ROUTE OPTIMIZATION OF ATM DELIVERY VEHICLES

by

Hatice Kübra Fenerci

B.S., Industrial Engineering, Doğuş University, 2010

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering

Boğaziçi University

2013

ACKNOWLEDGEMENTS

This graduation thesis would not have been possible without the support of some people. I wish to express my gratitude to my advisor, Prof. İlhan Or who was abundantly helpful and offered invaluable assistance, support and guidance.

I am also grateful to Assoc. Prof. Ekrem Duman without whose knowledge and assistance, this study would not have been successful.

Special thanks also to Tübitak, providing monetary support during my graduation study.

I would like to express my very great appreciation to Boğaziçi University Industrial Engineering Department staff, for their contribution to my academical vision and technical abilities.

I lovingly dedicate this thesis to my family including Şenay, Hüseyin, Büşra Fenerci and Gökhan Canel, who supported me at each step of my life.

ABSTRACT

ROUTE OPTIMIZATION OF ATM DELIVERY VEHICLES

The main objective of this study is optimizing routes of ATM delivery vehicles under some constraints. Since the problem is NP-hard, it is not possible to solve it in polynomial time and near optimum solutions through heuristic algorithms are found. The methodology developed involves construction of new heuristics based on the Savings and the Sweep Algorithms. Two new heuristics both manipulating and combining some general properties of these two algorithms and adding some new factors to the classical Savings algorithm are suggested. Then, initial tours such obtained are improved using 2-opt based algorithms. Two different versions of 2-opt algorithm are used and compared to each other. Then, a Switch based heuristic is developed and total cost of visiting an ATM one day before the suggested schedule is studied. In case it decreases total cost some ATMs are considered to be visited one day earlier. Instances are tested and results are reported for both the sample data retrieved from "www.branchandcut.org" and provided by a commercial bank (data provided by the bank is manipulated due to confidentiality issues). Trend analysis is tried to find the most suitable resulting algorithm for a spesific problem set. Algorithms are coded in SQL and executed on Intel(R) Core(TM)2 Duo CPU 2.40GHz computer.

ÖZET

ATM SEVKİYAT ARAÇLARININ ROTA OPTİMİZASYONU

Bu çalışmanın asıl amacı bazı kısıtlar altında ATM'lere para taşıyan araçların rotalarını optimize etmektir. Problem NP-hard olduğundan polinom zamanda optimize etmek güçtür ve sezgisel algoritmalar yardımıyla optimuma yakın sonuçlar bulunabilmektedir. Geliştirilen methodoloji, Savings ve Sweep sezgisel algoritmaları bazlı yeni sezgisel algoritmalar bulmayı kapsamaktadır. Bu algoritmaların bazı genel özelliklerini değiştirerek, birleştirerek ve klasik Savings algoritmasına bazı faktörler ekleyerek iki yeni sezgisel önerilmektedir. Daha sonra bu ilk algoritma 2-opt tabanlı algoritmalarla geliştirilmektedir. 2-opt algoritmasının iki yeni türevi kullanılmakta ve birbirleriyle kıyaslanmaktadır. Daha sonra, Switch bazlı bir algoritma geliştirilerek ATM'lerin önerilen çizelgeden bir gün önce ziyaret edilme durumundaki toplam maliyet çalışılmıştır. Toplam maliyeti düşürmesi durumunda bazı ATM'lerin bir gün önce ziyaret etmesi değerlendirilmiştir. Modeller hem "www.branchandcut.org" web sitesinden edinilen data hem de bir ticari banka tarafından sağlanan data ile test edilmekte ve sonuçlar raporlanmaktadır. Ticari banka tarafından sağlanan veriler gizlilik prensibinden ötürü değiştirilmiştir. Özel bir problem kümesi için en uygun algoritmayı bulma amacıyla trend analizi uygulanmaktadır. Algoritmalar SQL'de kodlanmakta ve Intel(R) Core(TM)2 Duo CPU 2.40GHz bilgissayarda çalıştırılmaktadır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

c_{ij}	transportation cost between two given points *i* and $_j$
d_a	total transportation cost
d_b	transportation cost
M	supply points
N	demand points
Nifinal	number of points in day i after all switches
Niinitial	number of points in day i at the beginning
Niswitch	number of switches from that point
Xij	the smallest shipping cost
CVRP or CVRPTW	capacitated vehicle routing problem (with or without time windows)
DSNo	dataset no
DPNo	number of demand points
IRP	inventory routing problem
NP	non-polynomial
TSP	travelling salesman problem
VRP	vehicle routing problem
VRPPD	vehicle routing problem with pickup and delivery
VRPTW	vehicle routing problem with time window

1. INTRODUCTION

Automatic Teller Machine (ATM) is a service that many people use every day without ever thinking about the complex logistics behind it. People use these devices to pay their rents, school fees, debt and general cash needs. ATM operations and services are very much reliable. Monetary transactions are easily tracked and recorded, while the bank is obliged to keep a record of all transactions forever. Also, ATM usage is strongly supported by the state since it is a great facilitator of white-market economy. As a result, customers cannot isolate themselves from ATM usage even when they do not use credit or debit cards.

From a customer point of view the requirements from an ATM machine are quite simple: to provide the required cash at all times. But for most banks, the stocking and maintenance of cash machines can represent a major cost. On the other hand, optimized ATM management allows banks to reap major savings and increase customer satisfaction, while reducing risk.

Actually, the challenges on ATM optimization models can be listed as follows:

- ATMs are not available all the time. They can be broken down or be out of cash, while cash supply trucks are far away from the spesific ATM.
- An unavailable ATM can also affect other product profitabilities of a bank, since it deteriorates customer goodwill and trust.
- Where to locate ATM’s is a critical issue. Utilization rates should be sufficiently high to justify the high operating costs.
- ATM machines incur significant operating costs in terms of maintenance, security, supply costs and most importantly, interest/opportunity cost of the cash stored in it.
- Although all daily demand estimations are closely monitored, cash reserve can be unexpectedly depleted due to a singular event such as a crisis, discount in a market or a marriage ceremony near the ATM.

In the past, the resupplying of ATMs was frustrating for both the bank and the customers. The machines were stocked according to what was felt to be necessary, or based on the experience of the bank manager, who ideally had to remember and consider all the possible fluctuations – the day of the week, season, holidays, major events, and so on. If the person responsible for stocking had incorrectly estimated the need for cash, the result was frustrated customers in front of empty cash points or excessive interest costs, since capital was tied up in the cash machines and the bank could not use it.

In this study, it is aimed to functionally address the vehicle routing part of the ATM cash flow optimization problem. The problem, as a whole involves the forecasting of daily cash demands, optimization of cash supplying frequency, and finding routes of ATM cash delivery trucks in order to minimize total cost over time.

For an ATM problem, forecasts are affected by many factors such as climate, date, location of the ATM and other factors such as economic crisis, terrorist attacts and so on. The solution of the forecasting problem involves finding optimum cash amount that must be stocked in a specific ATM so that ATM would not experience any stock-out. Optimization of supplying frequency problem aims at finding the optimal sequence of resupplying which satisfies the forecasting projections. As the last component of the ATM cash flow optimization problem, vehicle routing problem investigates routes of the delivery vehicles in order to identify a minimum cost solution.

In this study, all suggested algorithms are based on or are extensions of the ideas inherited from the “vehicle routing problem”, which is a direct extension of the “traveling salesman problem”. Additional factors consist of multiple origin points and multiple salesman, mainly delivery vehicles for the ATM problem. Some characteristics of the ATM cash delivery problem also resembles the “transshipment problem” in which transportation is allowed between supply and/or demand points. For instance, if a cash depot has a stock-out problem, necessary amount can be transshipped from other depots. On the other hand, this possibility is out of scope for this dissertation.

For initiating this study, cash stock levels and optimum service frequency information are obtained from other studies, and the only target is finding (near) optimum vehicle routes that satisfies minimization of the sum of delivery and interest costs. (Data provided by the bank is manipulated due to confidentiality issues.) Problem solution involves 2 phases. In the first step, “Savings” and “Sweep” based “tour building” algorithms are developed. Since obtaining a good initial solution does not guarantee the best final solution, in the next phase, “2-opt” based “tour improvement” algorithms are developed. As the study uses daily demands and ATM visiting frequency as inputs, transportation cost is not included until the VRP phase starts. In order to fine-tune daily visits, a Switch algorithm is developed and what-if analysis is conducted resulting in visiting some of the demand points in earlier days. Even if such “early visits” may increase stock keeping cost, decrease in transportation cost can be higher than this increased amount, so that algorithm may prefer to visit some ATMs earlier leading to a lower total cost.

In Chapter 2, the traveling salesman problem is presented including problem description, formulation and solution approaches suggested by previous studies. The base of our study, the “Vehicle Routing Problem” is presented in Chapter 3, pointing on its importance, literature background and key points. Then, in the Chapter 4, an extension of the “Vehicle Routing Problem”, and featuring a good representation of the ATM cash supply problem, named the “Inventory Routing Problem” is reviewed. In Chapter 5, new heuristics for the vehicle routing problem are introduced and discussed. In this context, first, two “tour building” algorithms based on the “Savings” and the “Sweep” algorithmts are proposed. Then, two “tour improvement” algorithms based on the “2-opt” principle are proposed. Test beds for these new algorithms and results obtained are also reported and discussed in this chapter.

Experiments are conducted using 30 test instances offered by a popular academic web site, named “www.branchandcut.org” and 5 days (manipulated) ATM cash delivery vehicles’ data. Trend analysis is conducted to find the most suitable data for spesific type of test instances. Regarding the ATM data of a commercial bank, day to day switches are examined and realized in case they decrease expenses. Algorithms are coded in SQL and tests are conducted on 2.40 GHz computer.

In Chapter 6, conclusions are drawn and advises for future studies are given.

2. THE TRAVELING SALESMAN PROBLEM

The travelling salesman problem (TSP) is an NP-hard problem extensively studied in operations research literature. Simply stated, given a list of nodes and their pairwise distances, the aim is to find the shortest possible way of visiting all nodes. Each node is visited only once and salesman returns to the origin node at the end.

The TSP has an integer programming formulation which aims at solving a completed direct graph, simply a tour building problem with minimal length. The TSP can be formulated as an integer linear programming problem. Integer programming formulation of the TSP for an "n" node (city) environment can be stated as follows:

$$\min x \quad \sum_{i=1}^{n} \sum_{j=1}^{n} x(i,j) d(i,j) \tag{2.1}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x(i,j) = 1, \; i = 1, 2, \ldots, n \tag{2.2}$$

$$\sum_{i=1}^{n} x(i,j) = 1, \; j = 1, 2, \ldots, n \tag{2.3}$$

$$\sum_{i,j \in S}^{n} x(i,j) \leq |S| - 1, \forall S \subset \{1,2,\ldots,n\} \tag{2.4}$$

$$x\,(i,j) \in \{0,1\} \tag{2.5}$$

where $x(i,j)$ are the set of binary decision variables i=1,2,….,n and j=1,2,….,n. ($x(i,j)$=1, if and only if the salesman goes from city i to city j, 0 otherwise.), $d(i,j)$ are the set of distance parameters between any node pair i and j, S = subset of cities, $|S|$ = cardinality of S (# of elements in S).

In the above formulation, each city must be "entered" and "exited" exactly once as enforced through constraint sets (2.2) and (2.3). However, just making sure that each node is visited once does not rule out "subtours", in other words, multiple (up to n/2) salesman visiting non-overlapping sets of nodes. Accordingly, constraint set (2.4) is added to eliminate subtours. This constraint set makes sure that the number of active arcs connecting nodes in any subset S of nodes is less than the number of nodes in that subset, thus making subtours impossible. Unfortunately, the number of such "subtour elimination constraints" is very large (one for every possible subset S); in addition this set of constraints destroys the "unimodularity" property of the first two set of constraints, thereby necessitating integrality constraints (constraint set 2.5).

As can be deducted from the formulation (2.1 - 2.5), every solution to the TSP is a solution to the assignment problem (AP). In the AP, there are a number of agents and a number of tasks. Any agent can be assigned to perform any task, incurring some cost that may vary depending on the assignment. If the numbers of agents and tasks are equal and the total cost of the assignment for all tasks is equal to the sum of the costs for each agent (or the sum of the costs for each task, which is the same thing in this case), then the problem is called the linear assignment problem. Commonly, when speaking of the assignment problem without any additional qualification, then the linear assignment problem is meant. Aim of the AP is assigning exactly one agent to each task and one task to each agent in such a way that the total cost of the assignment is minimized. The assignment problem is a special case of the transportation problem, which is a special case of the minimum cost flow problem, which in turn is a special case of a linear program. On the other hand, every solution of the AP is not a solution to the TSP. The AP is a relaxation of the TSP, and it gives a lower bound for the TSP.

This problem was first formulated by mathematicians in 1930. In the theory of computational complexity, the TSP is classified as an NP-complete problem. Thus, it is likely that the worst-case running time for any optimizing algorithm for the TSP increases exponentially with the number of nodes.

Although the problem is computationally burdensome, a large number of heuristics and exact methods are known, so that some instances with tens of thousands of nodes can be solved. These approaches include branch-and-bound algorithms, progressive improvement algorithms and heuristic and approximation algorithms. There are also many software packages available in the market to solve the TSP, such as Concorde, DynOpt, LKH, OpenOpt, tsp and TSPGA.

The TSP has several applications in many areas especially in planning and logistics. In logistics, numerous possibilities and practical examples of application of the TSP can be found such as distribution of food products from producers to shops, distribution of fuel to petrol stations, distribution of various products from producers or distributors to customers, visits of doctors at patients´ homes, and so on.

3. THE VEHICLE ROUTING PROBLEM

The vehicle routing problem (VRP) is a direct extension of the TSP where, simply stated, there are multiple origin nodes and multiple salesman in the VRP. A number of customers are served with a fleet of vehicles. First proposed by Dantzig and Ramser in 1959, the VRP is an important problem both for academicians and for professionals. While delivering ordered goods located at one or more central depots to customers, objective function is minimizing the cost of distributing all goods. Many methods have been developed for searching good solutions to the problem, but except all the smallest problems, finding a global minimum cost solution is computationally burdensome.

Several variations and specializations of the vehicle routing problem exist:

The Vehicle Routing Problem with Pickup and Delivery (VRPPD): A number of goods need to be moved from certain pickup locations to other delivery locations such that some nodes serve as both pickup and delivery nodes. The goal is to find optimal routes for a fleet of vehicles to visit the pickup and drop-off locations .

The Vehicle Routing Problem with LIFO: Similar to the VRPPD, except an additional restriction is placed on the loading of the vehicles: at any delivery location, the item being delivered must be the item most recently picked up. This scheme reduces the loading and unloading times at delivery locations because there is no need to temporarily unload items other than the ones that should be dropped off.

The Vehicle Routing Problem with Time Windows (VRPTW): The delivery locations have time windows within which the deliveries (or visits) must be made.

The Capacitated Vehicle Routing Problem (with or without Time Windows): CVRP or CVRPTW. The vehicles have limited carrying capacity of the goods that must be delivered.

Finding an optimal solution to a large NP-hard problem necessitates a heavy computational burden. Accordingly, approximating algorithms (or heuristics) are preferred for TSP or VRP type problems. Heuristic algorithms do not, in general, guarantee the determination of true optimal solutions, but they do generate “sufficiently good near optimal solutions”. Solving the VRP optimally takes too long, instead one normally uses approximation algorithms, or heuristics.

Heuristics consists of approaches and approximation algorithms that aim at finding possible good solutions quickly. Heuristics include two groups which are construction methods and improvement methods.

3.1. Construction Approaches to Solve the VRP

Construction methods build a feasible solution by selecting arcs with low (hopefully minimizing) cost. Best known approach is the Savings’ algorithm first suggested by Clarke and Wright (1964).

The classical Savings’ concept focuses on the cost savings obtained by joining two routes into one route. Large saving values indicate that it is attractive, with regard to costs, to visit all points in the two “sub-routes” in one single “aggregate route”. There are two versions of the Savings’ algorithm, a sequential and a parallel version. In the sequential version, exactly one aggregate route is built at a time, while in the parallel version more than one aggregate routes can be built at the same step.

In the first step of the Savings’ algorithm, the savings for all pairs of customers are calculated, and all pairs of customer points are sorted in descending order of the savings. Then, from the top of the sorted list of point pairs, one pair of points is considered at a time. When a pair of points “i-j” is considered, the two routes that visit i and j are combined (such that j is visited immediately after i on the resulting route), if the total demand on the resulting route does not exceed the vehicle capacity. Then, the procedure passes through the saving list sorted by ascending order. If a node pair j-k is encountered, k is added to the current route and the new aggregated route becomes “depot→ i→j→k→depot”. On the other hand, if a node pair “i-m” is encountered, m is added to the current route and the new aggregated route becomes “depot→ m→i→j→depot”. This aggregation continues as long as the vehicle capacity constraint is satisfied. In the sequential version, one must start from the top of the list every time a connection is established between a pair of points (since combinations that were not viable so far now may have become viable), while the parallel version only requires one pass through the list, since more than one node pair can be selected to build more than one route at a time.

There are many other studies related to the classical Savings’ method. Yellow, PC. (1970) embeds a route shape parameter into the basic Savings' formula to handle asymmetric distances.

Gaskell, T. (1975) studies the allocation of customers to routes. Five different algorithms, some of which are derived from Savings’ method are discussed and applied to six different datasets. The result is that no algorithm is better than other. In other words, the Clarke and Wright procedure is further strenghened.

Paessens, H. (1988) modifies the Savings' method for less CPU and memory usage. Performance is improved using a parametrical saving function. The suggested parametrical version helps decreasing both “computer storage of distances” and “determination time of the maximum saving value”.

Lysgaard, J. (1997) proposes a Sequential version of Clarke and Wright's Savings Heuristic and supports it with numerical examples.

Poot, A. and Kant, G. (2002) propose a Savings Based algorithm for the Extended Vehicle Routing Problem. The extension includes addition of non-standard model quality measures such as “visual attractiveness of the routing plan”. This algorithm is compared with a sequential insertion algorithm on real-life data. The results show that, the suggested Savings’ based algorithm performs better with respect to not only these non-standard quality measures, but also the traditional measures.

Altınel, IK. and Oncan, T. (2004) suggest a new algorithm for The Capacitated Vehicle Routing Problem, which is related to the Clarke–Wright’s Savings’ method. In this research, a demand parameter is embedded into the Savings' formula. Both distance and demand amounts are taken into consideration. The performance of this parametric heuristic is sensitive to fine-tuning. In addition, their approach requires much more computation time to solve an instance.

Battarra *et al.* (2008) also propose a genetic algorithm to set the parameter values. They reduce the time needed to solve an instance, but the quality of their solution is worse. Corominas, A. *et al.* (2010) propose to use an "empirically adjusted greedy heuristics", named "EAGH" to set the parameter values. Their experiments show the efficiency of EAGH; the algorithm obtains impressive results in a reasonably short time.

Regarding heuristic optimization procedures, a well-tuned set of parameters may need intensive computational requirements. In tuning the parameters of a Clarke-Wright type Heuristic via a Genetic Algorithm, M. Battarra, *et al.* (2007) use a 2-step parameter adjustment algorithm. At the first phase, they use genetic algorithm to determine a small parameter set. At the second phase, they use local search to improve this parameter set. This method is tested on Clarke and Wright's Savings' algorithm and the results are compared with an enumerative parameter-setting approach. The results show that, although the new parameter-setting procedure requires much shorter computational time, it produces results of the same quality as the enumerative approach.

Another widely used tour building heuristic is the "Sweep Method" suggested by Gillet, B. and L. Miller (1974). It may be applied to planar instances of the VRP. Feasible node clusters are created by rotating a ray centred at the depot and gradually increasing its "polar angle" in order to include customers in a vehicle route, until the capacity or route length constraint is attained. A new route is then initiated (with the rotating ray starting at the polar angle of the last node considered) and the process is repeated until the entire plane has been swept.

The clustering algorithm uses the "cluster first-route later" approach. One of the most common tour construction clustering algorithms is the "k-means variant" clustering algorithm, suggested by Kim, B., *et al.* (2004). In the k-means clustering algorithm, an initial centroid seed node for each route (cluster) is selected randomly as the first step and the remaining nodes are assigned to their nearest route (cluster). Then, a new centroid is calculated for each cluster. The nodes are clustered by considering the distances between the nodes and the centroids, in order words; a node is assigned to the cluster whose centroid is the closest to the node. When a node is assigned to a route, the capacity of the route is considered. The capacity of the route is defined by the maximum number of nodes that the vehicle is allowed to visit, the volume or weight that the vehicle is allowed to handle, and/or the allowable routing time. When a route has already reached its capacity, the node is assigned to the next closest route/cluster whose capacity is not full.

There are also other tour building heuristics including the “Nearest Neighbor” and the “insertion” approaches. In the “Nearest Neighbor” approach, the rule is to go next to the nearest unvisited customer, until all nodes are visited once. The “insertion” method, which has many variants in the literature, basically starts from a node and inserts a new node to the route in each step until all the nodes are considered under the vehicle capacity constraint.

3.2. Improvement Methods to Solve the VRP

Improvement methods start with a feasible solution (route) and try to improve it by exchanging arcs or nodes within or between the routes.

One of the earliest algorithms suggested for improving initial routes is the “2-opt” algorithm. Among simple local search algorithms, the most famous are 2-Opt, 3-Opt and OR-Opt. The 2-Opt algorithm was first proposed by Croes [1958], although the basic move had already been suggested by Flood [1956]. This move deletes two edges, thus breaking the tour into two paths, and then reconnects those paths in the other possible way.

In the “3-Opt” algorithm, the exchange replaces up to three edges of the current tour as demonstrated in Figure 3.1.

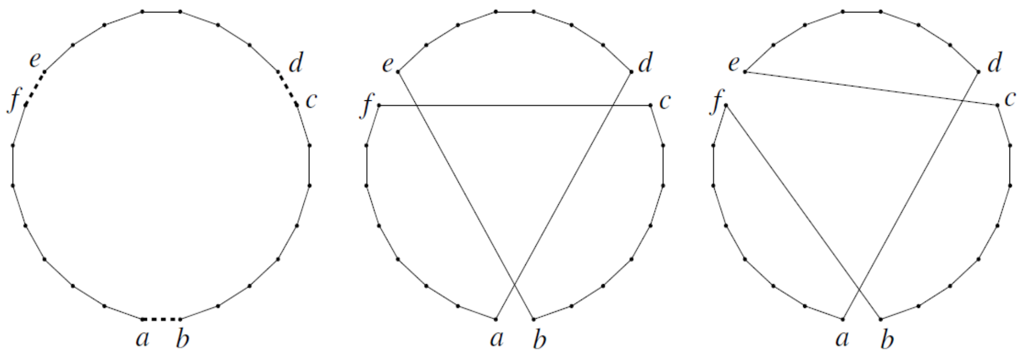

Figure 3.1. 2 Two Possible 3-Opt Moves.

One another best known route improvement method, called “OR-opt”, is developed by Or (1976). It attempts to improve the initial tour by moving a chain of three consecutive vertices to a different location, and possibly reversing it, until no further improvement can be obtained. In the following steps, the procedure is repeated for two, and then one single vertices.

The advantage of these heuristics is that they have polynomial running times, so using one of them almost always provides good solutions within a reasonable amount of time. On the other hand, they only do a limited search in the solution space, therefore run the risk of resulting in a local optimum (or near optimum).

3.3. Further Extensions to VRP Heuristics

In the literature, there are many other related studies some of which are summarized below:

Ballou, R. (1990) makes a comparison of several popular algorithms for vehicle routing and scheduling, including Cluster and Sweep methods which are redesigned to improve performance.

Eryavuz, M. and Gencer, C. (2001) try to minimize traveling distance of Balıkesir Military school busses using a version of the Savings’ algorithm called the “Randomized Savings” and VRP 328 Software. In the randomized Savings’ method, instead of selecting the node pair which has the highest cost saving, a random node pair (that has saving ≥ 0) is selected. Finally, authors try to improve Savings' results using 2-opt and OR-opt algorithms.

Cordeau *et al.* (2002) discuss several of the most important classical and modern heuristics which are summarized and compared in terms of accuracy, simplicity, flexibility and speed.

Asadi, *et al.* (2011) combine a genetic algorithm and the Hopfield Neural Network algorithm to find an efficient route and show that the results of the combined algorithm are more efficient that other similar algorithms.

Maden, *et al.* (2009) introduce another heuristic algorithm which minimizes the total travel time for vehicle routing in an environment where travel time between route pairs vary throughout the day. The variation of time is caused by congestion, which is generally greatest during morning and evening rush hours. Since the study is relatively new, there is more traffic information available that makes it possible to plan vehicle journeys taking account of the congestion that is predictable from the traffic patterns of the past. The approach does not take account of unexpected events that may cause congestion such as an accident, (but regular congestion due to volume of traffic or long-term road works are predicted from past data). The algorithm is used to schedule a fleet of delivery vehicles operating in the South West of the United Kingdom for a sample of days. The results show how conventional methods may still cause some long routes, while their approach reduces both total traveling time and Co2 emmisions.

Kim, BI., Kim, S. and Sahoo (2004) suggest a “Stable Marriage Based Balanced Clustering Algorithm”. Although insertion algorithms are commonly used in practice, routes having lots of overlapping are not desirable in real world applications. To handle this problem, the study extends an insertion algorithm also taking “visual attractiveness” of the routes into account. A shape metric is added to reduce overlapping in the route plan. The study reduces operating costs, provides better customer service, and determines appropriate prices. Sales and customer service also benefit because the solution is integrated the sales, customer service, and operations departments.

Taillard, *et al.* (2004) describe a tabu search tour improvement heuristic for the vehicle routing problem with soft time windows. Their Tabu Search method uses a mixed neighborhood structure and an advanced recovery procedure to generate high quality solutions. New starting points for the tabu search are produced with a combination of routes that are taken from different solutions.

Tarantilis, *et al.* (2004) investigate the open vehicle routing problem (OVRP), in which routes do not start and end at one central point but are open paths. This problem is very important for planning fleets of hired vehicles, a common practice in the distribution and service industry. To solve the problem, the author proposes a single-parameter metaheuristic method which uses a list of threshold values to guide an advanced local search is used. Computational results show that the proposed method outperforms previous approaches for the OVRP.

Malandraki and Daski (1992) suggest a mixed integer linear programming formulations for the TDVRP (Time Dependent Vehicle Routing Problems) and the TDTSP (The time dependent traveling salesman problem). They treat the travel time functions as step functions. The properties of the TDVRP make this problem unsuitable for most of the vehicle routing algorithms. A heuristic for the TDTSP without time windows using cutting planes is suggested. Some test results are reported.

Potvin and Rousseau (1993) describe an insertion algorithm for the Vehicle Routing and Scheduling Problem with Time Windows (VRSPTW). Numerical results on the standard set of problems of Solomon (retrieved from http://w.cba.neu.edu/~msolomon/problems.htm) and their comparisons are reported.

4. THE INVENTORY ROUTING PROBLEM

Inventory routing problem is a derivative of the vehicle routing problem which also takes inventory management into consideration. It provides integrated logistics solutions by simultaneously optimizing inventory management, vehicle routing and delivery scheduling. There is always a trade-off between inventory cost and transportation cost. If we transport goods more frequently, our transportation cost increases. So, the objective is the minimization of the sum of the inventory and transportation costs. For this tough problem type, time may be discrete or continuous, demand may be deterministic or stochastic, inventory holding costs may be accounted for in the objective function or not. In the IRP, three decisions have to be made:

- When to serve a customer?
- How much to deliver to a customer when it is served?
- Which delivery routes/vehicles to use?

The ATM cash delivery problem is basically an inventory routing problem. Due to high interest rate (i.e. inventory cost) of money, conducted study tries to optimize both inventory level and vehicle routing simultaneously. In Section 5.4.4., a switch based algorithm is suggested to minimize the sum of inventory and delivery costs.

The first studies published on the IRP were mostly variations of the Vehicle Routing Problem taking inventory cost into consideration. The paper of Bell *et al.* (1983) deal with the case where only transportation costs are included, demand is stochastic and customer inventory levels must be met. Some other early papers on the IRP are worthy of mention. Federgruen and Zipkin (1984) have modified the VRP heuristic of Fisher and Jaikumar (1981) to accommodate inventory and shortage costs in a random demand environment. Blumenfeld *et al.* (1985) have considered distribution, inventory and production set-up costs; Burns *et al.* (1985) have analyzed trade-offs between transportation and inventory costs, using an approximation of travel costs; Dror *et al.* (1985) have studied short term solutions. The latter study was extended to stochastic demand by Dror and Ball (1987). The paper of Dror and Levy (1986) adapts earlier VRP heuristics to the solution of a weekly IRP, while Anily and Federgruen (1990) have proposed the first clustering algorithm for the IRP. Most of these papers assume that the consumption rate at the customer locations is known and deterministic. Despite the large number of contributions on distribution and on inventory problems before this period, the integration of these two features proved dificult to handle, not only because of large computational requirements, but also because the available algorithms could not easily handle large and complex combinatorial problems, such as those combining routing and inventory management decisions.

5. NEW HEURISTICS FOR THE VEHICLE ROUTING PROBLEM: SOLUTION APPROACHES AND EXPERIMENTAL STUDIES

5.1. Necessity for New Heuristics

As discussed in Chapter 1, this study focuses on the “routing aspects” of the ATM cash delivery problem. In other words, the problem considered focuses on how (visiting what routes and what vehicles) to dispatch necessary supplies to the ATMs staring from and ending at given depots.

Desired cash stock levels and optimum service frequency information are treated as parameters to be obtained from other studies, and the only target is finding (near) optimum vehicle routes that satisfies minimization of the sum of delivery and interest costs. Delivery cost is affected by gasoline cost, driver cost, depreciation cost and total distance travelled. Interest cost, gasoline cost, driver cost and depreciation cost are given. As such, efficient solutions of the VRP are seeked. In addition, in order to fine-tune daily visits, a Switch algorithm is developed and “what-if” analysis is conducted through which “earlier visits to certain nodes” are considered (in order to save on transportation costs while enduring higher inventory/interest costs). Results may suggest visiting some of the demand points in earlier than originally scheduled.

On the other hand, as discussed in Chapter 3, the VRP is computationally burdensome, and a global optimum solution cannot be guaranteed in a reasonable time. Since time is usually a challenging issue for the problem of interest, various heuristics have been developed and applied to find near optimum solutions. Since none of the heuristics can be designed to find a global optimum solution, they are always open to be improved. Having a critical problem on hand, it is important to have even small contributions for near-optimality of the current solution. As a result of this justification, this thesis tries to improve and/or generate acceptable heuristics to solve the ATM cash delivery problem.

5.2. Suggested Heuristics for the VRP

5.2.1. Tour Building Heuristics

The tour building heuristics developed in this study are as follows. All the discussed algorithms are coded in t-SQL and tested through 30 test problems given in Appendix A. (This experimentation is discussed in Section 5.4.)

5.2.1.1 The "Sequental Savings' Algorithm (SSA). The basic savings concept expresses the cost savings obtained by joining two routes into one route as illustrated in figure 5.1, where point 0 represents the depot.

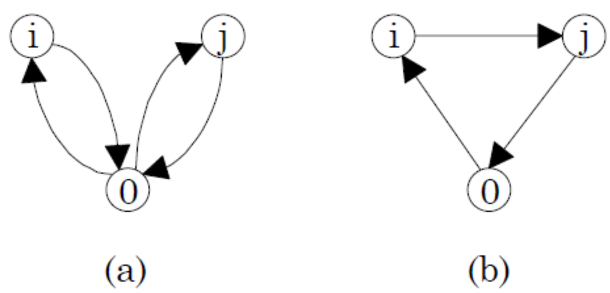

Figure 5.1. Illustration of the Savings' Concept.

The logic behind the procedure is as follows:

Denoting the transportation cost between two given points i and j by c_{ij} , the total transportation cost d_1 of visiting nodes "i" and "j" from depot node "0" or two different routes is :

$$d_1 = c_{0i} + c_{i0} + c_{0j} + c_{j0} \qquad (5.1)$$

Then, if these two routes are aggregated into one, such that nodes "i" and "j" are visited (from depot node "0" via one single route), the total transportation cost becomes d_2:

$$d_2 = c_{0i} + c_{ij} + c_{j0} \tag{5.2}$$

Accordingly, the "savings" associated with this particular "route aggregation" can be determined as:

$$s_{ij} = d_1 - d_2 = c_{i0} + c_{0j} - c_{ij} \tag{5.3}$$

Then, the point pairs are sorted in descending order of the savings. Subroutes with the largest Savings' values are joined first, in case their joint demand does not exceed the vehicle capacity. The ordered list is screened for possible expansions of this most recently generated "aggregated subtour". The second best Savings' nodes will be joined to the route if one of them matches with the nodes considered in the first step. If not, the procedure moves on to the next "savings". (If points with the second best Savings', which are the next pair in the list, are connected at this stage, more than one route would be built. Since the sequential version of the algorithm is limited to making only one route at a time, these point pairs will be disregarded and the algorithm proceeds to the next best Saving pairs.) The algorithm stops when no further aggregation is possible.

5.2.1.2 Savings with Angle Root Degree (SARD). Adding an angle parameter to the basic Savings' concept is inspired from another vehicle routing approach called the "Sweep Algorithm", suggested by Gillet and Miller in 1971. This "partition first, route later" approach partitions the customer locations into subsets in such a way that the total demand of each subset is less than the vehicle capacity. The algorithm computes the polar coordinates of each location with respect to the depot. The sites are then sorted by increasing the polar angle. Sites in the same subset have consecutive polar angles and thus are in the same "cone" projecting out of the depot. The number of subsets then become the number of routes (and thus the vehicles). The second phase of the sweep algorithm finds a shortest tour through the customer sites and the depot in each subset.

Suppose every vehicle has a capacity to serve 4 nodes, the following figure shows decision of the Sweep Algorithm:

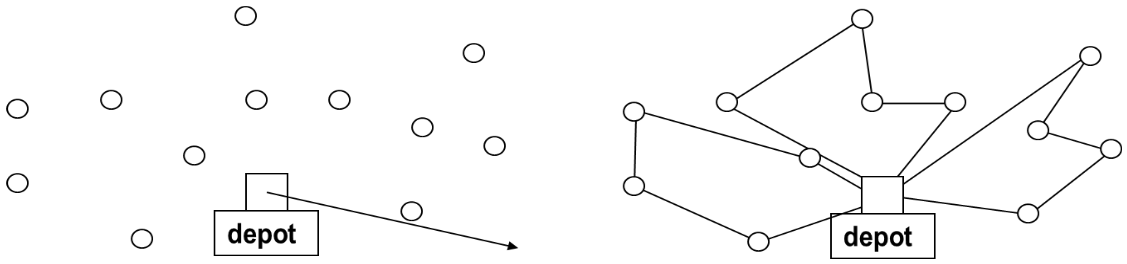

Figure 5.2. Illustration of the Sweep Concept.

In the SARD approach, the "route aggregation principle" of the Savings' approach is revised by increasing the priority of "similar angle" subroutes/nodes in the "subroutes/nodes to be aggregated" list. The results are quite promising.

Steps of the suggested "SARD" algorithm are as follows:

Step 1: Calculate all pairwise savings and sort in descending order.

Step 2: Calculate the angle (with respect to the depot) between all node pairs using the Cosine Theorem: *Angle = Degrees(ArcCos(Round(*$A^2 + B^2 - C^2$*) / (2 * A * B), 14))*

Step 3: Calculate the "angle-root degree" parameter: Divide all pairwise savings by roots (between 1-50) of the angle between nodes with respect to the depot. (This parameter will be called as angle-root degree from now on). If angle root degree is equal to 1, the algorithm becomes the classical Savings' algorithm.

Step 4: Build 50 aggregated routes with 50 different route degrees (c_i=1,2,…,50) where "revised savings" terms in each are determined by dividing original savings values by the i^{th} root of the associated polar angle between the considered nodes.

Step 5: Find the minimum cost angle-route degree for the considered dataset and drop the remaining 49 aggregated routes. For example, if the angle between 2 nodes is 50 degree (obtained from step 2) and optimal (least cost) root parameter is determined to be 15 (in step 4), all saving values will be divided by 50^1/15 to find the minimum "total cost".

Step 6: End.

5.2.1.3 Savings' with Angle Distance Factor (SADF). Recall the Savings' formula which calculates pairwise Savings':

$$s_{ij} = d_1 - d_2 = c_{i0} + c_{0j} - c_{ij} \tag{5.3}$$

The case below shows how and why this Savings' formula could overestimate the Savings' and lead to a wrong decision while joining the nodes.

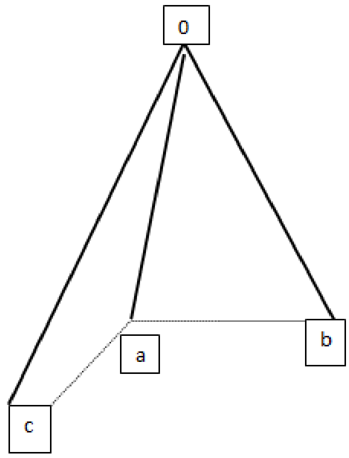

Figure 5.3. Illustration of the Savings' Selection Criterion.

Since the classical Savings' algorithm expresses the cost savings obtained by joining two routes into one route, it merges nodes "b" & "c" even though these nodes could be far away as exemplified in figure 5.3. On the other hand, overall savings obtained can be higher (when subsequent aggregation are considered) in joining nodes "a" & "b" rather than joining nodes "b" & "c". In order to avoid this behavior, both angle and distance between nodes are taken in to consideration. Suggested "SADF" algorithm has a higher tendency to merge subroutes/nodes, that have smaller angle corresponding to depot in conjunction with reasonably small (if not the smallest) original savings' value.

Steps of the suggested algorithm, named as "SADF" is as follows:

Step 1: Execute the following for each node pair.

Step 1a: Calculate the angle between node pairs (with respect to the depot) using the Cosine Theorem: *Angle = Degrees(ArcCos(Round(*A^2 *+*B^2 *-* C^2*) / (2 * A * B), 14))*

Step 1b: Calculate the factor named the "Angle Weight Percent". This is an experimentally determined integer between (1-5) for each dataset. (Suggested method tries all candidate integers between (1-5) and embeds 5 different "Angle Weight Percent" in the formula in step 1c.

Step 1c: Calculate the factor named as "Angle Factor" as:

*Angle Factor =1+((18-(((Angle/10))+1))*Angle Weight Percent)/100*

Angle between nodes could be an important factor for the VRP routes being constructed. On the other hand, if it would have a huge effect on savings calculation, the resultant savings would be far away from reality. If savings are directly multiplied by 1/angle, a node pair with 160 degree would have 16 times smaller savings than a node pair with 10 degree. On the other hand, nodes with 160 degree can be closer than the other node pair. In order to avoid this, node pairs should be multiplied by a scaling parameter. For each 10 degree, an experimentally determined step factor is used to embed the effect of the angle into the savings' formula. ((Angle/10)+1) is used to cluster node pairs into similar angle groups. For example, if the angle is 160 degree, the node pair is in the 17^{th} region (thus, the algorithm tends to give higher weight to node pairs with smaller angle). Angle weight percent is used to change the effect of the angle. If it is 5 for a node pair with 160 degree angle, the angle factor becomes 1,005. Changing angle weight percent to 1 results in an angle factor 1,001. (Angle/10) value is rounded to the closest smaller integer to avoid "0" angle factor error (and "0" savings error).

Step 1d: Determine the factor named the "Distance Weight" between (1-100). This is an experimentally determined integer between (1-100) for each dataset. (Suggested method tries all candidate integers between (1-100) and embeds 100 different "Distance Weight" in the formula in step 1e. The higher the distance weight, the more the "new saving" increases.

Step 1e: The “distance weight” is divided by the number of nodes in the selected region to find the step factor which is used to calculate the “distance factor”. This step factor changes from region to region depending on the number of nodes in that region. Then, the parameter named “distance factor” is calculated for each node pair. Setting “distance factor=1” to the closest node pair in that region, the step factor is added to the second closest node pair. Then, the step factor is added to other descendingly sorted node pairs. Thus, closer node pairs have a higher tendency to get higher savings values in the same region.

To demonstrate the “distance factor” idea, the following example is given:

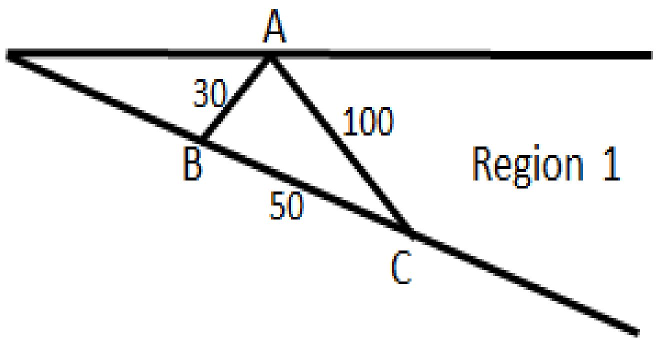

Figure 4.4. Demonstration of the “distance factor idea”.

Sorted pairs corresponding to distance are:

[AC]=100

[BC]=50

[AB]=30

For distance weight (explained in step 1d) =40

Number of node pairs (in region 1)=3

Step factor=40/3=13.33 (%13.33 effect)

Distance factors as follows:

(AC)=1 (since this is the closest pair)

(BC)=1+0.1333=1.1333

(AB)=1.1333+0.1333=1.2666

Step 1f: Calculate adjusted savings as: *Adjusted Saving=Saving * AngleFactor * DistanceFactor*

Step 2: Solve the VRP problem using the "SSA" method (explained in session 5.2.1.1) using "Adjusted Savings" obtained by step 6.

Step 3: End.

5.2.2. Tour Improvement Heuristics

The "2-opt" algorithm and its extension, named "Enhanced 2-opt" are coded in t-SQL and tested on 30 test instances given in Appendix A.

5.2.2.1 The 2-Opt Algorithm. The 2-opt algorithm, as discussed in session 3.2, is one of the most commonly used heuristics in the "route improvement" area which historically leads to considerable improvements for both randomly (arbitrarily) generated tours, experience based determined tours and tours obtained through various "tour building" algorithms.

As known, a good initial solution does not guarantee a good final solution. Since all the algorithms suggested in this study are heuristics, one can solve a VRP manually and find a less cost solution comparing to algorithms suggested in this thesis. In this study, the 2-opt algorithm is used to improve the initial solutions generated by the "Classical

Savings", "Savings' with Angle-Root Degree" and "Savings' with Angle Distance Factor" approaches. As discussed in session 3.2, the 2-opt is a simple local search algorithm, that cuts two edges and links them in other possible ways reordering the route, if it can find a more efficient solution at the end.

The 2-opt procedure developed and deployed in this study is as follows:

(i) Break 2 connections leaving 4 nodes not connected
(ii) Connect 4 nodes with only 2 edges but in a different way from the initial solution (hence, change the order by which the nodes are sequenced)
(iii) Check if the value of this new solution is better than the previous
(iv) Repeat the procedure for all edges of the initial route.

In order to better understand the mechanics of the procedure, consider the following initial solution generated by the Savings' Algorithm:

0→19→7→15→14→22→32→20→18→23→0

The 2-opt algorithm starts from the leftmost part of the route, which is 0-19 node pair and scans through the following options. 19-7 pair is skipped since the links to be exchanged ought not to share start/end points. When it finds a node pair which provides a less cost when it rejoins, it takes this edge and reorders the route.

Assume that it is desirable to replace 0-19 and 18-23 links with 0-18 and 19-23 (since $c_{0,18}+c_{19,23}$ is lower than $c_{0,19}+c_{18,20}$)

After this rearrangement, all parts of the route between node 19 and 18 should be traversed in reversed order in this new route in order to generate a directed graph. The new route becomes:

0→18→20→32→22→14→15→7→19→23→0

Another important remark is that, after deleting two edges, there is only one possible new connection option. For example, after deleting 0-19 and 18-23 pairs, the algorithm cannot join 0 to 23 and 19 to 18 since there is still an active consecutive arc between 0 and 23.

The algorithm keeps the initial route and tries all exchanges scanning the initial route starting from the leftmost part.

After phase 1 (0-19 node pair), exchanging other node pairs is tried as follows:

Phase 2: The link between (19-7) node pair is broken. New routing cost (rejoining this link with (15-14), (14-22), (32-20) or (23-0)) is calculated. (18-23 is omitted since the link does not exist after the first phase). Exchanges resulting lower total costs are executed. (If it encounters with a less cost on (15-14) it does not make any calculation for others; this node pair is directly selected.)

Assuming (22-32) is selected (to be rejoined with (19-7)), the new route becomes:

0→18→20→32→19→7→15→14→22→32→19→23→0

Phase 3: Similar rearrangements for the remaining node pairs are tried. After evaluation of rejoining options for all node pairs (in the first route), the final route is determined.

Note that the 2-opt process could run indefinitely if there is no strict stopping condition. Thus, the stopping condition is set as a “single pass” through the current links in the initial tour, in the order they appear in the current tour.

<u>5.2.2.2 The Enhanced 2-Opt Algorithm.</u> Another algorithm, which is named as the “Enhanced 2-opt”, is developed to improve the performance of the classical 2-opt algorithm implementation discussed in session 5.2.2.1. This enhancement computes all the pairwise interchange cost reductions and takes the highest cost reduction nodes to reorder

the route. In this case, the “Enhanced 2-opt Algorithm” passes are made through the (updated) tours.

The “Enhanced 2-opt” procedure developed and deployed in this study is as follows:

(i) break 1 connection leaving 2 nodes not connected
(ii) break 1 other connection such that reconnecting (2 disconnected nodes as a result of this step) with nodes in step i results in the smallest cost among all other reconnections
(iii) connect 4 nodes with only 2 edges but in a different way from the initial solution (hence, change the order by which the nodes are sequenced)
(iv) repeat steps i-iv (always starting from the latest route) until no less cost route can be built

The 2-opt concept described in Section 5.2.2.1 makes a decision about reordering/rejoining of arcs if total cost is smaller than existing total cost. On the other hand, there may be other rejoinining options (if it had a chance to see other node pairs which are after the selected node pair) which result in less cost. Additionally, the Enhanced 2-opt always make decisions on the final route (the route generated by reorderings). It tries all pairwise exchanges until there are no further cost improvements. Although its computational time is higher, it gives better results.

5.3. Data Acquisition

This study aims at obtaining a generic solution methodology, in order to be applied to other similar cases in future studies. So as to generate such a solution method, all tests are conducted with independent test instances gathered from an academic purposed web site named “www.branchandcut.org”.

In selecting test data of the problem, there are some key points that should be taken into consideration as listed below:

(i) Number and origin of the data sets: Since it's usually either impossible or impractical to track every member of a population, the next best option available is to sample the population. In statistical sampling, in order to attribute the result of the experiments to the whole population, at least 30 observations should be made. Accordingly, randomly selecting 30 test instances are assumed to be sufficient to give statistically significant results. So, 30 datasets besides the data set featuring real ATM data from a major bank in Turkey are used. The primary success criterion is set as the suggested heuristics giving better results than the basic Savings' algorithm in these 30 datasets. Then the successful heuristics could actually be deployed in real ATM cash delivery environments.

(ii) Number of demand points: Since problem tackled is NP hard, number of demand points range from 31 to 80. 31 is selected to be analogous to the number of data points of the original ATM problem. On the other hand, the developed methodology can be used in larger banks' data and ultimate objective of the study is suggesting applicable heuristics for all similar problems in a computationally acceptable time.

(iii) Demand: Some demand amounts are increased or decreased to see strength and weakness of the algorithms based on vehicle capacities. It is important to handle situations when demand of a point is greater than the capacity of the highest capacity truck. Although such a physical limitation seems odd in cash delivery vehicle, it is still a real limitation because of insurance issues (insurance coverage is possible for the transfer of limited funds per tour). Furthermore, such limitations are very much practical in other delivery environments.

(iv) Positions of demand points in the space: The general layout of the delivery points should not be similar in all of the instances as a specific algorithm can be successful in a specific geographical pattern and fail in another dispersion. So, this study tries to use different datasets which have different dispertion patterns.

Data selected based on these criteria is displayed in Appendix A. ID is the dataset number. NumDemand is the number of demand points in the data. Demand is the amount of goods that must be carried to the instance whose coordinates are indicated as “X” and “Y” in the table. From now on, all datasets will be called with its ID.

Additionally, although the purpose of this research is contributing to the “Vehicle Routing Problem” literature, its results are purposed to be used in the “ATM Cash Flow Optimization” project of a major private bank in Turkey. In order to test the algorithms in real life data, 5 days representative sample is demanded from the bank. Due to the high level of secrecy in ATM and depot coordinates, this particular dataset is modified and partially shared in Appendix C. Also, demand cash demands of provided ATMs are multiplied with some numbers, so that demands shared in appendix C do not reflect real demands provided by the private bank.

5.4. Experiments on Independent Test Instances

The aim of the experiments is finding a minimum cost daily route for all vehicles in an ATM cash holding depot. Despite the fact that all calculations are based on a single depot, the suggested procedure could also be used for multiple depots as the problem can be thought as multiple depot vehicle routing problems.

Although the result of the research will be used to determine the routing strategy of a commercial bank, academically, the solution approach can be extended to facilitate different vehicle (or goods) routing environments.

In order to advise a robust solution method, suggested algorithms are tested on both a sample data, and data provided by a commercial bank in Turkey.

5.4.1. Experimental Results for the Sample Data

Initial routes are built using the algorithms developed in Section 5.2. based on the classical “Savings’” concept discussed in Section 3.1.

Distances are calculated as Euclidian distances. Although all vehicles have the same capacity in the given example, in the generic heuristic method, vehicles are selected from highest to lowest capacity.

The classical Savings' results for Dataset 1 (in Appendix A) are given in Table 5.1 below.

Table 5.1. Tour Building Stages Associated with Dataset 1.

Cost=80.025 For Vehicle V1, Route 1: Depot -> 10 -> Depot
Cost=80.025 For Vehicle V2, Route 2: Depot -> 10 -> Depot
Cost=80.025 For Vehicle V3, Route 3: Depot -> 10 -> Depot
Cost=308.621 For Vehicle V4, Route 4: Depot -> 30 -> 21 -> 1 -> 27 -> 31 -> 17 -> 13 -> 2 -> 8 -> Depot
Cost=319.303 For Vehicle V5, Route 5: Depot -> 19 -> 7 -> 15 -> 14 -> 22 -> 32 -> 20 -> 18 -> 23 -> Depot
Cost=244.898 For Vehicle V6, Route 6: Depot -> 10 -> 16 -> 11 -> 26 -> 6 -> 28 -> Depot
Cost=204.056 For Vehicle V7, Route 7: Depot -> 9 -> 25 -> 3 -> 4 -> 24 -> 29 -> 5 -> Depot
Cost=23.4094 For Vehicle V8, Route 8: Depot -> 12 -> Depot

Next, all suggested algorithms except the "Enhanced 2-opt" are implemented on the 30 datasets mentioned in session 5.3, and the results are displayed in Table 5.3.

The "Enhanced 2-opt" algorithm is not tested for all instances, but tested for first 7 problem instances, since its computation time is remarkably higher than first suggested 2-opt algorithm. In other words, it is harder to implement in real life problems because of higher computational needs. As can be seen on Table 5.2, the "Enhanced 2-opt" gives a better result compared to the "2-opt" in all of the 7 test instances. However, improvement rates with respect to SSA and SARD varies in the range (0%-0.83%).

Table 5.2. Result of the Savings' Algorithm with "Enhanced 2-Opt" For The First 7 Problem Instances.

DSNo	**Savings (Classical)**	**2-Opt With Savings (Classical)**	**2-Opt Enhanced With Savings (Classical)**	**Savings (Angle-Root Degree)**	**2-Opt With Savings (Angle-Root Degree)**	**2-Opt Enhanced With Savings (Angle-Root Degree)**
1	1340.3	1305.2	1305.2	1363.1	1314.1	1304.4
2	989.9	989.9	989.9	986.3	974.5	974.5
3	593.0	591.1	591.1	589.6	584.8	584.3
4	1020.9	1005.0	1004.7	1024.5	1003.0	1000.0
5	1220.1	1200.4	1177.7	1255.4	1227.1	1164.3
6	1197.4	1194.5	1194.2	1165.8	1157.6	1132.8
7	1178.3	1129.7	1080.5	1094.6	1063.1	1056.1

In Table 5.3, "DSNo" is the "dataset no", "DPNo" is the "number of demand points in the dataset", "Best Cost" is the cost obtained implementing the "Algorithm" with the related "Best Cost Parameter" settings and "ExecDuration" is the amount of time (seconds) needed to run the corresponding algorithm on the specified problem on a Intel® Core™ 2 CPU 2 GHz Ram computer.

Also, "AW", "DW", "RD" are corresponding "Angle Weight", "Distance Weight" and "Root Degree" parameters as discussed in Section 5.2.1.

Table 5.3. Test Problem Results.

DSNo	**DPNo**	**Algorithm**	**BestCost**	**BestCostParameter**	**ExecDuration**
1	33	SSA	1340.4	---	3
1	33	2-Opt With SSA	1305.3	---	1
1	33	SARD	1363.1	RD:11	197
1	33	2-Opt With SARD	1314.2	---	2
1	33	SADF	1331.7	AW:1.DW:34	1082
1	33	2-Opt With SADF	1318.5	---	1
2	34	SSA	990.0	---	4
2	34	2-Opt With SSA	990.0	---	0

Table 5.3. Test Problem Results (cont.).

DSNo	DPNo	Algorithm	BestCost	BestCostParameter	ExecDuration
2	34	SARD	986.3	RD:13	185
2	34	2-Opt With SARD	974.6	---	0
2	34	SADF	969.7	AW:1,DW:24	1018
2	34	2-Opt With SADF	964.0	---	1
3	32	SSA	593.1	---	2
3	32	2-Opt With SSA	591.2	---	0
3	32	SARD	589.7	RD:28	106
3	32	2-Opt With SARD	584.8	---	1
3	32	SADF	593.1	AW:1,DW:1	711
3	32	2-Opt With SADF	591.2	---	1
4	35	SSA	1021.0	---	4
4	35	2-Opt With SSA	1005.1	---	1
4	35	SARD	1024.6	RD:46	173
4	35	2-Opt With SARD	1003.0	---	0
4	35	SADF	1019.9	AW:1,DW:1	1020
4	35	2-Opt With SADF	1004.8	---	1
5	34	SSA	1220.2	---	4
5	34	2-Opt With SSA	1200.5	---	0
5	34	SARD	1255.4	RD:31	159
5	34	2-Opt With SARD	1227.2	---	1
5	34	SADF	1187.7	AW:4,DW:1	1093
5	34	2-Opt With SADF	1184.4	---	0
6	35	SSA	1197.4	---	3
6	35	2-Opt With SSA	1194.6	---	1
6	35	SARD	1165.8	RD:21	168
6	35	2-Opt With SARD	1157.6	---	1
6	35	SADF	1165.8	AW:1,DW:8	1050
6	35	2-Opt With SADF	1157.2	---	1
7	38	SSA	1178.4	---	3
7	38	2-Opt With SSA	1129.7	---	1
7	38	SARD	1094.6	RD:48	185

Table 5.3. Test Problem Results (cont.).

DSNo	DPNo	Algorithm	BestCost	BestCostParameter	ExecDuration
7	38	2-Opt With SARD	1063.1	---	0
7	38	SADF	1162.7	AW:5,DW:34	1062
7	38	2-Opt With SADF	1134.6	---	1
8	38	SSA	1390.9	---	6
8	38	2-Opt With SSA	1376.6	---	0
8	38	SARD	1475.2	RD:41	238
8	38	2-Opt With SARD	1401.0	---	1
8	38	SADF	1374.7	AW:5,DW:15	1493
8	38	2-Opt With SADF	1359.8	---	0
9	38	SSA	1258.6	---	5
9	38	2-Opt With SSA	1209.3	---	0
9	38	SARD	1226.7	RD:41	232
9	38	2-Opt With SARD	1194.1	---	1
9	38	SADF	1197.8	AW:5,DW:16	1368
9	38	2-Opt With SADF	1178.4	---	0
10	40	SSA	1064.6	---	4
10	40	2-Opt With SSA	1040.6	---	1
10	40	SARD	1145.6	RD:41	224
10	40	2-Opt With SARD	1132.4	---	0
10	40	SADF	1064.6	AW:1,DW:1	1380
10	40	2-Opt With SADF	1040.6	---	1
11	40	SSA	1397.2	---	6
11	40	2-Opt With SSA	1357.3	---	0
11	40	SARD	1421.1	RD:48	377
11	40	2-Opt With SARD	1299.0	---	0
11	40	SADF	1322.2	AW:1,DW:48	1918
11	40	2-Opt With SADF	1300.9	---	1
12	45	SSA	1318.8	---	9
12	45	2-Opt With SSA	1318.2	---	0
12	45	SARD	1411.2	RD:11	452
12	45	2-Opt With SARD	1309.2	---	1
12	45	SADF	1319.5	AW:1,DW:1	2713

Table 5.3. Test Problem Results (cont.).

DSNo	DPNo	Algorithm	BestCost	BestCostParameter	ExecDuration
12	45	2-Opt With SADF	1318.7	---	2
13	46	SSA	1392.0	---	9
13	46	2-Opt With SSA	1392.0	---	1
13	46	SARD	1351.0	RD:49	420
13	46	2-Opt With SARD	1350.3	---	0
13	46	SADF	1343.1	AW:2.DW:1	2774
13	46	2-Opt With SADF	1340.5	---	0
14	46	SSA	1507.1	---	10
14	46	2-Opt With SSA	1446.2	---	1
14	46	SADF	1484.9	AW:2,DW:11	2678
14	46	2-Opt With SADF	1421.3	---	1
14	46	SARD	1703.4	RD:42	449
14	46	2-Opt With SARD	1547.6	---	1
15	47	SSA	1457.9	---	10
15	47	2-Opt With SSA	1440.6	---	2
15	47	SARD	1447.4	RD:13	510
15	47	2-Opt With SARD	1357.8	---	1
15	47	SADF	1397.0	AW:1,DW:5	3278
15	47	2-Opt With SADF	1347.8	---	1
16	49	SSA	1483.0	---	13
16	49	2-Opt With SSA	1459.0	---	0
16	49	SARD	1442.4	RD:33	560
16	49	2-Opt With SARD	1436.4	---	1
16	49	SADF	1383.6	AW:3,DW:1	3932
16	49	2-Opt With SADF	1346.8	---	1
17	54	SSA	1645.3	---	15
17	54	2-Opt With SSA	1586.7	---	2
17	54	SARD	1565.5	RD:49	881
17	54	2-Opt With SARD	1557.7	---	1
17	54	SADF	1492.1	AW:2,DW:5	4729
17	54	2-Opt With SADF	1489.4	---	0

Table 5.3. Test Problem Results (cont.).

DSNo	DPNo	Algorithm	BestCost	BestCostParameter	ExecDuration
18	55	SSA	1772.9	---	32
18	55	2-Opt With SSA	1770.9	---	0
18	55	SARD	1757.1	RD:29	973
18	55	2-Opt With SARD	1750.0	---	---
18	55	SADF	1742.1	AW:2,DW:3	---
18	55	2-Opt With SADF	1742.1	---	---
19	61	SSA	1910.6	---	35
19	61	2-Opt With SSA	1888.8	---	3
19	61	2-Opt With SADF	1972.2	---	---
19	61	SARD	1982.4	RD:44	1540
19	61	2-Opt With SARD	1953.7	---	2
19	61	SADF	1976.2	---	---
21	63	2-Opt With SARD	1788.7	---	1
21	63	SADF	1774.8	AW:1,DW:11	9895
21	63	2-Opt With SADF	1725.6	---	1
22	64	SSA	2015.1	---	53
22	64	2-Opt With SSA	2013.8	---	1
22	64	SARD	1988.4	RD:33	2251
22	64	2-Opt With SARD	1960.6	---	0
23	64	SSA	2196.3	---	45
23	64	2-Opt With SSA	2156.0	---	2
23	64	SARD	2104.9	RD:47	2776
23	64	2-Opt With SARD	2066.2	---	1
23	64	SADF	2062.5	AW:2,DW:1	12870
23	64	2-Opt With SADF	2043.1	---	1
24	65	SSA	2044.7	---	114
24	65	2-Opt With SSA	1994.5	---	2
24	65	SARD	2306.1	RD:9	2290
24	65	2-Opt With SARD	2138.3	---	1
25	66	SSA	1904.4	---	46
25	66	2-Opt With SSA	1854.8	---	3

Table 5.3. Test Problem Results (cont.).

DSNo	DPNo	Algorithm	BestCost	BestCostParameter	ExecDuration
25	66	SARD	2203.8	RD:21	2426
25	66	2-Opt With SARD	2070.4	---	1
25	66	SADF	1893.7	AW:3,DW:31	13753
25	66	2-Opt With SADF	1852.4	---	0
26	70	SSA	1877.6	---	75
26	70	2-Opt With SSA	1811.3	---	5
26	70	SARD	1960.6	RD:41	2580
26	70	SADF	1804.1	AW:5,DW:2	14925
26	70	2-Opt With SADF	1796.0	---	1
26	70	2-Opt With SARD	1917.3	---	1
27	81	SSA	2462.7	---	113
27	81	2-Opt With SSA	2353.5	---	1
27	81	SARD	2351.7	RD:23	5317
27	81	2-Opt With SARD	2274.2	---	1
28	36	SSA	1072.8	---	4
28	36	2-Opt With SSA	1040.7	---	2
28	36	SARD	1092.9	RD:42	164
28	36	2-Opt With SARD	1066.3	---	1
28	36	SADF	1050.2	AW:1,DW:8	1078
28	36	2-Opt With SADF	1034.0	---	1
29	39	SSA	1229.2	---	4
29	39	2-Opt With SSA	1158.0	---	1
29	39	SARD	1198.2	RD:26	271
29	39	2-Opt With SARD	1103.1	---	0
29	39	SADF	1138.1	AW:1,DW:8	1431
29	39	2-Opt With SADF	1112.0	---	0
30	40	SSA	886.9	---	5
30	40	2-Opt With SSA	841.4	---	1
30	40	SARD	897.6	RD:25	227
30	40	2-Opt With SARD	781.9	---	1
30	40	SADF	851.0	AW:1,DW:10	1350

Table 5.3. Test Problem Results (cont.).

DSNo	DPNo	Algorithm	BestCost	BestCostParameter	ExecDuration
30	40	2-Opt With SADF	789.1	---	1

As can be seen in Table 5.3, in most of the problem instances, "Savings' with Angle Distance Factor" algorithm outperforms both the "Angle-Root Degree" and the "Classical Savings'" algorithms. On the other hand, its computation time is remarkably higher than other suggested algorithms. In small datasets, the "Savings' with Angle Distance Factor" is the best algorithm to construct initial routes. In large samples, the "Savings' with Angle Root Degree", which shows the second best performance, might be implemented. The logical design and software codes of all algorithms are shared with the thesis via the "ReadMe" file.

A good starting solution does not guarantee a good final solution. After a tour improvement algorithm is applied, the resultant situation may change. For example, in DSNo 1, although the "Savings' with Angle-Distance Factor" outperforms all other suggested algorithms in the initial route, implementing 2-opt improves the classical "Savings'" relatively higher and the classical "Savings'" has the best performance at the end.

5.5. Experiments on the ATM Cash Flow Data

5.5.1. Handling of the Data

This study is conducted using "Euclidian distances" such that coordinates of the demand points, (latitude and longitude information, i.e. ATM locations) are provided by a private bank. 5 days' sample data including 192 individual ATMs is obtained. First day (including 30 ATMs) is used (in Section 5.5.2) to test the classical "Savings'" and the "2-opt" algorithms. All 5 days' sample is used to test the "Switch Algorithm" described in Section 5.6.

First, data is modified and cleaned (i.e. inconsistencies and missing information is handled). Then, the input module of the solver is modified to accept and process coordinate information. Longitude values are represented by “x”, and latitude values by “y” in this application. (As a first step of the study, depot coordinates are assumed to be 29-41.) Since, in Turkey, approximate distance between two longitude is 85 km and two latitudes is 111 km, coordinate information is converted to Euclidian distances to solve the problem.

5.5.2. Experimental Results of the ATM Cash Flow Data

A sample result using the “classical Savings’” and the “2-opt” algorithms of the first given day (in which homogeneous vehicle capacities are assumed to be 18 million TL) is displayed in Table 5.4 and Table 5.5:

Table 5.4. Proposed Vehicle Routes Generated by the Implementation of the Savings’ Algorithm (presented in Section 5.2.1.1) to ATM Data.

Nodes	**Vehicle 1**	**Vehicle 2**	**Vehicle 3**
1	Depot	Depot	Depot
2	Kütahya Bayindirlik	Profilo Avm	Çekmeköy Adese
3	Etimesgut Metrokent Avm	Küçükçekmece Yücel Petrol	Şişli 1
4	Serik Akdeniz Mah.	Nişantaşi Citys	Çerkezköy Saray
5	Fethiye Ovacik Ramos Otel	Marmara Ünv. Bahçelievler Kampüsü	Lüleburgaz Zorlulinen
6	Elmali Kalkan Kiziltaş	Demirören Istiklal Avm	Beylikdüzü Migros
7	Marmaris İçmeler	Marmara Ereğlisi	Avcilar Parseller
8	Kuşadasi Ephesia	Inegöl Milli Eğitim	Bahçelievler Mars Lojistik
9	Güzelbahçe Meydan	Kütahya Azerbaycan	Şişli 2
10	İzmir Metro Konak	Kuşadasi Belediye	Depot
11	Mudanya Korteks Berkun	Alanya Konakli Oasis	
12	Depot	Yenimahalle Onkoloji Hastanesi	
13		Gebze Organize	
14		Depot	

Table 5.4. Proposed Vehicle Routes Generated by the Implementation of the Savings' Algorithm (presented in Section 5.2.1.1) to ATM Data (cont.).

Routing Cost (Km)	1,679.14	1,831.39	305.20
Total Supply Amount (TL)	17,684,568	17,215,164	11,981,484
Remaining Vehicle Capacity (TL)	315,432	784,836	6,018,516

Table 5.5. Proposed Vehicle Routes Generated by the Implementation of the 2-opt Algorithm (presented in Section 5.2.2.1) to ATM Data.

Nodes	**Vehicle 1**	**Vehicle 2**	**Vehicle 3**
1	Depot	Depot	Depot
2	Kütahya Bayindirlik	Küçükçekmece Yücel Petrol	Çekmeköy Adese
3	Etimesgut Metrokent Avm	Marmara Ünv. Bahçelievler Kampüsü	Bahçelievler Mars Lojistik
4	Serik Akdeniz Mah.	Demirören Istiklal Avm	Avcilar Parseller
5	Elmali Kalkan Kiziltaş	Profilo Avm	Beylikdüzü Migros
6	Fethiye Ovacik Ramos Otel	Nişantaşi Citys	Lüleburgaz Zorlulinen
7	Marmaris İçmeler	Marmara Ereğlisi	Çerkezköy Saray
8	Kuşadasi Ephesia	Inegöl Milli Eğitim	Şişli 1
9	Güzelbahçe Meydan	Kütahya Azerbaycan	Şişli 2
10	İzmir Metro Konak	Kuşadasi Belediye	Depot
11	Mudanya Korteks Berkun	Alanya Konakli Oasis	
12	Depot	Yenimahalle Onkoloji Hastanesi	
13		Gebze Organize	
14		Depot	
Routing Cost (km)	1,635.05	1,807.84	303.58
Total supply amount (TL)	17,684,568	17,215,164	11,981,484
Remaining vehicle capacity (TL)	315,432	784,836	6,018,516

According to Table 5.4 and Table 5.5, (total) relative costs incurred as a results of the classical Savings' and the 2-opt algorithms are 3815.73 km and 3746.47 km. Although computational time increases in a reasonable amount, cost saving gained by applying the 2-opt algorithm is remarkable.

5.6. Extension of the Routing Problem: Switches Between Days

5.6.1. Why Do We Need Switches?

As discussed in Chapter 1, while the full scale ATM cash delivery problem includes decisions regarding visiting frequency and amount delivered to each node (ATM), as well as routing decisions, in this study, the first set of decisions (regarding frequency and amounts) are treated as parameters. In this section, this simplifying assumption is somewhat relaxed by considering "switches" between nodes in different (consecutive) days. In other words, reducing the visit frequency of individual nodes and changing the cash amount to be delivered to these nodes (because of the revised expected service time until the next scheduled visit) are considered.

For example, a vehicle can visit an ATM one day earlier than initially planned in case its opportunity cost is higher. Opportunity cost means the cost of any activity measured in terms of the value of the next best alternative forgone (that is not chosen). If it visits a demand point one day earlier, money keeping cost increases since it needs to load extra supply for an additional day. On the other hand, if the ATM is close to the ATMs planned to be visited on the earlier day, total routing cost of both days may be substantially lower. These switches between a day and the following day are considered, in case the routing cost decreases. If routing cost does not decrease, it is impossible to obtain a less cost alternative since, money keeping (inventory) cost certainly increases in the event of an early visit and delivery.

5.6.2. The Switch Algorithm

Steps of the proposed "Switch Algorithm" are as follows:

Step 0: The proposed delivery schedule of the n-day planning horizon. “n” is taken as input togather with the individual coordinates and amounts to be delivered of the ATM’s to be visited each day.

Step 1: Solve the VRP problem for the given n days (1,2,3,…n) and calculate total daily transportation costs and routes.

Step 2: Start from day “n” by setting current day c<=n. Evaluate moving each ATM i in any of the proposed vehicle routes of day c to one of the routes proposed for day c-1.

Step 2a: For each ATM, calculate the transportation cost improvement of moving that ATM from day “c” to day “c-1” as: *Decrease in Transportation Cost= (Old total transportation cost* $_{day(n)+day(n+1)}$*) - (New total transportation cost* $_{day(n)+day(n+1)}$ *)*

Note that in step 2a, the transportation cost differential is calculated in “distance” units instead of “money” units. (Transportation cost is calculated using the “Classical Savings’ algorithm). Switched ATM is put in the best cost position. The best cost position is found by solving the problem (with additional ATM for day c-1) using the classical Savings’ algorithm.

Step 2b: If the “transportation cost differential” ≥ 0 then, calculate the total monetary improvement of the “Switch” as: *Total Improvement = Decrease in transportation cost * 2 * (Gasoline Cost Per Liter / Kilometers drived by 1 liter gasoline) - (Demand of the switched ATM * Daily Interest Rate of money)*

In the above formula, it is assumed that labor cost and depreciation cost is as much as gasoline cost.

If the “total improvement” > 0 then, switch the ATM under consideration to day (c-1).

Else, do not switch the ATM.

Step 2c: Set c=c-1 and repeat 2a and 2b. Stop when c=0.

Number of savings calculations to be done in such a model (when n=5) is:

(i) 2* ($n_2$final+ $n_3$final+ $n_4$final+ $n_5$final)+4 or

(ii) ($n_2$initial+$n_2$switch+$n_3$initial+$n_3$switch+$n_4$initial+$n_4$switch+$n_5$initial+$n_5$switch)*2+4

$n_i$final: number of points in day "i" after all switches

$n_i$initial: number of points in day "i" at the beginning

$n_i$switch: number of switches from that point

Number of Savings' calculation increases by total number of nodes and days since for each switch trial, initial routes are built to calculate total transportation cost for 2 consecutive days. Thus, the suggested method is more reasonable (in terms of computation time) for small (or medium size) datasets.

5.6.3. Computational Studies

The "Switch" algorithm mentioned in Section 5.6.2 is applied to the 5 consecutive day ATM data provided by a private bank and presented in Appendix C.

Since the most disadvantageous part of the Switch algorithm is computation time, the algorithm is tested only for the classical Savings' based algorithms (the classical Savings', the classical Savings' with 2-Opt, the classical Savings' with Enhanced 2-Opt).

Additional parameters needed by the "Switch" method are set as follows:

- Average velocity of homogeneous delivery vehicles: 35km/h.
- Maximum allowed time for a vehicle to turn back to the depot: 9 hours.
- Daily break time for a driver: 60 min
- Maximum capacity of a single vehicle (Restricted by the insurance company): 18 Million TL.

- The sum of labor cost and depreciation cost is assumed to be as much as the gasoline cost.

- 1 liter of gasoline (for vehicles): 5 TL.

- A vehicle is able to go 10 kilometers using 1 liter gasoline.

- Daily interest rate is: 0.0002.

Additionally, since Euclidian distances cannot be used due to operational reasons, distances provided by Google Maps API are used in this example since they would provide a more accurate representation than Euclidian distances.

The initial routes obtained (before the Switch implementation) are:

Table 5.6. Result of the Savings Algorithm for the 1st day.

Date	**17.05.2012**
Nodes	**Vehicle 1**
1	Depot
2	Gebze Organize
3	Nişantaşı Citys
4	Şişli 1
5	Şişli 2
6	Küçükçekmece Yücel Petrol
7	Avcılar Parseller
8	Beylikdüzü Migros
9	Bahçelievler Mars Lojistik
10	Marmara Ünv. Bahçelievler Kampüsü
11	Demirören İstiklal Avm
12	Profilo Avm
13	Çekmeköy Adese
14	Depot
Routing Cost (Km)	191
Total Supply Amount (TL)	309,858.4

Table 5.6. Result of the Savings Algorithm for the 1st day (cont.).

Remaining Vehicle Capacity (TL)	17,690,141.6
Duration (hour)	5.46

Table 5.7. Result of the Savings Algorithm for the 2nd day.

Date	**18.05.2012**
Nodes	**Vehicle 1**
1	Depot
2	Pendik Aydınlı Yolu
3	Ataşehir İçerenköy Meydan
4	Kadıköy Kozyatağı Bayar Cad.
5	İdo Kabataş
6	Şişli Anthill Residance
7	Zekeriyaköy Meydan
8	Fatih Karagümrük Meydan
9	Bayrampaşa Şehir Parkı
10	212 İstanbul Avm
11	Beylikdüzü Atrium Çarşı
12	Atatürk Havalimanı
13	Marmara Forum Avm
14	Kadıköy Kozzy Avm
15	İdo Pendik
16	Depot
Routing Cost (Km)	192.34
Total Supply Amount (TL)	291,077.75
Remaining Vehicle Capacity (TL)	17,708.,922.03
Duration (hour)	5.50

Table 5.8. Result of the Savings Algorithm for the 3rd day.

Date	**19.05.2012**
Nodes	**Vehicle 1**
1	Depot
2	Tuzla Kipaş
3	Başakşehir Toki Kayaşehir
4	Arnavutköy Akpınar Sanayi Bölgesi
5	Silivri Ekol Ofset
6	Esenyurt Torium Avm
7	Gebze Ferro Döküm
8	Depot
Routing Cost (Km)	257.57
Total Supply Amount (TL)	220,576,3
Remaining Vehicle Capacity (TL)	17,779,423.7
Duration (hour)	7.36

Table 5.9. Result of the Savings Algorithm for the 4th day.

Date	**19.05.2012**
Nodes	**Vehicle 1**
1	Depot
2	Merter Meydan Avm
3	Depot
Routing Cost (Km)	38.98
Total Supply Amount (TL)	15,329.74
Remaining Vehicle Capacity (TL)	17,984,670.26
Duration (hour)	1.11

Table 5.10. Result of the Savings Algorithm for the 5th day.

Date	17.05.2012	
Nodes	**Vehicle 1**	**Vehicle 2**
1	Depot	Depot
2	Tuzla Kıran	Esenler Taksi Durağı
3	Ataşehir Ağaoğlu My Town	Depot
4	Üsküdar Koşuyolu Validebağ Sitesi	--
5	Zincirlikuyu Kanal	--
6	Fatih Bahçekapı Ticaret Merkezi	--
7	İdo Yenikapı	--
8	Bahçelievler Hizmet Hastanesi	--
9	Bahçelievler Çalış Cad.	--
10	Bakırköy Dünya Ticaret Merkezi Metro	--
11	Küçükçekmece Türk Telekom	--
12	Küçükçekmece Mng Teknik	--
13	Avcılar Karaköy Metraco	--
14	Avcılar Vestel	--
15	Beylikdüzü Migros	--
16	Mimaroba Bp	--
17	Silivri Tesco Kipa	--
18	Esenyurt Kiler Genel Müd.	--
19	Başakşehir Petrol	--
20	Bağcılar Automall Cıty	--
21	Güngören Soğanlı	--
22	Gaziosmanpaşa Yıldıztabya	--
23	Okmeydanı Memorıal	--
24	Şişli Bahçeşehir Üniversitesi	--
25	Show Tv Ayazağa	--
26	Astorıa Avm	--
27	Levent Metro	--
28	Başakşehir Kavacık	--
29	Taksim Metro	--
30	İdo Kadıköy	--
31	Kabataş Metro	--
32	Beykoz İskele	--
33	Beylikdüzü Optimum Avm	--
34	İstanbul Anadolu Bölgesi Kartal	--
Routing Cost (Km)	274.924	75.4

Table 5.10. Result of the Savings Algorithm for the 5th day (cont.).

Total Supply Amount (TL)	627,425	23,587.2
Remaining Vehicle Capacity (TL)	17,372,575	17,976,412.8
Duration (hour)	7.85	2.15

As can be seen, number of demand points (in Istanbul) for these 5 days are as 12, 14, 6, 1, 34.

Corresponding results after the Switch implementation are:

Table 5.11. The "Switch Algorithm" with Gasoline Cost.

Algorithm	**Node**	**Label**	**Switched From**	**Switched To**	**Demand Amount**	**Cost Savings (TL)**
Classical S.	G0015	Okmeydanı Memorıal	21.05.2012	20.05.2012	1516,104	2.12
Classical S.	G0607	Merter Meydan Avm	20.05.2012	19.05.2012	851,652	3.21
Classical S.	G0091	Bayrampaşa Şehir Parkı	18.05.2012	17.05.2012	918,774	11.38
Classical S. 2-opt	G0607	Merter Meydan Avm	20.05.2012	19.05.2012	851,652	29.90
Classical S. 2-opt	G0048	İdo Kabataş	18.05.2012	17.05.2012	930,798	2.23
Classical S. 2-opt	G0091	Bayrampaşa Şehir Parkı	18.05.2012	17.05.2012	918,774	6.86
Classical S. Enhanced 2-opt	G0607	Merter Meydan Avm	20.05.2012	19.05.2012	851,652	27.98
Classical S. Enhanced 2-opt	G0048	İdo Kabataş	18.05.2012	17.05.2012	930,798	13.66
Classical S. Enhanced 2-opt	G0455	Pendik Aydınlı Yolu	18.05.2012	17.05.2012	866,286	2.77

As can be seen in the above results, travel cost considerations (savings) lead to significant number of switches. Let's assume that gasoline cost per liter is 20 TL instead of 5 TL. It is expected to have more switched points in this scenario. Result for the classical Savings' is as below:

Table 5.12. Switch Results with Increasing Gasoline Cost.

Algorithm	Node	Label	Switched From	Switched To	Demand Amount	Cost Savings (TL)
Classical S.	G0015	OKMEYDANI MEMORİAL	21.05.2012	20.05.2012	1516,104	59.02
Classical S.	G0015	OKMEYDANI MEMORİAL	20.05.2012	19.05.2012	1516,104	0.16
Classical S.	G0607	MERTER MEYDAN AVM	20.05.2012	19.05.2012	851,652	140.31
Classical S.	G0048	İDO KABATAŞ	18.05.2012	17.05.2012	930,798	8.68
Classical S.	G0091	BAYRAMPAŞA ŞEHİR PARKI	18.05.2012	17.05.2012	918,774	56.14
Classical S.	G0110	BEYLİKDÜZÜ ATRİUM ÇARŞI	18.05.2012	17.05.2012	2507,634	53.31
Classical S.	G0143	212 İSTANBUL AVM	18.05.2012	17.05.2012	1006,668	27.17

As discussed at the beginning of this chapter, switching ATMs to earlier days (i.e. increased delivery frequency) can result in significant savings. Also, more flexible switching options of ATMs between days might be tried in future studies.

5.7. Trend Analysis of the Algorithms

This study suggests 3 route initialization and 2 route improvement heuristics. In a real life problem, it is a time consuming task to run all of the algorithms (6 possible combinations). In order to make an inference regarding dataset types, and associated more promising combination of solution algorithms, the performance of the considered algorithms over 30 datasets are analyzed regarding their "number of nodes" and "distance variation among nodes" over 30 datasets.

In table 5.8 and 5.10, integers [1-6] show the success order of the corresponding algorithm where 1 belongs to the most successful algorithm and 6 belongs to the least successful algorithm. Table 5.8 compares algorithms depending on number of nodes. Table 5.9 summarizes results of table 5.8 As seen in table 5.9, the “2-opt with SADF” has the best performance for each (number of nodes) interval. Its success ratio (ratio of being the best algorithm among datasets in this interval) is taking the highest value for datasets whose number of nodes are higher. On the other hand, the “2-opt with SARD” has a better performance on smaller datasets.

Table 5.10 uses “distance variability among nodes” as a comparison criteria. It may be concluded that the “2-opt with SARD” can be used for datasets with smaller distance variation (among nodes), yet, “2-opt with SADF” is more suitable for datasets with medium or large distance variation (among nodes).

Table 5.13. Trend Analysis of Algorithms: Dependence to the Number of Nodes.

	Algorithm					
Number of nodes	SSA	SARD	SADF	2-opt with SSA	2-opt with SARD	2-opt with SADF
32	4	2	4	3	1	3
33	5	6	4	1	2	3
34	4	6	2	3	5	1
34	5	4	2	5	3	1
35	4	6	2	3	5	1
35	5	6	4	3	1	2
36	5	6	3	2	4	1
38	6	5	3	4	2	1
38	4	6	2	3	5	1
38	6	2	5	3	1	4
39	6	5	3	4	1	2
40	5	6	4	3	1	2
40	2	4	2	1	3	1
40	5	6	3	4	1	2
45	4	6	5	2	1	3
46	5	4	2	5	3	1
46	4	6	3	2	5	1
47	6	5	3	4	2	1
49	6	4	2	5	3	1

Table 5.13. Trend Analysis of Algorithms: Dependence to the Number of Nodes (cont.).

	Algorithm					
Number of nodes	SSA	SARD	SADF	2-opt with SSA	2-opt with SARD	2-opt with SADF
54	6	4	2	5	3	1
55	5	3	1	4	2	1
61	2	4	6	1	3	5
63	6	5	2	4	3	1
64	4	2	4	3	1	3
64	6	4	2	5	3	1
65	2	4	2	1	3	1
66	4	6	3	2	5	1
70	4	6	2	3	5	1
81	4	2	4	3	1	3

Table 5.14. Percentage of Different Algorithms in Achieving the Best Performance.

	Success Rations			
Number of Nodes	2 opt with SADF	2 opt with SARD	2 opt with SSA	SADF
[32-40)	0,55	0,36	0,09	0
[40-50)	0,63	0,36	0,13	0
[50-81)	0,7	0,2	0,2	0,1

Table 5.15. Trend Analysis of Algorithms: Dependence to the Distance Variation.

		Algorithm					
DSNo	**Distance Variation**	SSA	SARD	SADF	2-opt with SSA	2-opt with SARD	2-opt with SADF
4	474,43	5	6	4	3	1	2
20	483,75	4	2	4	3	1	3
7	592,91	6	2	5	3	1	4
3	622,02	4	2	4	3	1	3
15	677,03	6	5	3	4	2	1
24	689,77	2	4	2	1	3	1
8	694,03	6	2	5	3	1	4
18	715,28	5	3	1	4	2	1
29	722,26	6	5	3	4	1	2
23	730,73	6	4	2	5	3	1
10	750,23	2	4	2	1	3	1
27	776,81	4	2	4	3	1	3
9	785,77	6	5	3	4	2	1

Table 5.15. Trend Analysis of Algorithms: Dependence to the Distance Variation (cont.).

	Algorithm						
DSNo	**Distance Variation**	**DSNo**	**Distance Variation**	**DSNo**	**Distance Variation**	**DSNo**	**Distance Variation**
14	789,16	4	6	3	2	5	1
22	809,49	4	2	2	3	1	1
26	813,34	4	6	2	3	5	1
19	819,19	2	4	6	1	3	5
25	824,09	4	6	3	2	5	1
12	855,63	4	6	5	2	1	3
17	891,1	6	4	2	5	3	1
1	920,06	5	6	4	1	2	3
21	922,91	6	5	2	4	3	1
6	963,46	4	6	2	3	5	1
11	964,67	5	6	3	4	1	2
5	971,94	4	6	2	3	5	1
13	1058,78	5	4	2	5	3	1
2	1085,38	5	4	2	5	3	1
16	1191,35	6	4	2	5	3	1
28	1304,78	5	6	3	2	4	1
30	1951,38	5	6	4	3	1	2

6. CONCLUSIONS

In this study, trigered by an interesting ATM cash delivery problem, some “initial tour building” and “tour improvement” heuristics for the VRP problem are discussed. New “initial tour building” methods based on the classical “Savings’” and the “Sweep” algorithms are suggested to improve the solution obtained by the classical Savings’ algorithm. The first algorithm features an angle factor so as to increase the tendency to merge node pairs which have smaller angles with respect to the depot. Some parameters are set dynamically and the algorithm searches for best cost parameters. Another suggested heuristic features both angle and distance considerations. Despite the fact that computational time of the second heuristic is larger, it gives comperatively better results in most of the test instances. Then, two route improvement algorithms are discussed and coded in t-SQL. Firstly, the “2-opt” algorithm is applied on the above mentioned 3 route initialization algorithms. Next, a new 2-opt algorithm named as the “Enhanced 2-opt algorithm” is proposed. This algorithm gives comperatively better results compared to the classical 2-opt. Since we have 6 different algorithmic combinations, a trend analysis is conducted to select best algorithmic procedure for each spesific test instance. Finally, regarding the ATM cash delivery problem, some “inventory routing” aspects are considered; namely the option of “earlier supply of cash” is investigated via node switches between days. A switch algorithm is proposed and tested through a set of a 5 day ATM visiting plan provided by commercial bank in Turkey. (Data provided by the bank is manipulated due to confidentiality issues). This experimentation shows that due to transportation cost savings, it may be more adventageous to visit some nodes (ATMs) earlier than originally scheduled (even if more inventory costs are to be incurred). On the other hand, some qualitative measures such as customer satisfaction are not taken into consideration while suggesting these switches.

During this research effort, simplicity and flexibility in the design of the suggested heuristics is incorporated. Simplicity makes the developed heuristics easy to understand and easy to implement. Simplicity also makes it easier to incorporate additional features.

Flexibility enables the heuristic to easily accommodate side constraints, so a set of related problems can be solved.

Test instances are tried to be selected in a stratified and objective way to provide flexibility. All experiments are conducted on both datasets in Appendix A, and ATM cash delivery data (Appendix C) since constructed heuristics are aimed to be used in other problem cases. Results are promising both in terms of simplicity/flexibility and performance.

APPENDIX A: SAMPLE DATASET

Table A.16. Sample Dataset.

ID	NumDem	Demand	X	Y
1	32	0	82	76
1	32	19	96	44
1	32	21	50	5
1	32	6	49	8
1	32	19	13	7
1	32	7	29	89
1	32	12	58	30
1	32	16	84	39
1	32	6	14	24
1	32	316	2	39
1	32	8	3	82
1	32	14	5	10
1	32	21	98	52
1	32	16	84	25
1	32	3	61	59
1	32	22	1	65
1	32	18	88	51
1	32	19	91	2
1	32	1	19	32
1	32	24	93	3
1	32	8	50	93
1	32	12	98	14
1	32	4	5	42
1	32	8	42	9
1	32	24	61	62
1	32	24	9	97
1	32	2	80	55
1	32	20	57	69
1	32	15	23	15
1	32	2	20	70
1	32	14	85	60
1	32	9	98	5
-	-	-	-	-
-	-	-	-	-

ID	NumDem	Demand	X	Y
2	33	0	42	68
2	33	5	77	97
2	33	23	28	64
2	33	14	77	39
2	33	13	32	33
2	33	8	32	8
2	33	18	42	92
2	33	19	8	3
2	33	10	7	14
2	33	18	82	17
2	33	20	48	13
2	33	5	53	82
2	33	9	39	27
2	33	23	7	24
2	33	9	67	98
2	33	18	54	52
2	33	10	72	43
2	33	24	73	3
2	33	13	59	77
2	33	14	58	97
2	33	8	23	43
2	33	10	68	98
2	33	19	47	62
2	33	14	52	72
2	33	13	32	88
2	33	14	39	7
2	33	2	17	8
2	33	23	38	7
2	33	15	58	74
2	33	8	82	67
2	33	20	42	7
2	33	24	68	82
2	33	3	7	48
-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y
3	31	0	17	76
3	31	25	24	6
3	31	3	96	29
3	31	13	14	19
3	31	17	14	32
3	31	16	0	34
3	31	9	16	22
3	31	22	20	20
3	31	10	22	28
3	31	16	17	23
3	31	8	98	30
3	31	3	30	8
3	31	16	23	27
3	31	16	19	25
3	31	10	34	7
3	31	24	31	7
3	31	16	0	37
3	31	15	19	23
3	31	14	0	36
3	31	5	26	7
3	31	12	98	32
3	31	2	5	40
3	31	18	17	26
3	31	20	21	26
3	31	15	28	8
3	31	8	1	35
3	31	22	27	28
3	31	15	99	30
3	31	10	26	28
3	31	13	17	29
3	31	19	20	26
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

ID	NumDem	Demand	X	Y
4	34	0	28	57
4	34	6	76	46
4	34	12	67	5
4	34	2	84	22
4	34	24	73	6
4	34	3	67	72
4	34	18	68	74
4	34	21	68	7
4	34	14	76	7
4	34	69	91	30
4	34	1	80	0
4	34	13	0	25
4	34	2	73	13
4	34	2	76	81
4	34	7	92	30
4	34	7	69	80
4	34	1	90	30
4	34	23	71	77
4	34	19	83	47
4	34	14	79	47
4	34	8	74	6
4	34	11	68	6
4	34	4	72	0
4	34	8	80	1
4	34	24	91	25
4	34	12	71	73
4	34	9	78	10
4	34	4	85	24
4	34	19	75	80
4	34	15	87	24
4	34	2	0	7
4	34	2	71	78
4	34	15	74	0
4	34	66	76	0

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y
5	33	0	34	31
5	33	26	34	31
5	33	17	70	80
5	33	6	81	70
5	33	15	85	61
5	33	7	59	55
5	33	5	45	60
5	33	15	50	64
5	33	16	80	64
5	33	17	75	90
5	33	1	25	40
5	33	21	9	66
5	33	66	1	44
5	33	25	50	54
5	33	16	35	45
5	33	11	71	84
5	33	7	1	9
5	33	17	25	54
5	33	17	45	59
5	33	22	45	71
5	33	10	66	84
5	33	25	11	35
5	33	16	81	46
5	33	7	85	10
5	33	21	75	20
5	33	11	15	21
5	33	21	90	45
5	33	11	15	0
5	33	21	31	26
5	33	22	10	95
5	33	25	6	6
5	33	2	51	5
5	33	22	26	36
-	-	-	-	-

ID	NumDem	Demand	X	Y
6	34	0	73	39
6	34	23	67	91
6	34	3	39	21
6	34	24	3	9
6	34	15	97	15
6	34	15	91	65
6	34	24	55	75
6	34	7	55	71
6	34	25	57	85
6	34	13	21	15
6	34	5	47	57
6	34	7	51	97
6	34	5	11	11
6	34	14	43	59
6	34	13	63	69
6	34	5	55	77
6	34	24	35	11
6	34	15	27	91
6	34	9	49	25
6	34	16	29	93
6	34	13	71	27
6	34	16	31	43
6	34	13	27	9
6	34	24	67	99
6	34	20	87	81
6	34	23	23	81
6	34	20	89	33
6	34	3	71	91
6	34	15	19	77
6	34	12	65	77
6	34	19	87	79
6	34	4	19	83
6	34	15	1	59
6	34	1	55	7

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
7	37	0	38	46	7	37	7	26	54
7	37	16	59	46	7	37	20	18	89
7	37	18	96	42	7	37	20	22	53
7	37	1	47	61	-	-	-	-	-
7	37	13	26	15	-	-	-	-	-
7	37	8	66	6	-	-	-	-	-
7	37	23	96	7	-	-	-	-	-
7	37	7	37	25	-	-	-	-	-
7	37	27	68	92	-	-	-	-	-
7	37	1	78	84	-	-	-	-	-
7	37	3	82	28	-	-	-	-	-
7	37	6	93	90	-	-	-	-	-
7	37	24	74	42	-	-	-	-	-
7	37	19	60	20	-	-	-	-	-
7	37	2	78	58	-	-	-	-	-
7	37	5	36	48	-	-	-	-	-
7	37	16	45	36	-	-	-	-	-
7	37	7	73	57	-	-	-	-	-
7	37	4	10	91	-	-	-	-	-
7	37	22	98	51	-	-	-	-	-
7	37	7	92	62	-	-	-	-	-
7	37	23	43	42	-	-	-	-	-
7	37	16	53	25	-	-	-	-	-
7	37	2	78	65	-	-	-	-	-
7	37	2	72	79	-	-	-	-	-
7	37	9	37	88	-	-	-	-	-
7	37	2	16	73	-	-	-	-	-
7	37	12	75	96	-	-	-	-	-
7	37	1	11	66	-	-	-	-	-
7	37	9	9	49	-	-	-	-	-
7	37	23	25	72	-	-	-	-	-
7	37	6	8	68	-	-	-	-	-
7	37	19	12	61	-	-	-	-	-
7	37	7	50	2	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
8	37	0	86	22	8	37	19	51	38
8	37	1	29	17	8	37	66	83	74
8	37	23	4	50	8	37	21	84	2
8	37	23	25	13	-	-	-	-	-
8	37	5	67	37	-	-	-	-	-
8	37	7	13	7	-	-	-	-	-
8	37	18	62	15	-	-	-	-	-
8	37	12	84	38	-	-	-	-	-
8	37	20	34	3	-	-	-	-	-
8	37	19	19	45	-	-	-	-	-
8	37	19	42	76	-	-	-	-	-
8	37	16	40	86	-	-	-	-	-
8	37	2	25	94	-	-	-	-	-
8	37	26	63	57	-	-	-	-	-
8	37	13	75	24	-	-	-	-	-
8	37	19	61	85	-	-	-	-	-
8	37	17	87	38	-	-	-	-	-
8	37	14	54	39	-	-	-	-	-
8	37	8	66	34	-	-	-	-	-
8	37	10	46	39	-	-	-	-	-
8	37	5	47	17	-	-	-	-	-
8	37	19	21	54	-	-	-	-	-
8	37	12	19	83	-	-	-	-	-
8	37	9	1	82	-	-	-	-	-
8	37	18	94	28	-	-	-	-	-
8	37	4	82	72	-	-	-	-	-
8	37	20	41	59	-	-	-	-	-
8	37	8	100	77	-	-	-	-	-
8	37	3	1	57	-	-	-	-	-
8	37	18	96	7	-	-	-	-	-
8	37	26	57	82	-	-	-	-	-
8	37	21	47	38	-	-	-	-	-
8	37	21	68	89	-	-	-	-	-
8	37	8	16	36	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
9	38	0	69	63	9	38	24	37	37
9	38	12	3	35	9	38	13	21	91
9	38	5	71	79	9	38	14	67	95
9	38	8	1	47	9	38	14	61	15
9	38	12	11	15	-	-	-	-	-
9	38	18	87	23	-	-	-	-	-
9	38	12	37	33	-	-	-	-	-
9	38	11	87	29	-	-	-	-	-
9	38	19	35	81	-	-	-	-	-
9	38	23	55	71	-	-	-	-	-
9	38	8	41	51	-	-	-	-	-
9	38	25	93	9	-	-	-	-	-
9	38	1	11	49	-	-	-	-	-
9	38	5	75	89	-	-	-	-	-
9	38	17	75	69	-	-	-	-	-
9	38	13	97	95	-	-	-	-	-
9	38	9	15	14	-	-	-	-	-
9	38	13	63	95	-	-	-	-	-
9	38	19	47	41	-	-	-	-	-
9	38	5	45	41	-	-	-	-	-
9	38	26	89	43	-	-	-	-	-
9	38	9	45	59	-	-	-	-	-
9	38	20	95	23	-	-	-	-	-
9	38	21	19	83	-	-	-	-	-
9	38	8	71	69	-	-	-	-	-
9	38	12	27	19	-	-	-	-	-
9	38	13	17	57	-	-	-	-	-
9	38	12	93	15	-	-	-	-	-
9	38	4	59	29	-	-	-	-	-
9	38	19	35	39	-	-	-	-	-
9	38	25	33	51	-	-	-	-	-
9	38	7	61	21	-	-	-	-	-
9	38	3	89	53	-	-	-	-	-
9	38	2	33	85	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
10	39	0	9	35	10	39	6	75	73
10	39	5	43	19	10	39	24	93	49
10	39	24	79	35	10	39	19	41	87
10	39	3	93	7	10	39	4	97	73
10	39	20	13	35	10	39	7	45	29
10	39	26	67	13	-	-	-	-	-
10	39	23	31	77	-	-	-	-	-
10	39	15	81	7	-	-	-	-	-
10	39	3	27	49	-	-	-	-	-
10	39	20	27	35	-	-	-	-	-
10	39	16	69	23	-	-	-	-	-
10	39	9	31	51	-	-	-	-	-
10	39	21	27	27	-	-	-	-	-
10	39	3	15	83	-	-	-	-	-
10	39	24	7	35	-	-	-	-	-
10	39	14	53	25	-	-	-	-	-
10	39	6	75	13	-	-	-	-	-
10	39	6	47	49	-	-	-	-	-
10	39	13	25	33	-	-	-	-	-
10	39	5	1	23	-	-	-	-	-
10	39	3	45	11	-	-	-	-	-
10	39	3	1	47	-	-	-	-	-
10	39	20	93	15	-	-	-	-	-
10	39	16	41	9	-	-	-	-	-
10	39	22	75	55	-	-	-	-	-
10	39	10	3	1	-	-	-	-	-
10	39	12	51	67	-	-	-	-	-
10	39	20	57	91	-	-	-	-	-
10	39	24	21	97	-	-	-	-	-
10	39	6	55	13	-	-	-	-	-
10	39	1	3	71	-	-	-	-	-
10	39	2	37	19	-	-	-	-	-
10	39	13	73	21	-	-	-	-	-
10	39	7	19	19	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
11	39	0	39	19	11	39	3	93	55
11	39	18	79	19	11	39	7	5	97
11	39	16	41	79	11	39	15	81	11
11	39	22	25	31	11	39	10	7	53
11	39	24	63	93	11	39	2	7	41
11	39	3	33	5	-	-	-	-	-
11	39	19	69	17	-	-	-	-	-
11	39	6	57	73	-	-	-	-	-
11	39	6	53	75	-	-	-	-	-
11	39	6	1	1	-	-	-	-	-
11	39	12	79	73	-	-	-	-	-
11	39	18	59	5	-	-	-	-	-
11	39	16	1	37	-	-	-	-	-
11	39	72	41	31	-	-	-	-	-
11	39	7	23	73	-	-	-	-	-
11	39	16	37	27	-	-	-	-	-
11	39	23	85	93	-	-	-	-	-
11	39	4	93	13	-	-	-	-	-
11	39	22	85	45	-	-	-	-	-
11	39	23	49	91	-	-	-	-	-
11	39	7	55	43	-	-	-	-	-
11	39	11	83	29	-	-	-	-	-
11	39	11	93	49	-	-	-	-	-
11	39	1	87	23	-	-	-	-	-
11	39	22	31	23	-	-	-	-	-
11	39	16	19	97	-	-	-	-	-
11	39	15	41	9	-	-	-	-	-
11	39	7	83	61	-	-	-	-	-
11	39	5	9	7	-	-	-	-	-
11	39	22	13	13	-	-	-	-	-
11	39	9	43	37	-	-	-	-	-
11	39	10	13	61	-	-	-	-	-
11	39	11	71	51	-	-	-	-	-
11	39	9	45	93	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
12	44	0	14	68	12	44	14	43	62
12	44	8	73	2	12	44	14	73	1
12	44	24	13	47	12	44	18	17	32
12	44	9	37	44	12	44	24	87	79
12	44	19	34	63	12	44	4	12	24
12	44	9	58	98	12	44	8	48	53
12	44	18	33	42	12	44	13	48	23
12	44	9	18	98	12	44	4	7	37
12	44	14	24	79	12	44	14	98	77
12	44	3	17	28	12	44	18	34	12
12	44	14	72	67	-	-	-	-	-
12	44	8	78	63	-	-	-	-	-
12	44	8	42	48	-	-	-	-	-
12	44	13	1	2	-	-	-	-	-
12	44	18	2	28	-	-	-	-	-
12	44	4	32	82	-	-	-	-	-
12	44	24	97	38	-	-	-	-	-
12	44	14	39	53	-	-	-	-	-
12	44	8	87	1	-	-	-	-	-
12	44	18	42	77	-	-	-	-	-
12	44	13	83	27	-	-	-	-	-
12	44	2	79	92	-	-	-	-	-
12	44	9	22	39	-	-	-	-	-
12	44	18	58	32	-	-	-	-	-
12	44	3	53	84	-	-	-	-	-
12	44	24	38	37	-	-	-	-	-
12	44	8	63	59	-	-	-	-	-
12	44	24	42	88	-	-	-	-	-
12	44	14	32	88	-	-	-	-	-
12	44	13	38	23	-	-	-	-	-
12	44	24	63	32	-	-	-	-	-
12	44	23	22	73	-	-	-	-	-
12	44	9	88	94	-	-	-	-	-
12	44	13	58	78	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	**NumDem**	**Demand**	**X**	**Y**	**ID**	**NumDem**	**Demand**	**X**	**Y**
13	45	0	31	73	13	45	22	7	81
13	45	19	11	67	13	45	17	96	88
13	45	2	52	96	13	45	22	2	35
13	45	12	81	29	13	45	17	32	94
13	45	20	97	62	13	45	8	95	94
13	45	6	71	5	13	45	23	9	11
13	45	17	6	56	13	45	5	96	16
13	45	8	48	50	13	45	3	90	68
13	45	14	91	17	13	45	18	33	31
13	45	2	49	68	13	45	12	6	59
13	45	8	85	29	-	-	-	-	-
13	45	7	74	98	-	-	-	-	-
13	45	22	56	37	-	-	-	-	-
13	45	14	13	81	-	-	-	-	-
13	45	17	66	80	-	-	-	-	-
13	45	23	96	55	-	-	-	-	-
13	45	15	36	17	-	-	-	-	-
13	45	21	32	23	-	-	-	-	-
13	45	2	6	13	-	-	-	-	-
13	45	24	64	30	-	-	-	-	-
13	45	10	87	5	-	-	-	-	-
13	45	20	75	61	-	-	-	-	-
13	45	6	40	72	-	-	-	-	-
13	45	21	1	44	-	-	-	-	-
13	45	10	60	95	-	-	-	-	-
13	45	6	27	49	-	-	-	-	-
13	45	13	15	33	-	-	-	-	-
13	45	21	46	53	-	-	-	-	-
13	45	24	28	43	-	-	-	-	-
13	45	11	3	9	-	-	-	-	-
13	45	16	1	100	-	-	-	-	-
13	45	8	53	46	-	-	-	-	-
13	45	11	98	8	-	-	-	-	-
13	45	11	6	25	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y
14	45	0	61	99
14	45	14	95	7
14	45	1	45	87
14	45	16	15	47
14	45	23	39	75
14	45	12	55	23
14	45	6	29	71
14	45	5	87	79
14	45	1	75	63
14	45	13	65	61
14	45	20	73	35
14	45	14	17	35
14	45	18	39	99
14	45	7	75	77
14	45	8	49	37
14	45	21	85	31
14	45	8	89	71
14	45	24	89	43
14	45	20	79	81
14	45	19	45	5
14	45	13	93	69
14	45	3	49	69
14	45	26	63	25
14	45	17	93	33
14	45	22	39	45
14	45	8	89	33
14	45	16	47	77
14	45	20	29	19
14	45	12	13	65
14	45	22	33	9
14	45	20	63	9
14	45	12	41	13
14	45	14	67	75
14	45	25	41	27

ID	NumDem	Demand	X	Y
14	45	17	49	77
14	45	19	57	81
14	45	20	45	5
14	45	15	83	7
14	45	2	81	61
14	45	9	57	81
14	45	10	93	89
14	45	6	17	13
14	45	11	89	27
14	45	21	7	25
14	45	24	35	35
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
15	46	0	75	55	15	46	7	9	91
15	46	12	7	75	15	46	18	11	27
15	46	26	77	1	15	46	3	59	41
15	46	1	51	25	15	46	23	67	1
15	46	20	81	25	15	46	1	77	39
15	46	2	59	37	15	46	17	47	29
15	46	13	93	45	15	46	13	3	89
15	46	20	43	21	15	46	6	33	87
15	46	7	35	53	15	46	22	17	45
15	46	10	77	63	15	46	20	91	41
15	46	15	37	13	15	46	21	23	3
15	46	7	37	51	15	46	2	97	61
15	46	24	27	31	-	-	-	-	-
15	46	10	95	31	-	-	-	-	-
15	46	12	87	43	-	-	-	-	-
15	46	23	23	65	-	-	-	-	-
15	46	13	9	51	-	-	-	-	-
15	46	19	73	81	-	-	-	-	-
15	46	9	3	1	-	-	-	-	-
15	46	12	41	61	-	-	-	-	-
15	46	6	29	81	-	-	-	-	-
15	46	9	51	95	-	-	-	-	-
15	46	22	49	25	-	-	-	-	-
15	46	18	81	53	-	-	-	-	-
15	46	19	7	51	-	-	-	-	-
15	46	20	21	5	-	-	-	-	-
15	46	24	91	35	-	-	-	-	-
15	46	10	17	81	-	-	-	-	-
15	46	4	61	69	-	-	-	-	-
15	46	20	27	97	-	-	-	-	-
15	46	15	83	23	-	-	-	-	-
15	46	13	21	93	-	-	-	-	-
15	46	12	59	31	-	-	-	-	-
15	46	3	27	53	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y
16	48	0	47	5
16	48	20	1	19
16	48	14	97	35
16	48	5	23	79
16	48	11	77	87
16	48	22	3	9
16	48	25	5	27
16	48	2	41	53
16	48	18	51	87
16	48	10	67	73
16	48	26	89	45
16	48	14	71	99
16	48	22	11	1
16	48	9	85	85
16	48	11	57	11
16	48	18	57	85
16	48	24	71	33
16	48	15	61	13
16	48	23	39	15
16	48	16	13	59
16	48	14	43	99
16	48	8	87	73
16	48	5	11	37
16	48	12	21	11
16	48	8	77	81
16	48	16	3	63
16	48	12	47	95
16	48	15	53	75
16	48	9	73	55
16	48	2	81	71
16	48	10	89	75
16	48	2	11	9
16	48	3	27	37
16	48	20	95	59

ID	NumDem	Demand	X	Y
16	48	3	63	63
16	48	13	37	21
16	48	25	33	47
16	48	23	23	63
16	48	8	13	55
16	48	16	47	93
16	48	9	45	43
16	48	14	83	7
16	48	4	69	91
16	48	13	13	11
16	48	7	37	15
16	48	16	53	59
16	48	18	97	83
16	48	16	75	31
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
17	53	0	24	63	17	53	16	18	8
17	53	2	35	60	17	53	23	53	86
17	53	12	79	46	17	53	2	88	51
17	53	14	3	45	17	53	18	77	51
17	53	2	42	50	17	53	22	58	89
17	53	17	3	40	17	53	1	12	44
17	53	20	29	96	17	53	8	70	88
17	53	2	47	30	17	53	4	36	17
17	53	26	54	77	17	53	26	85	23
17	53	7	36	30	17	53	21	93	30
17	53	24	83	86	17	53	2	68	67
17	53	23	30	6	17	53	15	71	34
17	53	13	55	29	17	53	25	56	73
17	53	25	13	2	17	53	22	37	37
17	53	20	1	19	17	53	19	78	20
17	53	3	98	1	17	53	3	88	69
17	53	18	75	10	17	53	3	95	42
17	53	23	39	23	17	53	5	44	71
17	53	6	62	91	17	53	13	18	1
17	53	2	96	9	-	-	-	-	-
17	53	13	27	87	-	-	-	-	-
17	53	22	14	16	-	-	-	-	-
17	53	3	52	49	-	-	-	-	-
17	53	6	95	21	-	-	-	-	-
17	53	7	30	6	-	-	-	-	-
17	53	1	18	40	-	-	-	-	-
17	53	18	82	90	-	-	-	-	-
17	53	18	50	79	-	-	-	-	-
17	53	10	48	49	-	-	-	-	-
17	53	2	82	73	-	-	-	-	-
17	53	9	64	62	-	-	-	-	-
17	53	10	34	78	-	-	-	-	-
17	53	8	83	6	-	-	-	-	-
17	53	30	3	77	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
18	54	0	61	5	18	54	2	31	13
18	54	24	85	53	18	54	2	69	33
18	54	9	17	57	18	54	4	91	47
18	54	15	49	93	18	54	13	13	69
18	54	17	69	11	18	54	18	65	75
18	54	2	87	15	18	54	9	91	27
18	54	19	49	39	18	54	19	9	85
18	54	10	87	23	18	54	3	15	19
18	54	17	19	83	18	54	14	7	37
18	54	20	69	87	18	54	19	61	11
18	54	16	69	43	18	54	21	59	83
18	54	8	49	67	18	54	4	85	69
18	54	12	17	61	18	54	6	15	29
18	54	3	45	61	18	54	22	1	13
18	54	23	21	53	18	54	13	1	83
18	54	4	71	37	18	54	10	85	31
18	54	23	53	23	18	54	18	95	25
18	54	20	77	63	18	54	5	5	33
18	54	2	89	7	18	54	9	51	11
18	54	19	21	83	18	54	36	51	85
18	54	2	77	25	-	-	-	-	-
18	54	23	85	95	-	-	-	-	-
18	54	23	43	93	-	-	-	-	-
18	54	5	75	25	-	-	-	-	-
18	54	12	1	43	-	-	-	-	-
18	54	15	7	7	-	-	-	-	-
18	54	9	81	69	-	-	-	-	-
18	54	13	23	57	-	-	-	-	-
18	54	18	81	15	-	-	-	-	-
18	54	16	77	35	-	-	-	-	-
18	54	7	49	3	-	-	-	-	-
18	54	6	21	93	-	-	-	-	-
18	54	2	41	37	-	-	-	-	-
18	54	8	71	91	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
19	60	0	27	93	19	60	9	31	61
19	60	16	33	27	19	60	5	59	69
19	60	2	29	39	19	60	9	29	15
19	60	7	7	81	19	60	2	93	83
19	60	11	1	59	19	60	14	63	97
19	60	9	49	9	19	60	19	65	57
19	60	17	21	53	19	60	11	15	69
19	60	21	79	89	19	60	21	31	97
19	60	23	81	83	19	60	20	57	9
19	60	10	85	11	19	60	21	85	37
19	60	6	45	9	19	60	18	21	29
19	60	19	7	65	19	60	48	53	11
19	60	18	95	27	19	60	1	15	77
19	60	20	81	85	19	60	17	41	69
19	60	13	37	81	19	60	42	45	17
19	60	5	69	69	19	60	2	13	25
19	60	11	15	95	19	60	4	63	57
19	60	24	89	75	19	60	24	95	5
19	60	2	33	93	19	60	18	55	91
19	60	3	57	83	19	60	21	3	31
19	60	1	11	95	19	60	11	47	7
19	60	5	3	57	19	60	9	61	69
19	60	20	45	11	19	60	18	85	35
19	60	23	43	61	19	60	22	89	81
19	60	24	35	43	19	60	9	45	47
19	60	18	19	83	19	60	23	65	93
19	60	19	83	69	-	-	-	-	-
19	60	2	85	77	-	-	-	-	-
19	60	17	19	39	-	-	-	-	-
19	60	17	83	87	-	-	-	-	-
19	60	9	1	13	-	-	-	-	-
19	60	11	15	39	-	-	-	-	-
19	60	2	83	17	-	-	-	-	-
19	60	6	41	97	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
20	61	0	61	37	20	61	12	79	61
20	61	23	93	57	20	61	8	73	73
20	61	17	15	67	20	61	18	35	95
20	61	12	23	43	20	61	12	5	43
20	61	6	53	5	20	61	72	19	45
20	61	22	13	75	20	61	2	71	39
20	61	3	29	73	20	61	5	35	63
20	61	24	47	37	20	61	14	27	73
20	61	24	23	71	20	61	11	31	21
20	61	11	67	45	20	61	19	47	9
20	61	7	21	49	20	61	16	87	45
20	61	12	93	43	20	61	19	1	49
20	61	8	67	13	20	61	3	1	77
20	61	14	69	25	20	61	12	63	73
20	61	20	53	35	20	61	10	79	71
20	61	16	25	39	20	61	20	21	55
20	61	16	85	69	20	61	7	65	23
20	61	4	81	27	20	61	13	65	47
20	61	9	77	79	20	61	16	97	23
20	61	18	45	43	20	61	23	23	71
20	61	14	31	75	20	61	22	5	81
20	61	10	49	99	20	61	18	53	27
20	61	19	63	9	20	61	6	57	85
20	61	22	47	37	20	61	12	89	23
20	61	19	33	47	20	61	27	51	65
20	61	9	39	69	20	61	9	13	49
20	61	18	49	3	20	61	15	91	41
20	61	2	49	87	-	-	-	-	-
20	61	18	87	39	-	-	-	-	-
20	61	11	37	91	-	-	-	-	-
20	61	19	19	33	-	-	-	-	-
20	61	18	97	35	-	-	-	-	-
20	61	15	31	5	-	-	-	-	-
20	61	4	35	25	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
21	62	0	2	48	21	62	2	58	41
21	62	26	64	71	21	62	18	93	41
21	62	18	33	83	21	62	19	8	74
21	62	16	77	89	21	62	18	13	87
21	62	8	96	61	21	62	4	34	68
21	62	7	53	55	21	62	16	84	97
21	62	11	38	10	21	62	26	35	20
21	62	4	7	40	21	62	3	96	79
21	62	9	7	1	21	62	23	73	87
21	62	9	73	68	21	62	16	21	1
21	62	16	4	24	21	62	10	28	93
21	62	7	82	56	21	62	9	55	94
21	62	6	30	56	21	62	21	56	84
21	62	1	38	62	21	62	24	68	99
21	62	2	42	53	21	62	24	30	53
21	62	22	5	49	21	62	19	85	49
21	62	23	1	85	21	62	12	85	2
21	62	4	94	4	21	62	16	0	30
21	62	3	50	70	21	62	6	12	71
21	62	20	35	69	21	62	2	26	55
21	62	7	14	41	21	62	2	72	75
21	62	1	94	73	21	62	2	71	21
21	62	2	23	96	21	62	23	33	49
21	62	12	96	88	21	62	16	51	2
21	62	22	74	94	21	62	26	93	7
21	62	6	68	52	21	62	21	70	22
21	62	11	73	22	21	62	2	54	53
21	62	12	92	96	21	62	7	26	17
21	62	2	4	48	-	-	-	-	-
21	62	14	24	10	-	-	-	-	-
21	62	14	52	72	-	-	-	-	-
21	62	2	61	46	-	-	-	-	-
21	62	9	38	9	-	-	-	-	-
21	62	20	90	15	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y
22	63	0	91	93
22	63	4	7	5
22	63	18	27	91
22	63	22	21	47
22	63	14	33	11
22	63	5	19	99
22	63	9	59	25
22	63	7	69	79
22	63	20	11	73
22	63	19	21	35
22	63	7	59	45
22	63	18	99	81
22	63	20	71	47
22	63	2	53	83
22	63	11	87	1
22	63	10	79	67
22	63	5	59	65
22	63	21	67	21
22	63	20	27	1
22	63	1	81	93
22	63	15	59	89
22	63	15	95	23
22	63	14	73	25
22	63	19	41	25
22	63	22	3	65
22	63	21	59	83
22	63	22	83	97
22	63	6	5	83
22	63	24	71	35
22	63	14	37	97
22	63	14	29	93
22	63	2	19	21
22	63	15	83	13
22	63	21	97	67

ID	NumDem	Demand	X	Y
22	63	15	31	83
22	63	6	65	17
22	63	23	19	63
22	63	14	59	23
22	63	26	17	25
22	63	15	45	27
22	63	15	89	7
22	63	23	19	41
22	63	7	23	39
22	63	22	23	21
22	63	26	83	61
22	63	20	11	93
22	63	3	17	11
22	63	2	35	11
22	63	15	21	59
22	63	2	21	69
22	63	21	71	13
22	63	12	63	13
22	63	4	49	21
22	63	10	83	31
22	63	23	41	97
22	63	4	85	15
22	63	24	77	73
22	63	17	57	1
22	63	2	83	11
22	63	20	1	75
22	63	18	45	71
22	63	19	41	55
22	63	8	45	13
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y
23	63	0	76	75
23	63	11	60	14
23	63	20	46	5
23	63	26	91	14
23	63	17	70	95
23	63	5	86	31
23	63	17	20	0
23	63	1	41	55
23	63	15	21	15
23	63	16	46	1
23	63	22	95	45
23	63	2	16	89
23	63	22	41	1
23	63	25	60	94
23	63	20	55	25
23	63	20	71	41
23	63	21	39	35
23	63	26	61	70
23	63	12	80	36
23	63	5	100	26
23	63	11	65	85
23	63	5	40	51
23	63	21	19	71
23	63	17	34	50
23	63	12	36	61
23	63	12	69	50
23	63	15	61	94
23	63	21	19	11
23	63	12	51	91
23	63	2	61	54
23	63	17	76	90
23	63	16	41	75
23	63	15	35	100
23	63	26	1	40

ID	NumDem	Demand	X	Y
23	63	12	15	91
23	63	22	21	11
23	63	16	79	81
23	63	21	34	36
23	63	16	74	99
23	63	2	75	14
23	63	6	65	54
23	63	2	55	10
23	63	6	100	6
23	63	10	99	91
23	63	11	25	86
23	63	1	75	16
23	63	63	30	45
23	63	10	21	85
23	63	21	75	80
23	63	7	71	35
23	63	1	56	81
23	63	6	25	76
23	63	26	85	76
23	63	17	60	34
23	63	10	41	44
23	63	25	6	55
23	63	25	60	54
23	63	25	40	96
23	63	16	20	71
23	63	5	94	45
23	63	22	31	41
23	63	17	40	49
23	63	6	56	80
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	**NumDem**	**Demand**	**X**	**Y**
24	64	0	97	33
24	64	10	57	81
24	64	15	1	33
24	64	23	55	57
24	64	23	29	37
24	64	24	21	39
24	64	17	93	37
24	64	1	5	91
24	64	4	25	11
24	64	2	47	37
24	64	5	87	25
24	64	18	67	65
24	64	9	71	89
24	64	8	67	15
24	64	23	45	79
24	64	13	71	57
24	64	4	29	1
24	64	18	59	79
24	64	16	93	83
24	64	26	47	41
24	64	16	51	41
24	64	4	23	93
24	64	23	87	95
24	64	8	39	45
24	64	26	45	7
24	64	16	85	51
24	64	5	35	93
24	64	2	47	79
24	64	21	59	91
24	64	23	83	51
24	64	8	49	65
24	64	5	21	55
24	64	8	51	21
24	64	26	69	43

ID	**NumDem**	**Demand**	**X**	**Y**
24	64	12	37	41
24	64	8	37	95
24	64	3	5	71
24	64	8	37	47
24	64	19	83	73
24	64	16	17	71
24	64	2	5	71
24	64	3	81	17
24	64	17	59	33
24	64	7	63	87
24	64	5	21	77
24	64	8	71	51
24	64	4	21	17
24	64	12	9	7
24	64	19	65	43
24	64	19	25	63
24	64	26	13	57
24	64	24	47	43
24	64	5	77	9
24	64	8	57	55
24	64	22	21	33
24	64	9	27	59
24	64	18	83	9
24	64	19	63	69
24	64	15	9	35
24	64	5	25	55
24	64	11	33	3
24	64	12	53	11
24	64	54	51	49
24	64	8	9	23
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
25	65	0	25	51	25	65	18	9	77
25	65	12	35	7	25	65	5	61	87
25	65	24	93	75	25	65	19	59	91
25	65	16	53	95	25	65	15	63	79
25	65	7	51	81	25	65	8	97	67
25	65	9	51	55	25	65	6	9	45
25	65	20	1	67	25	65	14	93	21
25	65	10	9	23	25	65	13	83	71
25	65	18	75	7	25	65	5	95	57
25	65	26	15	97	25	65	24	31	69
25	65	17	79	5	25	65	25	77	17
25	65	2	9	19	25	65	2	63	57
25	65	11	39	1	25	65	8	3	63
25	65	9	47	1	25	65	14	11	69
25	65	12	33	97	25	65	2	7	9
25	65	11	27	83	25	65	13	37	65
25	65	12	83	79	25	65	10	75	83
25	65	23	17	59	25	65	6	15	53
25	65	7	47	19	25	65	6	69	5
25	65	1	57	9	25	65	24	69	27
25	65	26	87	41	25	65	21	5	19
25	65	10	55	25	25	65	20	49	31
25	65	9	21	91	25	65	24	77	17
25	65	22	21	13	25	65	4	15	7
25	65	21	67	1	25	65	19	91	39
25	65	17	59	21	25	65	14	79	17
25	65	2	1	75	25	65	23	67	75
25	65	15	33	85	25	65	2	93	51
25	65	16	25	21	25	65	16	25	33
25	65	14	45	29	25	65	23	9	19
25	65	23	63	77	25	65	14	3	65
25	65	24	1	77	-	-	-	-	-
25	65	2	77	41	-	-	-	-	-
25	65	12	35	11	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y
26	69	0	59	44
26	69	2	9	23
26	69	1	84	68
26	69	6	36	93
26	69	9	87	9
26	69	16	34	16
26	69	5	40	98
26	69	3	72	43
26	69	9	0	60
26	69	12	64	90
26	69	1	11	8
26	69	1	46	7
26	69	18	33	54
26	69	10	23	84
26	69	5	67	18
26	69	5	34	93
26	69	9	19	25
26	69	16	14	9
26	69	12	70	64
26	69	6	58	50
26	69	6	5	78
26	69	20	95	39
26	69	23	38	54
26	69	39	50	73
26	69	17	48	46
26	69	8	64	4
26	69	2	39	28
26	69	13	79	30
26	69	17	61	36
26	69	3	28	50
26	69	2	51	91
26	69	7	68	59
26	69	23	14	9
26	69	10	94	87

ID	NumDem	Demand	X	Y
26	69	7	68	29
26	69	12	12	91
26	69	20	2	22
26	69	13	25	16
26	69	21	11	57
26	69	25	3	51
26	69	5	13	52
26	69	4	9	76
26	69	13	58	18
26	69	12	40	39
26	69	23	32	89
26	69	19	9	92
26	69	10	36	14
26	69	7	82	13
26	69	15	10	25
26	69	5	96	97
26	69	15	20	21
26	69	13	40	91
26	69	30	33	31
26	69	15	72	74
26	69	7	41	24
26	69	9	90	20
26	69	23	4	44
26	69	8	54	22
26	69	5	43	59
26	69	8	3	70
26	69	25	94	16
26	69	12	94	54
26	69	25	14	40
26	69	24	37	0
26	69	8	88	55
26	69	22	80	25
26	69	7	37	64
26	69	24	87	47

Table A.1. Sample Dataset (cont.).

ID	**NumDem**	**Demand**	**X**	**Y**
26	69	18	18	68
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

ID	**NumDem**	**Demand**	**X**	**Y**
27	80	0	92	92
27	80	24	88	58
27	80	22	70	6
27	80	23	57	59
27	80	5	0	98
27	80	11	61	38
27	80	23	65	22
27	80	26	91	52
27	80	9	59	2
27	80	23	3	54
27	80	9	95	38
27	80	14	80	28
27	80	16	66	42
27	80	12	79	74
27	80	2	99	25
27	80	2	20	43
27	80	6	40	3
27	80	20	50	42
27	80	26	97	0
27	80	12	21	19
27	80	15	36	21
27	80	13	100	61
27	80	26	11	85
27	80	17	69	35
27	80	7	69	22
27	80	12	29	35
27	80	4	14	9
27	80	4	50	33
27	80	20	89	17
27	80	10	57	44
27	80	10	57	44
27	80	9	60	25
27	80	2	48	42
27	80	9	17	93

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
27	80	1	21	50	27	80	5	52	82
27	80	2	77	18	27	80	9	46	6
27	80	2	2	4	27	80	9	3	26
27	80	12	63	83	27	80	5	46	80
27	80	14	68	6	27	80	12	94	30
27	80	23	41	95	27	80	2	26	76
27	80	21	48	54	27	80	12	75	92
27	80	13	98	73	27	80	19	57	51
27	80	13	26	38	27	80	6	34	21
27	80	23	69	76	27	80	14	28	80
27	80	3	40	1	27	80	2	59	66
27	80	6	65	41	27	80	2	51	16
27	80	23	14	86	27	80	24	87	11
27	80	11	32	39	-	-	-	-	-
27	80	2	14	24	-	-	-	-	-
27	80	7	96	5	-	-	-	-	-
27	80	13	82	98	-	-	-	-	-
27	80	10	23	85	-	-	-	-	-
27	80	3	63	69	-	-	-	-	-
27	80	6	87	19	-	-	-	-	-
27	80	13	56	75	-	-	-	-	-
27	80	2	15	63	-	-	-	-	-
27	80	14	10	45	-	-	-	-	-
27	80	7	7	30	-	-	-	-	-
27	80	21	31	11	-	-	-	-	-
27	80	7	36	93	-	-	-	-	-
27	80	22	50	31	-	-	-	-	-
27	80	13	49	52	-	-	-	-	-
27	80	22	39	10	-	-	-	-	-
27	80	18	76	40	-	-	-	-	-
27	80	22	83	34	-	-	-	-	-
27	80	6	33	51	-	-	-	-	-
27	80	2	0	15	-	-	-	-	-
27	80	11	52	82	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
28	35	0	78	95	28	35	13	14	25
28	35	12	93	43	-	-	-	-	-
28	35	3	57	4	-	-	-	-	-
28	35	2	2	80	-	-	-	-	-
28	35	13	10	17	-	-	-	-	-
28	35	17	31	8	-	-	-	-	-
28	35	12	10	87	-	-	-	-	-
28	35	1	97	50	-	-	-	-	-
28	35	26	16	93	-	-	-	-	-
28	35	13	98	48	-	-	-	-	-
28	35	15	103	47	-	-	-	-	-
28	35	20	38	9	-	-	-	-	-
28	35	20	100	51	-	-	-	-	-
28	35	3	60	11	-	-	-	-	-
28	35	3	15	19	-	-	-	-	-
28	35	12	39	15	-	-	-	-	-
28	35	25	102	47	-	-	-	-	-
28	35	2	103	59	-	-	-	-	-
28	35	15	10	82	-	-	-	-	-
28	35	24	39	9	-	-	-	-	-
28	35	2	97	52	-	-	-	-	-
28	35	7	18	97	-	-	-	-	-
28	35	15	32	13	-	-	-	-	-
28	35	2	96	45	-	-	-	-	-
28	35	13	11	21	-	-	-	-	-
28	35	9	15	96	-	-	-	-	-
28	35	12	10	81	-	-	-	-	-
28	35	26	13	24	-	-	-	-	-
28	35	17	0	8	-	-	-	-	-
28	35	26	103	59	-	-	-	-	-
28	35	9	33	11	-	-	-	-	-
28	35	14	13	94	-	-	-	-	-
28	35	9	63	5	-	-	-	-	-
28	35	25	3	87	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
29	38	0	64	75	29	38	1	46	4
29	38	18	16	97	29	38	21	0	85
29	38	10	2	79	29	38	17	51	2
29	38	16	28	79	29	38	2	5	80
29	38	12	16	3	29	38	1	46	4
29	38	21	39	20	29	38	21	0	85
29	38	23	35	65	29	38	17	51	2
29	38	15	43	21	29	38	2	5	80
29	38	25	44	0	-	-	-	-	-
29	38	3	37	67	-	-	-	-	-
29	38	6	17	100	-	-	-	-	-
29	38	7	34	80	-	-	-	-	-
29	38	4	45	7	-	-	-	-	-
29	38	20	11	83	-	-	-	-	-
29	38	25	35	88	-	-	-	-	-
29	38	20	21	98	-	-	-	-	-
29	38	12	41	72	-	-	-	-	-
29	38	3	51	0	-	-	-	-	-
29	38	12	17	9	-	-	-	-	-
29	38	14	36	66	-	-	-	-	-
29	38	26	20	105	-	-	-	-	-
29	38	9	35	85	-	-	-	-	-
29	38	22	37	75	-	-	-	-	-
29	38	20	17	11	-	-	-	-	-
29	38	9	46	6	-	-	-	-	-
29	38	13	40	23	-	-	-	-	-
29	38	21	43	73	-	-	-	-	-
29	38	10	40	21	-	-	-	-	-
29	38	8	29	85	-	-	-	-	-
29	38	5	45	6	-	-	-	-	-
29	38	14	0	24	-	-	-	-	-
29	38	7	42	66	-	-	-	-	-
29	38	21	30	81	-	-	-	-	-
29	38	20	17	9	-	-	-	-	-

Table A.1. Sample Dataset (cont.).

ID	NumDem	Demand	X	Y	ID	NumDem	Demand	X	Y
30	39	0	37	21	30	39	23	16	4
30	39	14	77	57	30	39	8	46	48
30	39	16	97	79	30	39	6	44	42
30	39	18	39	33	30	39	2	100	82
30	39	20	45	47	30	39	15	52	54
30	39	1	85	23	-	-	-	-	-
30	39	12	7	1	-	-	-	-	-
30	39	13	16	6	-	-	-	-	-
30	39	18	21	13	-	-	-	-	-
30	39	9	21	7	-	-	-	-	-
30	39	15	12	6	-	-	-	-	-
30	39	8	92	24	-	-	-	-	-
30	39	10	92	32	-	-	-	-	-
30	39	7	10	8	-	-	-	-	-
30	39	18	19	15	-	-	-	-	-
30	39	14	86	24	-	-	-	-	-
30	39	1	106	84	-	-	-	-	-
30	39	15	48	48	-	-	-	-	-
30	39	7	8	2	-	-	-	-	-
30	39	25	14	2	-	-	-	-	-
30	39	2	14	2	-	-	-	-	-
30	39	3	98	82	-	-	-	-	-
30	39	4	106	80	-	-	-	-	-
30	39	16	98	86	-	-	-	-	-
30	39	15	98	82	-	-	-	-	-
30	39	23	80	62	-	-	-	-	-
30	39	11	82	62	-	-	-	-	-
30	39	21	52	52	-	-	-	-	-
30	39	10	42	42	-	-	-	-	-
30	39	12	25	11	-	-	-	-	-
30	39	9	104	86	-	-	-	-	-
30	39	7	44	38	-	-	-	-	-
30	39	8	78	64	-	-	-	-	-
30	39	4	98	80	-	-	-	-	-

APPENDIX B: SSA CALCULATIONS FOR DATASET 1

Table A.17. SSA Calculations for Dataset 1.

From Node	To Node	Savings	Used	From Node	To Node	Savings	Used
13	2	207,4	1	14	22	167,3	1
13	17	201,8	1	13	14	167,1	0
1	31	199,3	0	1	8	166,8	0
13	31	199,1	0	25	31	166,7	0
17	2	195,8	0	11	26	165,2	1
17	31	195,7	1	14	8	165,1	0
27	31	193,6	1	15	31	164,6	0
1	13	193,4	0	28	31	164,0	0
2	31	189,5	0	15	28	163,7	0
17	27	189,2	0	25	27	163,6	0
13	27	189,1	0	2	32	163,3	0
1	27	187,5	1	21	26	163,1	0
1	17	187,4	0	15	27	162,3	0
22	32	186,3	1	14	17	162,0	0
2	8	184,3	1	21	31	161,7	0
20	32	183,9	1	22	8	161,6	0
13	8	183,6	0	17	22	161,2	0
2	27	182,5	0	13	32	160,7	0
1	2	181,9	0	21	25	160,1	0
1	21	181,5	1	14	32	159,7	0
17	8	180,8	0	27	28	159,6	0
18	20	179,9	1	13	25	159,2	0
18	32	179,6	0	17	25	159,2	0
21	6	178,9	0	15	17	158,0	0
20	22	178,2	0	13	15	157,7	0
1	28	175,4	0	2	20	156,1	0
31	8	174,9	0	13	28	155,8	0
18	22	174,3	0	14	31	155,8	0
1	25	173,5	0	14	20	155,3	0
2	22	173,2	0	17	28	155,2	0
27	8	172,5	0	15	21	155,1	0
26	6	171,0	1	21	27	154,4	0
13	22	170,7	0	22	31	154,2	0
21	28	170,6	0	14	27	153,5	0
14	2	169,6	0	13	21	153,4	0
1	15	169,6	0	13	20	153,4	0
15	25	168,9	0	14	18	153,0	0
25	28	168,6	0	18	2	152,9	0

Table A.2. SSA Calculations for Dataset 1 (cont.).

From Node	To Node	Savings	Used
2	25	152,8	0
32	8	152,6	0
15	2	151,9	0
17	32	151,5	0
1	6	151,4	0
17	21	150,7	0
11	6	150,4	0
22	27	150,2	0
13	18	150,1	0
28	6	149,5	1
2	28	148,4	0
1	14	147,7	0
20	8	147,1	0
30	6	146,7	0
25	8	146,6	0
15	8	146,5	0
1	22	145,9	0
17	20	145,2	0
18	8	144,5	0
31	32	144,5	0
2	21	143,9	0
26	30	142,7	0
17	18	142,3	0
21	30	141,7	1
28	8	141,4	0
27	32	140,9	0
11	21	140,8	0
25	6	139,4	0
20	31	138,3	0
1	32	136,2	0
11	30	135,7	0
18	31	135,6	0
20	27	135,3	0
15	6	135,2	0
31	6	135,0	0
21	8	134,3	0
1	26	134,1	0
18	27	132,8	0
26	28	132,4	0
11	16	131,9	1
16	26	131,3	0

From Node	To Node	Savings	Used
1	20	130,0	0
27	6	129,8	0
13	7	129,8	0
17	7	129,6	0
7	8	129,5	0
2	7	129,5	0
27	7	128,4	0
31	7	128,3	0
1	18	127,3	0
13	6	126,5	0
28	30	126,1	0
17	6	125,4	0
14	7	125,4	0
14	28	125,0	0
1	7	124,8	0
15	22	124,7	0
22	25	124,5	0
16	6	123,4	0
1	30	123,0	0
25	26	122,6	0
15	7	120,6	1
22	7	120,0	0
16	30	119,9	0
25	7	119,7	0
22	28	119,2	0
15	26	118,8	0
25	30	118,7	0
2	6	118,7	0
26	31	117,6	0
11	28	117,1	0
14	21	117,0	0
15	32	116,6	0
25	32	116,2	0
15	30	115,9	0
16	21	115,6	0
1	11	115,6	0
28	7	115,5	0
32	7	114,9	0
20	7	112,8	0
26	27	112,7	0
15	20	112,4	0

Table A.2. SSA Calculations for Dataset 1 (cont.).

From Node	To Node	Savings	Used	From Node	To Node	Savings	Used
14	15	130,8	1	6	8	112,1	0
14	25	130,4	0	20	25	112,0	0
21	22	111,7	0	11	17	93,9	0
18	7	111,7	0	13	4	93,5	0
30	31	111,5	0	3	8	93,4	0
28	32	110,8	0	6	7	93,4	0
15	18	110,5	0	16	25	92,9	0
18	25	110,0	0	13	3	92,7	0
13	26	109,1	0	17	4	92,1	0
11	25	108,7	0	17	3	91,1	0
27	30	108,5	0	15	16	90,6	0
17	26	108,3	0	22	6	90,5	0
21	7	107,4	0	4	7	90,0	0
20	28	106,6	0	31	4	89,4	0
11	15	105,5	0	27	4	89,4	0
18	28	104,6	0	3	7	88,0	0
17	30	104,2	0	3	31	88,0	0
13	30	104,0	0	27	3	87,9	0
21	32	103,0	0	11	2	87,7	0
11	31	101,9	0	23	26	86,4	0
2	26	101,6	0	11	23	86,2	0
16	28	99,5	0	16	23	85,8	0
22	3	98,7	0	16	31	85,7	0
3	32	98,6	0	1	4	85,0	0
20	3	98,4	0	23	30	84,9	0
20	21	98,3	0	23	6	84,7	0
18	3	98,3	0	14	24	84,2	0
2	30	98,3	0	22	24	84,1	0
11	27	98,3	0	24	4	84,0	1
22	4	97,6	0	11	8	83,5	0
32	4	96,9	0	1	3	83,2	0
14	4	96,9	0	24	32	83,2	0
14	3	96,9	0	30	7	83,2	0
14	6	96,8	0	16	27	83,2	0
20	4	96,6	0	2	24	83,0	0
18	4	96,5	0	20	24	82,9	0
18	21	96,2	0	18	24	82,8	0
1	16	96,0	0	32	6	82,7	0
26	8	95,7	0	24	8	82,7	0
3	4	95,0	1	24	3	82,6	0
2	4	94,8	0	14	30	82,2	0

Table A.2. SSA Calculations for Dataset 1 (cont.).

From Node	To Node	Savings	Used
30	8	94,5	0
4	8	94,2	0
2	3	94,2	0
11	13	93,9	0
21	23	81,2	0
24	7	80,7	0
25	4	80,5	0
26	7	80,2	0
10	11	80,0	0
10	16	80,0	1
10	26	79,9	0
24	27	79,4	0
24	31	79,3	0
16	17	79,2	0
20	6	79,1	0
15	3	79,1	0
10	23	79,0	0
13	16	78,7	0
25	3	78,3	1
10	30	77,7	0
18	6	77,5	0
10	6	77,4	0
28	4	77,0	0
1	24	75,9	0
22	30	75,6	0
19	21	75,2	0
22	26	74,8	0
28	3	74,6	0
23	28	74,2	0
19	28	74,2	0
19	6	74,0	0
16	2	73,8	0
10	21	73,8	0
15	24	73,6	0
19	30	73,1	0
19	25	73,0	0
24	25	72,9	0
15	19	72,5	0
1	19	72,5	0
11	7	72,2	0
16	8	71,2	0

From Node	To Node	Savings	Used
13	24	82,1	0
14	26	81,3	0
15	4	81,3	0
17	24	81,3	0
19	31	69,7	0
15	23	69,5	0
19	27	69,2	0
30	32	69,2	0
11	19	68,1	0
26	32	67,5	0
17	19	67,4	0
21	3	67,4	0
10	28	67,0	0
13	19	66,6	0
20	30	66,5	0
16	19	66,0	0
18	30	65,2	0
23	31	65,0	0
19	2	64,8	0
19	8	64,3	0
20	26	64,3	0
11	22	64,3	0
16	7	64,0	0
23	27	63,9	0
21	24	63,8	0
10	25	63,7	0
1	10	63,6	0
19	23	63,6	0
19	7	63,4	1
18	26	62,9	0
10	15	62,6	0
17	23	61,1	0
14	16	60,8	0
13	23	60,2	0
4	6	59,7	0
10	19	59,2	0
14	19	59,2	0
10	31	58,2	0
11	32	57,9	0
2	23	57,3	0
10	27	57,2	0

Table A.2. SSA Calculations for Dataset 1 (cont.).

From Node	To Node	Savings	Used
11	14	71,0	0
1	23	70,8	0
23	25	70,7	0
19	26	70,2	0
24	28	70,0	0
21	4	69,8	0
24	6	55,4	0
11	20	55,4	0
30	9	55,3	0
25	9	54,7	1
19	22	54,7	0
16	22	54,6	0
10	17	54,5	0
15	9	54,4	0
29	31	54,4	0
27	29	54,4	0
1	9	54,3	0
17	29	54,3	0
11	18	54,1	0
13	29	54,1	0
1	29	54,1	0
29	7	54,0	0
15	29	53,9	0
30	4	53,9	0
29	8	53,8	0
2	29	53,8	0
25	29	53,8	0
23	7	53,7	0
10	13	53,7	0
26	9	53,3	0
28	29	53,0	0
31	9	52,4	0
14	29	52,4	0
11	9	52,2	0
27	9	52,1	0
19	32	51,3	0
3	30	51,3	0
23	9	51,2	0
16	9	51,2	0
10	2	51,0	0
24	30	50,9	0

From Node	To Node	Savings	Used
3	6	57,0	0
21	9	56,4	0
19	9	56,3	0
23	8	56,3	0
6	9	55,7	0
28	9	55,6	0
24	29	49,5	1
14	23	49,4	0
16	32	49,1	0
29	4	49,1	0
10	9	49,0	0
2	9	48,9	0
8	9	48,7	0
29	32	48,7	0
7	9	48,7	0
20	29	48,3	0
10	7	48,2	0
18	29	48,0	0
19	4	48,0	0
29	3	47,8	0
19	29	47,3	0
19	24	47,3	0
29	6	47,2	0
16	20	47,1	0
26	3	47,0	0
24	26	46,5	0
16	18	46,2	0
19	3	45,8	0
29	30	45,6	0
14	9	45,1	0
11	4	44,7	0
22	23	44,2	0
10	14	43,8	0
29	9	42,7	0
11	24	42,5	0
26	29	42,3	0
11	3	42,2	0
22	9	41,7	0
16	4	40,3	0
11	29	40,3	0
23	32	40,3	0

Table A.2. SSA Calculations for Dataset 1 (cont.).

From Node	To Node	Savings	Used
17	9	50,8	0
21	29	50,7	0
22	29	50,3	0
19	20	50,2	0
13	9	50,2	0
10	8	50,1	0
18	19	49,6	0
26	4	49,6	0
18	23	38,4	1
18	9	38,1	0
23	29	37,9	0
16	3	37,9	0
3	9	36,8	0
23	4	36,4	0
23	24	35,8	0
10	32	35,4	0
10	29	35,3	0
23	3	34,3	0
10	20	34,2	0
10	18	33,7	0
10	4	32,5	0
10	24	32,2	0
10	3	30,6	0
27	5	28,8	0
31	5	28,8	0
17	5	28,8	0
29	5	28,8	1
13	5	28,8	0
5	7	28,8	0
5	8	28,7	0
2	5	28,7	0
1	5	28,6	0
15	5	28,5	0
25	5	28,4	0
14	5	28,2	0
28	5	28,0	0
24	5	27,6	0
22	5	27,3	0
4	5	27,2	0
21	5	26,8	0
3	5	26,7	0

From Node	To Node	Savings	Used
32	9	39,2	0
20	23	39,0	0
10	22	39,0	0
16	24	38,8	0
24	9	38,6	0
16	29	38,6	0
4	9	38,5	0
20	9	38,5	0
12	9	23,2	0
12	19	23,2	0
12	26	22,9	0
12	23	22,9	0
12	28	22,7	0
11	12	22,7	0
26	5	22,7	0
10	12	22,6	0
12	16	22,6	0
12	25	22,3	0
12	15	22,1	0
1	12	22,0	0
11	5	21,8	0
23	5	21,7	0
16	5	21,2	0
12	31	21,2	0
12	27	21,1	0
10	5	20,6	0
12	17	20,5	0
12	29	20,2	0
12	13	20,2	0
12	7	19,9	0
12	8	19,7	0
12	2	19,7	0
12	14	18,3	0
12	5	17,6	0
12	24	16,9	0
12	22	16,8	0
12	4	16,5	0
12	3	15,8	0
12	32	15,8	0
12	20	15,5	0
12	18	15,4	0

Table A.2. SSA Calculations for Dataset 1 (cont.).

From Node	To Node	Savings	Used
32	5	26,6	0
20	5	26,4	0
18	5	26,3	0
19	5	26,3	0
5	9	25,6	0
5	6	25,1	0
30	5	24,5	0
12	6	23,4	0
12	30	23,4	0
12	21	23,3	0

APPENDIX C: SAMPLE ATM DATA

Table A.18. Sample (Modified) ATM Data.

COORD_X	COORD_Y	SUPPLY_DATE	SUPPLY_AMOUNT_TL	ATM
40,7	29,8	21.05.2012	1,589,958	Adapazarı Kavaklar
36,8	30,6	21.05.2012	985,356	Aksu Lara Taksi Durağı
39,4	29,9	18.05.2012	8,550	Antalya Güzeloba
36,6	30,5	18.05.2012	30,744	Alanya Atatürk Caddesi
36,8	30,6	18.05.2012	2,610,378	Alanya Damlataş Plaj Yolu
36,8	30,8	18.05.2012	32,022	Alanya Greenbeach
41,4	27,9	17.05.2012	3,134,142	Alanya Konaklı Oasis
36,8	30,8	21.05.2012	1,786,896	Alanya Konaklı Pegasus
39,6	27,8	21.05.2012	1,253,646	Alanya Konyaaltı Sealıfe
36,8	30,6	20.05.2012	212,292	Alanya Lara Antalıum
39,1	34,2	18.05.2012	3,545,316	Alanya Obagöl
36,6	31,7	21.05.2012	1,912,788	Antalya Yeni Hal
41,3	41,3	18.05.2012	23,022	Arhavi Belediye
36,5	31,9	19.05.2012	1,348,992	Arnavutköy Akpınar Sanayi Bölgesi
40,9	29	21.05.2012	85,374	Ataşehir Ağaoğlu My Town
40,8	29,3	18.05.2012	148,941	Ataşehir İçerenköy Meydan
41	29	21.05.2012	4,248	Avcılar Karaköy Metraco
41	28,8	17.05.2012	2,141,748	Avcılar Parseller
40,8	29,3	21.05.2012	1,092,222	Avcılar Vestel
41	28,6	17.05.2012	3,061,404	Beylikdüzü Migros
40,9	27,9	21.05.2012	1,993,878	Beylikdüzü Migros
40,1	29	19.05.2012	208,8	Ayvalık Altınova Meydan
40,8	29,4	21.05.2012	1,151,352	Bağcılar Automall Cıty
41,2	29	21.05.2012	1,246,014	Bahçelievler Çalış Cad.

Table A.3. Sample (Modified) ATM Data (cont.).

COORD_X	COORD_Y	SUPPLY_DATE	SUPPLY_AMOUNT_TL	ATM
36,7	36,2	21.05.2012	18,504	Bahçelievler Hizmet Hastanesi
38,6	27,3	17.05.2012	11,637	Bahçelievler Mars Lojistik
38,3	26,8	17.05.2012	7,416	Marmara Ünv. Bahçelievler Kampüsü
40,1	32,9	20.05.2012	851,652	Merter Meydan Avm
38,6	27,3	18.05.2012	1,006,668	212 İstanbul Avm
40,9	29,1	18.05.2012	3,204,054	Atatürk Havalimanı
38,6	27,3	21.05.2012	54,369	Bakırköy Dünya Ticaret Merkezi Metro
38,3	27,1	18.05.2012	555,318	Marmara Forum Avm
39,9	32,7	20.05.2012	668,646	Özdilek İzmir Avm
40,3	27,9	18.05.2012	2,016,036	Bandırma İskele
36,8	30,7	21.05.2012	909,954	Başakşehir Kavacık
41	28,8	21.05.2012	1,292,706	Başakşehir Petrol
39,7	30,4	19.05.2012	2,256,156	Başakşehir Toki Kayaşehir
41	28,9	21.05.2012	95,859	Gaziosmanpaşa Yıldıztabya
40,9	28,8	21.05.2012	999	Kabataş Metro
38,6	27,3	21.05.2012	742,338	Taksim Metro
41	29	18.05.2012	2,507,634	Beylikdüzü Atrium Çarşı
41	29,9	21.05.2012	876,114	Beylikdüzü Optimum Avm
40,8	29,2	18.05.2012	930,798	İdo Kabataş
41	29	21.05.2012	1,364,724	Zincirlikuyu Kanal
39,6	27,9	21.05.2012	21,438	Biga Hastanesi
38,4	27,1	21.05.2012	1,274,328	Bornova Hayat Üçyol
41	29	21.05.2012	1,041,732	Şişli Bahçeşehir Üniversitesi
41	28,8	21.05.2012	1,469,862	Mimaroba Bp
41	29	21.05.2012	9,882	Balıkesir Subay Evleri Tansaş

Table A.3. Sample (Modified) ATM Data (cont.).

COORD_X	COORD_Y	SUPPLY_DATE	SUPPLY_AMOUNT_TL	ATM
39,7	37	21.05.2012	2,004,912	Çankaya Bahçelievler Başkent Ünv.
37	27,4	18.05.2012	627,678	Çankaya Panora Avm
36,8	30,7	21.05.2012	892,296	Çankaya Türk Japon Vakfı
41	28,9	17.05.2012	1,342,962	Çekmeköy Adese
41,3	41,4	18.05.2012	1,254,312	Çerkezköy Hema Endüstri
36,9	30,7	21.05.2012	19,152	Çerkezköy Kapaklı
38,4	27,1	21.05.2012	134,244	İzmir Kent Hastanesi
41	27,8	21.05.2012	16,326	Çorlu Termoteknik
40,1	32,9	18.05.2012	38,772	Çubuk Tav Esenboğa
38,4	27,1	21.05.2012	1,141,092	Derince Belediye
41,1	28,7	21.05.2012	1,104,192	Derince Çayırova Meydan
41,3	41,3	18.05.2012	20,718	Dörtyol Payas Belediye
36,2	29,4	17.05.2012	3,650,508	Elmalı Kalkan Kızıltaş
41,1	28,2	21.05.2012	13,104	Esenler Sapanca Taksi Durağı
41	29,1	21.05.2012	12,744	Esenyurt Kiler Genel Müd.
40,9	28,8	19.05.2012	2,844,864	Esenyurt Torıum Avm
40,2	28,9	17.05.2012	664,002	Etimesgut Metrokent Avm
41,1	28,9	21.05.2012	1,556,838	Fatih Bahçekapı Tivaret Merkezi
40,9	28,7	21.05.2012	381,798	İdo Yenikapı
40,9	28,7	18.05.2012	997,506	Karagümrük Meydan
41	28,9	21.05.2012	647,892	Balıkesir Kipa Avm
36,6	29,1	21.05.2012	1,442,772	Fethiye Lisesi
36,5	29,1	17.05.2012	2,178,846	Fethiye Ovacık Ramos Otel
38,4	27,2	21.05.2012	8,064	Ege Gaziemir Serbest Bölge
39,9	32,7	21.05.2012	1,430,244	Gaziemir Belediye

Table A.3. Sample (Modified) ATM Data (cont.).

COORD_X	COORD_Y	SUPPLY_DATE	SUPPLY_AMOUNT_TL	ATM
41	28,9	19.05.2012	1,237,878	Gebze Ferro Döküm
41	28,9	17.05.2012	1,023,624	Gebze Organize
41	29	18.05.2012	930,06	Nazilli Migros
41	29,1	21.05.2012	1,575,612	Güngören Soğanlı
38,3	27,1	17.05.2012	722,556	Güzelbahçe Meydan
40,2	29,5	21.05.2012	1,507,608	Harmancık Nilüfer Fsm Bulvarı
41	28,9	18.05.2012	8,478	Havran Meydan
41,3	27,9	18.05.2012	8,550	Hendek Park Shop Avm
41	28,9	18.05.2012	1,451,304	Hendek Park Shop Avm
41,3	41,4	18.05.2012	35,514	Hopa Belediyesi
37,9	34,6	17.05.2012	14,004	İnegöl Milli Eğitim
40,9	26,3	21.05.2012	1,319,238	İpsala Yeni Karpuzlu
41	28,5	21.05.2012	13,374	İdo Kadıköy
41	28,8	18.05.2012	1,109,088	Kadıköy Kozyatağı Bayar Cad.
40,9	29,1	18.05.2012	652,734	Kadıköy Kozzy Avm
37	27,4	18.05.2012	1530	İzmir Yeşilyurt
40,9	28,7	21.05.2012	973,332	Karatay Büsan Konya
39,9	32,8	21.05.2012	87,642	Karşıyaka Alaybey Çarşı
39,9	32,7	18.05.2012	1,644,768	Karşıyaka Kaymakamlık
38,3	38,2	21.05.2012	6,462	İstanbul Anadolu Bölgesi Kartal
40,9	29	21.05.2012	6,138	Kemer Bayturan A.Ş.
41,2	27,9	20.05.2012	1,839,006	Kemer Çamyuva
36,8	31	21.05.2012	1,338,102	Kemer Yenimahalle
40,2	27,2	21.05.2012	2,570,004	Kemer Zabıta
40,2	28,9	18.05.2012	997,722	Antalya Otogar

Table A.3. Sample (Modified) ATM Data (cont.).

COORD_X	COORD_Y	SUPPLY_DATE	SUPPLY_AMOUNT_TL	ATM
39,7	30,5	21.05.2012	107,892	Antalya Petrol Ofisi
38,4	27,1	18.05.2012	54,414	Kınık Soma
38,3	27,1	19.05.2012	4,493,592	Kınık Soma
38,7	35,4	21.05.2012	623,052	Kocasinan Kayseri Ted Koleji
40,9	28,8	21.05.2012	74,007	Kocasinan Mkp Bulvarı
37,9	32,5	21.05.2012	3,565,152	Koçarlı İncirliova
36,5	31,9	21.05.2012	8,442	Ege Sağlık
39,9	32,.	17.05.2012	659,574	İzmir Metro Konak
39,5	32,1	18.05.2012	697,788	İzmir Metro Üçyol
38,3	27,1	20.05.2012	26,622	Konak Tariş Genel Müdürlük
40,2	28,9	21.05.2012	2,273,112	Konyaaltı Banio Yapı Market
36,8	28,2	18.05.2012	123,732	Konyaaltı Opet
36,5	31,9	17.05.2012	279,342	Kuşadası Belediye
36,6	30,5	17.05.2012	5,424,066	Kuşadası Ephesıa
36,5	30,5	21.05.2012	22,698	Kuşadası G.Beğendi
36,5	32	21.05.2012	1,734,552	Kuşadası Kırcılar
36,5	31,8	21.05.2012	1,533,744	Kuşadası S Sitesi
39,5	27,1	21.05.2012	7,866	Küçükçekmece Mng Teknik
40,9	28,7	21.05.2012	837	Küçükçekmece Türk Telekom
41	28,8	17.05.2012	9,972	Küçükçekmece Yücel Petrol
41,3	27,5	17.05.2012	1,929,258	Lüleburgaz Zorlulinen
36,6	31,6	21.05.2012	1,660,122	Manavgat Alara Grand Bazaar
41	29	17.05.2012	2,517,246	Marmaris İçmeler
40,9	28,7	21.05.2012	1,661,724	Marmaris Karacanpoınt
41,3	41,4	21.05.2012	99,549	Kayseri 5m

Table A.3. Sample (Modified) ATM Data (cont.).

COORD_X	COORD_Y	SUPPLY_DATE	SUPPLY_AMOUNT_TL	ATM
41	28,9	21.05.2012	1,543,194	Balıkesir Gazi Bulvarı
40,1	26,4	18.05.2012	710,676	Çanakkale Tesco Kipa
36,9	30,7	18.05.2012	1,213,092	Çaykur Kemalpaşa
36,8	30,7	19.05.2012	601,272	Kütahya 30 Ağustos İöo
37,8	27,2	17.05.2012	1,192,554	Kütahya Azerbaycan
37,8	27,2	18.05.2012	1,162,674	Kütahya Azerbaycan
37,8	27,2	19.05.2012	956,322	Kütahya Azerbaycan
37,8	27,2	20.05.2012	58,689	Kütahya Azerbaycan
36,8	30,7	21.05.2012	935,244	Kütahya Azerbaycan
39,4	30	17.05.2012	556,002	Kütahya Bayındırlık
39,4	29,9	18.05.2012	349,452	Kütahya Bayındırlık
39,4	29,9	19.05.2012	96,822	Kütahya Bayındırlık
36,9	30,7	20.05.2012	551,808	Kütahya Bayındırlık
37,8	27,2	21.05.2012	845,532	Kütahya Bayındırlık
40,1	29	18.05.2012	12,942	Malatya Park Avm
38,1	27,3	18.05.2012	284,589	Manisa Zorpet
41	28,8	17.05.2012	1,106,532	Marmara Ereğlisi
40,8	29,3	21.05.2012	12,951	Niğde Meydan
36,8	30,7	18.05.2012	762,264	Petlas A.Ş
40,2	28,9	21.05.2012	1,444,302	Sivas Meydan
39,7	30,5	18.05.2012	3,377,52	Urfa Tarım Reformu
39	27,3	18.05.2012	2,820,906	Vestel Dijital
38,6	27,3	19.05.2012	61,794	Vestel Elektronik
38,4	27	18.05.2012	1,807,038	Vestel Elektronik Dış
38,4	27,2	21.05.2012	3,587,112	Vestel Hıgh End
39	27,3	19.05.2012	28,341	Vestel Klima

Table A.3. Sample (Modified) ATM Data (cont.).

COORD_X	COORD_Y	SUPPLY_DATE	SUPPLY_AMOUNT_TL	ATM
37,1	38,8	17.05.2012	2,979	Mudanya Korteks Berkun
40,7	30,4	18.05.2012	306,234	Mudanya Korteks Berkun
37,7	27,7	19.05.2012	287,154	Mudanya Korteks Berkun
38,7	35,5	20.05.2012	17,298	Mudanya Korteks Berkun
37,9	28,3	21.05.2012	268,956	Mudanya Korteks Berkun
38,7	35,5	18.05.2012	700,524	Mudanya Korteks İplik
36,5	30,5	21.05.2012	1,932,102	Antalya Akdeniz Bölgesi Muratpaşa
41	28,8	21.05.2012	8,856	Antalya Astur
39,8	32,8	21.05.2012	892,638	Antalya Büyükşehir Belediyesi
40,2	28,9	21.05.2012	646,668	Antalya Dedeman Otel
39,2	26,7	21.05.2012	13,986	Antalya Defterdarlık
39,8	32,7	21.05.2012	15,624	Antalya Deva Hastanesi
41	28,9	21.05.2012	1,087,452	Antalya Sampi Kavşağı
38,7	30,2	21.05.2012	836,082	Muratpaşa Antalya Minibüsçüler Odası
36,8	28,2	18.05.2012	909,81	Muratpaşa Elmaslar Market
40,9	28,6	18.05.2012	231,138	Muratpaşa Ventorama Avm
36,8	30,6	18.05.2012	9,918	Nilüfer Uludağ Ünv.
41	28,9	18.05.2012	918,774	Bayrampaşa Şehir Parkı
40,8	29,3	19.05.2012	1,911,672	Eskişehir Stadyumu
40,1	290	21.05.2012	1,692	Bursa Osmangazi Metro
40,1	29	21.05.2012	8,946	Bursa Şehreküstü Metro
40,9	28,8	18.05.2012	442,692	İdo Pendik
40,2	28,9	18.05.2012	527,778	Polatlı Kipa
36,9	30,7	17.05.2012	17,082	Çerkezköy Saray
41	28,7	18.05.2012	156,699	Zekeriyaköy Meydan

Table A.3. Sample (Modified) ATM Data (cont.).

COORD_X	COORD_Y	SUPPLY_DATE	SUPPLY_AMOUNT_TL	ATM
39,3	27,6	21.05.2012	57,132	Soma Savaştepe
39,6	27,8	17.05.2012	1,013,868	Serik Akdeniz Mah.
41,1	28,2	19.05.2012	2,466	Silivri Ekol Ofset
41	28,6	21.05.2012	161,172	Silivri Tesco Kipa
36,8	30,7	18.05.2012	33,408	Sinanpaşa Belediyesi
39,1	27,6	18.05.2012	2,412,792	Soma Belediye
39,1	27,6	21.05.2012	121,604	Soma Belediye
40,9	29	21.05.2012	1,110,906	Astorıa Avm
38,6	27,3	21.05.2012	5,382	Levent Metro
41	28,9	17.05.2012	884,214	Nişantaşı Cıtys
41	28,9	21.05.2012	1,516,104	Okmeydanı Memorıal
40,9	29	17.05.2012	814,878	Profilo Avm
41	28,8	21.05.2012	1,116	Show Tv Ayazağa
40,2	28,8	18.05.2012	920,538	Şişli Anthıll Resıdance
41	28,6	17.05.2012	979,902	Şişli 2
41	28,6	17.05.2012	136,251	Şişli 1
40,6	30,2	21.05.2012	1,114,542	Eskişehir Çarşı Ssk
41,1	28,6	21.05.2012	1,600,632	Eskişehir Kızılcıklı
39,8	32,8	21.05.2012	1,248,156	Torbalı Telekom
36,8	30,7	18.05.2012	866,286	Pendik Aydınlı Yolu
41	28,2	21.05.2012	80,631	Tuzla Kıran
40,8	29,2	19.05.2012	2,100,348	Tuzla Kipaş
41	28,6	21.05.2012	1,190,916	Beykoz İskele
41	28,9	21.05.2012	852,048	Üsküdar Koşuyolu Validebağ Sitesi
40,7	30,6	21.05.2012	740,286	Yenimahalle Carrefour Ümitköy

Table A.3. Sample (Modified) ATM Data (cont.).

COORD_X	COORD_X	COORD_X	COORD_X	COORD_X
36,9	30,6	17.05.2012	27,684	Yenimahalle Onkoloji Hastanesi
36,8	30,6	21.05.2012	24,354	Yenimahalle Onkoloji Hastanesi
40,2	28,9	21.05.2012	24,534	Yenimahalle Onkoloji Hastanesi Bahçe
38,4	27,1	17.05.2012	3,582	Demirören İstiklal Avm

REFERENCES

1. Altınel, K. and T. Oncan, “A New Enhancement of the Clarke and Wright Savings Heuristic for the Capacitated Vehicle Routing Problem”, *Journal of the Operational Research Society*, Vol. 56, pp. 954–961, 2005.

2. Anily, S. and A. Federgruen, “One Warehouse Multiple Retailer Systems with Vehicle Routing Costs”. *Management Science*, Vol. 36, No. 1, pp. 92-114, 1990.

3. Asadi, A. and A. Naserasadi, “A New Hybrid Algorithm for Traveler Salesman Problem Based on Genetic Algorithms and Artificial Neural Networks*”, International Journal of Computer Applications 1,* pp. 2945–3926, 2011.

4. Ballou, R., “A Continued Comparison of Several Popular Algorithms for Vehicle Routing and Scheduling”, *Journal of Business Logistics*, Vol. 11, No. 1, pp. 111-126, 1990.

5. Bard, J. F., H. Jaillet and M. Dror, “Decomposition Approach to the Inventory Routing Problem with Satellite Facilities”, *Transportation Science*, Vol. 32, No. 2, pp. 189-203, 1998.

6. Battara, M., B. Golden, D. Vigo, “Tuning a Parametric Clarke-Wright Heuristic via a Genetic Algorithm”, *Technical Report DEIS OR.INGCE 2006/1R*

7. Bell, W. Z., L. Dalberto, M. Fisher, “Improving the Distribution of Industrial Gases with an On-Line Computerized Routing and Scheduling Optimizer” *Interfaces,* 1 Vol. 13, No. 6, pp. 4-23, 1983.

8. Blumenfeld, D. E., L. Burns and D. Diltz, “Analyzing Trade-offs Between Transportation, Inventory and Production Costs on Freight Networks.”, *Transportation Research Part B: Methodological*, Vol. 19, No. 5, pp. 36-380, 1985.

9. Burns, L. D., D. Hall and D. Blumenfeld, “Distribution Strategies that Minimize Transportation and Inventory Costs”, *Operations Research*, Vol. 33, No. 3, pp. 469-490, 1985.

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

11. Croes, G. A. ‘‘A Method for Solving Traveling Salesman Problems’’, O*perations Research,* Vol. 6, pp. 791-812, 1958*.*

12. Cordeau, J. F., G. Laporte and F. Semet, “A Guide to Vehicle Routing Heuristics.”, *Journal of Operational Research Society,* Vol. 53, pp. 512–522, 2002.

13. Corominas, A., A.Garcia-Villoria and R. Pastor, “Fine-Tuning a Parametric Clarke and Wright Heuristic by Means of EAGH (Empirically Adjusted Greedy Heuristics)”, *Journal of the Operational Research Society,* Vol. 61, pp. 1309 -1314, 2010.

14. Dror, M. and M. O. Ball, “Inventory/Routing: Reduction From an Annual to a Short-Period Problem”, *Naval Research Logistics*, Vol. 34, No. 6, pp. 891-905, 1987.

15. Dror, M. and L. Levy, “A Vehicle Routing Improvement Algorithm Comparison of a Greedy and a Matching Implementation for Inventory Routing”, *Computers & Operations Research*, Vol. 13, No. 1, pp. 33-45, 1986.

16. Eryavuz, M. and C. Gencer, “Araç Rotalama Problemine Ait Bir Uygulama”, Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi, C.6, S.1 pp.139-155, 2001.

17. Federgruen, A. and P. Zipkin, “A Combined Vehicle-Routing and Inventory Allocation Problem.”, *Operations Research*, Vol. 32, No. 5, pp. 1019-1037, 1984.

18. Flood, M. M., ‘‘The Traveling-Salesman Problem’’, *Operations Research*, Vol. 4, pp. 61-75, 1956.

19. Gaskell, T., "Bases for Vehicle Fleet Scheduling", *Operations Research Quarterly,* Vol. 26, pp. 717-734, 1975.

20. Gillet, B. and L. Miller, "A Heuristic Algorithm for the Vehicle Dispactching Problem", *Operations Research*, Vol. 22, pp. 340-349, 1974.

21. Kim, B., S. Kim and S. Sahoo, "Balanced Clustering Algorithms for Vehicle Routing Problems," The Institute of Industrial Engineers Annual Conference, May 15-19, 2004.

22. Lysgaard, J., "Clarke & Wright's Savings Algorithm", 1997.

23. Maden, W., R. Eglese and D. Black, "Vehicle Routing and Scheduling with Time-Varying Data: A Case Study", *Journal of the Operational Research Society,* Vol. 61, pp. 515 –522, 2010.

24. Malandraki, C. and M. Daskin, "Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms." Trans Science, Vol. 26, pp. 185–200, 1992.

25. Or, I., "Traveling Salesman Type Combinatorial Problems and Their Relation to the Logistics of Regional Blood Banking" , Phd. Thesis, 1976.

26. Paessens, H., "The Savings Algorithm for the Vehicle Routing Problem.", *Journal of Operational Research,* Vol. 34, pp. 336–344, 1988.

27. Poot, A., G. Kant and A.P. Wagelmans, "A Savings Based Method for Real-life Vehicle Routing Problems", *Technical Report EI9938/A*, Erasmus University, Rotterdam, The Netherlands, 1999.

28. Potvin, J. Y. and J. M. Rousseau, "A Parallel Route Building Algorithm for the Vehicle Routing and Scheduling Problem with Time Windows." *Journal of Operational Research,* Vol. 66, pp. 331–340, 1993.

29. Taillard, E., P. Badeau and M. Gendreau, “A Metaheuristic for the Vehicle Routing Problem with Soft Time Windows”, *Transportation Science,* Vol. 31, pp. 170-18, 2004.

30. Tarantilis, C., C. Kiranoudis, “A Guided Tabu Search for the Heterogenous Vehicle Routing Problem”, *Journal of Operational Research Society,* Vol. 59, pp. 1698-169, 2007.

31. Tarantilis, C., C. Kiranoudis and G. Prastacos, “Solving the Open Vehicle Routing Problem Via a Single Parameter Metaheuristic Algorithm”, *Journal of the Operational Research Society,* Vol. 56, pp. 588–596, 2004.

32. Yellow, P.C., “A Computational Modification to the Savings Method of Vehicle Scheduling”, *Operational* Research, Vol. 21, pp.281–283, 1970.