item of supplier i (for 10 periods and for 20 suppliers)

SUPPLIERS	PERIODS									
	1	2	3	4	5	6	7	8	9	10
1	800	1000	1200	950	1100	550	670	800	1050	1000
2	600	800	550	1200	850	750	760	850	960	980
3	850	500	1100	650	1250	950	650	780	950	950
4	700	700	800	650	550	900	700	850	1100	680
5	950	600	700	900	1000	850	900	900	700	850
6	500	1000	850	900	550	800	1200	1000	950	950
7	450	520	800	700	650	900	990	950	850	850
8	900	300	550	650	700	850	1100	1050	700	750
9	1200	650	700	850	850	800	850	1000	1100	1050
10	860	650	850	900	900	950	900	950	900	850
11	800	700	750	700	950	980	850	800	940	950
12	900	850	850	800	800	900	870	900	550	960
13	700	650	750	700	950	750	900	840	850	800
14	950	800	900	850	850	700	900	900	800	900
15	950	850	1000	950	650	880	950	800	710	850
16	650	950	600	1000	500	600	850	400	750	850
17	800	500	400	700	1300	500	1200	600	800	850
18	1100	500	800	400	650	900	550	650	850	900
19	1050	900	500	950	500	700	400	1100	800	900
20	1000	650	1000	500	800	850	500	900	950	800

5.2 Test Results

In this section, the results obtained throughout the three parts of experiments are presented sequentially. The abbreviations used in presenting the results are given below:

- CPU time: It denotes the solution time (in seconds) of the method under consideration.
- TC: It is the total cost incurred in the solution obtained with the method under consideration.
- IGap %: 100*(opt-Heuristic)/opt: It denotes the gap between the optimal solution value of a MIP formulation (denoted by opt) and the solution value of heuristics.
- S-P-V: It is the supplier, period and number of vehicles used in test problem.
- MNC: It is the calculations of proposed modified nearest customer heuristic

Although the inventory level is eliminated from the formulations during the experiments, the total costs given in the results include the inventory holding cost incurred on the inventory level.

5.2.1 Differences of Total Cost With and Without Holding Inventory

In IRP problems, inventory holding is considered. Vehicles can transport only the monthly demand however, in our problem the vehicles can carry the monthly demand and inventory for next months. By this way, the total distance is decreased and the inventory holding cost is increased. The total cost with consideration of holding inventory and without holding inventory is calculated. The MIP is solved for 3 suppliers in different periods for 2 options. As there is no inventory in the solution of MIP for 3 suppliers and 2 period problem, the differences between holding inventory and not holding inventory is 0. The results for 2 periods and 3 suppliers problem is not presented on calculation table, because the differences is 0. The calculation results are presented in Table 5.8.

As a result, the gap between the total cost with holding inventory and the total cost without holding inventory is average 4.4%. In Figure 5.1 illustrates the differences between the cost with holding cost and without holding cost.

Table 5.8 Differences of total cost with and without inventory

S-P-V	TC With Inventory	TC Without Inventory	IGAP(%)
3-3-2	1960	2130	8.67%
3-4-2	2780	2840	2.15%
3-5-2	3510	3640	3.7%
3-6-2	4220	4350	3.08%
Average			**4.4%**

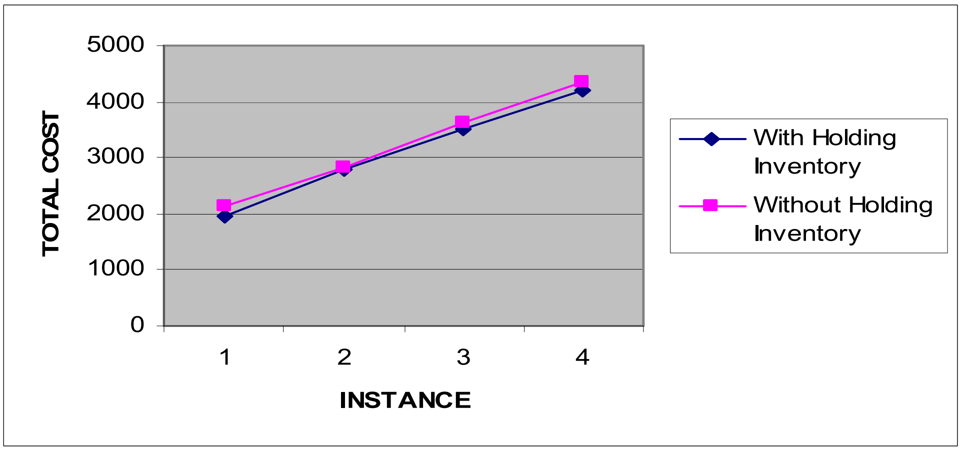

Figure 5.1 Total cost change with and without holding inventory

5.2.2 Results of Preliminary Experiment I

In this section, experiments are tested for a system that includes 3-8 suppliers and 2 periods. The purpose of these runs is to compare the solutions that are obtained by MIP and by developed heuristic. The gap between MIP and heuristic with increase of supplier number is presented.

8 test instances are generated according to the values presented in Section 5.1.The instances are solved with the LINGO and optimal solutions are obtained. The minimum

total cost obtained by solving MIP and heuristic, the times (in CPU seconds) required to solve the related models, and the integrality gap between the optimal solution of MIP and heuristic values are given in Table 5.9.

The average solution time is 14301 CPU time for the MIP formulation and it decreases to less than 5 seconds in all instances with the heuristic. The average solution time is more than 3.97 hours for MIP, which is a sign of the fact that we may not be able obtain optimal solutions for MIP in reasonable times for larger problems. The solution for 8-2-3 provides by LINGO is not optimal, the solution time is more than 15 hours and then best objective value after 15 hours is accepted as optimal.

Table 5.9 Results of PE-I

	MIP SOLVED BY LINGO			**MNC Heuristic**		
S-P-V	**Iterations**	**CPU Time**	**TC**	**CPU Time**	**TC**	**IGAP(%)**
3-2-2	1022	1	1420	3	1548	9%
4-2-2	96616	9	1440	4	1480	2.7%
****5-2-2**	413827	66	1552	3	1675	7.9%
6-2-3	20219113	3911	1759	3	1872	6.4%
7-2-3	113922266	25149	1864	3	2167	16.3%
***8-2-3**	235223965	56700	1896	3	2214	16.8%
Average		**14301**		**3.16**		**9.85%**

Solution report of LINGO for the problem 5-2-2 is given in Appendix C.

The heuristic provides solutions with integrality gaps 9.85 % on average. The gap in 8-2-3 is the largest one. On the other hand, the solution of MIP is not optimal for 8 suppliers. Figure 5.2 illustrates the differences between total cost in MIP and heuristic for experiment I.

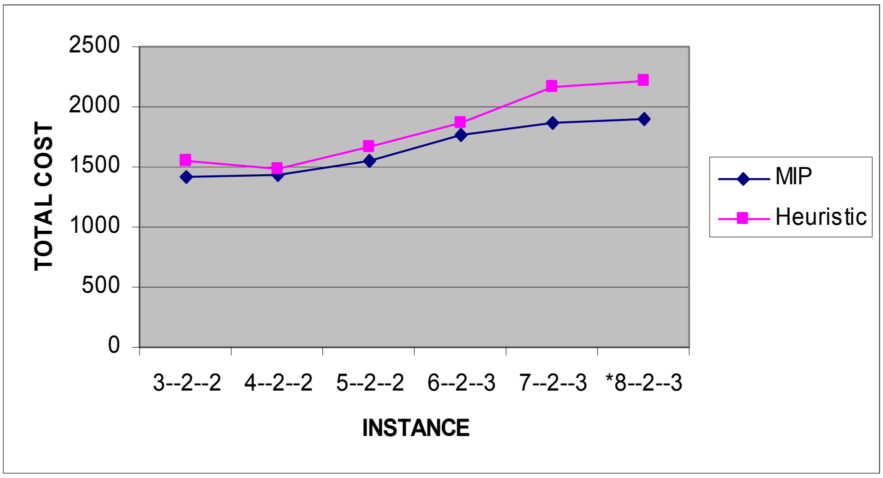

Figure 5.2 Total costs for MIP and heuristic in experiment I

5.2.3 Results of Preliminary Experiment II

The second part of preliminary experiments has been carried out considering distribution systems involving 3 suppliers and 2-6 periods. The aim of the runs performed with this problem set is to gain an insight into the changes in the performance of heuristic as the period increases.

5 test instances are generated according to the values presented in Section 5.1. In this experiment, the solution of MIP and heuristic for 3 suppliers, 2 vehicles and changing period from 2-6 are compared. The instances are solved with the LINGO and optimal solutions are obtained. The minimum total cost obtained by solving MIP and heuristic, the times (in CPU time seconds) required to solve the related models, and the integrality gap between the optimal solution of MIP and heuristic values are given in Table 5.10.

Table 5.10 Results of PE-II

	MIP SOLVED BY LINGO			MNC Heuristic		
S-P-V	**Iterations**	**CPU Time**	**TC**	**CPU Time**	**TC**	**IGAP(%)**
3-2-2	1022	1	1420	3	1548	9%
3-3-2	1940	2	1960	5	2232	13.8%
3-4-2	2596	1	2780	5	2961	6.5%
3-5-2	167501	39	3510	5	3843	9.4%
3-6-2	184190	192	4220	5	4491	6.45
Average		**47**		**4.6**		**9.02%**

The heuristic provides solutions with gaps that range between 6 % and 14% and of 9.02 % on average. The average solution time is 47 seconds for MIP and 4.6 CPU time for heuristic. In Figure 5.3, the results are presented by graphic.

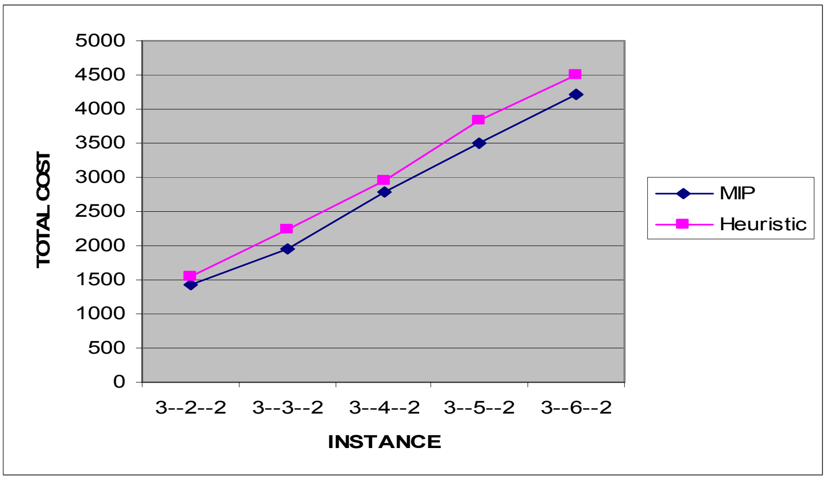

Figure 5.3 Total costs for MIP and heuristic in experiment II

5.2.4 Results of Main Experiment (ME)

In the main experiment involves test problems with 10 to 20 suppliers and 3 to 10 periods. These runs are carried out to test the performances of heuristic in large-scale problems.

Since the main experiments are performed for large instances, MIP formulation of models will not run for optimal solution because of long calculation times. The minimum total costs obtained from heuristic for 10 to 20 suppliers for 3 to 10 periods and CPU time to solve heuristic are given in Table 5.11.

Table 5.11 Results of ME

S-P	**CPU**	**TC**	**S-P**	**CPU**	**TC**
10-3	5	4404	10-4	7	5850
10-5	11	7509	10-6	12	8917
10-7	14	10450	10-8	15	11984
10-9	16	13505	10-10	16	14599
12-3	6	4536	12-4	7	6139
12-5	11	7743	12-6	13	9345
12-7	15	10888	12-8	15	12452
12-9	16	13942	12-10	17	15433
14-3	6	4818	14-4	8	7502
14-5	13	7950	14-6	14	9855
14-7	16	11606	14-8	20	13357
14-9	21	15109	14-10	21	16779
16-3	7	5229	16-4	8	6890
16-5	13	9036	16-6	18	10629
16-7	18	12636	16-8	22	14470
16-9	22	16349	16-10	25	18028
18-3	8	5083	18-4	9	6723
18-5	15	8586	18-6	17	10060
18-7	20	12249	18-8	22	14004
18-9	25	15942	18-10	28	17881
20-3	8	5707	20-4	13	7347
20-5	18	9266	20-6	21	10575
20-7	29	12999	20-8	31	15028
20-9	34	16882	20-10	36	18736

The longest CPU time is 36 seconds for 20 suppliers in 10 periods. It is less when compared to MIP solving time. In Figure 5.4, the CPU times in period 10 for different supplier numbers are illustrated. The CPU time is increased with an increase in the number of suppliers.

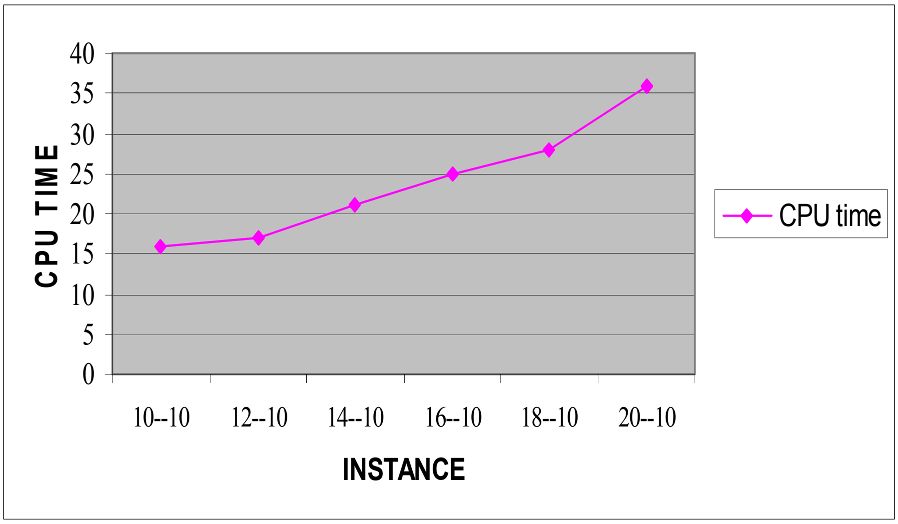

Figure 5.4 The CPU time in 10 periods for different supplier number

In Figure 5.5, the total costs in period 10 for different supplier numbers are illustrated. The total cost is affected from number of vehicles and vehicle capacity. The total cost for 16 suppliers is higher than the total cost for 18 suppliers in period 10. There are 5 vehicles for a system that has 16 suppliers. However, there are 4 vehicles with more capacity for a system that has 18 suppliers.

The CPUs for the problem with 16 suppliers for different periods are given in Figure 5.6. For same supplier number in different periods CPU time is increasing with increasing of period number.

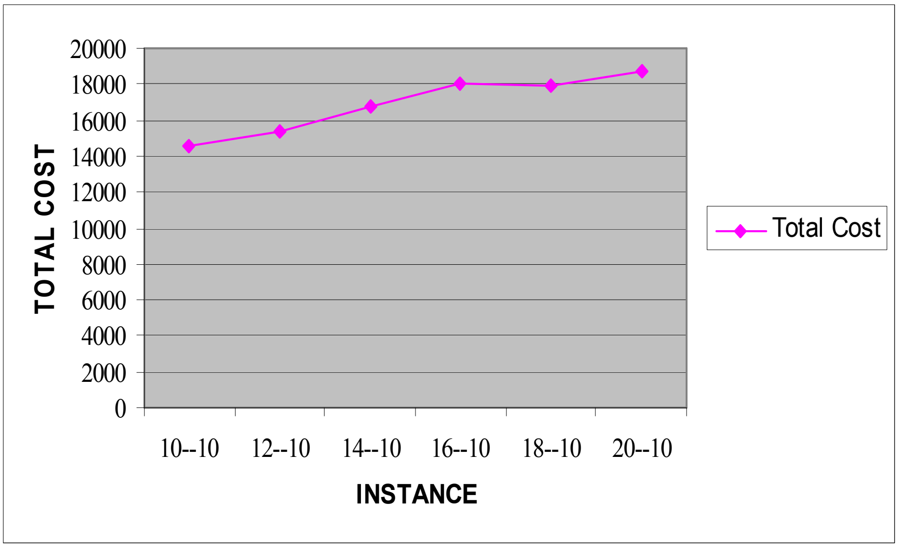

Figure 5.5 The total cost in 10 periods for different supplier number

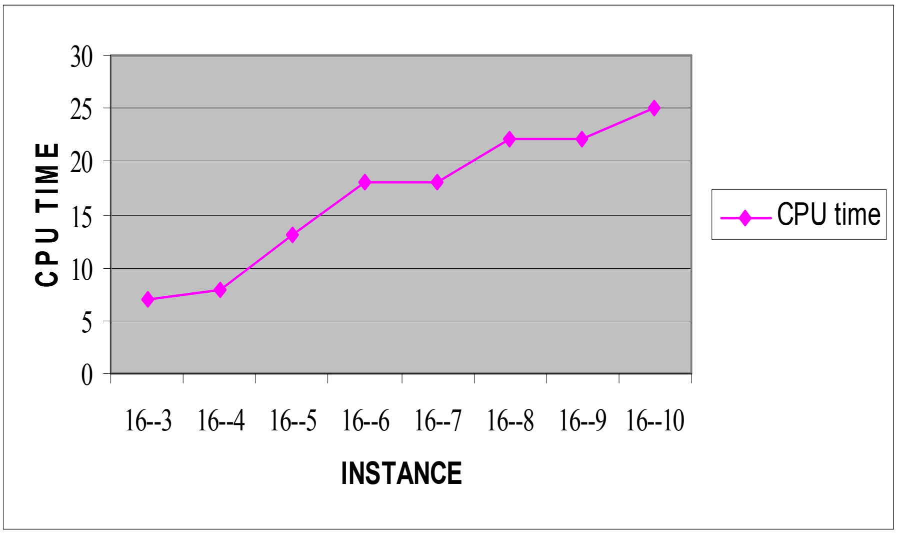

Figure 5.6 CPU for 16 suppliers in different periods

The total costs for the problem with 16 suppliers for different periods are given in Figure 5.7. The total cost is increasing with increasing of period. The number of vehicles and vehicle capacity are constant for same supplier number.

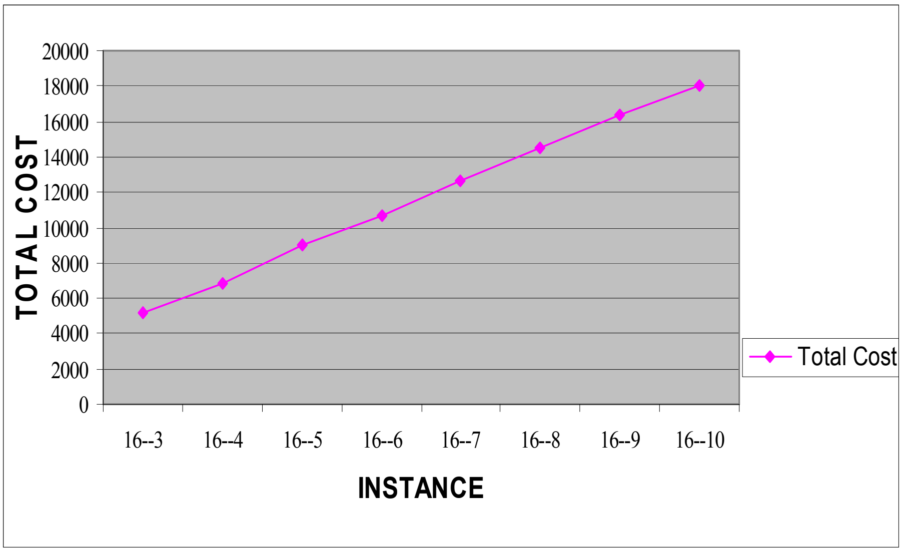

Figure 5.7 Total cost for 16 suppliers in different periods

CHAPTER SIX
CONCLUSION

6.1 Conclusion

In this study, an inventory routing problem with vehicle and storage constraint has been analyzed and an integrated mathematical model for the IRP has been developed. Our problem is developed from the problem of Sindhuchao et al (2005). There is storage constraint in warehouse in our problem that is common in real life. However, this is not considered on literature.

Optimal solutions for the problem could be identified with the MIP for small problem instances. The MIP provided for the IRP turned out to be a complex model due to several inherent vehicle routing problems that needed to be solved and large number of integer variables included in its formulation. Complexity of problem is increased as parabolic. On the other hand, the MIP for IRP includes vehicle routing problem (VRP). Thus, even if the inventory related decisions are excluded from the formulation, it is required to identify solutions to VRP to solve model. The VRP is known to be NP-hard. In addition to the routing difficulty, the complexity caused by other decisions included in our model motivates us to develop heuristic procedures to solve the IRP.

Considering difficulties, we developed a heuristic that provides solution to IRP. In this study, the nearest heuristic algorithm is used for vehicle routes and inventory algorithm is used for deciding the delivery amounts for every period. The procedure used to determine delivery amounts must take into consideration the tradeoff existing between inventory and transportation costs. In heuristic, we consider the receiving of next month demand as inventory to decrease the total cost of system. Heuristic is implemented by using Turbo C 2.0 programming language and MIP is solved by LINGO 9.0.

The developed heuristic is tested with 3 different experiments. In experiment I and II the result of heuristic is compared with the result of MIP for small instances. In experiment III, the heuristic performance for large problems is proved. In experiment I, a system that includes 3-8 suppliers and 2 periods is tested. The purpose of these runs is to check the gap between MIP and heuristic with increase of supplier number in same period. The average solution time for MIP is near 4 CPU hours however it is less than 5 seconds in heuristic. The heuristics provides solutions with integrality gaps 9.85 % on average. The results of experiments show that the heuristic performed well.

The second part of preliminary experiments has been carried out considering distribution systems involving 3 suppliers and 2-6 periods. The aim of the runs performed with this problem set is to gain an insight into the changes in the performances of heuristic identifying procedures as the period increases. The solution of MIP and heuristic for 3 suppliers, 2 vehicles and changing period from 2-6 are compared. The instances are solved with the LINGO and optimal solutions are obtained. The gap between the total costs given by MIP and heuristic ranges from 6 % to 14 % with an average of 9.02 %. The average solution times 47 CPU time for MIP and 4.6 CPU time for heuristic.

In the main experiment involves test problems with 10 to 20 suppliers and 3 to10 periods. These runs are carried out to test the performances of heuristic in large-scale problems. Since the main experiments are performed for large instances, MIP formulation of models will not run for optimal solution because of long calculation times. The longest CPU is 36 seconds for 20 suppliers in 10 periods. It is much less when compared to MIP solving time. The total cost is affected from number of vehicles and vehicle capacity. The total cost for 16 suppliers is higher than the total cost for 18 suppliers in 10 periods. For problem with 16 suppliers, there are 5 vehicles however for 18 suppliers there is 4 vehicles with more capacity.

For same supplier number in different periods CPU time is increasing with increasing of period number. The total cost is increasing with increasing of period in same supplier number.

6.2 Future Directions

In this study, one different item for each supplier is considered. For future directions, multiple items in one supplier can be considered. Extra variables should add to model to represent each item.

The backlog is not considered in this thesis. In real life, there is always a backlog possibility. The backlog can be considered in future researches. The backlog demand may have priority during pick up.

In our problem, the demands are deterministic; a problem in which the demands are stochastic can be developed.

REFERENCES

Abdelmaguid, T.F., Dessouky, M., & Ordonez, F. (2008) A constructive heuristic for the integrated inventory–distribution problem, *Proceedings of the ninth CairoUniversity International Conference on Mechanical Design and Production*

Anily, S.,& Federgruen, A. (1990). One warehouse multiple retailer systems with vehicle routing costs. *Management Science,* 36 (1), 92-114.

Anily, S.,& Federgruen, A. (1993). Two-echelon distribution systems with vehicle routing costs and central inventories. *Operations Research ,*41 (1), 37-47.

Anily, S. (1994). The general multi-retailer EOQ problem with vehicle routing costs. *European Journal of Operational Research,* 79, 451-473.

Baita, F., Ukovich, W., Pesenti, R., & Favaretto, D. (1998). Dynamic routing- inventory problems: A review. *Transportation Research,* 32A, 585-598.

Bell, W.J., Dalberto, L.M., Fisher, M.L., Greenfield, A.J., Jaikumar, R., Kedia, P., et al,(1983). Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces,* 13 (6), 4-23.

Bertazzi, L., Paletta, G., & Speranza, M.G. (2002). Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science,* 36 (1), 119-132.

Bertazzi, L., Paletta, G., & Speranza, M.G. (2005). Minimizing the total cost in an integrated vendor–managed inventory system. *Journal of Heuristics,* 11, 393–419.

Campbell, A.M., Clarke, L.W., & Savelsbergh, M.W.P.(2002). Inventory routing in practice. *The vehicle routing problem,* (2nd ed.) (309-330). Philadelphia: Society for Industrial and Applied Mathematics

Chan, L.M.A., Federgruen, A., & Simchi-Levi, D. (1998). Probabilistic analysis and practical algorithms for inventory-routing models. *Operations Research,* 46 (1), 96-106.

Chandra, P.(1993). A Dynamic distribution model with warehouse and customer replenishment requirements. *Journal of the Operational Research Society,* 44 (7), 681-692.

Chien, T.W., Balakrishnan, A., & Wong, R.T.(1989). An integrated inventory allocation and vehicle routing problem. *Transportation Science,* 23 (2), 67-76.

Dror, M., & Ball, M. (1987). Inventory routing: Reduction from an annual to a short period problem. *Naval Research Logistics* 34, 891-905.

Federgruen, A., & Simchi-Levi, D. (1995). Analysis of vehicle routing and inventory-routing problems. *Handbooks in OR & MS Vol. 8, edited by M.O. Ball et al*. Elsevier Science B.V., 297-373.

Fumero, F., & Vercellis, C. (1999). Synchronized development of production, inventory, and distribution schedules. *Transportation Science,* 33 (3), 330-340.

Gallego, G., & Simchi-Levi, D. (1990). On the effectiveness of direct shipping strategy for the one-warehouse multi-retailer systems. *Management Science,* 36 (2), 240-243.

Kleywegt,A., Nori,V.,& Savelsbergh, M.W.P.(2002). Dynamic programming approximations for a stochastic inventory, *Working Paper, School of Industrial and System Engineering, Georgia Institute of Technolocy*, Atlanta, GA,

Lee C-G, Bozer YA., &White III CC. (2003). A heuristic approach and properties of optimal solutions to the dynamic inventory routing problem. *Working Paper, University of Toronto, Toronto, Ontario, Canada*

Moin, N.H., & Salhi, S. (2007). Inventory routing problems: a logistical overview. *Journal of The Operational Research Society*, (58),1185-1194.

Savelsbergh, M., & Song, J.H. (2006). An optimization algorithm for the inventory routing problem with continuous moves. *Computers& Operations Research*, (35), 2266-2282.

Solomon, M.M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, (35), 254–265.

Sindhuchao, S.,Romejin, E., Akçalı, E., & Boondiskulchok, R. (2005). An integrated inventory-routing system for multi-item joint replenishment with limited vehicle capacity *Journal of Global Optimization*, (32), 93-118.

Yu, Y.,Chen, H., & Chu, F. (2008). A new model and hybrid approach for scale inventory routing problems. *European Journal of Operational Research,* (189), 1022–1040

Appendix A: LINGO 9.0 Commands For Model

```
MIN
=((35*X1011+80*X1121+75*X1131+5*X1141+35*X1151+35*X1101+85*X1021+80*X12
11+50*X1231+35*X1241+52*X1251+85*X1201+50*X1031+75*X1311+50*X1321+20*X1
341+40*X1351+50*X1301+45*X1041+28*X1051+28*X1501+45*X1401+35*X1421+5*X1
411+20*X1431+75*X1451+35*X1511+52*X1521+40*X1531+75*X1541+35*X2011+80*X
2121+75*X2131+5*X2141+35*X2151+35*X2101+85*X2021+80*X2211+50*X2231+35*X
2241+52*X2251+85*X2201+50*X2031+75*X2311+50*X2321+20*X2341+40*X2351+50*
X2301+45*X2041+45*X2401+28*X2051+28*X2501+35*X2421+5*X2411+20*X2431+75*
X2451++35*X2511+52*X2521+40*X2531+75*X2541+35*X1012+80*X1122+75*X1132+5
*X1142+35*X1152+35*X1102+85*X1022+80*X1212+50*X1232+35*X1242+52*X1252+8
5*X1202+50*X1032+75*X1312+50*X1322+20*X1342+40*X1352+50*X1302+45*X1042+
28*X1052+28*X1502+45*X1402+35*X1422+20*X1432+75*X1452+5*X1412+35*X1512+
52*X1522+40*X1532+75*X1542+35*X2012+80*X2122+75*X2132+5*X2142+35*X2102+
35*X2152+85*X2022+80*X2212+50*X2232+35*X2242+52*X2252+85*X2202+
50*X2032+75*X2312+50*X2322+20*X2342+40*X2352+50*X2302+45*X2042+28*X2052
+28*X2502+45*X2402+35*X2422+20*X2432+75*X2452+35*X2512+52*X2522+40*X253
2+75*X2542+5*X2412)*2+
100*X1011+100*X1021+100*X1031+100*X1041+100*X1051+
100*X2011+100*X2021+100*X2031+100*X2041+100*X2051+
100*X1012+100*X1022+100*X1032+100*X1042+100*X1052+
100*X2012+100*X2022+100*X2032+100*X2042+100*X2052+
+0.2*I121+
0.2*I221+0.2*I321+0.2*I421+0.2*I521);

X1011+X1021+X1031+X1041+X1051=1;
X1121+X1131+X1101+X1141+X1151<=1;
X1211+X1201+X1231+X1241+X1251<=1;
X1311+X1321+X1301+X1341+X1351<=1;
X1411+X1421+X1401+X1431+X1451<=1;
X1511+X1521+X1501+X1531+X1541<=1;

X2011+X2021+X2031+X2041+X2051=1;
X2121+X2131+X2101+X2141+X2151<=1;
X2211+X2201+X2231+X2241+X2251<=1;
X2311+X2321+X2301+X2341+X2351<=1;
```

```
X2411+X2421+X2401+X2431+X2451<=1;
X2511+X2521+X2501+X2531+X2541<=1;

X1012+X1022+X1032+X1042+X1052=1;
X1122+X1132+X1102+X1142+X1152<=1;
X1212+X1202+X1232+X1242+X1252<=1;
X1312+X1322+X1302+X1342+X1352<=1;
X1412+X1422+X1402+X1432+X1452<=1;
X1512+X1522+X1502+X1532+X1542<=1;

X2012+X2022+X2032+X2042+X2052=1;
X2122+X2132+X2102+X2142+X2152<=1;
X2212+X2202+X2232+X2242+X2252<=1;
X2312+X2322+X2302+X2342+X2352<=1;
X2412+X2422+X2402+X2432+X2452<=1;
X2512+X2522+X2502+X2532+X2542<=1;


X1121+X1211<=1;
X1131+X1311<=1;
X1141+X1411<=1;
X1151+X1511<=1;
X1231+X1321<=1;
X1241+X1421<=1;
X1251+X1521<=1;
X1341+X1341<=1;
X1351+X1531<=1;
X1451+X1541<=1;

X1122+X1212<=1;
X1132+X1312<=1;
X1142+X1412<=1;
X1152+X1512<=1;
X1232+X1322<=1;
X1242+X1422<=1;
X1252+X1522<=1;
```

```
X1342+X1342<=1;
X1352+X1532<=1;
X1452+X1542<=1;

X2121+X2211<=1;
X2131+X2311<=1;
X2141+X2411<=1;
X2151+X2511<=1;
X2231+X2321<=1;
X2241+X2421<=1;
X2251+X2521<=1;
X2341+X2341<=1;
X2351+X2531<=1;
X2451+X2541<=1;

X2122+X2212<=1;
X2132+X2312<=1;
X2142+X2412<=1;
X2152+X2512<=1;
X2232+X2322<=1;
X2242+X2422<=1;
X2252+X2522<=1;
X2342+X2342<=1;
X2352+X2532<=1;
X2452+X2542<=1;


X1011+X1021+X1031+X1041+X1051-X1101-X1201-X1301-X1401-X1501=0;
X1101+X1121+X1131+X1141+X1151-X1011-X1211-X1311-X1411-X1511=0;
X1201+X1211+X1231+X1241+X1251-X1021-X1121-X1321-X1421-X1521=0;
X1301+X1311+X1321+X1341+X1351-X1031-X1231-X1131-X1431-X1531=0;
X1401+X1411+X1421+X1431+X1451-X1041-X1241-X1141-X1341-X1541=0;
X1501+X1511+X1521+X1531+X1541-X1051-X1251-X1151-X1351-X1451=0;

X2011+X2021+X2031+X2041+x2051-X2101-X2201-X2301-X2401-X2501=0;
X2101+X2121+X2131+X2141+X2151-X2011-X2211-X2311-X2411-X2511=0;
```

```
X2201+X2211+X2231+X2241+X2251-X2021-X2121-X2321-X2421-X1521=0;
X2301+X2311+X2321+X2341+X2351-X2031-X2231-X2131-X2431-X2531=0;
X2401+X2411+X2421+X2431+X2531-X2041-X2241-X2141-X2341-X2541=0;
X2501+X2511+X2521+X2531+X2541-X2051-X2521-X2151-X2351-X2451=0;


X1012+X1022+X1032+X1042+X1052-X1102-X1202-X1302-X1402-X1502=0;
X1102+X1122+X1132+X1142+X1152-X1012-X1212-X1312-X1412-X1512=0;
X1202+X1212+X1232+X1242+X1252-X1022-X1122-X1322-X1422-X1522=0;
X1302+X1312+X1322+X1342+X1352-X1032-X1232-X1132-X1432-X1532=0;
X1402+X1412+X1422+X1432+X1532-X1042-X1242-X1142-X1342-X1352=0;
X1502+X1512+X1522+X1532+X1542-X1052-X1522-X1152-X1352-X1452=0;

X2012+X2022+X2032+X2042+X2052-X2102-X2202-X2302-X2402-X2502=0;
X2102+X2122+X2132+X2142+X2152-X2012-X2212-X2312-X2412-X2512=0;
X2202+X2212+X2232+X2242+X2252-X2022-X2122-X2322-X2422-X2522=0;
X2302+X2312+X2322+X2342+X2352-X2032-X2232-X2132-X2432-X2532=0;
X2402+X2412+X2422+X2432+X2452-X2042-X2242-X2142-X2342-X2542=0;
X2502+X2512+X2522+X2532+X2542-X2052-X2522-X2152-X2352-X2452=0;

Y1101<=2000*X1101;
Y1121<=2000*X1121;
Y1131<=2000*X1131;
Y1141<=2000*X1141;
Y1151<=2000*X1151;

Y1201<=2000*X1201;
Y1211<=2000*X1211;
Y1231<=2000*X1231;
Y1241<=2000*X1241;
Y1251<=2000*X1251;

Y1301<=2000*X1301;
Y1311<=2000*X1311;
Y1321<=2000*X1321;
Y1341<=2000*X1341;
```

```
Y1351<=2000*X1351;

Y1401<=2000*X1401;
Y1411<=2000*X1411;
Y1421<=2000*X1421;
Y1431<=2000*X1431;
Y1451<=2000*X1451;

Y1501<=2000*X1501;
Y1511<=2000*X1511;
Y1521<=2000*X1521;
Y1531<=2000*X1531;
Y1541<=2000*X1541;

Y2101<=2000*X2101;
Y2121<=2000*X2121;
Y2131<=2000*X2131;
Y2141<=2000*X2141;
Y2151<=2000*X2151;

Y2201<=2000*X2201;
Y2211<=2000*X2211;
Y2231<=2000*X2231;
Y2241<=2000*X2241;
Y2251<=2000*X2251;

Y2301<=2000*X2301;
Y2311<=2000*X2311;
Y2321<=2000*X2321;
Y2341<=2000*X2341;
Y2351<=2000*X2351;

Y2401<=2000*X2401;
Y2411<=2000*X2411;
Y2421<=2000*X2421;
Y2431<=2000*X2431;
```

```
Y2451<=2000*X2451;

Y2501<=2000*X2501;
Y2511<=2000*X2511;
Y2521<=2000*X2521;
Y2531<=2000*X2531;
Y2541<=2000*X2541;

Y1102<=2000*X1102;
Y1122<=2000*X1122;
Y1132<=2000*X1132;
Y1142<=2000*X1142;
Y1152<=2000*X1152;

Y1202<=2000*X1202;
Y1212<=2000*X1212;
Y1232<=2000*X1232;
Y1242<=2000*X1242;
Y1252<=2000*X1252;

Y1302<=2000*X1302;
Y1312<=2000*X1312;
Y1322<=2000*X1322;
Y1342<=2000*X1342;
Y1352<=2000*X1352;

Y1402<=2000*X1402;
Y1412<=2000*X1412;
Y1422<=2000*X1422;
Y1432<=2000*X1432;
Y1452<=2000*X1452;

Y1502<=2000*X1502;
Y1512<=2000*X1512;
Y1522<=2000*X1522;
Y1532<=2000*X1532;
```

```
Y1542<=2000*X1542;

Y2102<=2000*X2102;
Y2122<=2000*X2122;
Y2132<=2000*X2132;
Y2142<=2000*X2142;
Y2152<=2000*X2152;

Y2202<=2000*X2202;
Y2212<=2000*X2212;
Y2232<=2000*X2232;
Y2242<=2000*X2242;
Y2252<=2000*X2252;

Y2302<=2000*X2302;
Y2312<=2000*X2312;
Y2322<=2000*X2322;
Y2342<=2000*X2342;
Y2352<=2000*X2352;

Y2402<=2000*X2402;
Y2412<=2000*X2412;
Y2422<=2000*X2422;
Y2432<=2000*X2432;
Y2452<=2000*X2452;

Y2502<=2000*X2502;
Y2512<=2000*X2512;
Y2522<=2000*X2522;
Y2532<=2000*X2532;
Y2542<=2000*X2542;


Y1101+Y1301+Y1401+Y1501+Y1121+Y1131+Y1141+Y1151+Y1231+Y1241+Y1251+Y1321
+Y1341+Y1351+Y1201+Y1211+Y1311+Y1431+Y1421+Y1411+Y1451+Y1511+Y1521+Y153
1+Y1541<=2000;
```

```
Y2101+Y2301+Y2401+Y2121+Y2131+Y2141+Y2231+Y2241+Y2321+Y2341+Y2201+Y2211
+Y2311+Y2431+Y2421+Y2411+Y2501+Y2151+Y2251+Y2351+Y2451+Y2511+Y2521+Y253
1+Y2541<=2000;

Y1102+Y1302+Y1402+Y1122+Y1132+Y1142+Y1232+Y1242+Y1322+Y1342+Y1202+Y1212
+Y1312+Y1432+Y1422+Y1412+Y1502+Y1152+Y1252+Y1352+Y1452+Y1512+Y1522+Y153
2+Y1542<=2000;

Y2102+Y2302+Y2402+Y2122+Y2132+Y2142+Y2232+Y2242+Y2322+Y2342+Y2202+Y2212
+Y2312+Y2432+Y2422+Y2412+Y2502+Y2152+Y2252+Y2352+Y2452+Y2512+Y2522+Y253
2+Y2542<=2000;

Y1011+Y1211+Y1311+Y1411+Y1511-Y1101-Y1121-Y1131-Y1141-Y1151<=0;
Y1021+Y1121+Y1321+Y1421+Y1521-Y1201-Y1211-Y1231-Y1241-Y1251<=0;
Y1031+Y1131+Y1231+Y1431+Y1531-Y1301-Y1311-Y1321-Y1341-Y1351<=0;
Y1041+Y1141+Y1241+Y1341+Y1541-Y1401-Y1411-Y1421-Y1431-Y1451<=0;
Y1051+Y1151+Y1251+Y1351+Y1451-Y1501-Y1511-Y1521-Y1531-Y1541<=0;

Y2011+Y2211+Y2311+Y2411+Y2511-Y2101-Y2121-Y2131-Y2141-Y2151<=0;
Y2021+Y2121+Y2321+Y2421+Y2521-Y2201-Y2211-Y2231-Y2241-Y2251<=0;
Y2031+Y2131+Y2231+Y2431+Y2531-Y2301-Y2311-Y2321-Y2341-Y2351<=0;
Y2041+Y2141+Y2241+Y2341+Y2541-Y2401-Y2411-Y2421-Y2431-Y2451<=0;
Y2051+Y2151+Y2251+Y2351+Y2451-Y2501-Y2511-Y2521-Y2531-Y2541<=0;

Y1012+Y1212+Y1312+Y1412+Y1512-Y1102-Y1122-Y1132-Y1142-Y1152<=0;
Y1022+Y1122+Y1322+Y1422+Y1522-Y1202-Y1212-Y1232-Y1242-Y1252<=0;
Y1032+Y1132+Y1232+Y1432+Y1532-Y1302-Y1312-Y1322-Y1342-Y1352<=0;
Y1042+Y1142+Y1242+Y1342+Y1542-Y1402-Y1412-Y1422-Y1432-Y1452<=0;
Y1052+Y1152+Y1252+Y1352+Y1452-Y1502-Y1512-Y1522-Y1532-Y1542<=0;

Y2012+Y2212+Y2312+Y2412+Y2512-Y2102-Y2122-Y2132-Y2142-Y2152<=0;
Y2022+Y2122+Y2322+Y2422+Y2522-Y2202-Y2212-Y2232-Y2242-Y2252<=0;
Y2032+Y2132+Y2232+Y2432+Y2532-Y2302-Y2312-Y2322-Y2342-Y2352<=0;
Y2042+Y2142+Y2242+Y2342+Y2352-Y2402-Y2412-Y2422-Y2432-Y2452<=0;
Y2052+Y2152+Y2252+Y2352+Y2452-Y2502-Y2512-Y2522-Y2532-Y2542<=0;
```

```
Y1101+Y1121+Y1131+Y1141+Y1151+Y2101+Y2121+Y2131+Y2141+Y2151-I121=800;
Y1102+Y1122+Y1132+Y1142+Y1152+Y2102+Y2122+Y2132+Y2142+Y2152+I121=1000;
Y1201+Y1211+Y1231+Y1241+Y1251+Y2201+Y2211+Y2231+Y2241+Y2251-I221=600;
Y1202+Y1212+Y1232+Y1242+Y1252+Y2202+Y2212+Y2232+Y2242+Y2252+I221=800;
Y1301+Y1321+Y1311+Y1341+Y1351+Y2301+Y2321+Y2311+Y2341+Y2351-I321=850;
Y1302+Y1322+Y1312+Y1342+Y1352+Y2302+Y2322+Y2312+Y2342+Y2352+I321=500;
Y1401+Y1421+Y1411+Y1431+Y1451+Y2401+Y2421+Y2411+Y2431+Y2451-I421=700;
Y1402+Y1422+Y1412+Y1432+Y1452+Y2402+Y2422+Y2412+Y2432+Y2452+I421=700;
Y1501+Y1521+Y1511+Y1531+Y1541+Y2501+Y2521+Y2511+Y2531+Y2541-I521=950;
Y1502+Y1522+Y1512+Y1532+Y1542+Y2502+Y2522+Y2512+Y2532+Y2542+I521=600;

I121<=500;
I221<=800;
I321<=800;
I421<=700;
I521<=600;

@BIN( X1011);
@BIN( X1021);
@BIN( X1031);
@BIN( X1041);
@BIN( X1051);

@BIN( X1101);
@BIN( X1121);
@BIN( X1131);
@BIN( X1141);
@BIN( X1151);

@BIN( X1201);
@BIN( X1211);
@BIN( X1231);
@BIN( X1241);
@BIN( X1251);
@BIN( X1301);
```

```
@BIN( X1311);
@BIN( X1321);
@BIN( X1341);
@BIN( X1351);

@BIN( X1401);
@BIN( X1411);
@BIN( X1421);
@BIN( X1431);
@BIN( X1451);

@BIN( X1501);
@BIN( X1511);
@BIN( X1521);
@BIN( X1531);
@BIN( X1541);

@BIN( X2011);
@BIN( X2021);
@BIN( X2031);
@BIN( X2041);
@BIN( X2051);

@BIN( X2101);
@BIN( X2121);
@BIN( X2131);
@BIN( X2141);
@BIN( X2151);
@BIN( X2201);
@BIN( X2211);
@BIN( X2231);
@BIN( X2241);
@BIN( X2251);

@BIN( X2301);
@BIN( X2311);
```

```
@BIN( X2321);
@BIN( X2341);
@BIN( X2351);

@BIN( X2401);
@BIN( X2411);
@BIN( X2421);
@BIN( X2431);
@BIN( X2451);

@BIN( X2501);
@BIN( X2511);
@BIN( X2521);
@BIN( X2531);
@BIN( X2541);


@BIN( X1012);
@BIN( X1022);
@BIN( X1032);
@BIN( X1042);
@BIN( X1052);

@BIN( X1102);
@BIN( X1122);
@BIN( X1132);
@BIN( X1142);
@BIN( X1152);

@BIN( X1202);
@BIN( X1212);
@BIN( X1232);
@BIN( X1242);
@BIN( X1252);

@BIN( X1302);
```

```
@BIN( X1312);
@BIN( X1322);
@BIN( X1342);
@BIN( X1352);

@BIN( X1402);
@BIN( X1412);
@BIN( X1422);
@BIN( X1432);
@BIN( X1452);

@BIN( X1502);
@BIN( X1512);
@BIN( X1522);
@BIN( X1532);
@BIN( X1542);

@BIN( X2012);
@BIN( X2022);
@BIN( X2032);
@BIN( X2042);
@BIN( X2052);

@BIN( X2102);
@BIN( X2122);
@BIN( X2132);
@BIN( X2142);
@BIN( X2152);

@BIN( X2202);
@BIN( X2212);
@BIN( X2232);
@BIN( X2242);
@BIN( X2252);

@BIN( X2302);
```

```
@BIN( X2312);
@BIN( X2322);
@BIN( X2342);
@BIN( X2352);

@BIN( X2402);
@BIN( X2412);
@BIN( X2422);
@BIN( X2432);
@BIN( X2452);

@BIN( X2502);
@BIN( X2512);
@BIN( X2522);
@BIN( X2532);
@BIN( X2542);

@GIN( Y1101);
@GIN( Y1201);
@GIN( Y1301);
@GIN( Y1401);
@GIN( Y1501);

@GIN( Y1011);
@GIN( Y1021);
@GIN( Y1031);
@GIN( Y1041);
@GIN( Y1051);

@GIN( Y1121);
@GIN( Y1131);
@GIN( Y1141);
@GIN( Y1151);

@GIN( Y1231);
@GIN( Y1211);
```

```
@GIN( Y1241);
@GIN( Y1251);

@GIN( Y1311);
@GIN( Y1321);
@GIN( Y1341);
@GIN( Y1351);

@GIN( Y1411);
@GIN( Y1421);
@GIN( Y1431);
@GIN( Y1451);

@GIN( Y1511);
@GIN( Y1521);
@GIN( Y1531);
@GIN( Y1541);

@GIN( Y2101);
@GIN( Y2201);
@GIN( Y2301);
@GIN( Y2401);
@GIN( Y2501);

@GIN( Y2011);
@GIN( Y2021);
@GIN( Y2031);
@GIN( Y2041);
@GIN( Y2051);

@GIN( Y2121);
@GIN( Y2131);
@GIN( Y2141);
@GIN( Y2151);

@GIN( Y2231);
```

```
@GIN( Y2211);
@GIN( Y2241);
@GIN( Y2251);

@GIN( Y2311);
@GIN( Y2321);
@GIN( Y2341);
@GIN( Y2351);

@GIN( Y2411);
@GIN( Y2421);
@GIN( Y2431);
@GIN( Y2451);

@GIN( Y2511);
@GIN( Y2521);
@GIN( Y2531);
@GIN( Y2541);

@GIN( Y1102);
@GIN( Y1202);
@GIN( Y1302);
@GIN( Y1402);
@GIN( Y1502);

@GIN( Y1012);
@GIN( Y1022);
@GIN( Y1032);
@GIN( Y1042);
@GIN( Y1052);

@GIN( Y1122);
@GIN( Y1132);
@GIN( Y1142);
@GIN( Y1152);
```

```
@GIN( Y1232);
@GIN( Y1212);
@GIN( Y1242);
@GIN( Y1252);
@GIN( Y1312);
@GIN( Y1322);
@GIN( Y1342);
@GIN( Y1352);
@GIN( Y1412);
@GIN( Y1422);
@GIN( Y1432);
@GIN( Y1452);
@GIN( Y1512);
@GIN( Y1522);
@GIN( Y1532);
@GIN( Y1542);
@GIN( Y2102);
@GIN( Y2202);
@GIN( Y2302);
@GIN( Y2402);
@GIN( Y2502);
@GIN( Y2012);
@GIN( Y2022);
@GIN( Y2032);
@GIN( Y2042);
@GIN( Y2052);
@GIN( Y2122);
@GIN( Y2132);
@GIN( Y2142);
@GIN( Y2152);
@GIN( Y2232);
@GIN( Y2212);
@GIN( Y2242);
@GIN( Y2252);
@GIN( Y2312);
@GIN( Y2322);
```

```
@GIN( Y2342);
@GIN( Y2352);
@GIN( Y2412);
@GIN( Y2422);
@GIN( Y2432);
@GIN( Y2452);
@GIN( Y2512);
@GIN( Y2522);
@GIN( Y2532);
@GIN( Y2542);
@GIN( I121);
@GIN( I221);
@GIN( I321);
@GIN( I421);
```

Appendix B: Turbo C Commands For Model

```
#include "stdio.h"
#include "conio.h"
#include "stdlib.h"
#include "math.h"
#include "time.h"
#include "stddef.h"

char ch;

int mesafe[50][50]={0};
int hesapmesafe[50][50]={0};
int maliyethesap[5][100]={0};
int genelguzergah[50]={0};
int aracguzergah[5][50]={0};
int donemseltalep[50][50]={0};
int hesapdonemseltalep[50][50]={0};
int stokalani[50]={0};
int bufferstokalani[50]={0};
int arac_anlik_deger[5]={0};
int kalan_arackapasitesi[5]={0};

float tasimamaliyeti=0;
float stokmaliyeti=0;
float toplammaliyet=0;

time_t ilk,son;

int maliyetmesafe=0;
```

```
int maliyetstok=0;
int tedarikno=0;
int aracsayisi=0;
int period=0;
int aracsabitmaliyet=0;
int arackapasitesi=0;
int i,j,indeks=0;
int a=0;
int b=0;
int tedarikcikontrol=0;
int cikisflag=1;
int donembittiflag=1;

int minmesafe=30000;
int indekstedarik=0;
int indeksmesafe=0;

int indeksdonem=0;
int indeksperiod=0;
int indeksguzergah=0;
int indeksgenelguzergah=0;
int genelguzergahbuffer=0;
int guzergahbuffer=0;
int aracno=0;
int aractursayisi=1;
int ekranabasA=0;
int ekranabasB=0;
int ekranabasC=0;

void sirala(void)
```

```
{
 minmesafe=30000;
 for(j=0;j<=tedarikno;j++)
 { if((mesafe[indekstedarik][j] < minmesafe)&&(mesafe[indekstedarik][j] > 0))
     {minmesafe=mesafe[indekstedarik][j];
      indeksmesafe=j;
     }
  }

}

/*araç guzregah,anlık deger,kalan deger bas*/
void ekranabas(void)
{
   clrscr();
   /*
   printf("\n MESAFELER \n");
   for(i=0;i<=tedarikno;i++){
   for(j=0;j<=tedarikno;j++)
   {printf("%d  ",mesafe[i][j]);}
   printf("\n");          }
     */
   printf("\n GUZERGAH \n");
   for(i=0;i<=4;i++){
   printf("%d nolu aracin guzergahi----> ",i+1);
   for(j=0;j<=tedarikno;j++)
   {printf("%d    ",aracguzergah[i][j]);}
   printf("\n");    }

   printf("\n TALEPLER \n");
```

```
for(i=1;i<=tedarikno;i++){
printf("%d nolu tedarikci donemsel talepleri----> ",i);
for(j=0;j<period;j++)
{printf("%d  ",donemseltalep[i][j]);}
printf("\n");         }

printf("\n Arac Tasima Degerleri\n");
for(i=0;i<=aracno;i++){if(arac_anlik_deger[i]>2){
printf("\n %d nolu aracin tasima degeri---> %d",i+1,arac_anlik_deger[i]);}}

getch();
printf("\n");

printf("\n ALAN KAPASITELERI DEGERLERI\n");
for(i=1;i<=tedarikno;i++)
printf("\n%d.tedarigin anlik stok kapasitesi----> %d",i,bufferstokalani[i]);

if(maliyethesap[0][aractursayisi]>(aracsabitmaliyet+5)){printf("\n 1 nolu aracin
%d.donem maliyeti ---> %d
",aractursayisi,maliyethesap[0][aractursayisi]);toplammaliyet=(toplammaliyet+maliyeth
esap[0][aractursayisi]);}
if(maliyethesap[1][aractursayisi]>(aracsabitmaliyet+5)){printf("\n 2 nolu aracin
%d.donem maliyeti ---> %d
",aractursayisi,maliyethesap[1][aractursayisi]);toplammaliyet=(toplammaliyet+maliyeth
esap[1][aractursayisi]);}
if(maliyethesap[2][aractursayisi]>(aracsabitmaliyet+5)){printf("\n 3 nolu aracin
%d.donem maliyeti ---> %d
",aractursayisi,maliyethesap[2][aractursayisi]);toplammaliyet=(toplammaliyet+maliyeth
esap[2][aractursayisi]);}
```

```
    if(maliyethesap[3][aractursayisi]>(aracsabitmaliyet+5)){printf("\n 4 nolu aracin
%d.donem maliyeti ---> %d
",aractursayisi,maliyethesap[3][aractursayisi]);toplammaliyet=(toplammaliyet+maliyeth
esap[3][aractursayisi]);}
    if(maliyethesap[4][aractursayisi]>(aracsabitmaliyet+5)){printf("\n 5 nolu aracin
%d.donem maliyeti ---> %d
",aractursayisi,maliyethesap[4][aractursayisi]);toplammaliyet=(toplammaliyet+maliyeth
esap[4][aractursayisi]);}

    getch();
}

void maliyethesapla(void)
{
 int tempA=0;
 int tempB=0;

 for(i=0;i<=indeksguzergah;i++)
{tempA=(tempA+hesapmesafe[aracguzergah[aracno][i]][aracguzergah[aracno][i+1]]);}

maliyethesap[aracno][aractursayisi]=maliyetstok+((tempA*tasimamaliyeti)+aracsabitma
liyet);
 maliyetstok=0;
 ekranabasA=aracno+1;
 ekranabasB=aractursayisi;
 ekranabasC=maliyethesap[aracno][aractursayisi];
 /*printf("\n %d nolu aracin %d.turunun maliyet ---> %d
",aracno+1,aractursayisi,maliyethesap[aracno][aractursayisi]);
 getch();*/
}
```

```
int maliyetkontrol(void)
{
 int c=0;
 int d=1;
 int n=0;
 int karar=0;
 int totalmesafe=0;
 int alternatifmesafe=0;
 int farkmesafemaliyet=0;
 int farkstokmaliyet=0;
 int fark=0;

 while(d)
  {
   switch(c)
    {
     case 0:for(i=0;i<=indeksguzergah;i++)
{totalmesafe=(totalmesafe+hesapmesafe[aracguzergah[aracno][i]][aracguzergah[aracno]
[i+1]]);}
            for(n=0;n<=indeksguzergah;n++) {
                                 if((n!=guzergahbuffer)){
                                                  if((n+1)==guzergahbuffer)
{alternatifmesafe=alternatifmesafe+hesapmesafe[aracguzergah[aracno][n]][aracguzerga
h[aracno][n+2]];}
                                                  else
{alternatifmesafe=alternatifmesafe+hesapmesafe[aracguzergah[aracno][n]][aracguzerga
h[aracno][n+1]];}
                                                 }
                                }
```

```
            farkmesafemaliyet=(totalmesafe - alternatifmesafe)*tasimamaliyeti;
            clrscr();
            printf("\n totalmesafe= %d \n",totalmesafe);
            printf("\n alternatifmesafe= %d \n",alternatifmesafe);
            printf("\n farkmesafemaliyet= %d \n",farkmesafemaliyet);
            c=1;
            /*maliyetmesafe=(totalmesafe*tasimamaliyeti)+aracsabitmaliyet;*/
           break;
      case
1:farkstokmaliyet=donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem]
*stokmaliyeti*(indeksdonem-indeksperiod);
            printf("\n farkstokmaliyet= %d \n",farkstokmaliyet);
            c=2;
           break;
      case 2:fark=farkmesafemaliyet-farkstokmaliyet;
            if(fark>0) karar=1; else karar=0;
            d=0;c=0;
            printf("\n fark= %d \n",fark);getch();
            if(karar) maliyetstok=maliyetstok+farkstokmaliyet;
           break;
      default: d=0;
           break;
    }
  }

 return karar;
}

int stokkapasitekontrol(void)
{
```

```
  int donus=0;

if(donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem]<bufferstokalani
[aracguzergah[aracno][guzergahbuffer]])
  donus=1;
  else donus=0;

  return donus;

}

void initialroutin(void)
{
 for(i=1;i<=tedarikno;i++){if(donemseltalep[i][indeksperiod] < 0)

{bufferstokalani[i]=bufferstokalani[i]+hesapdonemseltalep[i][indeksperiod];}
                        }

 indeksdonem=indeksperiod;
 genelguzergahbuffer=1;
 aracno=0;
 b=0;
 aractursayisi++;

 kalan_arackapasitesi[0]=arackapasitesi;  /*1.arac*/
 kalan_arackapasitesi[1]=arackapasitesi;  /*2.arac*/
 kalan_arackapasitesi[2]=arackapasitesi;  /*3.arac*/
 kalan_arackapasitesi[3]=arackapasitesi;  /*4.arac*/
 kalan_arackapasitesi[4]=arackapasitesi;  /*5.arac*/
 arac_anlik_deger[0]=0;
```

```
 arac_anlik_deger[1]=0;
 arac_anlik_deger[2]=0;
 arac_anlik_deger[3]=0;
 arac_anlik_deger[4]=0;

 for(i=0;i<5;i++)
 for(j=0;j<50;j++)
 aracguzergah[i][j]=0;

 indeksguzergah=1;
 donembittiflag=1;
 cikisflag=1;

}
void kalanyuklemeler(void)
{
 int k=0;
while(cikisflag)
 {
   switch(b)
    {
     case 0:if(donemseltalep[genelguzergah[genelguzergahbuffer]][indeksdonem] > 0)
             { if((donemseltalep[genelguzergah[genelguzergahbuffer]][indeksdonem] <=
kalan_arackapasitesi[aracno])&&(genelguzergahbuffer<=indeksgenelguzergah))
                 {arac_anlik_deger[aracno]=arac_anlik_deger[aracno] +
donemseltalep[genelguzergah[genelguzergahbuffer]][indeksdonem];
                  kalan_arackapasitesi[aracno]=kalan_arackapasitesi[aracno] -
donemseltalep[genelguzergah[genelguzergahbuffer]][indeksdonem];
```

```
if(aracno==0)donemseltalep[genelguzergah[genelguzergahbuffer]][indeksdonem]=-
((aracturs ayisi*10)+1);

if(aracno==1)donemseltalep[genelguzergah[genelguzergahbuffer]][indeksdonem]=-
((aractursayisi*10)+2);

if(aracno==2)donemseltalep[genelguzergah[genelguzergahbuffer]][indeksdonem]=-
((aractursayisi*10)+3);

if(aracno==3)donemseltalep[genelguzergah[genelguzergahbuffer]][indeksdonem]=-
((aractursayisi*10)+4);

if(aracno==4)donemseltalep[genelguzergah[genelguzergahbuffer]][indeksdonem]=-
((aractursayisi*10)+5);

aracguzergah[aracno][indeksguzergah]=genelguzergah[genelguzergahbuffer];
indeksguzergah++;
                  }
               else{b=1;indeksdonem++;indeksguzergah--
;guzergahbuffer=indeksguzergah;k=1;}
              }
            genelguzergahbuffer++;
            if((genelguzergahbuffer > indeksgenelguzergah)&&(!k))
{b=1;indeksdonem++;indeksguzergah--;guzergahbuffer=indeksguzergah;}
         break;
    case 1:while(indeksdonem<period)
          {while(guzergahbuffer>0)
```

```
            {if((donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem] <=
kalan_arackapasitesi[aracno])&&(donemseltalep[aracguzergah[aracno][guzergahbuffer]
][indeksdonem] >0))
               {if(stokkapasitekontrol())
                 { if(maliyetkontrol())
                   {arac_anlik_deger[aracno]=arac_anlik_deger[aracno] +
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem];
                    kalan_arackapasitesi[aracno]=kalan_arackapasitesi[aracno] -
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem];

bufferstokalani[aracguzergah[aracno][guzergahbuffer]]=bufferstokalani[aracguzergah[ar
acno][guzergahbuffer]]-
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem];
                    if(aracno==0)
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem]=-
((aractursayisi*10)+1);
                    if(aracno==1)
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem]=-
((aractursayisi*10)+2);
                    if(aracno==2)
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem]=-
((aractursayisi*10)+3);
                    if(aracno==3)
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem]=-
((aractursayisi*10)+4);
                    if(aracno==4)
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem]=-
((aractursayisi*10)+5);
                   }
                 }
```

```
                }
              guzergahbuffer--;
           }
          indeksdonem++;guzergahbuffer=indeksguzergah;
          }
         if(indeksdonem>=period)   maliyethesapla();
         if(aracno<4)
{aracno++;indeksdonem=indeksperiod;indeksguzergah=1;b=0;genelguzergahbuffer--
;k=0;}
         else b=2;
         /*ekranabas();*/
         break;
    case 2:cikisflag=0;indeksperiod++;
         break;
    default:
         break;
 }
}
}
void aracayukle(void)
{
switch(a)
{
 case 0:if((donemseltalep[indeksmesafe][indeksdonem] <=
kalan_arackapasitesi[aracno])&&(tedarikcikontrol<tedarikno)&&(donemseltalep[indeks
mesafe][indeksdonem]>0))
       { arac_anlik_deger[aracno]=arac_anlik_deger[aracno] +
donemseltalep[indeksmesafe][indeksdonem];
         kalan_arackapasitesi[aracno]=kalan_arackapasitesi[aracno] -
donemseltalep[indeksmesafe][indeksdonem];
```

```
     if(aracno==0)donemseltalep[indeksmesafe][indeksdonem]=-
((aractur, sayisi*10)+1);
```

```
                { arac_anlik_deger[aracno]=arac_anlik_deger[aracno] +
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem];
                 kalan_arackapasitesi[aracno]=kalan_arackapasitesi[aracno] -
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem];

bufferstokalani[aracguzergah[aracno][guzergahbuffer]]=bufferstokalani[aracguzergah[ar
acno][guzergahbuffer]]-
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem];
                 if(aracno==0)
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem]=-
((aractursayisi*10)+1);
                 if(aracno==1)
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem]=-
((aractursayisi*10)+2);
                 if(aracno==2)
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem]=-
((aractursayisi*10)+3);
                 if(aracno==3)
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem]=-
((aractursayisi*10)+4);
                 if(aracno==4)
donemseltalep[aracguzergah[aracno][guzergahbuffer]][indeksdonem]=-
((aractursayisi*10)+5);
                }
                                  }
              }
        guzergahbuffer--;
        }
      indeksdonem++;guzergahbuffer=indeksguzergah;
     }
```

```
        if(indeksdonem>=period)   maliyethesapla();
        if(tedarikcikontrol<tedarikno){ aracno++;indeksdonem=0;indeksguzergah=1;a=0;}

if(tedarikcikontrol>=tedarikno){a=2;indeksdonem=0;indeksperiod=1;indeksgenelguzerg
ah--;}

         break;

 case 2:ekranabas();
        initialroutin();
        kalanyuklemeler();
        if(indeksperiod==period)  {ekranabas();a=3;}
        break;

 case 3:ekranabas();
        break;

default:break;
}
}

void baslangic(void)
{
 kalan_arackapasitesi[0]=arackapasitesi;  /*1.arac*/
 kalan_arackapasitesi[1]=arackapasitesi;  /*2.arac*/
 kalan_arackapasitesi[2]=arackapasitesi;  /*3.arac*/
 kalan_arackapasitesi[3]=arackapasitesi;  /*4.arac*/
 kalan_arackapasitesi[4]=arackapasitesi;  /*5.arac*/
 aracguzergah[0][0]=0; /*1.arac*/
 aracguzergah[1][0]=0; /*2.arac*/
```

```
 aracguzergah[2][0]=0; /*3.arac*/
 aracguzergah[3][0]=0; /*4.arac*/
 aracguzergah[4][0]=0; /*5.arac*/
 indeksguzergah=0;
 indeksguzergah++;
 genelguzergah[indeksgenelguzergah]=0;
 indeksgenelguzergah++;

 for(i=1;i<=tedarikno;i++)  bufferstokalani[i]=stokalani[i];
}

main()
{
 clrscr();
 printf("\n TEDARIKCI SAYISINI GIRINIZ=");
 scanf("%d",&tedarikno);
 for(i=0;i<=tedarikno;i++)
  { for(j=0;j<=tedarikno;j++)
    { if(j>i){
      printf("\n %d nolu tedarikci ile %d nolu tedarikci aras□ mesafe giriniz=",i,j);
      scanf("%d",&mesafe[i][j]);

mesafe[j][i]=mesafe[i][j];hesapmesafe[i][j]=mesafe[i][j];hesapmesafe[j][i]=hesapmesafe
[i][j];}
    }
  }

 for(i=0;i<=tedarikno;i++) {mesafe[i][0]=0;}

 clrscr();
```

```
printf("\n MESAFELER \n");
for(i=0;i<=tedarikno;i++){
for(j=0;j<=tedarikno;j++)
{printf("%d  ",mesafe[i][j]);}
 printf("\n");          }
 getch();

clrscr();
printf("\nDONEM SAYISINI GIRINIZ =");
scanf("%d",&period);
for(i=1;i<=tedarikno;i++)
 { for(j=0;j<period;j++)
    { printf("\n %d nolu tedarikcinin %d donemine ait talebi giriniz =",i,j+1);
      scanf("%d",&donemseltalep[i][j]);hesapdonemseltalep[i][j]=donemseltalep[i][j];
    }
 }
clrscr();
printf("\n DONEMSEL TALEPLER \n");
for(i=1;i<=tedarikno;i++){
for(j=0;j<period;j++)
 {printf("%d  ",donemseltalep[i][j]);}
  printf("\n");         }
  getch();

clrscr();
printf("\n STOK MALIYETINI GIRINIZ=");
scanf("%f",&stokmaliyeti);
printf("\n TASIMA BIRIM MALIYETINI GIRINIZ=");
scanf("%f",&tasimamaliyeti);
printf("\n ARAC MALIYETINI GIRINIZ=");
```

```
 scanf("%d",&aracsabitmaliyet);
 printf("\n ARAC KAPASITESINI GIRINIZ=");
 scanf("%d",&arackapasitesi);

 clrscr();
 printf("\nDEPO STOK ALAN KAPASITESINI GIRINIZ \n");
 for(i=1;i<=tedarikno;i++)
 { printf("\n %d nolu tedarigin alan kapasitesi =",i);
   scanf("%d",&stokalani[i]);
 }
 clrscr();
 printf("\n ALAN KAPASITELERI DEGERLERI\n");
 for(i=1;i<=tedarikno;i++)
 printf("\n %d",stokalani[i]);
 getch();

 ilk=time(NULL);
 baslangic();
 sirala();
while(1){
aracayukle();
if(a==3){son=time(NULL);
printf("\n \n");
printf("\n TOPLAM MALIYET= %f",toplammaliyet);
printf("\n TOPLAM SURE= %d sn.",(unsigned)difftime(son,ilk));
printf("\n PROGRAM BITMISTIR\n");
printf("\n PROGRAMI SONLANDIRMAK ICIN Q TUSUNA BASINIZ");
if((ch=getch())=='q')  break;}
```

Appendix C: Solution Report of LINGO

If the value of variable X is 1 ; it means the vehicle visits the supplier and receive the delivery quantity for the supplier in index 1 . For example : X1151 = 1 vehicle in period 1 visits supplier 5 after the supplier 1 and carry the product of supplier 1.

The route of vehicles are as following according to solution report.

Vehicle 1 in period 1 visits 0-4 -1-5-0 , vehicle 2 in period 1 visits 0-1-4-2-3-0 , vehicle 1 in period 2 visits 0-5-1-0, vehicle 2 in period 2 visits 0-3-2-4-0 , in sequentially . The delivery amounts are meets the demands**.**

```
Global optimal solution found.
   Objective value:                              1552.000
   Extended solver steps:                           17730
   Total solver iterations:                        413827
```

Variable	Value	Reduced Cost
X1011	0.000000	170.0000
X1121	0.000000	160.0000
X1131	0.000000	150.0000
X1141	0.000000	10.00000
X1151	1.000000	70.00000
X1101	0.000000	70.00000
X1021	0.000000	270.0000
X1211	0.000000	160.0000
X1231	0.000000	100.0000
X1241	0.000000	70.00000
X1251	0.000000	104.0000
X1201	0.000000	170.0000
X1031	0.000000	200.0000
X1311	0.000000	150.0000
X1321	0.000000	100.0000
X1341	0.000000	40.00000

X1351	0.000000	80.00000
X1301	0.000000	100.0000
X1041	1.000000	190.0000
X1051	0.000000	156.0000
X1501	1.000000	56.00000
X1401	0.000000	90.00000
X1421	0.000000	70.00000
X1411	1.000000	10.00000
X1431	0.000000	40.00000
X1451	0.000000	150.0000
X1511	0.000000	70.00000
X1521	0.000000	104.0000
X1531	0.000000	80.00000
X1541	0.000000	150.0000
X2011	1.000000	170.0000
X2121	0.000000	160.0000
X2131	0.000000	150.0000
X2141	1.000000	10.00000
X2151	0.000000	70.00000
X2101	0.000000	70.00000
X2021	0.000000	270.0000
X2211	0.000000	160.0000
X2231	1.000000	100.0000
X2241	0.000000	70.00000
X2251	0.000000	104.0000
X2201	0.000000	170.0000
X2031	0.000000	200.0000
X2311	0.000000	150.0000
X2321	0.000000	100.0000
X2341	0.000000	40.00000
X2351	0.000000	80.00000
X2301	1.000000	100.0000
X2041	0.000000	190.0000
X2401	0.000000	90.00000
X2051	0.000000	156.0000
X2501	0.000000	56.00000

X2421	1.000000	70.00000
X2411	0.000000	10.00000
X2431	0.000000	40.00000
X2451	0.000000	150.0000
X2511	0.000000	70.00000
X2521	0.000000	104.0000
X2531	0.000000	80.00000
X2541	0.000000	150.0000
X1012	0.000000	170.0000
X1122	0.000000	160.0000
X1132	0.000000	150.0000
X1142	0.000000	10.00000
X1152	0.000000	70.00000
X1102	1.000000	70.00000
X1022	0.000000	270.0000
X1212	0.000000	160.0000
X1232	0.000000	100.0000
X1242	0.000000	70.00000
X1252	0.000000	104.0000
X1202	0.000000	170.0000
X1032	0.000000	200.0000
X1312	0.000000	150.0000
X1322	0.000000	100.0000
X1342	0.000000	40.00000
X1352	0.000000	80.00000
X1302	0.000000	100.0000
X1042	0.000000	190.0000
X1052	1.000000	156.0000
X1502	0.000000	56.00000
X1402	0.000000	90.00000
X1422	0.000000	70.00000
X1432	0.000000	40.00000
X1452	0.000000	150.0000
X1412	0.000000	10.00000
X1512	1.000000	70.00000
X1522	0.000000	104.0000

X1532	0.000000	80.00000
X1542	0.000000	150.0000
X2012	0.000000	170.0000
X2122	0.000000	160.0000
X2132	0.000000	150.0000
X2142	0.000000	10.00000
X2102	0.000000	70.00000
X2152	0.000000	70.00000
X2022	0.000000	270.0000
X2212	0.000000	160.0000
X2232	0.000000	100.0000
X2242	1.000000	70.00000
X2252	0.000000	104.0000
X2202	0.000000	170.0000
X2032	1.000000	200.0000
X2312	0.000000	150.0000
X2322	1.000000	100.0000
X2342	0.000000	40.00000
X2352	0.000000	80.00000
X2302	0.000000	100.0000
X2042	0.000000	190.0000
X2052	0.000000	156.0000
X2502	0.000000	56.00000
X2402	1.000000	90.00000
X2422	0.000000	70.00000
X2432	0.000000	40.00000
X2452	0.000000	150.0000
X2512	0.000000	70.00000
X2522	0.000000	104.0000
X2532	0.000000	80.00000
X2542	0.000000	150.0000
X2412	0.000000	10.00000
I121	0.000000	0.2000000
I221	100.0000	0.2000000
I321	0.000000	0.2000000
I421	0.000000	0.2000000

I521	0.000000	0.2000000
Y1101	0.000000	0.000000
Y1121	0.000000	0.000000
Y1131	0.000000	0.000000
Y1141	0.000000	0.000000
Y1151	575.0000	0.000000
Y1201	0.000000	0.000000
Y1211	0.000000	0.000000
Y1231	0.000000	0.000000
Y1241	0.000000	0.000000
Y1251	0.000000	0.000000
Y1301	0.000000	0.000000
Y1311	0.000000	0.000000
Y1321	0.000000	0.000000
Y1341	0.000000	0.000000
Y1351	0.000000	0.000000
Y1401	0.000000	0.000000
Y1411	475.0000	0.000000
Y1421	0.000000	0.000000
Y1431	0.000000	0.000000
Y1451	0.000000	0.000000
Y1501	950.0000	0.000000
Y1511	0.000000	0.000000
Y1521	0.000000	0.000000
Y1531	0.000000	0.000000
Y1541	0.000000	0.000000
Y2101	0.000000	0.000000
Y2121	0.000000	0.000000
Y2131	0.000000	0.000000
Y2141	225.0000	0.000000
Y2151	0.000000	0.000000
Y2201	0.000000	0.000000
Y2211	0.000000	0.000000
Y2231	700.0000	0.000000
Y2241	0.000000	0.000000
Y2251	0.000000	0.000000

Y2301	850.0000	0.000000
Y2311	0.000000	0.000000
Y2321	0.000000	0.000000
Y2341	0.000000	0.000000
Y2351	0.000000	0.000000
Y2401	0.000000	0.000000
Y2411	0.000000	0.000000
Y2421	225.0000	0.000000
Y2431	0.000000	0.000000
Y2451	0.000000	0.000000
Y2501	0.000000	0.000000
Y2511	0.000000	0.000000
Y2521	0.000000	0.000000
Y2531	0.000000	0.000000
Y2541	0.000000	0.000000
Y1102	1000.000	0.000000
Y1122	0.000000	0.000000
Y1132	0.000000	0.000000
Y1142	0.000000	0.000000
Y1152	0.000000	0.000000
Y1202	0.000000	0.000000
Y1212	0.000000	0.000000
Y1232	0.000000	0.000000
Y1242	0.000000	0.000000
Y1252	0.000000	0.000000
Y1302	0.000000	0.000000
Y1312	0.000000	0.000000
Y1322	0.000000	0.000000
Y1342	0.000000	0.000000
Y1352	0.000000	0.000000
Y1402	0.000000	0.000000
Y1412	0.000000	0.000000
Y1422	0.000000	0.000000
Y1432	0.000000	0.000000
Y1452	0.000000	0.000000
Y1502	0.000000	0.000000

Y1512	600.0000	0.000000
Y1522	0.000000	0.000000
Y1532	0.000000	0.000000
Y1542	0.000000	0.000000
Y2102	0.000000	0.000000
Y2122	0.000000	0.000000
Y2132	0.000000	0.000000
Y2142	0.000000	0.000000
Y2152	0.000000	0.000000
Y2202	0.000000	0.000000
Y2212	0.000000	0.000000
Y2232	0.000000	0.000000
Y2242	700.0000	0.000000
Y2252	0.000000	0.000000
Y2302	0.000000	0.000000
Y2312	0.000000	0.000000
Y2322	500.0000	0.000000
Y2342	0.000000	0.000000
Y2352	0.000000	0.000000
Y2402	700.0000	0.000000
Y2412	0.000000	0.000000
Y2422	0.000000	0.000000
Y2432	0.000000	0.000000
Y2452	0.000000	0.000000
Y2502	0.000000	0.000000
Y2512	0.000000	0.000000
Y2522	0.000000	0.000000
Y2532	0.000000	0.000000
Y2542	0.000000	0.000000
Y1011	0.000000	0.000000
Y1021	0.000000	0.000000
Y1031	0.000000	0.000000
Y1041	0.000000	0.000000
Y1051	0.000000	0.000000
Y2011	0.000000	0.000000
Y2021	0.000000	0.000000

```
Y2031        0.000000          0.000000
Y2041        0.000000          0.000000
Y2051        0.000000          0.000000
Y1012        0.000000          0.000000
Y1022        0.000000          0.000000
Y1032        0.000000          0.000000
Y1042        0.000000          0.000000
Y1052        0.000000          0.000000
Y2012        0.000000          0.000000
Y2022        0.000000          0.000000
Y2032        0.000000          0.000000
Y2042        0.000000          0.000000
Y2052        0.000000          0.000000

  Row    Slack or Surplus      Dual Price
    1        1552.000         -1.000000
    2        0.000000          0.000000
    3        0.000000          0.000000
    4        1.000000          0.000000
    5        1.000000          0.000000
    6        0.000000          0.000000
    7        0.000000          0.000000
    8        0.000000          0.000000
    9        0.000000          0.000000
   10        0.000000          0.000000
   11        0.000000          0.000000
   12        0.000000          0.000000
   13        1.000000          0.000000
   14        0.000000          0.000000
   15        0.000000          0.000000
   16        1.000000          0.000000
   17        1.000000          0.000000
   18        1.000000          0.000000
   19        0.000000          0.000000
   20        0.000000          0.000000
   21        1.000000          0.000000
```

22	0.000000	0.000000
23	0.000000	0.000000
24	0.000000	0.000000
25	1.000000	0.000000
26	1.000000	0.000000
27	1.000000	0.000000
28	0.000000	0.000000
29	0.000000	0.000000
30	1.000000	0.000000
31	1.000000	0.000000
32	1.000000	0.000000
33	1.000000	0.000000
34	1.000000	0.000000
35	1.000000	0.000000
36	1.000000	0.000000
37	1.000000	0.000000
38	1.000000	0.000000
39	0.000000	0.000000
40	1.000000	0.000000
41	1.000000	0.000000
42	1.000000	0.000000
43	1.000000	0.000000
44	1.000000	0.000000
45	1.000000	0.000000
46	1.000000	0.000000
47	1.000000	0.000000
48	0.000000	0.000000
49	1.000000	0.000000
50	0.000000	0.000000
51	0.000000	0.000000
52	1.000000	0.000000
53	1.000000	0.000000
54	1.000000	0.000000
55	1.000000	0.000000
56	1.000000	0.000000
57	1.000000	0.000000

58	1.000000	0.000000
59	1.000000	0.000000
60	0.000000	0.000000
61	0.000000	0.000000
62	1.000000	0.000000
63	1.000000	0.000000
64	1.000000	0.000000
65	1.000000	0.000000
66	0.000000	0.000000
67	0.000000	0.000000
68	0.000000	0.000000
69	0.000000	0.000000
70	0.000000	0.000000
71	0.000000	0.000000
72	0.000000	0.000000
73	0.000000	0.000000
74	0.000000	0.000000
75	0.000000	0.000000
76	0.000000	0.000000
77	0.000000	0.000000
78	0.000000	0.000000
79	0.000000	0.000000
80	0.000000	0.000000
81	0.000000	0.000000
82	0.000000	0.000000
83	0.000000	0.000000
84	0.000000	0.000000
85	0.000000	0.000000
86	0.000000	0.000000
87	0.000000	0.000000
88	0.000000	0.000000
89	0.000000	0.000000
90	0.000000	0.000000
91	0.000000	0.000000
92	0.000000	0.000000
93	0.000000	0.000000

94	1425.000	0.000000
95	0.000000	0.000000
96	0.000000	0.000000
97	0.000000	0.000000
98	0.000000	0.000000
99	0.000000	0.000000
100	0.000000	0.000000
101	0.000000	0.000000
102	0.000000	0.000000
103	0.000000	0.000000
104	0.000000	0.000000
105	0.000000	0.000000
106	1525.000	0.000000
107	0.000000	0.000000
108	0.000000	0.000000
109	0.000000	0.000000
110	1050.000	0.000000
111	0.000000	0.000000
112	0.000000	0.000000
113	0.000000	0.000000
114	0.000000	0.000000
115	0.000000	0.000000
116	0.000000	0.000000
117	0.000000	0.000000
118	1775.000	0.000000
119	0.000000	0.000000
120	0.000000	0.000000
121	0.000000	0.000000
122	1300.000	0.000000
123	0.000000	0.000000
124	0.000000	0.000000
125	1150.000	0.000000
126	0.000000	0.000000
127	0.000000	0.000000
128	0.000000	0.000000
129	0.000000	0.000000

130	0.000000	0.000000
131	0.000000	0.000000
132	1775.000	0.000000
133	0.000000	0.000000
134	0.000000	0.000000
135	0.000000	0.000000
136	0.000000	0.000000
137	0.000000	0.000000
138	0.000000	0.000000
139	0.000000	0.000000
140	1000.000	0.000000
141	0.000000	0.000000
142	0.000000	0.000000
143	0.000000	0.000000
144	0.000000	0.000000
145	0.000000	0.000000
146	0.000000	0.000000
147	0.000000	0.000000
148	0.000000	0.000000
149	0.000000	0.000000
150	0.000000	0.000000
151	0.000000	0.000000
152	0.000000	0.000000
153	0.000000	0.000000
154	0.000000	0.000000
155	0.000000	0.000000
156	0.000000	0.000000
157	0.000000	0.000000
158	0.000000	0.000000
159	0.000000	0.000000
160	0.000000	0.000000
161	1400.000	0.000000
162	0.000000	0.000000
163	0.000000	0.000000
164	0.000000	0.000000
165	0.000000	0.000000

166	0.000000	0.000000
167	0.000000	0.000000
168	0.000000	0.000000
169	0.000000	0.000000
170	0.000000	0.000000
171	0.000000	0.000000
172	0.000000	0.000000
173	1300.000	0.000000
174	0.000000	0.000000
175	0.000000	0.000000
176	0.000000	0.000000
177	1500.000	0.000000
178	0.000000	0.000000
179	0.000000	0.000000
180	1300.000	0.000000
181	0.000000	0.000000
182	0.000000	0.000000
183	0.000000	0.000000
184	0.000000	0.000000
185	0.000000	0.000000
186	0.000000	0.000000
187	0.000000	0.000000
188	0.000000	0.000000
189	0.000000	0.000000
190	0.000000	0.000000
191	0.000000	0.000000
192	400.0000	0.000000
193	100.0000	0.000000
194	100.0000	0.000000
195	0.000000	0.000000
196	0.000000	0.000000
197	475.0000	0.000000
198	375.0000	0.000000
199	225.0000	0.000000
200	475.0000	0.000000
201	150.0000	0.000000

202	0.000000	0.000000
203	0.000000	0.000000
204	400.0000	0.000000
205	0.000000	0.000000
206	0.000000	0.000000
207	0.000000	0.000000
208	600.0000	0.000000
209	0.000000	0.000000
210	200.0000	0.000000
211	500.0000	0.000000
212	0.000000	0.000000
213	0.000000	0.000000
214	0.000000	0.000000
215	0.000000	0.000000
216	0.000000	0.000000
217	0.000000	0.000000
218	0.000000	0.000000
219	0.000000	0.000000
220	0.000000	0.000000
221	0.000000	0.000000
222	0.000000	0.000000
223	0.000000	0.000000
224	500.0000	0.000000
225	700.0000	0.000000
226	800.0000	0.000000
227	700.0000	0.000000
228	600.0000	0.000000