DESIGN OF A BLOOD BANK NETWORK: CONSIDERING FACILITY LOCATION, INVENTORY AND ROUTING DECISIONS

by

Doğuş Özkök

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Industrial Engineering

Koç University

August, 2013

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Doğuş Özkök

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Assist. Prof. Onur Kaya

Assoc. Prof. Sibel Salman

Assist. Prof. Fadime Üney Yüksektepe

Date:

To my family

ABSTRACT

In this thesis, we design a supply chain network for blood bank distribution by integrating strategic, tactical and operational decisions. These decisions are usually studied separately in classical literature. However, we formulated a mixed integer nonlinear programming (MINLP) model to combine three decisions to minimize total system cost. In blood distribution network of Istanbul, hospitals keep their own inventory and procure bloods from main blood banks via weekly shipments. In the proposed model, some of the hospitals are selected as local blood banks (LBBs) and serve the hospitals which are assigned to them. Thus, our MINLP model solves a complex problem which aims to find optimal number and location of LBBs, assignment of hospitals to opened LBBs and the weekly and daily routes from the main blood bank to LBBs and from LBBs to hospitals. We use exact and approximate solution methods to solve this NP-hard problem. Firstly, small sized instances are solved by using commercial solvers. However, for mid and large sized problems, exact solution failed to find solutions in polynomial time due to increasing complexity. Therefore, we propose a tabu search based heuristic approach to find optimal and near optimal solutions. The performance of the solution methods are analyzed by comparing with each other and the current system costs on 65 test instances.

ÖZETÇE

Bu tezde, kan bankası dağıtımı için stratejik, taktik ve operasyonel kararlar entegre edilerek bir tedarik zinciri şebeke tasarımı yapılmıştır. Bu kararlar, klasik literatürde genellikle birbirinden bağımsız olarak çalışılmaktadır. Ancak, burada kurgulanan karışık tamsayılı doğrusal olmayan programlama (KTDOP) modelinde toplam maliyetin enküçüklenmesi için üç karar birleştirilmiştir. İstanbul'daki kan dağıtım şebekesinde, her hastane kendi envanterini tutmakta ve haftalık sevkiyatlar ile ana kan bankasından kan tedarik etmektedir. Önerilen modelde ise hastanelerin bazıları yerel kan bankası seçilerek, kendilerine atanacak olan hastanelere hizmet edecektir. Bunun için, önerilen KTDOP modeli optimum yerel kan bankası sayısı ve lokasyonu, hastanelerin hangi yerel kan bankalarına atanacağı ve ana kan bankasından yerel kan bankasına, yerel kan bankasından hastanelere yapılacak günlük ve haftalık sevkiyat rotalarnn belirlenmesi gibi karmaşık bir problemi çözmeye çalışmaktadır. Bu NP-zor problemin çözülmesi için tam ve yaklaşık çözüm metotlar kullanılmıştır. Öncelikle, küçük boyutlu problemler ticari çözücüler kullanlarak çözülmüştür. Ancak, orta ve büyük boyutlu problemlerde artan karmaşıklık ile birlikte, tam çözümlü metotlar polinom zamanda sonuç bulmakta başarısız olmuştur. Bu nedenle, en iyi ya da en iyiye yakın çözümler bulunması için sezgisel çözüm yaklaşmlarından benzetilmiş tavlama metodu uygulanması önerilmektedir. Çözüm metotlarının performansları 65 test problemi kullanılarak, birbirleri ve mevcut model maliyetleri ile kıyaslanarak analiz edilmiştir.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my supervisor Assoc. Prof. Onur Kaya for his support and valuable advices during my thesis. As an advisor, he gave me both freedom and guidance during our study. I am especially grateful for his patience and suggestions.

I am also grateful to members of my thesis committee for critical reading of this thesis and for their insightful comments.

I would like to express my appreciation to Yiğit Can Ören, Aslı Pnar Yapıcı, Emin Rodoslu, Zeynep Özsarp, Aylin Polat, Yasin Arslan, Pelin Atıcı, Gökhan Kirlik, Önay Batur, Caner Canyakmaz and all of my friends at Koç University for their friendship and all the good times that we had during two years. A special thanks to Yahya Yavuz not only for his valuable friendship and good personality, but also for his patience and support.

My deep appreciation also goes to Can Öz for his endless help and worth friendship during my master study and thesis. Without him this thesis would definitely not have been possible. Therefore, I'll remember his contributions to this thesis for the rest of my life.

Finally I thank my family, Aynur Özkök, İsmet Özkök and C. Barıs Özkök for their morale support and encouragements. They always believe in me even the times I lost faith and motivation in myself. Thus, I dedicate this thesis to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

SCNDP	Supply Chain Network Design Problem
TRC	Turkish Red Crescent
LBB	Local Blood Bank
MINLP	Mixed Integer Non Linear Programming
DC	Distribution Center
VRP	Vehicle Routing Problem
FLP	Facility Location Problem
IRP	Inventory Routing Problem
LRP	Location Routing Problem
LAP	Location Allocation Problem
WLRP	Warehouse Location Routing Problem
MDVDP	Multi Depot Vehicle Dispatch Problem
WLAP	Warehouse Location Allocation Problem
MDRAP	Multi Depot Routing Allocation Problem
MDLRP	Multi Depot Location Routing Problem
EOQ	Economic Order Quantity
POT	Power of Two
FPP	Fixed Partition Policy
SA	Simulated Annealing
TS	Tabu Search
GE	Genetic Algorithm
ACO	Ant Colony Optimization
VNS	Variable Neighborhood Search
CO	Combinatorial Optimization

Chapter 1

INTRODUCTION

In today's competitive business environment, public and private sector companies have to pay attention to their organizations related with the entire supply chain decisions to increase their efficiency and effectiveness. Hence, managing supply chain and network design decisions have become a major challenge for these firms as they try to reduce their costs and improve their service level. The main decisions of a typical supply chain, are to determine the number, location, and size of the facilities, how to procure items from suppliers and how to distribute them from suppliers to demand points. Consequently, to have an improved system, effective supply chain strategies should include different level of decisions and use an integrated approach.

The supply chain, which is also called as the logistics network, traditionally has three decision levels according to their planning horizon. These are strategic, tactical and operational level decisions. Strategic decisions include logistic network design, facility location, capacity sizing, warehouse layout and fleets sizing that have long-lasting effects. Strategic decisions usually use aggregated data that are based on forecasting. Tactical decisions deal with moderate capital investments, also production and distribution planning and resource allocation which are made on an annual, semi-annual or seasonal time basis. These decisions are based on forecasted disaggregated data. Operational decisions include day to day operations or in real time and have a narrow scope such as daily shipments, vehicle dispatching or order picking. These are the low cost operations and are based on very detailed data.

In our study, we aim to integrate the different level decisions to design a blood bank network of a blood distribution system by considering facility location, inventory decisions and vehicle routing aspect. We realized that design and analysis of blood bank networks from an integrated supply chain network optimization perspective is missing in literature.

The motivation of this thesis comes from the importance of blood banks and blood distribution system. Blood banks are the vital part of the health service systems. Therefore, the applications of blood banks have significant effect on the success of medical treatment procedures. Main functions of a blood bank are blood procurement, cross-matching, storage, distribution, quality control and outdating. Moreover, blood banks are responsible for normal and emergency case blood demand fulfillment of their regions that include hospitals and clinics.

Human blood is the only material that can be used in medical treatments which is voluntarily supplied from people (donors). Hence, blood is a scarce resource that needs special attention. There are 8 different blood types whose frequency changes with respect to population and regional differences. The blood is composed of more than 4000 different kinds of components. Red cells, white cells, platelets and plasma are the most important ones. These components are derived from the main blood after relevant processing.

Our main focus is the red blood cells which are the perishable and expensive blood type. Its lifespan is around 20-30 days. When they are not used within that time, they are considered as outdated and must be destroyed. They are given to patients who suffer from blood loss. The demand for this blood type is highly variable since it is needed in several medical situations. This leads hospitals to face overstock or stockout problems. When they order more than their need, they destroy many units of this expensive blood product. On the contrary, they demand emergency shipments from blood banks which bring significant costs to the system. Hence, a better distribution network for blood banks is needed.

In Turkey, blood services were initiated by Turkish Red Crescent (TRC) in 1950s. TRC operates more than 60 blood centers and many blood stations spread out over the country. They also collect blood using mobile units. Blood is processed at the blood banks or at donation centers. Then blood banks supply hospitals within their regional area. Hospitals demand blood using fax or telephone. Blood bank sends several vehicles which have different routes and fulfill the demand of the hospitals. In the current system, hospitals keep their own inventory that leads to overstock and stockout risks mentioned above. Furthermore, managing inventory and maintaining high service level is a problem for the entire system. Thus, we suggest localization of blood banks to increase efficiency, and benefit from risk pooling advantages. In the proposed distribution network, some of the hospitals will be selected as local blood bank (LBB) to monitor and serve the nearby hospitals. The daily demand of hospitals will be consolidated and satisfied by these local centers via daily shipments. The main blood bank will supply local blood banks via weekly shipments. Therefore, there will be two different vehicle types and the routing problem of daily and weekly shipments.

The framework of the blood bank distribution system is also applicable, with some modifications, to the optimization of other supply chain network problems. Hospitals can be seen as retailers, local blood banks as distribution centers and main blood bank as a supplier. Then, our problem turns to a SCNDP with a goal to find the optimal number and locations of distribution centers, assignment of retailers to the open DCs and inventory levels of a perishable product that will be kept at the open DCs. Moreover, optimal routes of the weekly vehicles that supply the DCs and the daily vehicles that supply the retailers are found.

In conclusion, this study introduces a model for the general supply chain network design problem(SCNDP). In this model, we formulate a mixed integer nonlinear programming problem (MINLP) to combine the strategic, tactical and operation level decisions for finding an optimal network design. Our strategic decision is to find the optimal number and location of distribution centers (DCs). In tactical level, assignment of the retailers to DCs and inventory levels in the DCs is studied. Finally, daily and weekly transportation route decisions are made at the operational level. The problem can be solved optimally for small sized instances by using GAMS's Boron and Cplex solver. However, the complex nature of the problem makes it impossible to solve medium and large sized instances. Therefore, we applied heuristic methods to solve these instances.

The thesis includes six chapters. The relevant literature review for supply chain network design problems are studied in Chapter 2. Chapter 3 introduces the current distribution system and proposed mathematical model and assumptions of the mixed integer nonlinear programming (MINLP) formulations in detail. We also give some insights on computational complexity. Following this chapter, Chapter 4, we described our solution methodologies. We try to find exact and approximate solutions to small size instances of our problem. In addition, lower bounds are found by using piecewise linear approximation techniques. Furhermore, we develop heuristic algorithms to solve the medium and large sized instances of the problem. In Chapter 5, we present details about data generation and discuss the result of computational studies. Finally, in Chapter 6, we summarize our study and provide some future research directions.

Chapter 2

LITERATURE REVIEW

Supply chain network design problems have been widely studied in different sub-problems such as facility location, location allocation, inventory decisions, vehicle routing and scheduling. All of these problems are the parts of a general problem with the objective to find the best possible solution to maintain the flow of goods/materials from suppliers to demand points by deciding the network structure while minimizing overall system costs.

SCNDP deals with strategic decisions like opening a facility or a distribution center which influence tactical and operational decisions since the opened facility will affect the inventory levels and distribution quantities. Therefore, a lot of integrated decisions are involved in these problems. However, in the literature only a part of these problems related to the complex network problem are modeled by using simplifying assumptions. For instance, the vehicle routing problem (VRP) which considers only the routing part or Facility Location Problems (FLPs) have been extensively studied. There is also integration of two sub problems such as the inventory-location model, the inventory routing problems (IRP) and the location routing Problem (LRP) in literature. But the most suitable problem type with our model is LRP which is the problem of finding the optimal number of depots and vehicles by determining the optimal routes from depots to customers while minimizing overall costs. In this chapter, the relevant literature is discussed in three main sections. In first section, inventory-location models are reviewed. In section two, inventory routing problems are discussed. In the last section, we explored LRPs which are the most relevant problems to our model.

2.1 Inventory-Location Models

Location literature has mostly ignored the inventory related costs and has focused on finding the optimal number of facilities, their positions and assignment to these facilities. On the other hand, inventory models focus on finding optimal replenishment strategies and safety stock decisions by assuming that the number and location of the facilities are known. The early studies that combine these two decisions are defined by Barahona and Jensen[1], and Erlebacher and Meller[2].

Barahona and Jensen[1] proposed an integer programming model for plant location with inventory costs. They used the Dantzig-Wolfe decomposition to solve linear programming relaxation. They were able to derive integer solutions from fractional solutions within 4 percent optimality.

Erlebacher and Meller[2] present a location-inventory model which is a mixed integer non-linear model. They proposed heuristics to solve the problem since it is NP-hard. The well-known Frito-Lay Inc.'s modified data are used in numerical example part of the study. According to computational results, heuristics performed good solutions in the existence of the demand variation and the spatial dispersion.

Shen, Coullard and Daskin [3],[4] contributed to the inventory-location literature by adding working inventory and safety stock costs to distribution center location models. In [3], they present a non-linear mixed integer problem and solved it by using Lagrangian relaxation algorithm. Moreover, a number of improvement heuristics were outlined for the problem. Computational results indicated that the Langrangian relaxation algorithm produced a better solution than set partitioning and column generation method for this problem. In [4], they converted the same model to a set covering integer-programming model. They solve the new model via the column generation method.

Miranda and Garrido[5] have proposed a simultaneous approach to incorporate economic order quantity (EOQ) and safety stock decisions with facility location models. A non linear mixed integer programming model was developed considering stochastic demand, also risk pooling. They solved the model using a heuristic solution approach that is based on Lagrangian relaxation and a sub-gradient method. According to computational results, they stated that reduction in total cost is higher when holding cost, ordering cost, lead time/service level increases.

A recent study was done by Sourirajan, Ozsen and Uzsoy [6]. The difference in their model from others is the ability to capture the tradeoff between risk-pooling and lead times. They introduce a model for a single product distribution network problem and solved it by using a lagrangian heuristic. They obtained near optimal solutions. Romeijn, Shu and Teo [7] have studied a multi-echelon supply chain distribution network design problem with a single product that was distributed from a single supplier to multiple DCs and from DCs to retailers. They formulated a set covering model under a single sourcing policy to solve the problem. They solved the model via column generation method for practical size problems and obtained solutions effectively in a reasonable time.

2.2 Inventory Routing Problems

The inventory routing problem (IRP) is the integration of the inventory control decisions and vehicle routing into a cost efficient distribution system. This problem can be seen as an extension of VRP. Due to the NP-hardness of the VRP problem, most of the papers in this field have practical heuristic solutions even though some theoretical solutions exist.

One of the earliest studies came from Federgruen and Zipkin [8]. In their problem, integration of inventory allocation and routing were provided by a single product, single period problem with random demand in retailers. They designed a non linear mixed integer programming model and solved it by a generalized Bender's decomposition approach.

Another early study was conducted by Chien, Balakrishnan and Wong [9]. They modeled a mixed integer program to integrate inventory allocation and vehicle routing. In their problem the central depot has a supply capacity and demand need not be satisfied in each customer. Therefore, they added a penalty cost to their formulation. The solution procedure consists of a Lagrangian method that gave upper and lower bounds. They produced good solutions with a small gap between upper and lower bounds using heuristics.

Anily and Federgruen [10] researched a problem with a depot and multiple retailers which faced deterministic constant demands. They assumed demand at each retailer is the same. Inventory was held in retailers not in the depot. They aimed to find inventory rules and routing patterns that minimize the inventory and routing costs in the long run. They presented a two stage heuristic. In first stage they found a lower bound by partitioning the demand points into groups. In second stage these groups combined into larger families of demand groups. Then, efficient routing patterns were constructed using regional partitioning heuristics within larger families. Finally, they showed that their heuristic is asymptotically optimal under mild probabilistic assumptions.

Viswanathan and Mathur [11] worked a different and extended version of [10]. They deal with multiple products and periods of their problem. A new heuristic was developed to solve the multi product version by generating nested joint replenishment policies. Moreover, computational studies showed that their heuristics worked well with a single product case.

Lee, Bozer and White [12] studied a class of IRP that has multiple suppliers and an assembly plant in an automotive part supply chain. They dealt with a finite horizon, multi supplier, multi period single assembly plant network problem. The objective was to minimize inventory and routing costs over the planning horizon. They formulated a mixed integer programming model and decomposed the problem into two, vehicle routing and inventory control. First they used a simulated annealing heuristic to generate and evaluate vehicle routes. Then, a linear program calculated the optimum inventory levels for a given set of vehicle routes. They stated that the optimal solution was mainly dominated by the transportation cost regardless of the unit inventory carrying cost.

IRPs arise in many industries and have attracted many researchers. One of these industries was the gas companies which were studied by Campbell and Savelsbergh [13]. In this study, they developed a model by using a vendor managed resupply policies and the stockout of customers were not allowed. They developed a two phase solution approach. First, a delivery schedule is created by using integer programming. In phase two, delivery routes are constructed by using routing and scheduling heuristics. Computational results indicated that this procedure is effective while solving real life large scale instances.

In a recent study, Zhao, Wang and Lai [14] studied integration of inventory control and vehicle routing schedules. In this problem, a single warehouse serves retailers which are geographically dispersed and retailers face a deterministic, customer specific demand rate. The objective was determining inventory policies and routing strategies for the distribution systems. A fixed partitioning policy was proposed for this problem by using the power of two (POT) principle to warehouse and retailer replenishment intervals. They developed a tabu search algorithm to find the optimal retailer partitioning region. Computational results showed the robustness and effectiveness of the algorithm.

Li, Chu and Chen [15] researched a slightly different problem than others by using three-level distribution systems with a vendor, a warehouse and multiple retailers. Inventories are kept in the warehouse and retailers. The demand is replenished directly from the vendor or through the warehouse. They proposed a decomposition solution approach based on the fixed partition policy (FPP). Given fixed partitioning, the problem decomposed into three problems. They developed efficient algorithms for the sub problems. A genetic algorithm was proposed and found near optimal fixed partitions for the problem.

Readers can find a detailed review of IRP in Moin and Salhi [16].

2.3 Location Routing Problems

LRP is the combination of VRP and LAP and interdependence between these two problems was not recognized until the 1970s [17]. In LAP, the objective is to find the optimal number of depots and their locations from given potential sites and allocate customers to these opened facilities while minimizing opening depot costs and assignments costs of customers. In VRP, the objective is to find optimal delivery routes with a given depot to its assigned customers. Various exact and heuristic solutions approaches have been developed as solution methods for LRP. The most common solution techniques used in the literature are heuristics since solving LRP is NP-hard.

Or and Pierskalla [18], Jacobsen and Madsen [19], Madsen [20], Perl and Daskin [17], Balakrishnan, Ward and Wong [21] are the earliest studies in this field. In [18], they deal with the transportation of blood from regional blood banks to hospitals. In this problem, they need to decide how many blood banks to set up, where to locate them and how to allocate hospitals to regional blood banks, also how to route periodic supply operations. They presented algorithms to solve these problems while minimizing total transportation costs and system costs.

Jacobsen and Madsen [19], designed a newspaper distribution system by considering three main decisions, number of distribution points (depots), design of tours from supplier to depots and routes from depots to retailers. They proposed three different procedures and gave a comparison between them by using exactly same cost calculation in three heuristics. Madsen [20] gave a survey of solving combined location routing models. Three new heuristics are developed, implemented and compared with the same distribution problem in [19]. Computational results showed that an alternate location-allocation-savings procedure and a saving-drop procedure is promising.

Perl and Daskin [17] dealt with a warehouse location routing problem (WLRP) which is not studied in literature widely. They proposed a mixed integer linear programming formulation to solve this WLRP. The objective of their model is to minimize the sum of fixed warehousing cost, trunking cost, variable warehousing and delivery cost. They cannot solve the WLRP directly with current known techniques since problem is large and complex. They proposed a heuristic method for solving WLRP with decomposition of the problem into three sub-problems which are multi depot vehicle dispatch problem (MDVDP), warehouse location-allocation problem(WLAP), and multi-depot routing allocation problem (MDRAP). The heuristic solved these sub-problems sequentially either optimally or heuristically. Computational results showed that proposed heuristic can generate good solutions to WLRP.

Balakrishnan, Ward and Wong [21] discussed various modeling approaches for location routing problems that includes mathematical programming formulations, approximations and modified models. They presented advantages and disadvantages of these approaches and gave directions for future research.

Hansen et al. [22] modeled a different version of the integer linear programming formulation of Perl and Daskin [17] to provide a better formulation a new set of flow variables and flow constraints. They used this model to help a company which tries to choose location of a plant to start a new production. Their model is used as a long term strategic decision tool. Moreover, their heuristic improved solutions to benchmark problem.

Sirivastava and Benton [23] studied another aspect of location-routing heuristics. They investigated impact of environmental factors that affect distribution system design. Savings-drop heuristics, savings-add heuristic and cluster-route procedure methods are used to see the effects of these environmental factors. Their research showed that performance of alternative location routing heuristics was affected by key environment factors. Therefore, implementation and use of solution procedures should be based on systems environment. Siribastava [24] developed three new location-routing model and compared with [23] by using the same heuristic methods. Results of this study indicated that new models were superior to existing models.

Chien [25] proposed an approximate approach for LRPs that used sequential procedures that incorporated two robust route length estimators. Procedure firstly generated and improved the location-allocation. Secondly, minimum cost routes are designed depending on the location-allocation results. They evaluated three heuristics to provide approximate solutions to problem. Computational study showed that sequential procedure could produce good solutions to practical sized problems when capacities were not restrictive.

Nagy and Salhi [26],[27] proposed a different method called "Nested". In this nested method unlike iterative and sequential approaches, they treat routing as a sub-problem of a main location problem. This technique used ideas from computational geometry to define the concept of proximity. Furthermore a tabu search was used to improve the solutions of current method. They obtained encouraging results with less computational effort.

Salhi and Fraser [28] proposed an iterative method that alternates between location phase and routing phase until a suitable stopping condition is met. In their study the vehicles in LRP have the different capacities which was unique at that time, however the number of vehicles for each type was unlimited that is typically assumed in literature. In their heuristic, firstly number of depots and their locations are found. Secondly, vehicle fleet combinations are decided and then vehicle routes are determined. They compared their solutions with the sequential method. All of the cases showed that their solutions were not worse than the sequential method.

Tuzun and Burkle [29] presented a new approach for LRP. They used a tabu search which is one of the meta heuristic methods, to find solutions for LRP. In their tabu search method, they used two-phase approach, route-first and location-allocation second. This architecture allowed them to search the solution space efficiently without using too much time. To our best knowledge this study was the first one that compares two different LRP heuristics. The results of the comparative study showed that tabu search heuristics perfomed significantly better than other heuristic and also consumed less computational time.

Wu, Low, Bai [30] presented another meta-heuristic method to solve a multi-depot location-routing problem (MDLRP). They divided the original problem into two sub-problems and solved sequentially by using a simulated annealing algorithm (SA). In first phase, LAP is solved. Then, VRP is solved in the second phase. The results showed that their method gave good quality solutions in a short time.

Ambrossino and Scutella [31] researched a complex distribution network design problem with facility location, warehousing, transportation and inventory decisions. They proposed two different mathematical models for several realistic scenarios. Some formulations extend the models proposed in [17] and other formulations were based on flow variables and constraints. The computational results showed that optimal solution was found only in one instance, other best solutions were not good and far from lower bound.

Beside the heuristics methods, some exact methods were also used to solve LRPs in literature. Most of the contribution in this field comes from Laporte's research. Laporte and Nobert [32] used an exact method for the problem of locating a single depot considering several potential points while minimizing sum of depot operation costs and routing costs. They proposed an integer linear program for the problem and used a constraint relaxation technique to solve it. Integrality is achieved by branch and bound. Moreover, Laporte et al. [33], [34] studied the same problem with the uncapacitated and capacitated multi-facility LRP by using a similar approach as in [32]. In [34], they managed to solve the problems optimally up to 20 sites within a reasonable number of iterations.

In another study that was perfomed by Laporte et al. [35], a transformation was given from the MDVRP and LRP to an equivalent constraint assignment problem. They solved it by using a branch and bound method up to 80 nodes within a reasonable time.

Laporte and Dejax [36] developed two solution approaches to dynamic location-routing problems. The first one is an exact method for the small-size problems. In this method, problem was represented by a suitable network and solved to optimally by using an integer linear programming model. In the second approach, they determined a shortest path on a directed graph and obtained a global solution by approximating some of the system costs.

In a recent work [37], Akca et al. proposed a branch and price algorithm based on a set-partitioning formulation of the LRP. They investigated different pricing algorithms to increase the speed of the pricing sub-problems' solutions. They also described a heuristic algorithm for the LRP based on their exact solution algorithm. The computational results indicated that the branch and price algorithm can solve optimally instances up to 40 customers and 5 depots.

Belenguer et al. [38] developed a branch and cut algorithm for solving LRP with capacity constraints on depots and vehicles. This method based on an integer formulation using only binary variables and different families of valid inequalities such as depot capacity constraints, path constraints, co-circuit constraints, and inequalities derived from CVRP. They solved to optimality all instances considered by [37] involving up to 40 costumers and 5 depots and three instances with 50 customers.

In summary, our study differs from the literature by combining three decision levels in a complex supply chain network. The strategic location decisions, tactical inventory decisions and operational weekly and daily routing decisions are integrated with our proposed model. Furthermore, we use all demand points as potential location zones whereas in classical literature, potential location zones are limited with several points or nodes which makes the problem easier in those cases.

Chapter 3

MODEL FORMULATION

3.1 Problem Definition

In classical SCNDPs, there are three layers: suppliers, distribution centers and demand points. Suppliers send their products to DCs and DCs fulfill the demand of the customers. A pre-determined level of inventory is kept at a DC to maintain a desired customer service level. In our blood bank network design problem, we receive inspiration from the classical SCNDP with some modifications. Our suppliers are the main blood banks, which supplies blood to local blood banks (DCs) and LBB serve hospitals (demand points) on daily basis shipments. There are weekly shipments between main blood bank and local blood banks. A weekly vehicle distributes demand of LBB according to its route.

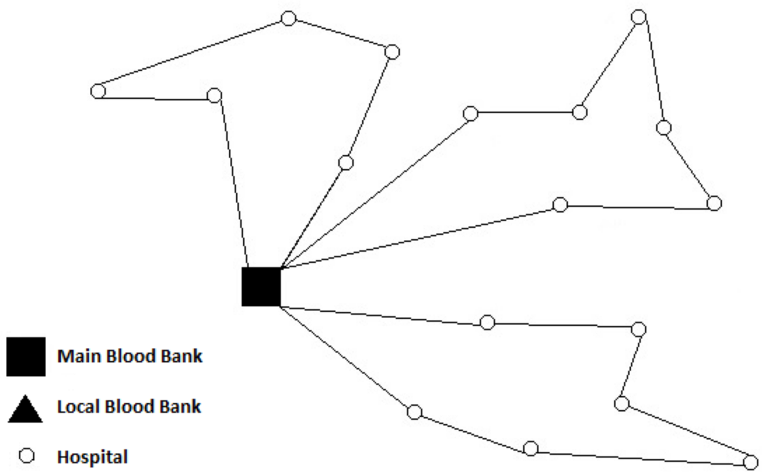

Figure 3.1: Present Distribution Network Model

Present blood distribution network in Istanbul (Figure 3.1) shows that hospitals keep their own inventory and receive shipments directly from main blood bank in weekly basis. There are several regions that a distribution vehicle supplies the demand of hospitals within regions. Our new model aims to change the current system by adding a new layer (local blood banks) and using the advantage of risk pooling effects. Furthermore, routing of weekly and daily shipments between layers is considered.

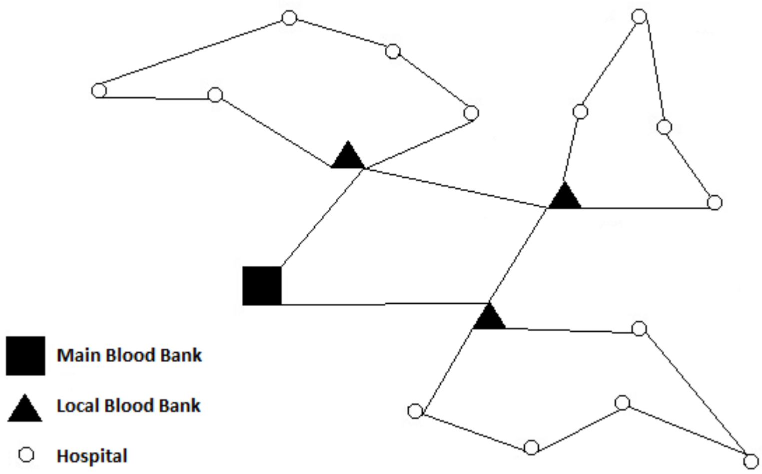

Figure 3.2: New Distribution Network Model

In our proposed model, locations of hospitals and main blood bank are known and we firstly, try to find the hospitals that will become a local blood bank and serve the nearby hospitals. The cost of turning a hospital to local blood bank is included as a fixed annual cost and it covers building a new storage area (warehouse) which will be sufficient to cover demand of nearby hospitals as well as the safety stock. Other relevant costs such as the construction costs, new equipments costs, land costs and daily shipment vehicle costs are also included within this fixed location cost. We also have another cost item related with LBBs, which is variable cost. Variable cost is directly proportional with the amount of blood that passes through the LBBs and aims to cover the cost of employees that work in LBBs such as nurses, officers, drivers, etc.

The main purpose of turning a hospital to a LBB is to use the risk pooling advantages which is proven by Eppen [39]. In Eppen's work, he compares the decentralized and centralized policy for retailer supply management. The only costs considered in his model is one period holding and penalty costs. He uses the assumption that customer demands are normally distributed with a mean μ_i and standard deviation σ_i for retailer i. Also, assuming that correlation coefficient of demands at retailer i and j is ρ_{ij}, then expected cost under decentralized mode for a single period model with N retailers is:

$$K\sum_{i=1}^{N}\sigma_i \tag{3.1}$$

On the other hand, expected total system cost under the centralized model is:

$$K\sqrt{\sum_{i=1}^{N}\sigma_i^2+\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sigma_i\sigma_j\rho_{ij}} \tag{3.2}$$

where K is a constant depending on holding and penalty costs and standard normal loss function.
Thus, if the demands of the N retailers are independent, $\rho_{ij} = 0$ and optimal costs become:

$$K\sqrt{\sum_{i=1}^{N}\sigma_i^2} \tag{3.3}$$

which is less than 3.1

Eppen's result shows us that cost savings can be achieved by consolidation of hospital demands in a LBB under mentioned assumptions.

In our model setting, after deciding the LBBs, we try to assign hospitals to LBBs which directly affect the inventory levels at LBBs and distribution costs of the entire system. Hospitals are not keeping their own inventory and their demands are fulfilled by daily shipments from LBBs. Inventory holding costs at LBBs consist of cycle stock level and safety stock levels.

In classical network design problems, for simplicity, shipments between DCs and retailers are assumed as direct shipments. However, our model captures these shipments by routing. A daily shipment vehicle travels through assigned hospitals of a LBB and returns back to LBB after satisfying their demands. Our assumption is that the daily vehicle has the capacity to fulfill all hospitals demand in its route on a single shipment. In addition to daily shipments, our model considers weekly shipments between the main supplier and LBBs. The weekly replenishment vehicle is dispatched from the main supplier and satisfies the demand of the LBBs on its route. Therefore, we try to find daily and weekly shipment routes.

As mentioned throughout this section, proposed model's main advantage comes from the holding inventory costs by centralizing the safety stock locations. However, emerging daily routes which occur between newly located LBBs and the hospitals which are assigned to them increases the system costs. Therefore, there is a tradeoff between inventory holding costs and daily routing costs.

In conclusion, our problem finds answers to make the following decisions:

- Which hospitals should be selected as LBB,
- How to allocate hospitals to LBB,
- Vehicle routes of daily shipments from LBB to hospitals,
- Vehicle routes of weekly shipments from the main blood bank to LBBs,
- Quantity of blood units distributed from the main supplier to LBBs and LBBs to hospitals,
- Inventory levels at LBBs,

under the following assumptions:

- The location of the main blood bank and hospitals are known
- The main blood bank has no capacity limit
- The LBBs have no capacity limit
- There is only one main blood bank
- All hospitals can be selected as LBB
- Demands of hospitals are normally distributed and independent from each other
- Maximum number of weekly replenishment vehicles are available and the capacity of each vehicle is known.
- Each LBB has exactly one daily replenishment vehicle.
- Daily replenishment vehicle capacity is enough to satisfy all hospitals of its route.
- Each route is served by one daily/weekly vehicle.
- Each route begins and ends at the same point.
- LBBs and hospitals replenish using a single sourcing strategy, i.e. each LBB will be replenished from a single supplier and each hospital will be replenished from a single LBB.

3.2 MINLP Formulation of the SCNDP

The problem is modeled to find the optimal number and locations of local blood banks, assignment of hospitals to the local blood banks, inventory levels at the LBB and routes of the weekly and daily vehicles which supply the LBB and hospitals. The problem parameters and decision variables are stated as follows:

Index Set

O: denotes the main blood bank indexed by o,

I: set of retailers (hospitals) indexed by i,

J: set of candidate DC (local blood bank) sites indexed by j,

L: set of all replenishment vehicles, indexed by l,

$\tilde{I}=$ $I \bigcup J \bigcup O$

Parameters

μ_i: mean daily demand at hospital i,

σ_i: variance of mean daily demand at hospital i,

d_{ij}: distance between location i and j,

f_j: fixed annual cost of locating a local blood bank at hospital i,

c_j: variable cost for a unit at local blood bank j,

h: weekly inventory holding cost at a local blood bank,

t: Lead time in days,

α: probability of stocking out during a replenishment cycle,

β: weight factor associated with the transportation cost of weekly replenishment vehicle,

θ: weight factor associated with the transportation cost of daily replenishment vehicle,

γ: cost of owning and running a replenishment vehicle,

C_r: Capacity of weekly replenishment vehicle,

C_d: Capacity of daily replenishment vehicle,

Decision Variables

$$X_j = \begin{cases} 1 \text{ If hospital j is selected as a local blood bank} \\ 0 \text{ otherwise.} \qquad j \in J \end{cases}$$

$$Y_{ij} = \begin{cases} 1 \quad \text{If hospital i is served by a local blood bank j} \\ 0 \quad \text{otherwise.} \qquad \forall \text{i} \in I, \forall \text{j} \in J \end{cases}$$

$$U_{ikj} = \begin{cases} 1 \quad \text{If hospital k succeds hospital i in a daily route of blood bank j} \\ 0 \quad \text{otherwise.} \qquad \forall i, k \in I, \forall j \in J \end{cases}$$

$$W_{jm} = \begin{cases} 1 \quad \text{If local blood bank m succeeds blood bank j in a weekly replenishment cycle} \\ 0 \quad \text{otherwise.} \qquad \forall j, m \in J \end{cases}$$

$$Z_{jl} = \begin{cases} 1 & \text{If local blood bank j is assigned to replenishment vehicle l} \\ 0 & \text{otherwise.} \qquad \forall j \in J, \forall l \in L \end{cases}$$

D_j: demand assigned to local blood bank j,

S_j: standard deviation of demand assigned to local blood bank j,

SS_j: safety stock at local blood bank at hospital i,

FD_{ij}: truckload carried out from hospital i on a daily route of blood bank j,

FW_{jl}: truckload carried out from blood bank j of replenishment vehicle l,

The objective function becomes:

Minimize

$$\sum_{\forall j \in J} f_j X_j + h(\sum_{\forall j \in J} [\frac{D_j}{2} + SS_j]) + \sum_{\forall j \in J} c_j D_j + \sum_{\forall l \in L} \gamma Z_{ol} + \sum_{\forall j,m \in \tilde{I}} \beta d_{jm} W_{jm} + \sum_{\forall j \in J} \sum_{\forall i,k \in I} \theta d_{ik} U_{ikj} \tag{3.4}$$

subject to

$$\sum_{\forall i \in I} Y_{ij} \leq X_j, \qquad \forall j \in J \tag{3.5}$$

$$\sum_{\forall j \in J} Y_{ij} = 1, \qquad \forall i \in I \tag{3.6}$$

$$\sum_{\forall k \in I} U_{ikj} = Y_{ij}, \qquad \forall i \in I, \forall j \in J \tag{3.7}$$

$$\sum_{\forall k \in I} U_{kij} = Y_{ij}, \qquad \forall i \in I, \forall j \in J \tag{3.8}$$

$$\sum_{\forall k \in I} U_{ikj} = \sum_{\forall k \in I} U_{kij}, \qquad \forall i \in I, \forall j \in J \tag{3.9}$$

$$X_j = \sum_{\forall i \in I} U_{jij}, \qquad \forall j \in J \tag{3.10}$$

$$X_j = \sum_{\forall i \in I} U_{ijj}, \qquad \forall j \in J \tag{3.11}$$

$$X_j = \sum_{\forall l \in L} Z_{jl}, \qquad \forall j \in J \tag{3.12}$$

$$X_j = \sum_{\forall m \in \tilde{I}} W_{mj}, \qquad \forall j \in J \tag{3.13}$$

$$\sum_{\forall m \in \tilde{I}} W_{mj} = \sum_{\forall m \in \tilde{I}} W_{jm}, \qquad \forall j \in J \tag{3.14}$$

$$\sum_{\forall j \in I} W_{oj} = \sum_{\forall l \in L} Z_{ol} \tag{3.15}$$

$$W_{mj} + Z_{ml} - Z_{jl} \leq 1, \qquad \forall m \in \tilde{I}, \forall j \in \tilde{I} \tag{3.16}$$

$$W_{jm} + Z_{ml} - Z_{jl} \leq 1, \qquad \forall m \in \tilde{I}, \forall j \in \tilde{I} \tag{3.17}$$

$$D_j = \sum_{\forall i \in I} \mu_i Y_{ij}, \qquad \forall j \in J \tag{3.18}$$

$$\sum_{\forall j \in J} Z_{jl} D_j \leq C_r, \qquad \forall l \in L \tag{3.19}$$

$$S_j = \sqrt{t \sum_{\forall i \in I} \sigma^2 Y_{ij}}, \qquad \forall j \in \ J \tag{3.20}$$

$$SS_j = F^{-1}(CSL) S_j, \qquad \forall j \in J \tag{3.21}$$

$$FD_{i,j} - FD_{kj} - C_d U_{ikj} \geq \mu_k - C_d - C_d X_k, \qquad \forall i, k \in I, \forall j \in J \tag{3.22}$$

$$FW_{ml} - FW_{jl} - C_r W_{mj} \geq D_j - C_r, \qquad \forall m, j \in J, \forall l \in L \tag{3.23}$$

$$X_j, Y_{ij}, U_{ikj}, W_{mj}, D_{jl}, Z_{jl} \in \{0, 1\} \qquad \forall i, j, k, l \tag{3.24}$$

$$D_j, S_j, SS_j, FD_{ij}, FW_{jl} \geq 0 \qquad \forall i \in I, \forall j \in J, \forall l \in L \tag{3.25}$$

The objective function (3.4) tries to minimize the fixed cost of locating a LBB ($\sum_{\forall j \in J} f_j X_j$), cost of owning replenishment vehicles ($\sum_{\forall l \in L} \gamma Z_{ol}$), inventory costs at LBBs ($h(\sum_{\forall j \in J}[\frac{D_j}{2} + SS_j])$), variable costs of LBBs ($\sum_{\forall j \in J} c_j D_j$), transportation cost of weekly deliveries

($\sum_{\forall j,m \in \tilde{I}} \beta d_{jm} W_{jm}$) and daily deliveries ($\sum_{\forall j \in J} \sum_{\forall i,k \in I} \theta d_{ik} U_{ikj}$) while satisfying the constraint set between (3.5) - (3.25).

Constraint (3.5) indicates that hospital-local blood bank assignment can only be done in the hospitals which are selected as local blood bank. Constraint (3.6) ensures that all hospitals must be assigned to a local blood bank. Constraint set (3.7) and (3.8) states that if hospital i is assigned to LBB j, then in a daily route of LBB j, a retailer k succeed by retailer i, and i is succeeded by retailer k. Constraint (3.9) guarantees that number of daily vehicle that enters and leaves hospital i is the same. Constraint set (3.10) and (3.11) indicates that if a LBB j is opened than there is a daily vehicle enters the LBB k and respectively a vehicle leaves the k. Constraint (3.12) shows that a weekly replenishment vehicle is assigned to every LBB. Similarly, (3.13) states that there must be a inbound flow from a LBB or main blood bank in a weekly replenishment route if LBB j is opened. Constraint (3.14) guarantees that a LBB j is proceeded by local blood bank or main blood in a weekly replenishment route is proceed to a LBB or main blood bank in the same weekly replenishment route. Constraint (3.15) states that number of weekly replenishment vehicle leaves from main blood bank must be equal to vehicles which are assigned to LBBs. Constraint set (3.16) and (3.17) force to combine routing and allocation components such that if LBB m succeed LBB j in a weekly replenishment route, same vehicle must be assigned to LBB m and j. Constraint (3.18) is the calculation of aggregated demand in LBB j. Constraint (3.19) guarantees that capacity of the weekly replenishment vehicle cannot exceed the total demand assigned to itself. Constraint set (3.20) and (3.21) indicates the standard deviation and safety stock calculations of a LBB j. Constraint set (3.22) and (3.23) is the classical sub-tour elimination constraints and prevent calculation of unnecessary moves of daily and weekly replenishment routes. Finally, remaining (3.24) and (3.25) are the nonnegativity and integrality constraints.

Chapter 4

SOLUTION METHODOLGY

As we stated in Literature Review, our problem can be seen as the combination of LRP, IRP and the inventory-location problems. Therefore, it has more complexity than its sub-problems. Our literature review research showed us that even the classical LRP problems are NP-hard and finding an exact solution in polynomial time is not easy. Moreover, our model has non-linear terms and binary variables in square root operator which make it more difficult to solve. Nevertheless, we try to find exact solutions in part 4.1 to our model for small-sized problems using GAMS/BARON. In addition to that, in part 4.2 we modified our model to eliminate nonlinearity in constraints 3.19 and 3.20 by adding new variables and using piecewise linear approximation. Then we use GAMS/CPLEX solver to find a lower bound to our inventory cost part of the objective function.Lastly, we used simulated annealing heuristic to find good solutions to mid and large sized problem instances. Details of the heuristic method are explained in 4.3

4.1 Exact Solution Method

Exact solution method evaluates all possible feasible solutions among the solution space and finds the global optimum for the problem. In our case, this method is only useful for small-size instances since the combination of the possible solutions increases exponentially with the increasing number of hospitals.

In this problem, any of the hospitals can be selected as a LBB. The hospitals can be assigned to any LBB and not necessarily to the closest LBB. The solver considers routing costs and inventory holding costs while making the assignments of hospitals. Therefore, for all opened LBB, it solves a VRP and gives the best route of daily blood distribution.

Hence, the existence of binary variables within root operator to find safety stock level, rapid increase of the number of constraints and the number of decision variables that needs to be answered limit the exact solution to small size instances. Finding optimal solution for the problem will be very time consuming with this computational complexity. Therefore, we firstly, solve the problem by linearizing the non-linear constraints. Then, we developed heuristic algorithms to find good solutions in a reasonable time.

4.2 Piecewise Approximation Method

We introduce the mathematical formulation for SCNDP in chapter 3 by describing the objective and constraints. In the model, there are two constraints with non-linear terms that makes problem even more difficult to solve in exact methods.

In Constraint 3.19 nonlinearity is encountered from the multiplication of the two decision variables, Z_{jl} and D_j. As mentioned before, Z_{jl} is the assignment of weekly vehicle l to LBB j and D_j is the total demand of LBB j.

$$\sum_{\forall j \in J} Z_{jl} D_j \leq C_r, \qquad \forall l \in L$$

This constraint indicates that the demand assigned to weekly replenishment vehicle can't exceed the vehicle capacity.

In order to deal with the nonlinearity, a new decision variable and two constraints are introduced:

DR_{jl} :total demand of local blood bank j which is assigned to weekly replenishment vehicle l,

$$\sum_{\forall j \in J} D_{jl} \leq C_r, \qquad \forall l \in L \tag{4.1}$$

$$D_{jl} \geq D_j - C_r(1 - Z_{jl}), \qquad \forall j \in J, \forall l \in L \tag{4.2}$$

$$D_{jl} \geq 0, \qquad \forall j \in J, \forall l \in L \tag{4.3}$$

Constraint 4.1 impose the capacity restriction for the weekly replenishment vehicles and Total demand distributed by vehicle l can not exceed the vehicle capacity. Constraint 4.2

guarantees that if vehicle l is assigned to LBB j, then total amount distributed by weekly replenishment vehicle l is greater than or equal to total demand of LBB j. Last constraint 4.3 indicates the nonnegativity.

Thus, we replaced constraint 3.19 with 4.1, 4.2 and 4.3 to deal with the first non-linear term.

Second non-linearity which is in constraint 3.20, comes from the binary variable in the square root operator.

$$S_j = \sqrt{t \sum_{\forall i \in I} \sigma^2 Y_{ij}}, \qquad \forall j \in \ J$$

Standard deviation of the the demand assigned to LBB j is calculated with this formulation. To get rid of the nonlinearity, we suggest to use piecewise linear approximation to find approximate S_j values.

Piecewise approximation technique can be used with different approximation schemes depending on the number of segments in the function. The number of breakpoints in the function determines the approximation error and the computational complexity. Therefore, number of pieces in the function have been chosen wisely to keep the balance between error range and complexity.

As you can see from Figure 4.1, we define a new index set $t = 1, 2, ...10$ to divide variance range into 10 parts. Then, slope of the function in every segment is calculated.

Thus, our new parameters become:

$alin_t$:	denote the break points,
Δ_t:	the slope of the linear approximation function in part t,
S':	value inside the root operator,

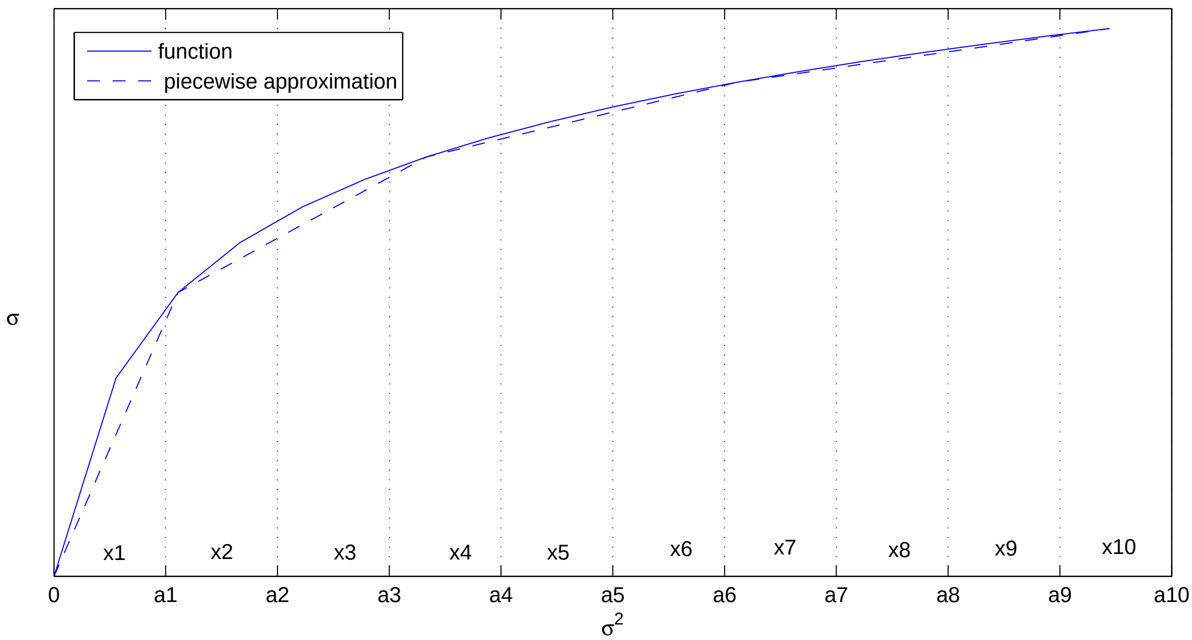

Figure 4.1: Piecewise Approximation Scheme

These parameters are calculated as follows:

$$alin_t = \sum_{\forall j \in J} var(j)\frac{t}{10}, \tag{4.4}$$

$$\Delta_t = \frac{\sqrt{\sum_i var(i)\frac{t}{10}} - \sqrt{\sum_i var(i)\frac{t-1}{10}}}{alin_t - alin_{t-1}}, \tag{4.5}$$

$$S' = L_e \sum_{\forall i \in I} \sigma^2 Y_{ij}, \tag{4.6}$$

Then, two new decision variable introduced:

$Xlin_t$: is the amount in part t,

$$Ylin_t = \begin{cases} 1 \text{ If part t is fully filled} \\ 0 \text{ otherwise.} \end{cases}$$

To find the approximate value of S_j by

$$S_{approx} = \sum_{t=1}^{10} Xlin_t \Delta_t \tag{4.7}$$

while the constraint set below is satisfied.

$$S' = \sum_{t=1}^{10} Xlin_t, \tag{4.8}$$

$$Xlin_t \leq alin_t - alin_{t-1}, \tag{4.9}$$

$$Xlin_1 \leq alin_1, \tag{4.10}$$

$$M(Ylin_t) \geq Xlin_{t+1}, \tag{4.11}$$

$$Xlin_1 \geq alin_1 Ylin_1, \tag{4.12}$$

$$Xlin_t \geq (alin_t - alin_{t-1})Ylin_t \tag{4.13}$$

In order to have a better understanding, we will show the piecewise approximation method in an example.

Let's assume that we have a inside root value of 35 and the sum of all variances is 50. For simplicity we use five break points (instead of 10) in range 0 to 50.

Then our parameters become:

S'=35

$alin_1 = 10$, $alin_2 = 20$, $alin_3 = 30$, $alin_4 = 40$, $alin_5 = 50$,

$\Delta_1 = 0.316$, $\Delta_2 = 0.131$, $\Delta_3 = 0.1$, $\Delta_4 = 0.085$, $\Delta_5 = 0.075$,

and the constraint set below is defined to find the $Xlin$ and $Ylin$

S'=$Xlin_1$+$Xlin_2$+$Xlin_3$+$Xlin_4$+$Xlin_5$ $= 35$

$Xlin_1 \leq 10,$

:

$Xlin_5 \leq 10,$

$999(Ylin_1) \geq Xlin_2,$

:

$999(Ylin_4) \geq Xlin_5,$

$Xlin_1 \geq 10Ylin_1,$

:

$Xlin_5 \geq 10Ylin_5,$

$Xlin$ and $Ylin$ values become:

$Xlin_1 = Xlin_2 = Xlin_3 = 10,$

$Xlin_4 = 5$

$Xlin_5 = 0,$

$Ylin_1 = Ylin_2 = Ylin_3 = 1,$

$Ylin_4 = Ylin_5 = 0,$

Approximation of the values becomes:

S_{approx} =10(0.316) + 10(0.131) + 10(0.1) + 5(0.085)

=5,895

Therefore, approximation method found value of $\sqrt{35}$ as 5.895 whereas the real value of $\sqrt{35}$ is 5.916.

As it can seen from the example above, when we use piecewise linearization of the square root function, the total inventory cost will be lower than the actual cost. Thus, the exact solution of the problem with piecewise linearization will be a lower bound for the original problem. In addition, the solution obtained by the piecewise linearization is a feasible solution for the original problem. Thus, if we take this solution and calculate the actual cost of the system, it will give us an approximate result and this solution can be used as an approximate solution for the original problem.

4.3 Heuristic Solution Method

In previous sections, we clarified that the complex nature of the problem makes it impossible to solve our model in a reasonable time for mid and large size problems. Due to the exponential grow of decision variables and constraints, even the small size problems is hard to solve with exact methods. Therefore, we use heuristics methods to find good solutions for our problem.

Throughout this chapter, we give insights about Simulated annealing (SA) method which is one of the popular meta-heuristic methods in the literature. Then, we explained proposed simulating annealing algorithm (SA) for SCNDP in details.

4.3.1 Simulated Annealing

The Simulated Annealing (SA) is one of the oldest among meta-heuristics and the first algorithm that have the strategy (hill climbing moves) to escape from local optima. It uses a stochastic relaxation which has its origins in statical mechanics (Metropolis et al.,[44]). The SA firstly used for solving combinatorial optimization by Kirkpatrick et al. [45] in 1983 and Cerny [46] in 1985. Since then, SA has been applied successfully to variety of highly complicated combinatorial optimization (CO) problems as well as various real-world problems. We refer interested readers to Osman and Laporte [41] where they can find many different application of the SA in literature.

The general framework of a SA algorithm can be seen from the figure 4.2. In this figure, s denotes solution and T denotes temperature, both are initially set. Then, new solution s' is selected from neighborhood of solutions N(s) and improving solutions are accepted with respect to a probability function until a finishing condition met.

The SA algorithm search for a optimum or near optimum solutions by using slow cooling procedure described above. Firstly, it starts with a random or heuristic based initial solution S and initial temperature T_i. At each iteration a new solution is taken from the neighborhood $N(S)$ of the current solution by using predefined moves. Then, objective

```
s ← GenerateInitialSolution()
T ← T_0
while termination conditions not met do
  s′ ← PickAtRandom(N(s))
  if (f(s′) < f(s)) then
    s ← s′              % s′ replaces s
  else
    Accept s′ as new solution with probability p(T, s′, s)
  endif
  Update(T)
endwhile
```

Figure 4.2: General SA Algorithm (Blum and Roli [43])

cost of the new solution $S^{'}$ is compared with the current cost. If there is an improvement in objective cost, the new solution is always accepted and new solution becomes current solution. On the other hand, a new solution without improvement may also be accepted with a small probability computed by the Boltzmann distribution.

$$exp(\frac{\Delta}{kT})$$

where Δ is the difference between new and current solution $C(S^{'}) - C(S)$, k is a physical constant known as Boltzmanns constant and T is the temperature of the current state. This algorithmic mechanism of SA allow uphill moves that may help escaping from local optimum to reach global optimum. The value of the T is the critical in this part since it determines the occurrence of uphill moves. T starts with high values at the beginning of the search and probability of accepting uphill moves is high. This lead to exploration of the search space (intensification). Then T is decreased gradually to small value close to zero which permits exploitation of the search space (diversification). When T is constant,lower δ values leads to higher probability of accepting uphill moves.

4.3.2 SA Heuristics Algorithm for SCNDP (SA-SCNDP)

In operations research literature, SA methodology is used in location, location-allocation, vehicle routing, location routing problems and many more as it is shown in [41]. Therefore, we decided to use a SA based approach to solve our SCNDP in this study. As mentioned in chapter 3, we try to solve a complex problems with decisions of the location of a LBB, assignment of hospitals to opened LBBs, routing of the weekly and daily vehicles together. Thus, our algorithm needs to solve a location, allocation and multiple routing problems all in once.

We design the SA algorithm in a way that it uses four consecutive phase to find the best solution to SCNDP. These are Initialization, Location-Allocation, Weekly Route Improvement and Daily Route Improvement phases. In location-allocation and daily route improvement phases, SA methodology is used to find improved solutions by searching neighborhood of current solutions. Before, explaining the SA phases,algorithms and procedures used in SA-SCNDP in detail, we firstly introduce our solution representation as follows:

Solution Representation

In our algorithm, we use 4 different matrix representation to define our decision variables, $X_j, Y_{ij}, U_{ikj}, W_{mj}, D_{jl}, Z_{jl}$. X matrix is used for the location decisions. Y is for the assignments and WR for weekly routes and DR for daily routes
Example below show a case with 5 hospitals and 1 main blood banks solution representation.

$$X = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

0^{th} node represents the main blood bank and always zero and 2^{nd}, 4^{th} are the hospitals which was selected as LBB.

$$Y = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

shows that1^{st} and 3^{rd} hospital is assigned to hospital 2 (LBB) and 5^{th} is assigned to hospital 4.

$$WR = \begin{pmatrix} 0 & 2 & 4 & 0 \end{pmatrix}$$

means that one vehicle begins its route from main blood bank (0) and first it visits hospital 2, then 4 and return to main blood bank. If there is more than one route, then we'll have more than one row in the matrix. Therefore, row number represents the weekly number used in solution.

$$SR=\begin{pmatrix} 2 & 3 & 1 & 2 \\ 4 & 5 & 4 & 0 \end{pmatrix}$$

indicates that there are two daily routes that begin and end at the same LBBs that are represented by hospital numbers 2 and 4.

SA-SCNDP Phases

The initialization phase starts with the *Randomfacilityopen* procedure to open LBBs randomly with respect to minimum facility limitation. Then, *Allocation* procedure assign the hospitals to opened LBBs by looking at the shortest distances. After that, *BigRoute Algorithm* and *SmallRoute Algorithm* construct the initial routes for weekly and daily vehicle routes respectively. Finally, *Savings Algorithm* and *2 Opt Algorithm* is applied to daily routes to have better initial solutions.

In location-allocation phase, initial solution is improved by *Add-Drop* and *Swap* moves. At the beginning of this part, SA parameters are set and initial solution from first phase is accepted as current and best solution. At each iteration one of the moves are applied upon the dynamic probability selection rule. Probability of the moves are close to each other at the beginning of the iterations, but probabilities change dynamically at upcoming iterations. If the selected move is applied and current solution is improved, then the probability of the move increases and vice versa. Moves are applied according to SA methodology as explained in previous chapter. Best solution found during the iterations are recorded and become the input of the weekly improvement and daily improvement phases.

Third phase begins with the best solution found in location-allocation phase. Optimal number and location of LBB and hospital assignment are taken as input. Bigroute algorithm construct the best weekly distribution route from main blood bank to opened LBBs. After that 2-Opt Algorithm is applied to this route until there is no improvement in route. Thus, this phase can be seen a local search for route improvement.

In final phase, depending on the number of the opened LBBs two of the following cases occur. In the first case, if there is only one opened LBB, then due to our assumption of each LBB has only one daily replenishment vehicle, we'll have only one route. Therefore, we use Savings algorithm and then same route improvement logic as third phase by using 2 Opt Algorithm. In second case, we have more than one opened LBB and cost improvements can occur due to insertion of one hospital to another LBB's serving route. Thus, second SA procedure algorithm begins with this case. Best solution from second and third phase taken as input. Initial smallroute and related costs such as daily routing costs, inventory holding cost and variable costs are aggregated and accepted as initial cost. At each iteration, an insert move is applied after the feasibility check. Every feasible move that improve the current costs is accepted. Non-improving moves are also accepted with a small probability. Best daily distribution route found during the iterations are returned as final solution.

SA-SCNDP procedure is explained step by step as follows:

Step 1: Calculate minimum LBBs needed by considering vehicle capacity and total demand,Open at least LBBs by using *Randomfacilityopen* procedure,

Step 2: Use *Allocation* procedure to assign hospitals to opened closest LBBs by considering weekly replenishment vehicle capacities.

Step 3: Construct weekly and daily vehicle routes by using Bigrouting(), SmallRouting(),use Savings and 2-Opt algorithm to improve the routes.

Step 4: Set cooling parameters, random selection parameters, best cost and initial cost.

Step 5: Apply Add-Drop move or Swap move according to random selection probability.

Step 6: Compute $\Delta = C(S^{'}) - C(S)$ (the difference between new and current solution)

Step 7: If $\Delta < 0$ or $\Delta > 0$ and $exp(\frac{\Delta}{kT}) > x$, where x is a random number between [0,1] Then, set new cost as the current cost. $S = S^{'}$ and increase the selection probability of applied move by 1%

Else, increase the nonimprovement counter by 1 and decrease the selection probability of applied move by 5%

Step 8: If new cost is less than the best cost found so far, S_{best}

Then, set new cost as the best cost. $S_{best} = S^{'}$ and increase selection probability of applied move by 5%

Step 9: Set nrep=nrep+1, iteration=iteration+1

Step 10: If max nrep number is reach withing the current temperature go to Step 11,

Else go to Step 5.

Step 11: Update the current temperature by $T_{k+1} = \alpha T_k$, where $\alpha \in (0, 1)$ and go to Step 5.

Step 12: Terminate SA procedure for Location-Allocation phase if the stopping conditions are met.

Step 13: Construct weekly routing by using best solution found between step4-step12.

Step 14: Calculate weekly route cost and set to currentRoutecost.

Step 15: Apply 2 opt algorithm to current weekly routing solution.

Step 16: If currentRoutecost is improved,

Then, set new cost as current cost and new solution as current weekly route solution and the go to step 15.

Else, terminate the Weekly route improvement phase and return the current weekly route solution.

Step 17: Check the daily route number of the best solution found.

If there is more than one daily route go to step 18

Else go to step 28.

Step 18: Set cooling parameters, random selection parameters, best cost and initial cost.

Step 19: Apply Insertion move

Step 20: Check feasibility of the new solutions.

If new solution is feasible go to step 21.

Else increase the nonfeasibiliy counter by 1 and go to step 24.

Step 21: Compute $\Delta = C(S^{'}) - C(S)$

Step 22: If $\Delta < 0$ or $\Delta > 0$ and $exp(\frac{\Delta}{kT}) > x$, where x is a random number between [0,1]

Then, set new cost as the current cost. $S = S^{'}$

Step 23: If new cost is less than the best cost found so far, S_{best}

Then, set new cost as the best cost. $S_{best} = S^{'}$.

Step 24: Set nrep=nrep+1, iteration=iteration+1

Step 25: If max nrep number is reach withing the current temperature go to Step 26,

Else go to Step 19.

Step 26: Update the current temperature by $T_{k+1} = \alpha T_k$, where $\alpha \in (0, 1)$ and go to Step 19.

Step 27: Terminate SA procedure for SmallRouting phase if the stopping conditions are met.

Step 28: Calculate daily route cost and set to currentRoutecost.

Step 29: Apply 2 opt algorithm to current daily routing solution.

Step 30: If currentRoutecost is improved,

Then, set new cost as current cost and new solution as current daily route solution and the go to step 15.

Else, terminate the Daily route improvement phase and return the daily route solution.

The moves, procedures and algorithms which are mentioned in SA-procedure above, explained in detail as follows:

Initialization

The aim of this phase is to construct initial solutions that lead our algorithm to find optimal or near optimal solutions in short time. Therefore, the algorithm uses Randomfacilityopen procedure to open LBBs randomly. We use this initialization since it's fact and intuitive. We initialize with the lower bound on facility number, since the number of LBB that needs to be opened can be calculated with the knowledge of the total demand and weekly replenishment capacity of the replenishment trucks in advance. Then, hospitals are assigned to nearest opened LBB by considering the capacity limits. Allocation procedure is applied for this assignments.Improvement algorithms like savings and 2-opt are also used to start with good initial weekly and daily routes. Details of the procedures and algorithms are explained as follows:

Randomfacilityopen procedure

This procedure takes weekly replenishment capacity, total demand as input and returns the opened facility matrix, X.
Firstly, summation of the all hospital demands are divided to replenishment vehicle capacity to find a lower bound on opened LBB number. Then, facilities are opened randomly while considering the minimum number of LBB that should be opened to have a feasible solution.

Allocation Procedure

Allocation procedure uses inputs as opened LBB matrix X, vehicle capacity and distance matrix between hospitals. The aim of this procedure to assign hospitals which are not selected as LBB to nearest LBB without violating the weekly replenishment capacity. The weekly capacity is important for feasibility of assignment and the final solution. therefore, total demand of a LBB consisting the have to be less than weekly replenishment vehicle's capacity.

Firstly, procedure checks the capacity limits and total demand. If there is enough opened LBB, a distance matrix between LBBs and hospitals are constructed. The hospital with the shortest distance to a LBB is selected as a candidate. Then, demand of the candidate hospital is added to LBB's total demand. If the demand is less than capacity limits then assignment is made. This procedure repeats until all of the hospitals assigned to a LBB. Finally, procedure returns the assignment matrix Y for the upcoming phases.

BigRoute Algorithm

Bigroute algorithm basically construct the weekly route (called as bigroute) of the replenishment truck. A replenishment route starts from the main blood bank. It distributes the weekly demand of the LBBs on its route and returns the starting point, main blood bank. There could be several routes and therefore many trucks depending on the problem structure. Therefore, number of feasible routes determines the replenishment truck number. The algorithm takes assignment matrix Y, vehicle capacity, distance matrix and demands of the hospitals as input and returns the feasible weekly delivery routes.

Bigroute algorithm has mainly two phases. In the first phase, it uses a classical Clark-Wright savings algorithm [55] to construct the savings matrix between main blood bank and LBBs. Then, depending on the savings, it begins to construct feasible routes case by case. The Clark-Write Savings algorithm will be explained in upcoming Savings Algorithm subsection, therefore we mentioned only the routing phase of the algorithm in this part.

At the beginning, the algorithm constructs a separate route for every opened LBB based on the direct shipment from main blood bank to LBB. Then, savings algorithm reveals the saving costs of LBBs in a matrix form depending on the distance between locations. Designing of new routes begin with LBBs with the highest savings number. An example of the savings matrix used in algorihm shown below.

$LBB1$	$LBB2$	$Savings$
4	7	6011
2	4	2924
2	7	1789

Three of the following cases can occur before combining the two of LBB in a weekly rote,

- LBB1 and LBB2, both have a direct route from only the main blood bank
- One of the LBB has a route that consists of other LBBs, and the other one has only route including the main blood bank.
- Both have different routes that consists of other LBBs.

In the first case, algorihm check the weekly routing vehicle capacity and summation of total demand of LBB1 and LBB2. If there is enough capacity then, it combines them in a route. For example, node 4 and node 7 has the highest savings as it shown in matrix above. They have only direct route with main blood bank, 0-4-0 and 0-7-0. Then, after capacity check, it construct the new route 0-4-7-0.

In second case, one of the LBBs in saving matrix is assigned to a route and other one is only assigned to main blood bank. Algorithm checks again the capacity of the route consisting route and demand of the unassigned LBB. If it is feasible, unassigned LBB is added to the route depending on the position of other LBB. If you look at the previous example, we have a route 0-4-7-0 and in second iteration combining node 2 and 4 seems the optimal. Node 2 has only route with main blood bank 0-2-0, on the other hand node 4 have a different route. After,capacity check new route 0-2-4-7-0 can be constructed. If, weekly routing vehicle don't have enough capacity then we have, two different route 0-4-7-0, and 0-2-0 and number of weekly vehicle is determined as 2.

In final case two of the LBBs have different routes, then algorithm checks the capacity of the two routes. If it is feasible, then it combines them. For example we have two routes 0-4-7-0 and 0-2-5-0. Let's say combining 2 and 4 gives the highes savings and weekly routing vehicle have enough capacity. Then, algorithm construct the 0-5-2-4-7-0 route as the new solutions.

In the end, all of the node couples in savings matrix is analyzed case base case with respect to methods explained above. Then, algorithm construct and returns the BR matrix which consist the all weekly routes.

Smallroute Algorithm

The purpose of this algorithm is to construct the daily route (called as smallroute)of the daily distribution truck. Every LBB have exactly one daily truck due to our assumption mentioned in chapter 3. The daily truck begins its tour from the LBB and delivers daily demand of the hospitals which are assigned to that LBB and returns to LBB. The algorithm takes opened facility matrix X and allocation assignment matrix Y as input and returns the daily routes.

The algorithm firstly calculates the number of daily routes in the problem by checking the number of LBBs given in opened facility matrix. Then, it uses allocation assignment matrix to find how many hospital assigned to each LBB. The daily route number and max hospital assignment are used to determine the size of the daily route matrix DR at the end of the algorithm. Secondly, allocation assignments are accepted as the daily routes and some modification are done depending on the position of the LBB in the allocation matrix. Three cases can occur as follows:

- The selected hospital (LBB) is the first element in the row
- The selected hospital (LBB) is the last element in the row
- The selected hospital (LBB) is neither first nor last element in the row.

A daily route begins and ends at the same point. Therefore, in the first case, LBB added to last element of the row and a daily route completed. In second case all elements in the row shifted to right and LBB assined as the first element of the row. In last case, LBB is swapped with the first element of the row and also add to last element. An example is presented below to have a better understanding of this modification phase.

Let's assume that hospital 2, 5 and 8 are selected as LBB and hospitals 1, 3, 4, 6, 7 are assigned to these three LBB. Then we have a pre-route matrix constructed by looking assignment matrix,

$$\begin{pmatrix} 1 & 2 & 3 \\ 5 & 6 & 9 \\ 4 & 7 & 8 \end{pmatrix}$$

In first route, hospital 2 (LBB) is not the first nor the last row so we swap it with the first element and then enlarge the matrix size by one and add hospital 2 as last element. Therefore we have a daily route 2-1-3-2. (Case3), In second route hospital 5 (LBB) is the first element, so we enlarge the row size by one and added hospital 5 to last row. Then, 5-6-9-5 becomes our second daily route (Case1). In last route, hospital 8 is the LBB and last element in the row, algorithm increase the row size by one, shift all element to right and add hospital 8 to first element's position in the row. Then, we have 8-4-7-8 as our last daily route. Lastly algorithm return the daily route matrix DR as follows.

$$\begin{pmatrix} 2 & 1 & 3 & 2 \\ 5 & 6 & 9 & 5 \\ 8 & 4 & 7 & 8 \end{pmatrix}$$

As it seen from the example, the algorithm do no consider the distance between the hospitals, daily routes found in this stage are not the optimal ones.Therefore, savings algorithm and 2-opt algorithm are applied to daily routes every time after using this algorithm to improve current routes.

Savings Algorithm

Savings algoritm is used to minimize the distance traveled by daily vehicles in their routes. Algorithm takes the daily route matrix and distance matrix as input and return the daily route matrix by rearranging the routes based on savings concepts. In this part, classical clark-wright algorithm [55] is applied which is a heuristics algorithm that yields good solutions for VRPs.

The idea of the savings algorithm express that the savings can occur by joining two routes into one route. The figure shows the before and after case of joining routes.

In this figure, 0 represents the LBB, i and j are the hospitals which are assigned to LBB. The algorithm firstly construct the savings matrix by using following formula:

$$S(i,j) = Dist(0,i) + D(0,j) - D(i,j) \tag{4.14}$$

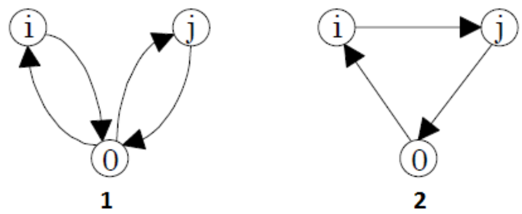

Figure 4.3: Savings Concept

Then, all i and j pairs are sorted in descending order of savings value. Each of the routes are considered one by one and combined until there is no hospital pairs to assign that route. Since there is no capacity constraint on daily vehicle, algorithm always returns one route for one LBB. Same steps are applied to all LBBs in the SR matrix.

2-Opt Algorithm

2-opt algorithm is an improvement method that start from a given solution and tries to improve this solution by erasing two edges that cross and reconnect the resulting two paths by edges that do not cross. The figure 4.4 illustrates 2-opt move to a route. In our problem, 2 opt takes one or multiple routes represented by a matrix SR or BR and return same matrix with an in-route improvement if possible.

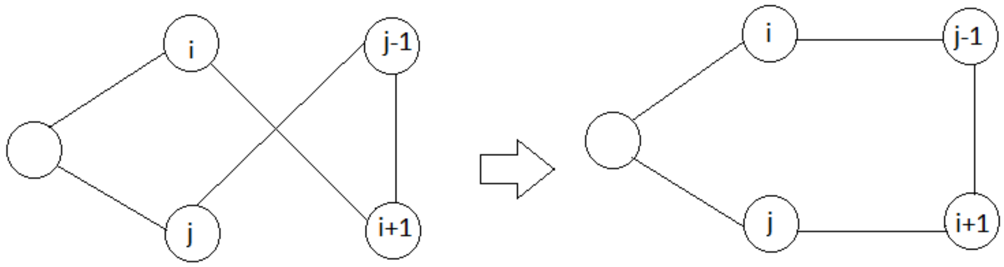

Figure 4.4: 2-Opt Move

The algorithm firstly checks the number of hospitals in the route and if there is more than 3 hospital which means, there must be at least 3 edge to continue. Then, it begins to calculate the distance improvement of erasing 2 of the edges and constructing the two possible edges with following formulas:

$$D_{old} = Dist(i, i+1) + D(j-1, j) \tag{4.15}$$

$$D_{new} = Dist(i, j-1) + D(i+1, j) \tag{4.16}$$

$$D_{imp} = D_{new} - D_{old} \tag{4.17}$$

All of the possible edges are calculated in this manner and the move that gives the best D_{imp} value is applied. The iteration ends with only one improvement move.

Location-Allocation

In this phase, initial solution is improved by using SA algorithm. Two move is designed to search the neighborhood solution of the initial solution by changing the location decisions matrix X. *Add-Drop* and *Swap* moves is used depending on their selection probability. The aim is to favor the best move for their time. For instance, Probability of Add-Drop move is expected to be high at the beginning of the iterations to find minimum number of LBBs. On the other hand, after reaching the correct number of facilities, swap move would lead to find correct locations of the LBBs.

As previously mentioned, at the end of a location move whether it is a Add-Drop or Swap , allocation procedure begins to assign hospitals to LBBs. Then, Bigroute, Smallroute ,savings and 2-opt algorithms applied consequently as it's applied in the initialization phase to calculate new routes and finally new solution cost. This procedure continues until SA algorithm for this phase stops and return the best solution found during the iterations.

Add-Drop Move

Add-Drop is a location move that opens or close the selected hospital/LBB depending their current state. Move randomly selects one of the hospitals in location decision matrix and checks if it is open or close. If selected hospital is closed which means not selected as LBB, it change it's state and opens that hospital. In other case, when selected hospital is a LBB, then it firstly checks the total demand and vehicle capacity and if the move is feasible, LBB is closed.

Swap Move

Swap move open one of the hospitals and close one that is currently closed simultaneously. Number of open facilities remain constant and move allow to find correct hospitals to be opened. Therefore, feasibility check is not necessary in this move.

Weekly Route Improvement

Weekly route improvement starts with the result of the location-allocation phase. Best weekly route found in location-allocation phase becomes the current solution of this process. Improvement in the route derived from the 2-Opt algorithm which is explained in previous subsection.

2-opt algorithm makes only one move after searching the best improvement route within the route. Therefore, we applied 2-opt algorithm consequently until there is no improvement in weekly route cost.

Daily Route Improvement

The goal of the daily route improvement phase is to improve the daily route which is constructed from the best solution of the location allocation phase. As it is explained earlier, location allocation phase determines the optimal number of LBBs and the assignment of hospitals to opened LBBs. Therefore, daily routes between LBB and hospitals was constituted in this phase by applying *SmallRoute* algorithm to best solution. This solution becomes the current solution of the daily route improvement.

Current solution is improved in two ways depending on the number of the opened LBB. If there is only one LBB, thus only one route, weekly route improvement methodology is applied to this route and 2-opt algorithm provides the within route improvement. In second case, there could be more than one LBB and more route accordingly. The route improvements achieved by SA algorithm by using only insert move in this phase. Insert move allow us to search improvements between routes if it exists. Since, hospitals are assigned to nearest open LBBs, route structure and the cost only considers the minimum distance, but there's also a inventory holding cost aspect. Therefore, Insert move can assign a hospital to a LBB which is not the closest one. Insert move is applied until SA algorithm ends. At the end of this phase optimal daily route is returned.

Insert Move

One of the hospitals is randomly chosen and randomly inserted to any other open LBB if there is enough capacity in that LBB's route.

Chapter 5

COMPUTATIONAL STUDIES

This chapter presents the generation of the data, selection of the parameters of our heuristic algorithms and the computational results through numerical experiments. As mentioned in previous chapters, three methods are used to solve and evaluate the SCNDP model. Exact and piecewise approximate methods are coded in GAMS whereas the proposed SA based meta-heuristic was implemented in C language. All of the methods was run on a PC Intel Core i5-2520M CPU (2.50 GHz) and 4 GB RAM.

The SA-SCNDP Algorithm is evaluated by comparing exact and approximate solution results since there was no study directly related with our problem to benchmark with. The performance of the SA-SCNDP algorithm is evaluated in terms of solution quality which is best solution found and computational efficiency in CPU time.

The computational studies chapter include three sections. Firstly,in section 5.1 we explained how we generate our data and give information about our test instances. Then, in Section 5.2.2,we presents the parameters settings for SA-SCNDP heuristics. Lastly, results of the numerical experiments are discussed in section 5.3.

5.1 Data Generation

In model formulation part, current model and proposed model of the Istanbul's blood distribution network are explained in detail. Since, both models are not studied before, we generated new test instances with different parameter values and problem sizes. In order to prove our new model's efficiency we need to know real parameter values of the current system, but we are unable to get the information of parameters from TRC officials. Nevertheless, we try to generate data as close as the real parameter values.

In test bed, we created 13 instances for each problem size. This problem sizes are expressed as ixj=(10x10),(20x20), (50x50), (75x75) and (100x100) where i is the number of hospitals and j is the number of candidate LBBs. Since all hospitals can be selected as LBB in our model, number of hospitals are always equal to nunber of LBB j, thus (i=j) for all instances. While we are creating the different instances, we firstly select instance 1 as our original setting and change only one parameter value to create another instance. Three level is used to change one parameter as Medium, High and Low shown as M, H, L. The structure of the instance setting can bee seen in Table 5.1.

Table 5.1: Parameters Settings

Instance	CSL	Demand	Std. Deviation	Facility	Holding	Routing
1	M	M	M	M	M	M
2	M	M	M	M	M	H
3	M	M	M	M	M	L
4	M	M	M	M	H	M
5	M	M	M	M	L	M
6	M	M	M	H	M	M
7	M	M	M	L	M	M
8	M	M	H	M	M	M
9	M	M	L	M	M	M
10	M	H	M	M	M	M
11	M	L	M	M	M	M
12	H	M	M	M	M	M
13	L	M	M	M	M	M

The location of the main blood bank and 100 hospitals in asian side of the Istanbul are collected in terms of latitude and longitude. Then,distance matrix for 101 points are calculated by finding distance between two points using formula below:

$$D_{i,j} = K\sqrt{(latitute_i - latitute_j)^2 + (longitude_i - longitde_j)^2} \tag{5.1}$$

where K is a constant which allow us to convert distance in meters and set as 141. After all distance are generated, we check that all distances' are satisfying the triangle inequality. Distance table for size 10 is given in table 5.2 in terms of meters.

The demand of each hospital generated by using the yearly blood demand data of 80 hospitals which is found in the annual health statistics report 2006 of the ministry of health.

Table 5.2: Node 10 Distance Table

(ixj)	0	1	2	3	4	5	6	7	8	9	10
0	0	2815	1516	14917	11281	2370	12572	3944	8517	2841	8881
1	2815	0	2168	12103	8936	2374	9836	1616	6324	1926	6128
2	1516	2168	0	13948	9873	865	11929	3700	7084	1373	8220
3	14917	12103	13948	0	7208	13587	4027	11344	8760	12803	6470
4	11281	8936	9873	7208	0	9172	8535	9213	2811	8501	7160
5	2370	2374	865	13587	9172	0	11798	3988	6366	784	8112
6	12572	9836	11929	4027	8535	11798	0	8682	8825	11046	3711
7	3944	1616	3700	11344	9213	3988	8682	0	6864	3514	5044
8	8517	6324	7084	8760	2811	6366	8825	6864	0	5711	6301
9	2841	1926	1373	12803	8501	784	11046	3514	5711	0	7372
10	8881	6128	8220	6470	7160	8112	3711	5044	6301	7372	0

Firstly, we calculated 2012's yearly demand data by assuming 5% increase at each year's blood demand from 2006 to 2012. Moreover, we generated 20 new demand without changing the average and standard deviation of 80 hospitals demands. The values converted to weekly demand data and set as the medium level for demand parameter. High and low demand levels are generated by taking twice and half of the medium demand. Finally, we generated three different level of standard deviation by multiplying demand with 0.5, 0.3 and 0.1 for the high, medium and low levels respectively. Tables presents the demand and std.deviations can be found in Appendix.

The establishment cost of a LBB is calculated by assuming that one of the rooms in a hospital turn into a warehouse/stocking room or the current room where they keep their blood stock can be expanded. In both cases, cost of the opening a LBB in that hospital would be less than constructing a new warehouse building. Therefore, we assumed that the cost of new warehouse area would be 120000 TL. The weekly payment of this investment calculated as 300 TL/week by considering these payments continues for the next 10 years.

As we stated in earlier chapters, cost of blood is high due to expensive processes during the procurement, quality control and mandatory tests. Therefore, we estimated 1500 TL cost for per blood unit. Then, holding cost is calculated due to weekly opportunity cost of this price by considering %10 interest rate. Thus, holding cost set to 3. The weight factor

Table 5.3: Parameter levels used in experiments

Parameters	Levels		
	High	Medium	Low
LBB locating cost	600	300	150
Inventory holding cost	9	3	1
Transportation cost of weekly vehicle	0.010	0.005	0.0025
Transportation cost of daily vehicle	0.006	0.003	0.0015
Customer service level	0.99	0.97	0.95

associated with the transportation cost of weekly and daily trucks is calculated based on the oil prices and the fuel consumption of the vehicles. Basically, the weekly trucks consumes more than daily truck and costs more. Therefore, we suggest to set 0.005 and 0.003 for the weekly and daily transportation costs respectively. The probability of a stock out during replenishment is set to 0.03. Therefore, desired customer service level (CSL) is become 0.97 for which the corresponding $F^{-1}(CSL)$ is equal to 1.88. High and low CSL is set to 0.99 and 0.95. Table 5.3 represents the different levels of the parameters mentioned above.There are also constant cost parameters which are used throughout the computational experiments. One of them is the variable cost per unit passes through a LBB which is 0.4. The lead time in days is set to 1. Owning a weekly replenishment vehicle is 325. The daily and weekly replenishment vehicles are assumed uncapacitated for simplicity. All constant parameters presented in table 5.4.

Table 5.4: Constant parameters values used in experiments

Parameters	Value
Variable cost	0.4
Lead time in days	1
Owning a replenishment vehicle	325

5.2 Parameters

5.2.1 Piecewise Approximation Parameters

In piecewise approximation method, the number of the pieces, also the number of breakpoints, in the function determines the approximation error and the computational complexity. Therefore we tried three different parameter values 10, 20 and 50 to find the best solution in terms of solution quality and CPU time. Test results with selected parameter values for instance 1 are shown in Table 5.5

Table 5.5: Test results for number of pieces in approximation function

	Number of pieces =10			Number of pieces =20			Number of pieces =50		
Problem Size	LB	GAMS Cost	CPU time	LB	GAMS Cost	CPU time	LB	GAMS Cost	CPU time
10x10	2624,67	2624,67	32,49	2624,67	2624,67	43,12	2624,67	2624,67	75,12
20x20	4055,65	4473,86	7200	4055,65	4473,86	7200	4055,65	4473,86	7200

As you can see from table below, number of pieces only increases the solution time for 10x10 size problem and have no effect on solution quality. Since for all of the parameters, optimal solution is found regardless of the piece number. In 20x20 sized problem, method couldn't find optimal in a reasonable time, therefore we put a time limit, 3 hr, and again the number of pieces have no effect on solution quality. Therefore, we selected number of pieces parameters as 10.

5.2.2 Heuristic Parameters

The parameters setting for heuristic algorithms is highly critical since the solution quality of the problem depends on the correctness of these values. The generic decisions of the Simulated Annealing algorithm is the initial temperature values, cooling schedule,cooling rate, replication numbers and the stopping condition values. These decisions controls the balance between diversification and intensification during the search. In order to test the effectiveness of heuristic parameters, we selected instance 1 parameters for all problem sizes. Heuristic parameters initially set as follows:

The initial temperature value T_i must be high enough to allow almost free exchange of neighboring solutions. Therefore, we try four different high temperature values for each

Table 5.6: Initial value of parameters in SA-SCNDP

T_i	100
T_f	0.01
cooling schedule	$T_{k+1} = \alpha T_k$, $\alpha = 0.90$
maximum replication number (Nrep)	L/5

population size. Tested initial temperatures are shown in Table 5.7.

Table 5.7: Candidate values used as T_i for SA

Problem Size	Candidate Initial Temperature Values
10x10	100, 250, 500 ,1000
20x20	100, 250, 500 ,1000
50x50	100, 250, 500 ,1000
75x75	100, 250, 500 ,1000
100x100	100, 250, 500 ,1000

The selection of the suitable cooling schedule is highly important for the performance of the SA algorithm. T values are updated according to a defined cooling schedule. Therefore, a good cooling schedule will lead to near optimal/optimal solutions. In our case, one of the most used cooling schedule that follows a geometric law is selected as the cooling schedule since it's practical, fast and easy to use in applications. We tested four different cooling rates as $\alpha = 0.85, 0.90, 0.95, 0.99$. The effect of different T_o's and α values can be seen in Table 5.8. Each value shown in the table are averaged over 30 experiments.

Table 5.8 indicates that initial temperature value increase effects positively on the objective value up to some level. It can be seen that initial temperatures which is more than 250 has worsened the objective value almost all of the problem sizes. Therefore, T_i is set as 250 and used for all of the problem sizes throughout the experiments. We see that the initial temperature does not significantly effect the CPU time. On the other hand, the increase in cooling rate, change the CPU time. The increase in cooling rate leads to slow cooling which means increase in the iteration number. Moreover, the dominant α level on solution quality is found 0.95 and set as the cooling rate for all problem size.

Table 5.8: Preliminary test results for T_i and cooling rate "α" setting

		$\alpha = 0.85$		$\alpha = 0.90$		$\alpha = 0.95$		$\alpha = 0.99$	
Problem Size	T_i	Cost	Runtime	Cost	Runtime	Cost	Runtime	Cost	Runtime
10x10	100	2708	0,04	2682	0,04	2638	0,05	2625	0,08
	250	2626	0,04	2632	0,04	2624	0,05	2624	0,08
	500	2624	0,05	2624	0,04	2625	0,06	2659	0,10
	1000	2632	0,05	2624	0,04	2624	0,06	2633	0,12
20x20	100	5113	0,04	5092	0,04	4795	0,07	4805	0,35
	250	5013	0,03	4820	0,04	4591	0,07	4838	0,35
	500	5046	0,03	4932	0,05	4758	0,07	5444	0,45
	1000	5127	0,04	4897	0,05	4860	0,08	5832	0,45
50x50	100	15815	0,62	15768	0,98	14967	1,55	15403	7,52
	250	15748	0,64	15431	1,02	14686	1,99	15973	9,67
	500	16670	0,59	17835	0,98	16839	2,02	16855	9,83
	1000	18195	0,61	18099	1,22	17982	2,24	18122	13,45
75x75	100	24456	0,98	23941	1,75	23584	3,82	24319	15,52
	250	23579	1,21	24416	2,11	23358	3,98	24724	18,34
	500	24888	1,28	25058	2,32	25995	4,13	26598	19,16
	1000	26241	1,37	26244	2,64	27920	5,30	28075	25,21
100x100	100	34361	1,91	33744	3,20	32750	6,40	32841	32,80
	250	33787	2,23	34218	3,20	34063	6,90	33604	35,30
	500	35291	2,50	33505	3,90	36329	7,20	36995	41,20
	1000	37021	2,53	37750	4,10	38781	8,90	38014	48,40

The number of maximum iterations (Nrep) at each temperature is imposed to limit the time spend at very low temperatures. The iteration numbers are chosen with respect to length of the solution representation (L). Thus, Nrep number is unique for each of the problem sizes. The effect of the three different Nrep values are shown at table 5.9.

From Table 5.9, it can be observed that the increase in nrep number, results in an increase on computational times. Moreover, this increase also yield slightly better results in terms of objective function values. Therefore, L/2 which represents the half of the solution representation is selected as the nrep for all problem sizes.

The stopping condition of the system is expressed in terms of the final value of temper-

Table 5.9: Test results for Nrep settings

	L/10		L/5		L/2	
Problem Size	Cost	RunTime	Cost	RunTime	Cost	RunTime
10x10	2627	0,02	2624	0,02	2624	0,03
20x20	4701	0,07	4690	0,13	4934	0,28
50x50	15797	0,54	15706	0,92	15297	1,78
75x75	24891	1,36	25266	2,93	24898	5,42
100x100	34908	3,36	34982	6,13	34346	16,43

ature up to this point. However, we also tested two different stopping condition to see the effects on objective value. The total number of iterations performed and maximum number of consecutive iterations while best solution does not improve. The parameters for stopping conditions can be seen in Table 5.10.

According to the preliminary test results, stopping condition with maximum number of iterations gives slightly better result than maximum number of non-improving iteration number despite the fact that it consumes more time. Moreover, it is clear that the increasing number of iterations, yields an increase in CPU time but it does not provide better cost values at all time. Therefore, we selected best number of iterations which gives the most effective objective values for each of the problem size. The stopping condition values which used for the rest of the experiments are shown in Table 5.11

The final parameter values selected for SA heuristic are summarized in Table 5.12 . The parameters used for all instances and problems sizes throughout the computational experiments.

Table 5.10: Preliminary test results for stopping condition setting

	Maximum iterations			Maximum non-improving iterations		
Problem Size	Iteration #	Cost	Runtime	Iteration #	Cost	Runtime
10x10	500	2630	0,02	250	2630	0,02
	1000	2624	0,03	500	2624	0,03
	2000	2624	0,04	1000	2624	0,04
20x20	1000	5092	0,14	250	5102	0,11
	2000	4839	0,32	500	4923	0,15
	4000	4892	0,48	1000	5080	0,43
50x50	2500	14882	1,11	250	15782	0,62
	5000	15332	2,14	500	15420	0,70
	10000	15763	4,32	1000	15448	0,91
75x75	3500	24206	4,23	500	25068	2,18
	7000	23588	7,07	750	24440	2,76
	15000	24822	12,25	1500	24242	3,67
100x100	5000	35032	10,78	1000	35185	6,13
	10000	34707	17,82	1500	35102	9,34
	20000	34024	38,21	2000	34734	14,92

5.3 Computational Results

In this section, results of the numerical experiments are explained in terms of solution quality and computational run time. The solutions of the exact, piecewise and heuristics solutions are compered for small size problems. Moreover, current distribution system which is explained in Chapter 3 is also studied and total system costs are compared with the proposed system costs.

For the mid-size problems, lower bound of the piecewise approximation solution is used for comparing the efficiency of the heuristics solutions. The percentage deviation of objective function from lower bounds is used as a performance parameter for each problem instance. The performance of the heuristics algorithm evaluated by using the best, average and worst solution of the objective function values obtained by SA-SCNDP algorithm over 30 experiments. The deviation from LB is calculated as follows.

Table 5.11: Final values used as stopping condition

Problem Size	Max. # of iterations
10x10	1000
20x20	2000
50x50	5000
75x75	7000
100x100	10000

Table 5.12: Final value of parameters in SA-SCNDP

T_i	250
T_f	0.01
cooling schedule	$T_{k+1} = \alpha T_k$, $\alpha = 0.95$
maximum replication number (Nrep)	L/2

$$\%deviation = \frac{ObjectiveValue - LB}{ObjectiveValue} \tag{5.2}$$

Lastly, computational results and effect of parameters on test instance are shown in large size problems. Since, the exact and approximate methods do not provide optimal or lower bound solutions for large size problems, comparison is made between the current system costs and proposed cost which is calculated as:

$$\%difference = \frac{CurrentSystemCost - ObjectiveValue}{CurrentSystemCost} \tag{5.3}$$

5.3.1 Results

Firstly, we try to solve all problem sizes for instance one by applying three solutions methods that we mentioned earlier. All parameters' levels are selected as medium in instance one as it shown in parameters setting (Table 5.1). Solutions obtained by the exact, piecewise and best solution of heuristic algorithm is represented in Table 5.13. The best solution of the heuristic algorithm is found after 30 replications. For problem size 50 and more, we applied 8 hour time limit for exact and approximate solution methods.

Table 5.13: Test Results for Instance 1

	Current System	Exact Solution		Piecewise Method				Heuristic Solution	
Problem Size	Cost	Cost	CPU time	LB	GAMS Cost	Actual Cost	CPU time	Cost	CPU time
10x10	2865,19	2624,67	3240	2624,67	2624,67	2624,67	32,49	2624,67	0,03
20x20	5658,00	-	-	4055,65	4473,86	4473,86	7200	4463,09	0,21
50x50	23607,73	-	-	9532,82	14838,42	15274,07	28800	14204,24	1,63
75x75	39232,07	-	-	-	-	-	28800	22846,87	7,12
100x100	57446,91	-	-	-	-	-	28000	31195,36	27,23

As we mentioned earlier, system costs of the current distribution network is theoretically calculated to compare with the proposed new system. The related costs with the current system includes the inventory holding costs of the all hospitals, the weekly routing costs, replenishment vehicle costs and variable costs. First column of table above represents the total current system costs for all problem sizes for instance 1.

Exact solution method is able to solve our problem optimally only for problem size 10x10. For the medium and large sized problem instances we use piecewise and heuristic methods to find solutions. As results of 10x10 problem indicates, exact method is the slowest method in terms of CPU time. On the other hand heuristic solution found same optimal solution less than 1 second.

As it can be seen, piecewise approximation has three solution representation (LB, GAMS and Actual) which GAMS and Actual gave the same results for 10x10 and 20x20 problem size. When we solve the problem in Cplex solver, it gives best possible and final solution which construct LB and GAMS solution in our table. Then, we manually recalculate the safety stocks to find actual inventory costs to use in our solution rather than using approximate costs. However, for 10x10 and 20x20 problem size, model opens only one LBB as an optimal solution. Therefore, approximation methods estimates safety stock values precisely rather than giving an approximate value which is less than actual cost. Moreover, the difference between actual and gams can be seen for 50x50 problem size which is also prove our claim that the piecewise approximation is a LB to actual cost. As the problem size is increased further, piecewise methods begins to fail and not able to find solutions. In terms of

CPU time, approximation gives faster solutions than exact but slower than heuristic method.

Table 5.13 indicates that the heuristic solution is the only method which solves the problem with all problem sizes in a reasonable time. In terms of computational results, heuristics manage the find optimal solution for 10x10 and find better solutions than then piecewise approximation for 20x20 and 50x50 problem size. Furthermore, for the problem size 75x75 and 100x100, heuristic is the only method which give solution for this instance. Therefore, comparison between current system and proposed system is made by using heuristic method's solution. The improvements achieved at every problem size for instance one is shown in Table 5.14.

Table 5.14: Cost Improvements % of the Proposed Network Model

	Current System	Proposed System	
Problem Size	Cost	Cost	% Imp.
10x10	2865,19	2624,67	8,39%
20x20	5658,00	4463,09	21,12%
50x50	23607,73	14204,24	39,83%
75x75	39232,07	22846,87	41,76%
100x100	57446,91	31195,36	45,70%

As table above indicates, cost improvements are achieved by using the proposed model. The improvement percentage increases as the problem size increase. Thus, we conclude that risk pooling advantage for this problem greater than the cost disadvantage caused by the daily routing for this instance. Cost improvements change from 8,39% to 45,70%.

Lastly, in Table 5.15, hospitals which are selected as a blood bank and the number of hospitals assigned to opened LBBs of the best solutions are presented. The results show us that the number of LBB increases as the number of hospitals increase. In 10x10 and 20x20 sized problems, only one hospital is selected as a LBB and all other hospitals assigned to this LBB. The assumption of each route served by a one daily vehicle makes problem possible to serve 19 hospitals in a daily route. In 50x50 sized problem, problem is solved with heuristic method and three hospitals are selected as LBB and on average 15 hospitals are assigned to one LBB. For 75x75 and 100x100 sized problems, 6 and 11 LBBs are located. In these solutions, some of the LBBs are selected without hospitals assignments due to their high demand or their location with respect to main blood bank and other hospitals.

Table 5.15: Blood bank and Hospital Assignments

Problem Size	# of opened hospitals	LBB	# of hospitals assigned to LBB
10x10	1	Hospital 2	9
20x20	1	Hospital 19	19
50x50	3	Hospital 19	16
		Hospital 28	13
		Hospital 42	18
75x75	6	Hospital 16	8
		Hospital 30	12
		Hospital 31	13
		Hospital 43	18
		Hospital 63	18
		Hospital 71	0
100x100	11	Hospital 2	1
		Hospital 15	24
		Hospital 23	1
		Hospital 24	15
		Hospital 30	9
		Hospital 42	19
		Hospital 58	1
		Hospital 60	4
		Hospital 68	15
		Hospital 71	0
		Hospital 88	0

5.3.2 Sensitivity Analysis of parameters

In data generation, we introduce 13 different problem instance. All instances are created by keeping all parameters of the original setting (instance 1) same and change only one parameter value as high or low to create new instance.(Table 5.1) To examine the effects of different parameter levels, we tested all problem size with 13 instances and presented difference of solutions methods and differences of current and proposed system costs.

Since the commercial solvers are able to solve MINLP optimally for small size problems (10x10), we use the problem instances of the small size problem to compare the exact, piecewise approximate and heuristics solution methods. The table 5.16 represents the objective value (costs) of the proposed system and the CPU time in terms of seconds.

Table 5.16: Optimal Solutions of 10x10 Problem Size

	Current System	Proposed System					
		Exact Solution		Piecewise Method		Heuristics Solution	
Instances	Cost	Cost	CPU time	Cost	CPU time	Cost	CPU time
1	2865,19	2624,67	3240	2624,67	32,49	2624,67	0,03
2	2996,26	3352,78	7260	3352,78	58,81	3352,78	0,03
3	2725,41	2260,62	3120	2260,62	29,37	2260,62	0,03
4	6543,13	4274,99	3582	4274,99	33,91	4274,99	0,03
5	1573,21	2074,56	9960	2074,56	76,18	2074,56	0,03
6	2865,19	2924,67	2178	2924,67	15,06	2924,67	0,03
7	2865,19	2474,67	16860	2474,67	148,77	2474,67	0,03
8	4071,19	3175,85	1197	3175,85	37,16	3175,85	0,03
9	1576,93	2075,26	4005	2075,26	31,12	2075,26	0,03
10	5142,55	3897,98	708	3897,98	29,75	3897,98	0,03
11	1661,03	1088,69	2477	1088,69	53,46	1088,69	0,03
12	3282,92	2831,54	3566	2831,54	33,13	2831,54	0,03
13	2600,06	2529,21	3045	2529,21	28,59	2529,21	0,03

As it can be seen from the Table 5.16 all of the methods solve the problem optimally for all of the instances. However, the computational time causes the main difference between the methods. Exact method which is found by using GAMS Baron solver is the slowest method among them. The CPU time of this method varies from 20 minute to 5 hour depending on the problem instances. On the other hand, piecewise approximation method which is constructed by eliminating the non-linear terms of the MINLP gives faster solutions than exact method. One can expect that there should be a difference between exact and approximate method in terms of cost but for this problem size model opens only one LBB. Therefore, approximation model estimates safety stock value precisely rather than giving a approximate value. Lastly, we observe that heuristic solutions solve problem faster than all of other methods. CPU time shows that it solve the problem less than 1 second.

Another comparison is made between the solution of the proposed model and current system. As you can see from table5.17, proposed model gives better costs than the current system for the 7 of the 13 instances. For the instance 2, 5, 6 and 9, it would be wise the stay with the current system and do not open any LBB. Since, our model force to open LBB, cost associated with this four instance is higher than the current system. The comparison results also show us that for problem size 10x10, high routing cost, low inventory holding cost, high LBB location cost and low variance structure makes the proposed model worse than the current system. Because, cost savings from inventory aggregation are not better than the daily routing cost and LBB opening cost increase.

The computational complexity of the problems increases due to the increasing problem size. As mentioned above tables, exact methods are struggling even for the small size problems when we compare to approximation and heuristics methods. Thus, we are unable to use exact method for medium and large size problems to find solutions. Therefore, we begin to use lower bounds by using the approximation method and compare the quality of the solutions obtained by heuristics solutions. Table 5.18 represents the Lower bounds found by GAMS Cplex with 2hr time limit and the heuristic solutions for problem size (20x20). Heuristic solutions are shown as the best, average and the worst solutions of 30 replications for each instance. Furthermore, % difference from current system and % deviation from the lower bounds are shown in Table 5.19 to evaluate the performance of both Piecewise and Heuristic methods.

According to the results of the Table 5.18, average values of the heuristic solutions is

Table 5.17: Current vs Proposed Model for problem size 10x10

	Current System	Proposed System	
Instances	Cost	Cost	% Difference
1	2865,19	2624,67	8,39%
2	2996,26	3352,78	-11,90%
3	2725,41	2260,62	17,05%
4	6543,13	4274,99	34,66%
5	1573,21	2074,56	-31,87%
6	2865,19	2924,67	-2,08%
7	2865,19	2474,67	13,63%
8	4071,19	3175,85	21,99%
9	1576,93	2075,26	-31,60%
10	5142,55	3897,98	24,20%
11	1661,03	1088,69	34,46%
12	3282,92	2831,54	13,75%
13	2600,06	2529,21	2,72%

better than the solutions of the Piecewise methods obtained in two hours for the most of the instances. The best solutions found in 30 replications of the heuristic algorithm also indicates that the heuristic's best solutions are superior than the piecewise methods almost for all instances. Moreover, the worst solutions found by heuristic is not too far from the piecewise solutions. On the other hand, heuristics algorithm did not find better values than the lower bound found by piecewise method in any of the instances. In terms of the computational time, we clearly see that heuristic algorithm provide fastest solutions. The time spent to find a solution for 20x20 size problem is always below one second for all instances.

Table 5.18: LB and Heuristic Solutions of the 20x20 Problem Size

	Current System	Piecewise Method				Heuristic Solution			
Instance	Cost	Lower Bound	Gams Cost	Actual Cost	CPU Time	Cost (Average)	Cost (Best)	Cost (Worst)	CPU Time
1	5658,00	4055,65	4473,86	4473,86	7200	4561,34	4463,09	5000,23	0,21
2	6144,69	5015,67	8057,83	8295,51	7200	6780,37	6413,28	6910,32	0,21
3	5414,66	3352,33	3975,61	3975,61	7200	3790,54	3546,04	4432,21	0,21
4	12956,64	6298,67	6715,11	6715,11	7200	6794,97	6727,06	7196,76	0,21
5	3225,13	3105,48	4254,98	4276,96	7200	3889,42	3708,43	4645,31	0,21
6	5658,00	4509,40	5164,06	5164,06	7200	4873,69	4763,09	5392,36	0,21
7	5658,00	3567,32	4529,64	4529,64	7200	4442,59	4313,09	4744,77	0,21
8	8107,62	4694,97	5979,97	6067,58	7200	5313,97	5218,51	5591,11	0,21
9	3230,70	3022,69	4826,60	4911,26	7200	3860,89	3710,01	4167,14	0,21
10	11112,76	5866,66	6957,32	7132,97	7200	6604,50	6464,67	7411,16	0,21
11	3033,49	2801,24	3818,62	3818,62	7200	3552,51	3461,35	4198,85	0,21
12	6568,23	4199,99	5104,16	5104,16	7200	4944,91	4746,88	5312,68	0,21
13	5231,13	3829,92	4512,21	4512,21	7200	4473,23	4332,12	4896,31	0,21

When we compare the solutions of heuristic and piecewise method with the current system costs, we see that proposed model is better than the current for the nine of the instances. Current system still better off in some cases such as routing cost is high, inventory holding cost is low, standard deviation of average weekly demand is low and the average weekly demand is low. The performance of the proposed system can be observed from table 5.19 more easily. For the instance one which all of parameter levels are medium, proposed model is better %20,929 than the current according to piecewise solution. As it can be recall from Table 5.17 the difference between current and proposed was %8,39 for 10x10 problem sized and when we look at the best solution of the heuristic algorithm, the percentage increases and %21,119. Therefore, we can conclude that with the current cost parameters, proposed model is better than current system. Furthermore,we conclude that cost savings increases with the increasing problem size.

Table 5.19: Performance of Piecewise and Heuristics methods for problem size 20x20

	% Difference from Current System Cost				% Deviation from Lower Bound			
Instance	Piecewise Method	Heuristic Solution			Piecewise Method	Heuristic Solution		
		Average	Best	Worst		Average	Best	Worst
1	20,929%	19,383%	21,119%	11,625%	9,348%	11,086%	9,129%	18,891%
2	-35,003%	-10,345%	-4,371%	-12,460%	39,538%	26,027%	21,792%	27,418%
3	26,577%	29,995%	34,510%	18,144%	15,678%	11,561%	5,463%	24,364%
4	48,172%	47,556%	48,080%	44,455%	6,202%	7,304%	6,368%	12,479%
5	-32,614%	-20,597%	-14,985%	-44,035%	27,390%	20,156%	16,259%	33,148%
6	8,730%	13,862%	15,817%	4,695%	12,677%	7,475%	5,326%	16,374%
7	19,943%	21,481%	23,770%	16,141%	21,245%	19,702%	17,291%	24,816%
8	25,162%	34,457%	35,635%	31,039%	22,622%	11,648%	10,032%	16,028%
9	-52,018%	-19,506%	-14,836%	-28,986%	38,454%	21,710%	18,526%	27,464%
10	35,813%	40,568%	41,827%	33,309%	17,753%	11,172%	9,250%	20,840%
11	-25,882%	-17,110%	-14,105%	-38,416%	26,643%	21,148%	19,071%	33,286%
12	22,290%	24,715%	27,730%	19,115%	17,714%	15,064%	11,521%	20,944%
13	13,743%	14,488%	17,186%	6,401%	15,121%	14,381%	11,592%	21,779%

The best cost improvement can be seen in instance 4 which is the case that inventory holding cost is triple of the medium level. On the other hand worst difference is observed in instance 9 which the standard deviation of average weekly demand is at low level. Thus, we conclude that the holding cost is the most effective indicator that determines the performance of proposed system for the problem.

The performance of the heuristic algorithm with respect to lower bounds are also shown in Table 5.19. Again for the instance 1, heuristic algorithm's best solution is % 9,129 away from the LB whereas the piecewise solution is %9,348 away. On the other hand worst and average solution of the heuristic is worst than the piecewise solution. The minimum gap between LB and best of heuristic is obtained for instance 6 which is %5,326 away whereas piecewise solution is % 12,677 away from the optimal. On the other hand, greatest gap is seen in instance 2 with % 21,792. When, we consider all of the instances together, on average piecewise solutions are %21, best of the heuristics are %12,60 , average of heuristics % 15,36 and worst of the heuristics are %23,19 away from the LB.

The model is also solved for the 50x50, 75x75 and 100x100 problem size by using SA-based heuristic algorithm. The performance of them model is evaluated by comparing the cost of the current system with the best solution found after 30 replications of heuristic solution. As Table 5.20 presents, proposed system is better than current in all of the instances. In 50x50 improvements range from 14,59% to 59,12%. Improvement percentage improves as the number of hospitals in the system increase as it seen from the table below.

Table 5.20: Test Results for 50x50, 75x75 and 100x100

	50x50 Problem Size			75x75 Problem Size			100x100 Problem Size		
	Current System	Proposed System		Current System	Proposed System		Current System	Proposed System	
Instance	Cost	Cost	% Difference	Cost	Cost	% Difference	Cost	Cost	% Difference
1	23607,71	14204,22	39,83%	39232,07	22846,87	41,76%	57446,91	31195,36	45,70%
2	24747,14	17590,91	28,92%	41556,95	28338,28	31,81%	60903,71	38117,32	37,41%
3	23038,00	12137,42	47,32%	38069,64	21396,72	43,80%	55718,52	30551,79	45,17%
4	57255,11	23408,57	59,12%	95143,67	41530,42	56,35%	140410,36	56738,93	59,59%
5	12391,92	10583,86	14,59%	20594,88	17645,32	14,32%	29792,44	25029,53	15,99%
6	23607,71	16125,25	31,69%	39232,07	28389,24	27,64%	57446,91	33393,02	41,87%
7	23607,71	13950,86	40,91%	39232,07	22917,94	41,58%	57446,91	32922,12	42,69%
8	34884,90	18340,08	47,43%	57947,39	29055,31	49,86%	85213,00	39148,61	54,06%
9	12397,50	10350,89	16,51%	20600,46	16622,18	19,31%	29798,02	23893,54	19,82%
10	46557,80	24293,63	47,82%	77414,46	39829,15	48,55%	111917,74	54798,77	51,04%
11	12180,97	10082,46	17,23%	20418,81	15557,22	23,81%	30516,46	22136,09	27,46%
12	27825,41	16574,05	40,44%	46309,43	25961,46	43,94%	68242,08	34580,88	49,33%
13	21661,24	13573,86	37,34%	36066,48	21916,65	39,23%	53043,25	31492,21	40,63%

Moreover, we analyze the effect of the parameters on the total cost by using the problems instances setting which we showed in Table 5.1. The total cost of the current system and proposed system is analyzed by using the 50x50 problem size. Figure 5.1 and 5.2 presents the cost structures.

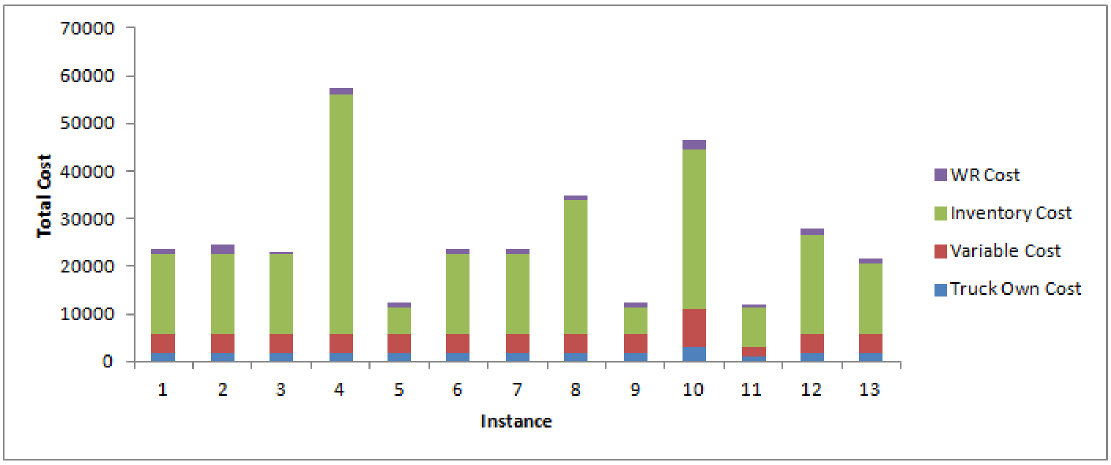

Figure 5.1: Bar Chart Graph of Current System Costs for 50x50 Problem Size

As it seen from above figure, the total cost of the current system is composed of four main cost components including replenishment truck owning cost, variable cost, inventory cost and weekly routing costs. Since daily routing cost in current system is not considered and there is no need for LBB establishment, facility cost and daily routing cost is set 0 for all of the instances in current system. On the other hand, in proposed model, all of the six cost components are included in total system cost.

In instance 1, we set all parameter levels as medium and inventory, variable, truck own and weekly route cost becomes % 70,91, % 16,94, % 6,85 and % 5,30 of the current total cost respectively. However, in proposed model, inventory cost becomes % 39,46, variable cost %28,30, daily routing cost % 19,51, facility cost % 8,45, truck owning cost %2,29 and weekly route cost %2,00 of the total cost. As expected, inventory cost which is the main cost component of the system, decreases when we change our decentralized structure to more centralized structure by LBBs are holding inventory instead of hospitals. The weekly routing and truck owning cost are also diminished since we use less replenishment trucks and only couple of LBB in a weekly routes instead of all hospitals. Despite the fact that, percentage of variable cost increases,in reality variable cost remain same. Since the variable cost directly related with the total demand of the system and not changed, decrease in total system cost lead an increase in variable cost percentage. Finally, as explained facility cost and daily routing costs emerge with the new system.

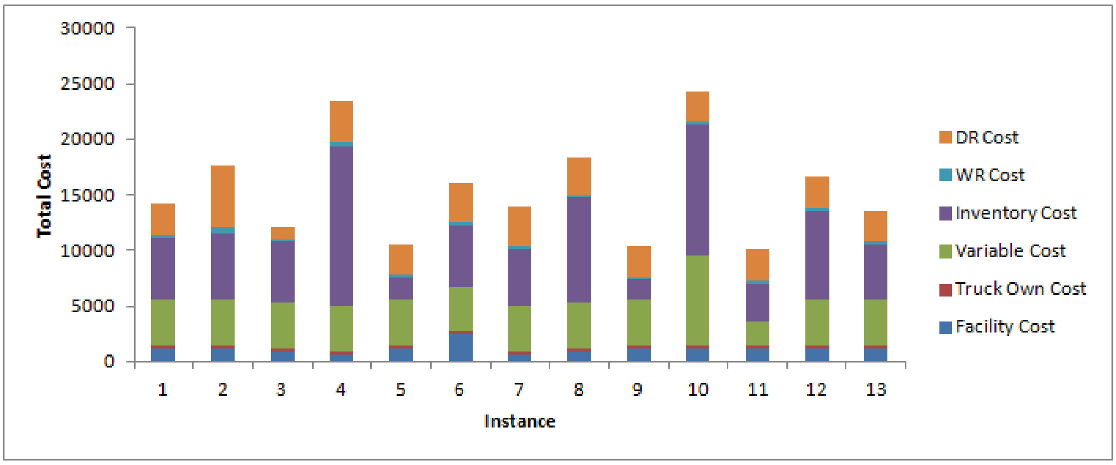

Figure 5.2: Bar Chart Graph of Proposed System Costs for 50x50 Problem Size

When we consider all of the instances, inventory cost ranges between 45-87%, variable cost 7-32% , truck owning 3-13% and weekly routing cost 2-10% as the percent of total cost in current system structure. On the other hand, in proposed model, inventory cost range between 18-52%, variable cost 17-38%, daily routing cost 9-32%, facility cost 3-15%, truck owning 2-3%, and weekly routing cost 1-3% as the percent of the total cost. The percentage of the total cost components for all of the instances can be seen in Appendix.

Chapter 6

CONCLUSIONS AND FUTURE RESEARCH

6.1 Conclusions

Supply chain network design problems traditionally have strategic, tactical and operational decision levels. These decisions mainly include facility locations, size of the facilities, procurement and distribution of items, daily shipments etc. In literature, this decisions have been studied in different sub-problems such as facility location, LRP, IRP and VRPs.

In this study, we aim to integrate these three different decision levels and design a complex supply chain network for blood distribution in Istanbul. In proposed model, we modeled a mixed integer nonlinear programming model that finds optimal number and locations of LBBs, assign hospitals to open LBBs, decide safety stock levels in opened LBBs and routes weekly and daily distributions together. The objective of our model is to minimize total system costs which include fixed cost of locating a LBB, cost of owning weekly replenishment vehicles, inventory costs at LBBs, variable cost of per unit passes through a LBB and transportation cost of weekly and daily deliveries while satisfying the pre-determined customer service level. Our problem differentiated from classical SCNDP by the integration of location, allocation, inventory and routing decisions are done together. Moreover, instead of having a couple of potential hospitals or LBB zones, all of the hospitals in problem can be selected as LBB. Therefore, our model is more complex and different than the problems seen in literature and strongly NP-hard.

We use three different methods to solve our model. Firstly, we use exact and piecewise-approximation methods by using commercial solves like GAMS Baron and CPLEX. We realized that exact methods can only solve the small size problem instances. Therefore, we modify non-linear terms of the model and apply an piecewise approximation to find lower bounds and approximate solutions. Furthermore, we provide a simulated annealing based heuristic (SA-SCNDP) to solve the model for mid and large size problem instances. The SA-SCNDP use four consecutive phase, initialization, location-allocation, weekly route improvement and daily route improvement to find optimal/near optimal solutions to the problem.

The performance of the SA-SCNDP heuristic are evaluated at three level with computational experiments. At first optimal solutions found by exact methods are compared with the Heuristic algorithm. Since, problem was not studies before, we don't have available data for the problem. Therefore, we generate new data set which can also be used in further studies. Computational study show that our heuristic solve the problem optimally for small-size problems of 11 instances and much faster than the exact methods. Then, for mid-size problems best-worst and average solutions obtained by heuristic is compared with the lower bounds found by the piecewise approximation method. Results indicate that heuristic solutions are better than the piecewise solutions for almost all of instances. Finally, for large size problems effects of parameters are examined in terms of the components of total system costs for current distribution network and proposed distribution network.

In summary, we manage to model a complex supply chain network for blood distribution that integrates facility location, allocation, inventory decisions and routing decisions. We solve the MINLP model by using GAMS and heuristic methods. The SA-based heuristic algorithm provide optimal and near optimal solutions in less than 1 minute. The framework of the blood distribution system which we modeled is also applicable with some modifications, to optimization of other supply chain network problems.

6.2 Future Research

The proposed model studied in this thesis is a relaxed distribution network which is suitable for real-word applications. There are many assumptions which is mentioned in chapter 3. One of them is the average weekly blood demands of the hospitals which considered deterministic and assumed as known. The problem with stochastic demand and uncertainty would be a good case for further studies and more suitable for real-life cases.

The SA based heuristic solution obtained optimal and good solutions for small and mid size problems but we cannot test or compare the success of the algorithm for large problem size. Therefore, new heuristic algorithms like tabu search (TS), genetic algorithm (GE), ant colony optimization (ACO) or variable neighborhood search (VNS) can be applied to obtain different solutions than our findings. Alternatively, approximation techniques which gives better lower bounds for mid and large problem size may be found for further studies.

BIBLIOGRAPHY

[1] F. Barahona, D. Jensen, Plant location with minimum inventory, Mathematical Programming 83 (1998) 101-111.

[2] S.J. Erlebacher, R.D. Meller, The interaction of location and inventory in designing distribution systems, IIE Transactions 32 (2000) 155-166.

[3] M.S. Daskin, C. Coullard, Z.-J.M. Shen, An inventory-location model: Formulation, solution algorithm and computational results, Annals of Operations Research 110 (2002) 83-106.

[4] Z.-J.M. Shen, C. Coullard, M.S. Daskin, A joint location-inventory model, Transportation Science 37 (2003) 40-55.

[5] P.A. Miranda, R.A. Garrido, Incorporating inventory control decisions into a strategic distribution network design model with stochastic demand, Transportation Research Part E: Logistics and Transportation Review 40 (2004) 183-207

[6] K. Sourirajan, L. Ozsen, and R. Uzsoy, "A Single Product Network Design Model with Lead Time and Safety Stock Considerations," IIE Transactions on Scheduling and Logistics 39, 411-424 (2007).

[7] H.E. Romeijn, J. Shu, C.-P. Teo, Designing two-echelon supply networks, European Journal of Operational Research 178 (2007) 449-462.

[8] A. Federgruen, P.Zipkin, An allocation and distribution model for perishable products, Opms Res 34 (1986) 75-82.

[9] T.W. Chien , A. Balakrishnan, R.T. Wong, An integrated inventory allocation and vehicle routing problem, Transport Science 23 (1989) 67-76..

[10] Anily, S., Federgruen, A., 1990. One warehouse multiple retailers systems with vehicle routing costs. Management Science 36 (1), 92-113.

[11] S. Viswanathan, K. Mathur, " Integrating routing and inventory decisions in one warehouse multiretailer, multiproduct distribution systems. Management Science 43(1997):294-312

[12] C-H. Lee, Y. A. Bozer, C.C. White III, "A heuristic approach and properties of optimal solutions to the dynamic inventory routing problem," Working Paper,2003

[13] Campbell, A., Savelsbergh, M., 2004. A decomposition approach for the inventoryrouting problem. Transportation Science 38 (4), 488-502.

[14] Zhao, Q.-H., Wang, S.-Y., Lai, K.-K., 2007. A partition approach to inventory-routing problem. European Journal of Operational Research 177 (2), 786-802

[15] Li, J., Chu, F., Chen, H., " A solution approach to the inventory routing problem in a three level distribution system", European Journal of Operational Research,vol.210,pp.736-744,2011

[16] Moin, N.H., Salhi, S., 2007. Inventory routing problems: A logistical overview. Journal of the Operational Research Society 58 (9), 1185-1194..

[17] Perl, J., Daskin, M.," A warehouse location problem", Transportation Research Quarteryly, vol.19B(5),pp.381-389,1985..

[18] Or I, Pierskalla WP. "A transportation location-allocation model for regional blood banking", AIIE Transactions 1979 p.11:86 95.

[19] Jacobsen S.K., Madsen O.B. "A comparative study of heuristics for a two-level routing-location problem." European Journal of Operational Research 1980;5:378 87.

[20] Madsen, O.B.G., " Methods for solving combined two-level location-routing problems of realistic dimensions", European Journal of Operations Research, vol.12, pp. 295-301, 1983..

[21] Balakrishnan. A., Ward, J.E., Wong, R.T., 1987. Integrated facility location and vehicle routing models: Recent work and future prospects. American Journal of Mathematical and Management Sciences 7 (l,2), 35-61.

[22] Hansen, P.H, Hegedahl, B., Hjortkjaer, S., Obel, B., " A heuristic solution to the warehouse location-routing problem", European Journal of Operations Research, vol. 76, pp. 111-127, 1994.

[23] Srivnstava. R.. Benton, W.C., 1990. The location-routing problem: Consideration in physical distribution system design, Computers and Operations Research 6 , 427-435.

[24] Srisvastava. R., 1993. Alternate solution procedures for the location- routing problem. Omega 21 (4), 497-506.

[25] Chien, T.W., "Heuristic Procedures for practical-sized uncapacitated location-capacitated routing problems", Decision Sciences, vol.24, pp. 995-1021, 1993.

[26] Nagy G, Salhi S. Nested heuristic methods for the location-routing problem. Journal of Operational Research Society 1996;47:1166 74.

[27] Nagy G, Salhi S. A nested location-routing heuristic using route length estimation. Studies in Locational Analysis 1996;10:109 vol.27.

[28] Salhi S, Fraser M. An integrated heuristic approach for the combined location vehicle feet mix problem. Studies in Locational Analysis 1996;8:3 vol.21.

[29] Tuzun, D., Burke, L.I., " A two-phase tabu search approach to the location routing problem", European Journal of Operational Research, vol. 166, pp. 87-99, 1999.

[30] Wu, T.H., Low,C., Bai, J.W., "Heuristic solutions to multi-depot location-routing problems", Computers and Operations Research, vol. 29, pp. 1393-1415, 2002.

[31] Ambrossino, D., Scutella, M.G., " Distribution Network Design : New problems and related models", European Journal of Operations Research, vol. 165, pp. 610-624, 2005.

[32] Laporte, G., Nobert, Y.. 1981. An exact algorithm for minimizing routing and operating costs in depot location. European Journal of Operational Research 6, 224-226.

[33] Laporte, G.,Nobert, Y., Pelletier, P., 1983. Hamiltonian location problems.Europen Journal of Operations Research 12 , 80-87.

[34] Laporte, G., Nobert, Y., Arpin, D., "An exact algorithm for solving a capacitated location-routing problems", Transportation Science,vol.6,pp 293-310, 1986.

[35] Laporte, G., Nobert, Y., Taillefer, S., "Solving a family of multi-depot vehicle routing and location-routing problems", Transportation Science,vol.22,pp 161-172, 1988.

[36] Laporte, G., Dejax, P.J., "Dynamic location-routing problems", Journal of Operational Research Society, vol.40, pp.471-482, 1989.

[37] Akca Z, Berger RT, Ralphs TK. "A branch-and-price algorithm for combined location and routing problems under capacity restrictions." In: Chinneck JW, Kristjansson B, Saltzman MJ, editors. Operations research and cyber-infra- structure, vol. 47. US: Springer; 2009. p. 309-30

[38] Belenguer, J.-M., E. Benavent, C. Prins, C. Prodhon, R. Wolfler Calvo.2011. A branch-and-cut method for the capacitated location-routing problem. Comput. Oper. Res. 38(6) 931-941.

[39] Eppen, G., "Effects of centralization on expected costs in a multi-location newsboy problem", Management Sci. vol.25, p.498-501, 1979.

[40] Glover, F., "Future paths for integer programming and links to artificial intelligence", Comput. Oper. Res, vol. 13, p.533549, 1986.

[41] Osman, I.H., Laporte, G., "Metaheuristics: A bibliography. Ann. Oper. Res., vol.63,p.513623, 1996.

[42] Stützle, T., "Local Search Algorithms for Combinatorial ProblemsAnalysis, Algorithms and New Applications", DISKIDissertationen zur K unstliken Intelligenz. infix, Sankt Augustin, Germany, 1999.

[43] Blum, C., Roli, A., "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison", ACM Computing Surveys, vol.35, no. 3, pp. 268308, 2003.

[44] Metropolis, N., A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, "Equation of State Calculation by Fast Computing Machines", Journal of Physical Chemistry vol.21, p.1087-1092, 1953.

[45] Kirkpatrick, S., Gelart, C.D. and Vecchi, P.M., "Optimization by Simulated Annealing", Science vol.220, p.671-680, 1983.

[46] Cerny, V., "A thermodynamical approach to the travelling salesman problem. An efficient simulated annealing algorithm", Journal of Optimization Theory and Applications, vol.45, pp.41, 1985.

[47] Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor, 1975.

[48] Osman, I.,H. and Christofides, N., "Capacitated Clustering Problems by Hybrid Simulated Annealing and Tabu Search", International Transactions in Operational Research, vol. 1, no. 3, pp.317-336,1994.

[49] Liu, C.M., Kao, R.L., and Wang, A.H., "Solving location-allocation problems with rectilinear distances by stimulated annealing", Journal of the Operational Research Society vol. 45, pp. 1304-1315, 1994.

[50] Osman, I.H., "Metastrategy simulated annealing and tabu search algorithms for combinatorial optimization problems", Ph.D. Thesis, The Management School, Imperial College, London, 1991.

[51] Osman, I.H., "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problems", Annals of Operations Research vol.41, p.421, 1993.

[52] A. van Breedam,"Improvement heuristics for the vehicle routing problem based on simulated annealing", European Journal of Operational Research vol.86, pp.480, 1995.

[53] Janssens, G.K. and Breedam, A.V., "A simulated annealing postprocessor for the vehicle routing problem",in:*Applications of Modern Heuristic Methods*, Ed. V. Rayward-Smith (Alfred Waller,Henley-on-Thames, 1995).

[54] Yu, V.F., Lin, S.W., Lee, W., Ting, C.J., "A simulated annealing heuristic for the capacitated location routing problem", Computers and Industrial Engineering, vol.58, pp. 288299, 2010.

[55] Clarke, G., Wright, J.W., "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", Operations Research, Vol. 12, pp. 568-581, 1964.

Appendix A

Table A.1: Weekly Demands for Node 10 Problem

Hospitals	High Demand	Medium Demand	Low Demand
1	800	400	200
2	56	28	14
3	122	61	31
4	84	42	21
5	182	91	46
6	80	40	20
7	200	100	50
8	36	18	9
9	320	160	80
10	352	176	88

Table A.2: Standard Deviation of Weekly Demands for Node 10 Problem

Hospitals	High Demand Std.Dev			Medium Demand Std.Dev			Low Demand Std.Dev		
	H	M	L	H	M	L	H	M	L
1	200	120	40	400	240	80	100	60	20
2	14	8	3	28	17	6	7	4	1
3	31	18	6	61	37	12	16	9	3
4	21	13	4	42	25	8	11	6	2
5	46	27	9	91	55	18	23	14	5
6	20	12	4	40	24	8	10	6	2
7	50	30	10	100	60	20	25	15	5
8	9	5	2	18	11	4	5	3	1
9	80	48	16	160	96	32	40	24	8
10	88	53	18	176	106	35	44	26	9

Table A.3: Main Blood Bank and Hospital Locations & Medium Level Demands

Hospital	Latitude	Longtitude	Demand (M)	Hospital	Latitude	Longtitude	Demand (M)
0	41,015144	29,024942	0	51	40,918969	29,308176	111
1	41,007302	29,043303	400	52	41,017909	29,169838	457
2	41,019947	29,034559	28	53	41,025006	29,058409	212
3	40,976399	29,123383	61	54	40,84836	29,294363	469
4	41,023893	29,104468	42	55	40,866964	29,267766	576
5	41,023686	29,039419	91	56	41,011709	29,135789	139
6	40,963729	29,097784	40	57	40,98682	29,099242	883
7	40,996031	29,045362	100	58	41,117553	29,098341	255
8	41,0249	29,084555	18	59	41,004152	29,025139	73
9	41,020926	29,044247	160	60	41,0054	29,02123	222
10	40,98074	29,077699	176	61	41,015879	29,037911	105
11	40,907302	29,159968	58	62	40,97004	29,103277	419
12	41,004678	29,03487	100	63	40,915848	29,172285	46
13	40,988605	29,026619	42	64	41,00912	29,036848	109
14	40,995091	29,04283	65	65	41,018523	29,04666	412
15	40,890645	29,177842	60	66	40,950047	29,138815	84
16	40,920777	29,137588	30	67	40,877725	29,228822	147
17	40,879646	29,261506	395	68	41,01223	29,041647	108
18	40,987997	29,061778	109	69	40,925358	29,134462	239
19	40,997618	29,032896	111	70	40,965345	29,265737	100
20	40,900993	29,16736	94	71	41,17474	29,616924	194
21	40,878754	29,236658	177	72	40,976044	29,082843	310
22	41,024099	29,084126	109	73	40,975203	29,085389	181
23	41,025928	29,021888	84	74	40,985768	29,066211	627
24	40,963146	29,084437	396	75	40,949191	29,140358	164
25	40,883101	29,236057	28	76	40,916347	29,170444	213
26	41,006816	29,07285	35	77	41,118048	29,098794	187
27	41,021444	29,119906	481	78	40,946495	29,143907	575
28	40,946438	29,125158	353	79	40,918669	29,220479	1092
29	40,945644	29,125528	125	80	40,848245	29,305687	857
30	41,052415	29,076219	161	81	41,033115	29,102955	404
31	40,982619	29,064943	27	82	40,969757	29,10333	323
32	40,976885	29,093299	776	83	40,880587	29,237183	111
33	40,969343	29,257801	825	84	40,825038	29,325482	149
34	40,966711	29,269617	525	85	40,816174	29,271615	304
35	40,909896	29,203001	881	86	41,002427	29,019313	282
36	40,970559	29,260164	14	87	40,979015	29,133092	166
37	40,883775	29,244404	33	88	41,142742	29,369588	10
38	40,91031	29,138875	241	89	41,00326638	29,0641525	358
39	41,027134	29,1151	140	90	40,97661088	29,08107875	38
40	41,028559	29,114006	79	91	40,93506613	29,14496925	466
41	41,025378	29,02104	186	92	40,977077	29,10874375	234
42	40,941381	29,129112	9	93	40,971759	29,1688215	549
43	40,98181	29,04179	102	94	40,97111325	29,13306138	147
44	41,009309	29,036817	661	95	40,98482575	29,11617013	80
45	40,966143	29,10332	81	96	40,961579	29,17846613	261
46	41,008807	29,212046	84	97	41,0031015	29,07428413	160
47	40,884022	29,243937	564	98	40,991636	29,18873938	471
48	40,990297	29,077742	254	99	41,0052295	29,17560875	182
49	41,021323	29,119563	264	100	40,90476	29,21382975	461
50	41,016895	29,094146	174				

Table A.4: Percentage of the cost components for Current System

Instance	1	2	3	4	5	6	7	8	9	10	11	12	13
Truck Own Cost	6,88%	6,57%	7,05%	2,84%	13,11%	6,88%	6,88%	4,66%	13,11%	6,28%	8,00%	5,84%	7,50%
Variable Cost	17,03%	16,24%	17,45%	7,02%	32,44%	17,03%	17,03%	11,52%	32,42%	17,27%	16,54%	14,45%	18,56%
Inventory Cost	71,26%	67,98%	73,03%	88,15%	45,25%	71,26%	71,26%	80,55%	45,28%	72,27%	69,17%	75,62%	68,68%
WR Cost	4,83%	9,21%	2,47%	1,99%	9,19%	4,83%	4,83%	3,27%	9,19%	4,18%	6,29%	4,09%	5,26%

Table A.5: Percentage of the cost components for Proposed System

Instance	1	2	3	4	5	6	7	8	9	10	11	12	13
Facility Cost	8,45%	6,82%	7,42%	2,56%	11,34%	14,88%	4,30%	4,91%	11,59%	4,94%	11,90%	7,24%	8,84%
Truck Own Cost	2,29%	1,85%	2,68%	1,39%	3,07%	2,02%	2,33%	1,77%	3,14%	1,34%	3,22%	1,96%	2,39%
Variable Cost	28,30%	22,85%	33,12%	17,17%	37,98%	24,93%	28,81%	21,92%	38,83%	33,09%	19,98%	24,25%	29,61%
Inventory Cost	39,46%	33,74%	46,00%	61,49%	18,67%	34,06%	36,97%	52,04%	18,19%	48,66%	33,77%	48,11%	36,64%
WR Cost	2,00%	3,24%	1,39%	1,58%	2,84%	2,15%	2,13%	0,99%	1,74%	0,80%	2,90%	1,72%	2,10%
DR Cost	19,51%	31,50%	9,40%	15,81%	26,11%	21,97%	25,46%	18,37%	26,50%	11,17%	28,22%	16,72%	20,41%

VITA

DOĞUŞ ÖZKÖK was born in Istanbul, Turkey on January 8, 1988. He received his B.Sc. degree in Industrial Engineering from Isik University, Istanbul, in 2010. In September 2010, he joined the Industrial Engineering Department of Koc University, as a teaching and research assistant. His research focus on supply chain network design, supply chain management and heuristic algorithms