JOINT OPTIMIZATION OF CASH MANAGEMENT AND ROUTING FOR NEW GENERATION AUTOMATED TELLER MACHINE NETWORKS

by

Şeyma Batı

B.S., Computer Engineering, Boğaziçi University, 2011

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 Computer Engineering
Boğaziçi University
2016

JOINT OPTIMIZATION OF CASH MANAGEMENT AND ROUTING
FOR NEW GENERATION AUTOMATED TELLER MACHINE NETWORKS

APPROVED BY:

Prof. Fatih ALAGÖZ (Thesis Supervisor)	....
Assoc. Prof. Didem GÖZÜPEK (Thesis Co-supervisor)	....
Prof. Tuna TUĞCU	....
Assoc. Prof. Berk CANBERK	....
Assist. Prof. Mehmet Şükrü KURAN	....

DATE OF APPROVAL: 10.06.2016

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis advisor Prof. Dr. Fatih Alagöz for his supervision and guidance and co-advisor Assoc. Prof. Didem Gözüpek for her guidance, encouragement, patience and insight throughout the research. I also would like to thank Provus Inc. for providing us with real data collected from ATMs and for the invaluable information about the operations in an ATM network. The last but not the least, I would like to thank my family for their invaluable support during my study.

ABSTRACT

JOINT OPTIMIZATION OF CASH MANAGEMENT AND ROUTING FOR NEW GENERATION AUTOMATED TELLER MACHINE NETWORKS

Cash related costs constitute a large portion of management cost of an Automated Teller Machine (ATM) network. Cash should be loaded to or taken from ATM devices in certain intervals in order to both meet customer satisfaction and to be able to generate additional revenue from excess cash through daily interest rates. Unlike classical ATMs, new generation ATMs have a single tape for cash withdrawal and deposit; this property imposes new restrictions on ATM cash management. Moreover, recycle ATMs are costly and hence their deployment should be planned carefully. In this thesis, our aim is to optimize the ATM networks in terms of cash related costs. We formulate an optimization problem as an integer linear program, which jointly decides on when to visit an ATM, how much amount of money to load to which ATM and which road should be followed for the distribution of cash to the ATMs. We also decide on which ATMs in the network should be replaced by a recycle ATM. We then propose a polynomial-time heuristic algorithm and compare it with the optimization formulation in terms of cash cost and the recycle ATM decision. We demonstrate through performance evaluation that our heuristic algorithm is suitable for practical implementation.

ÖZET

YENİ NESİL OTOMATİK VEZNE MAKİNELERİNDE NAKİT YÖNETİMİ VE ROTALAMANIN BÜTÜNLEŞİK OPTİMİZASYONU

Nakit ile ilgili maliyetler Otomatik Vezne Makinesi yönetim maliyetinin önemli bir kısmını oluşturmaktadır. Müşteri memnuniyetini sağlamak ve ihtiyaç fazlası nakitten günlük faiz ile bankaya ek gelir sağlayabilmek için belirli aralıklarla nakit yüklemesi veya alınması gerekmektedir. Standart makinelerin tersine geri dönüşümlü adı verilen yeni tip Otomatik Vezne Makinelerinde para yatırma ve çekme işlemi aynı para kasedi üzerinden yapılmakta ve bu özellik nakit yönetimine yeni kısıtlamalar getirmektedir. Bununla birlikte geri dönüşümlü makineler yüksek maliyetli olduğu için planlaması dikkatli yapılmalıdır. Bu çalışmada, amacımız otomatik vezne makinesi ağlarını nakdi maliyetler açısından optimize etmektir. Makineye ne zaman, ne miktarda para yükleneceği ve nakit dağıtımında hangi rotanın izleneceğine bütünleşik olarak karar veren optimizasyon problemi tamsayı lineer programlama olarak formüle edilmiştir. Aynı zamanda nakit maliyetini azaltmak amacıyla hangi makinelerin geri dönüşümlü makineler ile değiştirilmesi gerektiği kararı da çalışmamızın çıktıları arasındadır. Otomatik Vezne Makinesi nakit yönetimi problemi için polinom zamanlı buluşsal algoritma tasarlanmış ve optimizasyon formülasyonu ile nakit maliyeti ve geri dönüşümlü makine ile değiştirme kararı açısından karşılaştırma yapılmıştır. Performans sonuçlarına göre tasarlanan buluşsal algoritmanın pratikte kullanılabilir olduğu görülmüştür.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

B	Number of working hours in minutes
C_i	Maximum amount of cash that can be held in ATM i
c_{ihk}	Money paid to CIT k for visiting ATM i on day h
D_{ih}	Deposit amount for ATM i on day h
f_h	Daily interest rate on day h
$\mathcal{H}$	Days in the scheduling period
$\mathcal{M}$	Set of CITs
$\mathcal{N}$	Set of ATMs
R	Cost of deploying a recycle ATM
t_{ijh}	The time it takes to go from ATM i to ATM j on day h
v_k	Capacity of CIT vehicle k in terms of cash value
W_{ih}	Withdrawal amount for ATM i on day h
δ	Service time for an ATM in minutes

LIST OF ACRONYMS/ABBREVIATIONS

API	Application Programming Interface
ATM	Automated Teller Machine
CIT	Cash in Transit
ILP	Integer Linear Programming
TSP	Travelling Salesman Problem

1. INTRODUCTION

Cash management for automated teller machines (ATM) is a key service area for financial institutions such as banks. Cash-related costs constitute around 35-60% of the overall costs of running an ATM [1]. As the size and complexity of ATM networks increases, it becomes critical for financial institutions to optimize ATM cash flows to improve return on cash assets, reduce operation costs, and deliver high quality service to their customers. The factor that reduces the return on cash assets, referred to as the *idle cash cost*, is due to the more than necessary amount of cash residing in ATMs. Idle cash in ATM constitutes a cost to the financial institution since the institution cannot generate additional revenue by investments such as daily interest.

There are two types of ATM machines, referred to as *classical* and *recycle ATMs* [2]. While classical ATMs have separate tapes for cash withdrawal and deposit, recycle ATMs, also called as new generation ATMs, have a single tape for both operations. Recycle ATMs are costly; therefore, their deployment requires rigorous analysis. Transfer of cash between cash center and ATM points is carried out by firms called *"Cash in Transit (CIT)"*. Banks pay the CITs a certain amount of money in return for their service and this payment constitutes the logistic costs, which are major components of operational costs. Optimal ATM cash management involves the analysis of idle cash cost and logistic cost. A vital, yet unexplored, issue in ATM cash management stems from the tradeoff between these costs: an ATM cash management system should minimize the overall idle cash and logistic cost while at the same time providing the customers with a quality of service by ensuring that ATMs do not run out of cash, i.e., by deciding on the optimum amount of money that should be placed in the ATMs to satisfy the customer demands [3].

In this thesis, we formulate an optimization problem whose objective is to minimize the cash management cost. We consider a system consisting of CITs and cash centers as well as classical and recycle ATMs. To the best of our knowledge, this thesis is the first study that focuses on cash management optimization of recycle ATMs.

Furthermore, we decide on the route of the CIT vehicles. Since armored vehicles of the CITs have a certain upper limit for the amount of money to carry due to reasons such as security, CIT routes should be determined together with the amount of money to be loaded to or taken from the ATMs. To the best of our knowledge, this thesis is the first study that focuses on the joint optimization of the cash management and the routing of CIT vehicles.

In this thesis, we first formulate an integer linear programming (ILP) problem that jointly optimizes cash decisions, i.e. when to load how much amount of cash to which ATMs, and the routing of CIT vehicles. We then propose a polynomial-time heuristic algorithm and conduct simulations using synthetic data we generated and real ATM data obtained from a private company (Provus Inc.). Our simulation results indicate that our heuristic algorithm yields close solutions to the values obtained from the execution of our ILP formulation using optimization software CPLEX.

The remainder of this thesis is organized as follows: In Chapter 2 we explain the motivation for this work and summarize the related work in the literature as well as our contributions. We formulate our optimization problem as an ILP in Chapter 3 and describe our proposed heuristic algorithm in Chapter 4. We present the simulation results in Chapter 5 and then conclude in Chapter 6.

2. RELATED WORK AND SUMMARY OF CONTRIBUTIONS

2.1. Related Work

Before discussing the related work in the literature, let us point out that in the remainder of this thesis, *withdraw* and *deposit* refer to the customer actions, whereas *take* and *load* refer to the CIT vehicle actions.

Most studies about cash management in the literature focus on estimation of daily cash demands for ATMs. For instance, the work in [4] proposes a method based on simulated annealing to estimate the amount of cash load for ATMs such that the maintenance cost of ATMs is minimized. ATM maintenance cost function consists of idle cash costs (related to interest rate), cash loading costs and constant ATM-service costs, while neglecting the routing of CIT vehicles. The work in [5] suggests an application of fuzzy ARTMAP Network for analyzing and forecasting daily cash requirement in ATM assuring prompt cash availability and dispensing service. In [6], the prediction of cash demand for groups of ATMs with similar day-of-the week cash demand patterns is done to improve ATMs' cash demand forecast by using neural networks. The authors of [7] propose to use a local learning model of the pseudo self-evolving cerebellar model articulation controller associative memory network to produce accurate forecasts of ATM cash demands. Another study in forecasting cash demand in ATMs uses Neural Networks and Least Square Support Vector Machine [8]. The work in [9] also uses artificial neural networks and neuro-fuzzy models for demand forecasting. In [10], a local linear wavelet neural network is used for time-series prediction.

The work in [3] focuses on cash management in ATMs in the compensation of credit card transactions. They formulate a stochastic programming problem and analyze its several special cases. The short-term model with fixed costs results in an integer linear programming problem, whereas the mid-term model with fixed and staircase costs leads to a multistage stochastic problem. Another work in [11] presents a general model of cash management, viewed as an impulse control problem for a stochastic money flow process. This process is respresented by a superposition of a Brownian motion and a compound Poisson process, controlled by two-sided target-trigger policies. The study in [12] applies a stochastic single-period inventory management approach to analyze optimal cash management policies with fuzzy cash demand based on fuzzy integral method so that total cost is minimized.

The authors in [13] develop a policy for cash management using Miller and Orr model, which does not define a single ideal point for cash balance, but an oscillation range between a lower bound, an ideal balance, and an upper bound. They use genetic algorithms and particle swarm optimization.

The work in [14] focuses on cash inventory management for out of working hours, during which replenishment of the ATMs is impossible. They propose inventory models and policies under both full and imperfect information. The study in [15] proposes grouping ATMs into nearby-location clusters and also optimizing the aggregates of daily cash withdraws in the forecasting process.

There are a few studies in the literature that focus on the routing of CIT vehicles. However, these studies do not take the cash management into account. For instance, the works in [16] and [17] model the routing of CIT vehicles as some type of vehicle routing problem and address it using a genetic algorithm. Besides, the work in [18], treats the routing of CIT vehicles and cash management as two separate problems, while focusing mainly on demand forecasting in cash management. In [19], they develop a vehicle routing system based on a discrete particle swarm optimization method to support the decision of vehicle routes. Also in paper [20], particle swarm optimization algorithm is used for grain logistics vehicle routing problem. In [21], they study a possibility of finding the optimal solution of vehicle routing problem and offer a practical implementation of the Nearest Insertion Algorithm. There are many different variants of the vehicle routing problem [22]. In [23], an ant colony optimization is proposed for capacitated vehicle routing problem. Also in [24], capacitated vehicle routing problem is studied as a hysteretic optimization problem. In [25] a heuristic algorithm as a tour construction type procedure with an embedded improvement procedure is developed for the periodic vehicle routing problem. In another work in [26] delivery and pickup vehicle routing problem is studied by using tabu search algorithm. While most of the studies on vehicle routing focus on minimizing the sum of total distance, the work in [27] focuses on the minimization of fuel consumption.

There are also studies on inventory management in literature other than ATM domain. In [28], a generic problem description and a mathematical model are proposed to address the production-inventory-distribution-routing problem which integrates lot-sizing, inventory management, distribution planning, and vehicle routing problems. In [29], they present a mixed integer programming model for fleet deployment including inventory management at the ports along trade routes. A rolling horizon heuristic is proposed which solves the problem by iteratively solving sub-problems with shorter planning horizon. In [30], they propose a framework for reducing the total operation cost while satisfying the service level constraints. The performances of each inventory in the system are estimated by kriging models in a region-wise manner which reduces the computational time during both sampling and optimization. The work in [31] presents a multi-objective approach to solve one version of the Inventory Routing Problem by simultaneously minimizing both the inventory and transportation costs. The method proposed in this work is based on a pareto evolutionary algorithm and includes aspects associated with the representation of candidate solutions, genetic operators and local search. The study in [32] aims to design the organization of the inventory pool and to propose an inventory reserve strategy that lies in the order processing procedure to handle the risk pooling effect. They propose reserve strategies based on the marginal cost approach for supporting the requests with higher service priority.

In paper [33], they suggest the application of genetic algorithms as means for searching and generating optimal upload strategies to minimize the daily amount of stocked money and to assure cash dispensing service.

There are also works done on joint optimization in the literature. In [34], joint optimization of maintenance strategy and production control policy is studied. In [35], they study the problem of jointly determining the promotion optimizing and inventory control of multiple beer product variants in a Chinese retail supermarket based on demand forecasting. In [36], they consider the joint management of finished goods inventory and demand for a product in a make-to-stock production system. The production process is random with controllable mean rate, and the demand process is stochastic with changeable mean rate dependent on the sale price being high or low. The management issue is how to dynamically adjust the production rate and the sale price to maximize the long run total discounted profit. The work in [37] offers a heuristic algorithm for joint optimization of performance, energy, and temperature in allocating tasks to multi-core processors.

The work in [38] discusses the differences between joint inventory supply chain management and the general inventory supply chain management. They make quantitative analysis of the key parameters of the supply chain, studying the impact of bottlenecks and the bullwhip effect on the whole supply chain.

In [39], they present an evolutionary approach to the joint management of inventory and routing in a retail chain. With this purpose, they design an ad-hoc evolutionary algorithm which includes a non-standard individual representation and two mutation operators specific to this particular problem.

2.2. Summary of Contributions

Our contributions can be summarized as follows:

- To the best of our knowledge, this thesis is the first one in the literature that focuses on joint optimization of cash management and routing for ATM networks.
- We have not encountered any previous analytical study considering recycle ATMs in the literature.
- Besides deciding on the cash amount in ATMs, we also decide on which of the

classical ATMs should be replaced by recycle ones in order to lower the cash management cost.

2.3. Practical Implications

For efficient cash management in an ATM network, necessary amount of cash should be held in each ATM because having insufficient amount of cash leads to customer dissatisfaction. On the other hand, since the money held in an ATM is in cash, it is not possible for the banks to invest that money and generate additional income through daily interest rates. Therefore, having more than necessary amount of cash in the ATMs has a financial cost for the banks. Furthermore, the route of the CITs should be decided in an optimal way such that the cash collected from ATMs is delivered to the cash center (e.g. central bank) within working hours so that additional income can be generated through daily interest rates; otherwise, the cash is counted as idle.

CIT firms carry out the loading/unloading of cash to the ATMs; this action is referred to as the replenishment of the ATMs. Financial institutions such as banks pay the CIT firms a certain amount of money for their service. We call this cost *CIT cost*. Daily replenishment of the ATMs decreases the customer dissatisfaction and the idle cash cost; however, it increases the CIT cost. On the other hand, replenishing the ATMs in long intervals decreases the CIT cost, but increases the idle cash cost. As a result, the frequency of ATM replenishment is an important decision.

In this thesis, we address these tradeoffs by formulating an optimization problem that determines the route of each CIT vehicle, which ATMs should be visited on which day by each CIT vehicle, and the amount of cash to be loaded to or taken from each ATM so that the overall cost of ATM cash management is minimized. Our model takes the ATM type (recycle/classical) into consideration and also determines what the type of each ATM should be; in other words, ATM type is a decision variable in our formulation. The heuristic algorithm used in this thesis can be utilized in real life situations for banks to lower their ATM cash management costs. In order to apply the algorithms to a real life ATM network, estimated withdrawal and deposit cash amounts of ATMs should be known beforehand. Also, time to travel between each ATMs should also be given. Once this information is obtained, algorithm can be run for each ATM network. In Chapter 4, we investigate the time complexity analysis of our heuristic algorithm and prove that it has polynomial time complexity. Therefore it is suitable for practical implementation.

3. PROBLEM FORMULATION

Our optimization problem aims to find a schedule that decides on which days the ATMs should be visited, what amount of cash should be loaded to the ATMs, and what the route of the CIT vehicles should be such that the total cost is minimized. For each ATM, the cash amount to be loaded is calculated for each day in the scheduling period, which is a tunable parameter and is usually six or seven days in practice. We assume that the daily cash need for each ATM is forecasted beforehand. Therefore, the daily cash amount forecasted to be in withdrawal and deposit box of each ATM, the daily interest rate and the amount of money charged by CIT for each visit of an ATM are input parameters to our optimization problem. Table 3.1 and 3.2 show the input and decision variables, respectively, of our ILP formulation.

Table 3.1. Table for input variables.

$\mathcal{N}$	Set of ATMs, where $\mathcal{N}$ = 1, 2,..., N
$\mathcal{M}$	Set of CITs, where $\mathcal{M}$ = 1, 2, ..., M
$\mathcal{H}$	Days in the scheduling period, where $\mathcal{H}$ = 1, 2, ..., H
t_{ijh}	The time it takes to go from ATM i to ATM j on day h (in minutes)
v_k	Capacity of CIT vehicle k in terms of cash value (e.g. Euro)
f_h	Daily interest rate on day h
c_{ihk}	Money paid to CIT k for visiting ATM i on day h
W_{ih}	Withdrawal amount for ATM i on day h
D_{ih}	Deposit amount for ATM i on day h
C_i	Maximum amount of cash that can be held in ATM i
B	Number of working hours in minutes
δ	Service time for an ATM in minutes
R	Cost of deploying a recycle ATM

The objective in our optimization problem is to minimize the overall cost of ATM cash management, which consists of *logistic cost*, *idle cash cost* and recycle ATM cost. Idle cash cost is due to more than necessary amount of cash in deposit and withdrawal box of ATMs. The first term of Equation 3.1 models the idle cash cost,

Table 3.2. Table for decision variables.

x_{ijhk}	$= \begin{cases} 1, & \text{if ATM } j \text{ is visited after ATM } i \text{ by CIT } k \\ & \text{on day } h \\ 0, & \text{otherwise} \end{cases}$
y_{ihk}	$= \begin{cases} 1, & \text{if ATM } i \text{ is visited by CIT } k \text{ on day } h \\ 0, & \text{otherwise} \end{cases}$
w_{ih}	The remaining cash in withdrawal box of ATM i on day h
d_{ih}	The remaining cash in deposit box of ATM i on day h
z_{ihk}	Cash amount to be loaded to ATM i on day h by CIT k
a_{ihk}	Cash amount to be taken from ATM i on day h by CIT k
z_{ih}	Cash amount to be loaded to ATM i on day h
a_{ih}	Cash amount to be taken from ATM i on day h
r_i	$= \begin{cases} 1, & \text{if ATM } i \text{ is a recycle ATM} \\ 0, & \text{otherwise} \end{cases}$
ϕ_{ih}	$r_i \times \sum_{e=1}^{h} a_{ie}$
$\Upsilon_{ihh'}$	$\prod_{e=h}^{h'} \sum_{k=1}^{M} (1 - y_{iek})$
u_i	the extra variables used for subtour elimination

which is proportional to the daily interest rate, whereas, the second term of Equation 3.1 models the logistic cost, which is due to the money paid to the CIT firm for ATM visits. The last term of Equation 3.1 models the cost of deploying a recycle ATM. Accordingly, the objective function of our ILP formulation is as follows:

$$\min \sum_{i=1}^{N} \left(\sum_{h=0}^{H} (w_{ih} + d_{ih}) \times f_h \right) + \sum_{k=0}^{M} \sum_{i=0}^{N} \sum_{h=0}^{H} (c_{ihk} \times y_{ihk}) + \sum_{i=1}^{N} (r_i \times R) \tag{3.1}$$

CIT vehicles have to start the route from a presepecified center node, which is usually the central bank, because the cash to be loaded to the ATMs must be taken from a cash center. We model this requirement as follows:

$$\sum_{j=1}^{N} x_{0jhk} = 1; \forall k \in \mathcal{M}, h \in \mathcal{H} \tag{3.2}$$

where 0 refers to the index of the center node.

CIT's have to return to the center node after visiting the ATMs in order to bring the collected cash to the central bank. This requirement necessitates a closed loop to be constructed as follows:

$$\sum_{i=0}^{N} x_{iphk} - \sum_{j=0}^{N} x_{pjhk} = 0; \forall p \in \mathcal{N}, k \in \mathcal{M}, h \in \mathcal{H} \tag{3.3}$$

Each ATM should be visited by at most one CIT. We can model this requirement as follows:

$$\sum_{k=0}^{M} y_{ihk} \leq 1; \forall i \in \mathcal{N}, h \in \mathcal{H} \tag{3.4}$$

A CIT can visit an ATM only if the ATM is on the route of the related CIT:

$$y_{ihk} = \sum_{j=0}^{N} x_{jihk}; \forall i \in \mathcal{N}, h \in \mathcal{H}, k \in \mathcal{M} \tag{3.5}$$

The cash capacity v_k of the CIT vehicle k stems from its physical and security requirements. The cash amount carried by CIT k on day h should not exceed its capacity. The first term of Equation 3.6 is the total amount of cash to be loaded to ATMs assigned to CIT k on day h. At the begining of the route, at least that amount of cash must exist in the vehicle. Second term of Equation 3.6 states that after each ATM visit on the route of the CIT vehicle, the amount of cash in the vehicle decreases by the amount of cash loaded to that ATM and increases by the amount of cash taken from that ATM. At each point on the route of the CIT vehicle, vehicle capacity v_k must not be exceeded:

$$\sum_{i=1}^{N} z_{ihk} + \sum_{j=0}^{N} \sum_{e=0}^{i} x_{jehk} \times (a_{ehk} - z_{ehk}) \leq v_k; \forall h \in \mathcal{H}, k \in \mathcal{M} \tag{3.6}$$

The amount of cash loaded to ATM i on day h is equal to the sum of the cash loaded to ATM i on day h by all CITs:

$$z_{ih} = \sum_{k=0}^{M} z_{ihk}; \forall i \in \mathcal{N}, h \in \mathcal{H} \tag{3.7}$$

The amount of cash taken from ATM i on day h is equal to the sum of cash taken from ATM i on day h by all CITs:

$$a_{ih} = \sum_{k=0}^{M} a_{ihk}; \forall i \in \mathcal{N}, h \in \mathcal{H} \tag{3.8}$$

If ATM i is visited on day h by CIT k, then the amount of cash loaded to ATM i cannot exceed the ATM cash capacity. Otherwise, the amount of cash loaded to ATM i equals zero:

$$z_{ihk} \leq C_i \times y_{ihk}; \forall i \in \mathcal{N}, h \in \mathcal{H}, k \in \mathcal{M} \tag{3.9}$$

Likewise, if ATM i is visited on day h, then the amount of cash taken from ATM i is at most the ATM cash capacity. Otherwise, the amount of cash taken from ATM i is zero:

$$a_{ihk} \leq C_i \times y_{ihk}; \forall i \in \mathcal{N}, h \in \mathcal{H}, k \in \mathcal{M} \tag{3.10}$$

For classical ATMs, the remaining amount of cash in the withdrawal box of ATM i on day h is equal to the difference between the total amount of cash loaded to ATM i until day h and the total amount of cash withdrawn from ATM i until day h. For recycle ATMs, on the other hand, the remaining amount of cash in ATM i on day h is equal to the difference between the total amount of cash loaded and deposited to ATM i until day h and the total amount of cash withdrawn and taken from ATM i until day h. These constraints can be modeled as follows:

$$w_{ih} = \sum_{e=1}^{h}(z_{ie} - W_{ie}) + r_i \times (\sum_{e=1}^{h} D_{ie} - \phi_{ih}); \forall i \in \mathcal{N}, h \in \mathcal{H} \tag{3.11}$$

Note that the definitions of the decision variables ϕ_{ih}, r_i and a_{ie} pose a non-linear relationship between the decision variables. For recycle ATMs, CIT vehicle can take more than necessary amount of cash residing in the withdrawal box while visiting the ATM. The term ϕ_{ih} refers to the cash amount taken from ATM i on day h if ATM i is a recycle ATM. Nonlinear term, ϕ_{ih}, can be transformed into linear terms by the use of linear constraints [40] as follows:

$$\phi_{ih} \leq C_i \times h \times r_i; \forall i \in \mathcal{N}, h \in \mathcal{H} \tag{3.12}$$

$$\phi_{ih} \geq 0; \forall i \in \mathcal{N}, h \in \mathcal{H} \tag{3.13}$$

$$\phi_{ih} \leq \sum_{e=1}^{h} a_{ie}; \forall i \in \mathcal{N}, h \in \mathcal{H} \tag{3.14}$$

$$\phi_{ih} \geq C_i \times h \times (r_i - 1) + \sum_{e=1}^{h} a_{ie}; \forall i \in \mathcal{N}, h \in \mathcal{H} \tag{3.15}$$

For classical ATMs, the remaining amount of cash in deposit box on day h is equal to zero if ATM i is visited by a CIT on day h; otherwise, it is equal to the total amount of cash deposited to ATM i after the last visit of ATM i by a CIT. For recycle ATMs, since there is no separate deposit box, the remaining amount is equal to zero:

$$d_{ih} = (1 - r_i) \times \sum_{e=0}^{h} \Upsilon_{ieh} \times D_{ie}; \forall i \in \mathcal{N}, h \in \mathcal{H} \tag{3.16}$$

Note that the definitions of the decision variables $\Upsilon_{ihh'}$ and y_{iek} pose a non-linear relationship between the decision variables. We can linearize this relationship through the following set of constraints:

$$\Upsilon_{ihh'} \leq 1 - y_{iek}; \forall i \in \mathcal{N}, h, h' \in \mathcal{H}, h \neq h' \tag{3.17}$$

$$\Upsilon_{ihh'} \geq h - h' + \sum_{e=h}^{h'} \sum_{k=1}^{M} (1 - y_{iek}); \forall i \in \mathcal{N}, h, h' \in \mathcal{H}, h \neq h' \tag{3.18}$$

The time spent in traveling between ATMs and during the cash loading process has to be smaller than the total number of working hours; i.e., CIT should return to the center node within working hours. The time spent in giving service to the ATMs is assumed to be constant and equal to δ. We can model this constraint as follows:

$$\sum_{i=0}^{N} \sum_{j=0}^{N} t_{ijh} \times x_{ijhk} + \sum_{i=1}^{N} \delta \times y_{ihk} \leq B \tag{3.19}$$

The constraints in 3.20, 3.21 and 3.22 refer to the Miller–Tucker-Zemlin (MTZ) formulation of the TSP [41–43]. They provide subtour elimination in the routing of the CIT vehicles.

$$u_1 = 1; \tag{3.20}$$

$$2 \leq u_i \leq N; \forall i \neq 1; \tag{3.21}$$

$$u_i - u_j + 1 \leq (N-1) \times (1 - x_{ijhk}); \forall i \neq 1, \forall j \neq 1, \forall k, \forall h \tag{3.22}$$

The following set of constraints model the decision variables of our ILP formulation:

$$x_{ijhk}, y_{ihk}, r_i \in \{0, 1\} \tag{3.23}$$

$$w_{ih}, d_{ih}, a_{ih}, z_{ih}, a_{ihk}, z_{ihk}, u_i \in \{\mathbb{Z}^+ \cup \{0\}\} \tag{3.24}$$

4. PROPOSED ALGORITHM

4.1. Heuristic Algorithm

The problem formulated in Chapter 3 is a computationally very difficult problem. Therefore, designing efficient heuristic algorithms that find approximate solutions with acceptable time and space complexity is vital [44]. To this end, we propose in this chapter a polynomial-time heuristic algorithm.

Optimization software CPLEX [45] can be used to generate solutions for ILP problems. When CPLEX finds an optimal solution, it indicates this situation as an output. Finding an optimal solution might take too long time especially when the problem size is very large. In such a case, the optimality tolerance parameter called epgap can be set so that the computation ends when a solution within the provided epgap percentage of the optimal solution is found. This way, CPLEX can be used to efficiently find a (not necessarily optimal) solution to the ILP formulation in Equations 3.1 - 3.24 in Chapter 3. Nevertheless, in a real life setting, CPLEX or any other optimization software may not be available. In addition, the network may be so large and dense that the running times of the optimization software become too high. Moreover, CPU and memory of the computer(s) may be insufficient to run the optimization software for such large networks. In such cases, it becomes inconvenient to use CPLEX. Therefore, we propose a polynomial-time heuristic algorithm and we then compare the performance of our heuristic algorithm with the solutions obtained from CPLEX.

As the problem size gets larger, CPLEX running times become too high and we have to either set a time limit for the optimizer or change epgap parameter to a higher value. We set the time limit to 18.000 seconds in our experiments in Chapter 3. Default value of epgap is 0.0001 and it can take any value between 0.0 and 1.0 [46]. When we set the time limit, epgap parameter value is 0.01. This way, we obtain CPLEX solutions that are either optimal or near optimal so that we can have a baseline to compare our heuristics with.

4.1.1. Calculation of ATM visit days and cash amount

First, which day to visit and the amount of cash to load to each ATM are calculated separately. The amount of cash to be deposited to (D_{ih}) and withdrawn from (W_{ih}) ATMs are given as input to this stage. Furthermore, the cost of CIT for visiting the ATMs (c_{ihk}) and the daily interest rate (f_h) are also given as input. Daily cash management cost of ATMs (tc_{ihk}) can be calculated by the following formula:

$$tc_{ihk} = (w_{ih} \times f_h) + (c_{ihk} \times y_{ih}) \tag{4.1}$$

Total cost depends on which days the ATMs are visited (y_{ih}). Calculation of the cash amount to be loaded to ATMs is done for a scheduling period, which is given as input to the algorithm. In practice, the scheduling period is a small and constant number, which is usually 6 or 7 days. If we consider 7 days ahead, there are total of 2^7 possible solutions. Therefore, in Stage 1, line 6 states that the number of possible solutions to investigate is 2^7. Since there is a finite number of possible solutions, the amount of cash to load to ATMs, the remaining amount of cash in ATM (w_{ih}), the idle cash cost due to more than necessary amount of cash (mc), the amount of money to be charged by the CIT (cc), and the total cost (tc_{ihk}) can be calculated for each possible solution. In line 7, b_j shows the binary form of j where bit values indicate whether ATM is visited or not on that day. In line 8, the cash amount to be loaded to ATM for that possible solution is calculated by using the withdrawal and deposit amounts. In line 9, the remaining cash value for each day is calculated. Line 10 calculates the idle cash cost, which is proportional to the interest rate and the remaining amount of cash in the ATM at the end of the day. Line 11 calculates the CIT cost, which is equal to zero if the ATM is not visited on that day. Line 12 calculates the total cash management cost for that day, which is stated in Equation 4.1. In lines 13, 14 and 15, the solution that gives the minimum total cost among all possible 2^7 solutions is stored. The algorithm is executed for each day. As the output, we find the days (within the N days) to visit the ATM and what amount of cash to be loaded to that ATM. The same algorithm is executed for each ATM.

Require: $W_{ih}, D_{ih}c_{ihk}, f_h$
Ensure: y_{ih}
$tc_{ihk}, mc, cc \leftarrow 0$
$tc_{ihk}^{min} \leftarrow$ A very large number
for each ATM i **do**
  for $j = 0$ `to` 2^7 **do**
    $b_j \leftarrow binary(j)$
    `findLoadAmount(`b_j, W_{ih}, D_{ih}`)`
    `findRemainingAmount(`w_{ih}`)`
    $mc \leftarrow w_{ih} \times f_h$
    $cc \leftarrow c_{ihk} \times y_{ih}$
    $tc_{ihk} \leftarrow mc + cc$
    if $tc_{ihk} \leq tc_{ihk}^{min}$ **then**
      $tc_{ihk}^{min} \leftarrow tc_{ihk}$
      $b_j^{min} \leftarrow b_j$
    end if
    for each h **do**
      $y_{ih} \leftarrow b_j$
    end for
  end for
end for
return y_{ih}

Figure 4.1. Algorithm for CIT visit days.

4.1.2. Candidate route construction for CITs

Stage 1 determines when to visit each ATM, i.e. which ATMs will be visited within a given day. Stage 2 decides on the routes of the CITs that pass through these predetermined ATMs. The CIT routes should satisfy the following criteria: (i) The distribution of the cash to ATMs must be completed within the working hours (B). (ii) CITs must start the route from a center node and return to it within the given time period.

In Stage 2, lines 6 through 11 state that, for the first half of the working hours, our algorithm starts the route from the center node and moves as far away from the center node as possible. In contrast, for the second half of the working hours, our algorithm makes the route return to the center node as we state in lines 12 through 17. In order to decide on the next hop in the first half, our algorithm selects the node that is closer to the center among the two nodes that are closest to the current node. The algorithm marks the visited ATMs as it proceeds. For each CIT, a route with candidate ATM nodes is constructed similarly. x_{ijk} in line 19 is the binary variable showing whether route of CIT k includes the edge from ATM i to j.

4.1.3. Assignment of ATMs to the CITs

As the output of second stage, different route sets are given; i.e., for each CIT, a route with candidate ATM nodes are constructed. Each unvisited ATM must be assigned to one of the routes that passes through it. The routes might intersect; however, each ATM must be visited and served by exactly one CIT; i.e., ATMs must be assigned to only one route. In order to assign the ATMs to the routes and to pick the route with minimum cost we construct an edge-weighted bipartite graph [47] $G = (\mathcal{A}, \mathcal{R}, E)$, where $\mathcal{A} = \{1, ..., A\}$ is the set of ATMs, $\mathcal{R} = \{1, ..., R\}$ is the set of routes, and there is an edge $e \in E$ between an ATM $a \in \mathcal{A}$ and a route $r \in \mathcal{R}$ if ATM a is on the route r. Let c_r be the cost of visiting an ATM on route r, i.e. c_r equals c_{ihk} shown in Table 3.1 denoting the money paid to the CIT. Since Stage 2 determines for each CIT a route with candidate ATMs, each route here corresponds to

Require: t_{ij}, B

Ensure: x_{ijk}

for each CIT k in $\mathcal{M}$ **do**

  $totalTime \leftarrow 0$

  Pick the closest node to the center

  while($totalTime \leq B/2$)

  $totalTime \leftarrow totalTime + t_{ij}$

  Pick the node that is further from the center among the next two nodes that are closest to the current node

  $x_{ijk} \leftarrow 1$

  end

  while($totalTime \leq B$)

  $totalTime \leftarrow totalTime + t_{ij}$

  Pick the node that is closer to the center among the next two nodes that are closest to the current node

  $x_{ijk} \leftarrow 1$

  end

end for

return x_{ijk}

Figure 4.2. Algorithm for CIT routes.

a CIT and the cost of using that route equals the money paid to that CIT. We set the weights of all edges incident to vertex $r \in \mathcal{R}$ to c_r. We then define another variable Δ_r, which denotes the maximum number of ATMs that route r can visit. We set Δ_r to a reasonable number (as follows) by considering the service time for each ATM (δ in Table 3.1), time to travel between visited ATMs (t_{ijh} in 3.1) and the working hour limit (B in Table 3.1). Let Y_{hk} be the total number of ATMs on the route of CIT k on day h and X_{hk} be the total amount of time spent in traveling between the ATMs on the route of CIT k on day h. Note that Y_{hk} and X_{hk} can be stated as follows:

$$Y_{hk} = \sum_{i=0}^{N} y_{ihk}; \forall h \in \mathcal{H}, k \in \mathcal{M} \tag{4.2}$$

$$X_{hk} = \sum_{i=0}^{N} \sum_{j=0}^{N} x_{ijhk} \times t_{ijh}; \forall h \in \mathcal{H}, k \in \mathcal{M} \tag{4.3}$$

Then we calculate Δ_r as follows:

$$\Delta_r = \frac{B - (\delta \times Y_{hk}) - X_{hk}}{\delta} \tag{4.4}$$

We then solve the ILP in 4.5-4.7. Let x_{ar} be a binary decision variable related to y_{ihk} in Table 3.2 as in the following. Here the subscript a in x_{ar} corresponds to ATM i and the subscript r corresponds to the route of CIT k on day h. In other words,

$$x_{ar} = \begin{cases} 1; & \text{if edge between ATM } a \text{ and} \\ & \text{route } r \text{ is selected} \\ 0; & \text{otherwise} \end{cases}$$

The objective function in 4.5 aims to minimize the total cost of assigning ATMs to the routes. The goal here is related to the transportation cost of the objective function 3.1 in Chapter 3. The constraint in Equation 4.6 ensures that each ATM is assigned to only one route and the constraint in Equation 4.7 ensures that at most Δ_r ATMs can be assigned to each route r.

$$\min \sum_{a=1}^{A} c_r \times x_{ar} \tag{4.5}$$

$$\sum_{r=1}^{R} x_{ar} = 1; \forall a \in A \tag{4.6}$$

$$\sum_{a=0}^{A} x_{ar} \leq \Delta_r; \forall r \in R \tag{4.7}$$

4.2. Computational Complexity

In Stage 1, scheduling period is constant in all experiments; hence there exists a finite number of CIT visit alternatives for each ATM. Each day, an ATM is either visited by a CIT or not; therefore, there are a total of 2^7 alternatives since we take the scheduling length as 7 days. In line 8 and 9 of Stage 1, finding the remaining amount of cash in the ATM has constant time complexity because it is related to the scheduling length, which is also constant. We make the calculation for each CIT visit alternative for each ATM. We use exhaustive search in this stage; since the scheduling length is always constant in our algorithm (7 days), this step takes linear time.

In Stage 2, for each CIT, the algorithm scans the nodes in the ATM network. The time to construct a route is restricted by the working hour limit B. For each node in the network, the algorithm scans at most $N-1$ other nodes, where N is the number of ATMs. Hence, the route of each CIT can be constructed in quadratic time.

In Stage 3, the calculation of Δ_r clearly takes polynomial time. We now show that the ILP in 4.5-4.7 is also solvable in polynomial time. Let $\mathbf{I}$ be a function associating an interval of natural numbers for each vertex in $\mathcal{A}$ and $\mathcal{R}$. We then set $\mathbf{I}(a) = [1, 1]$ $\forall a \in \mathcal{A}$ and $\mathbf{I}(r) = [0, \Delta_r]$ $\forall r \in \mathcal{R}$. The problem of finding a sub(multi)graph that maximizes the total edge weights while respecting the constraints about the interval of allowed degrees for each vertex is known to be solvable in polynomial time [48, 49]. In particular, if the (multi)graph is bipartite (as it is in our case), then the solution for the ILP representing this problem is equal to the solution of its linear program because the incidence matrix of a bipartite graph is totally unimodular [48]. Therefore, if we update the edge weights c_r as $c_r \leftarrow M - c_r$, where M is a sufficiently large number so that the resulting weights are nonnegative, then the corresponding maximization problem gives our desired solution in polynomial time.

5. NUMERICAL EVALUATION

5.1. Input Generation

In this chapter, we evaluate via simulations the performance of our heuristic algorithm under various parameter settings by comparison with the solutions obtained from the execution of the ILP formulation in 3.1 - 3.22 using CPLEX optimization software and Java. In particular, we compare our heuristic algorithm and CPLEX solutions in terms of the total cost of cash management, the number of recycle ATMs, and the cost per ATM.

In the simulations, we use both synthetic data and real ATM data provided by Provus, a payment processing company in Istanbul, Turkey. The real data consists of ATMs of PTT (the national post and telegraph directorate of Turkey) which are operated by Provus. We use the data of 16 ATMs in Ataşehir and Kadıköy region in the Anatolian side of Istanbul and 106 ATMs in the European side of Istanbul. We use the actual withdrawal and deposit amounts between December 2013 and May 2014 as well as the actual x-y coordinates of the ATM. We obtain the travel times between each pair of ATMs by using Google Maps Distance Matrix API [50]. We set the scheduling period to 1 week; i.e., using real data we evaluate the performance of our proposed methods for 25 weeks. Therefore, the figures for real data display the results for 25 samples.

For synthetic data, we generate three ATM networks with 25, 50, and 100 ATMs. Each network is connected and randomly generated. Travel times between each pair of ATMs are set to be uniformly random between 5 and 60. Table 5.3 shows the ranges for the amount of withdrawal and deposit for synthetic data with 25 ATMs, while Table 5.4 shows the corresponding ranges for 50 and 100 ATMs. For each of the three networks, we run 10 independent simulations and take their average as the obtained result depending on what the evaluated metric in that experiment is, i.e. average total cost of cash management etc. In all experiments, we set the CIT cost, service time for an ATM, interest rate, CIT capacity, working hours, and scheduling period to constant values, which are specified in Table 5.1.

Table 5.1. Parameter values.

Parameter	Value
Scheduling length	7 days
Daily interest rate	11.25
Money paid to CIT per day for visiting an ATM	100 TL
Service time for an ATM	5 min
Working hours	720 min
CIT capacity	10.000.000 TL
Cost of deploying a recycle ATM	500 TL

5.2. Simulation Results

In the first set of experiments, we analyze the impact of the number of CITs on feasibility. A solution is infeasible unless it satisfies all of the constraints in 3.2 - 3.22 specified in Chapter 3. For instance, the solution is infeasible if at least one ATM cannot be visited within the restricted working hours. For the real data with 16 ATMs, both CPLEX and heuristic algorithm always yield feasible solutions even with 1 CIT. For the real data with 106 ATMs, CPLEX returns an infeasible solution whereas the heuristic algorithm yields a solution that can leave some of the ATMs, which were originally required to be visited, as unvisited. We refer to such a solution as a *partial solution*. Therefore, in addition to the fact that heuristic algorithm is in general more useful in practice than CPLEX solutions in terms of running time and not requiring a commercial optimization software, in a practical scenario where partial solutions are permissible, heuristic algorithm is practically more useful than CPLEX also from this aspect. Figure 5.1 illustrates the performance in terms of the ratio of visited ATMs for the real data of 106 ATMs with 1, 2, and 3 CITs. Both heuristic algorithm and CPLEX visit all ATMs in the case with 3 CITs. For 1 and 2 CITs, CPLEX yields an infeasible solution, whereas our heuristic algorithm can generate partial solutions with the demonstrated ratio of visited ATMs. Moreover, we observe that increasing the number of CITs has an important role in increasing the ratio of visited ATMs and

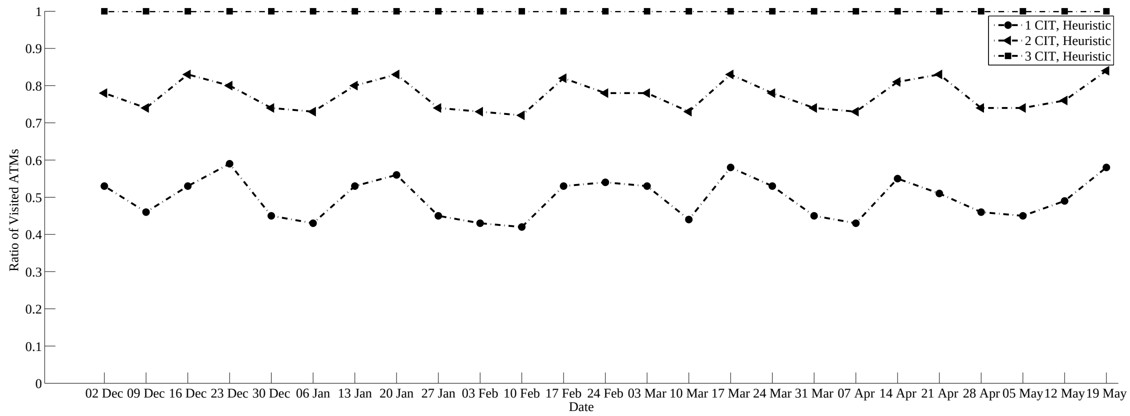


Figure 5.1. Ratio of visited ATMs with 1, 2 and 3 CITs in heuristic algorithm solution for real data with 106 ATMs.

eventually obtaining a feasible solution.

We then investigate the relation between the number of CITs and the total cost of cash management by using real data with 16 and 106 ATMs. In our experiments, we vary the number of CITs from 1 to 5. In Figure 5.2, we compare the total cost in CPLEX solution and heuristic algorithm with 1 and 5 CITs for 16 ATMs. We do not demonstrate the cost of 2, 3 and 4 CITs in the figure for better visual quality; instead, we show the average cost values for these cases in Table 5.2. For 106 ATMs, since CPLEX gives infeasible solution for 1 and 2 CITs, we compare the results with 3, 4 and 5 CITs. In Figure 5.3, we show the results for only 3 and 5 CITs, again for better visual quality. For 4 CITs, average costs of CPLEX and heuristic algorithm are 22.948 and 24.949, respectively. In Figure 5.2 and Figure 5.3, we observe that our proposed algorithm yields close performance to CPLEX for both 16 and 106 ATMs. Furthermore, once a feasible solution is found, increasing the number of CITs result in higher cost values.

Table 5.2. Average cost for real data of 16 ATMs.

Number of CITs	CPLEX	Heuristic
2	3713,93	4222,52
3	4075,58	4722,52
4	4492,83	5222,52

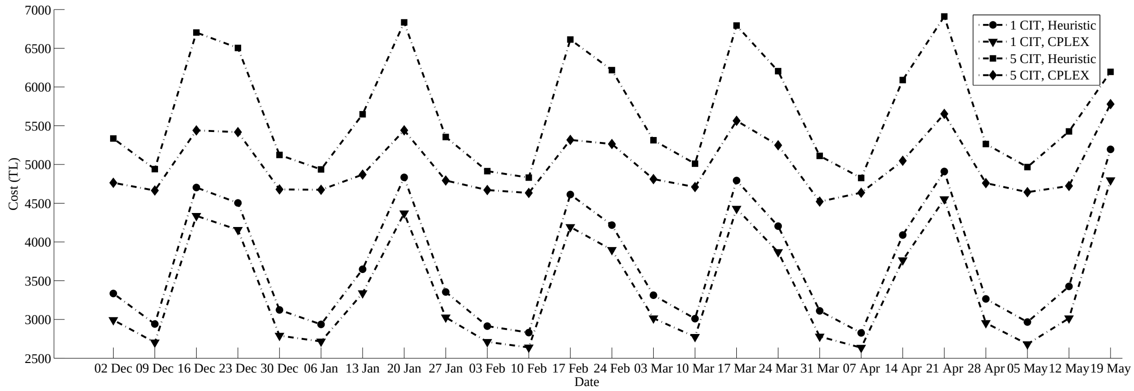


Figure 5.2. Comparison of CPLEX and heuristic algorithm for real data with 16 ATMs.

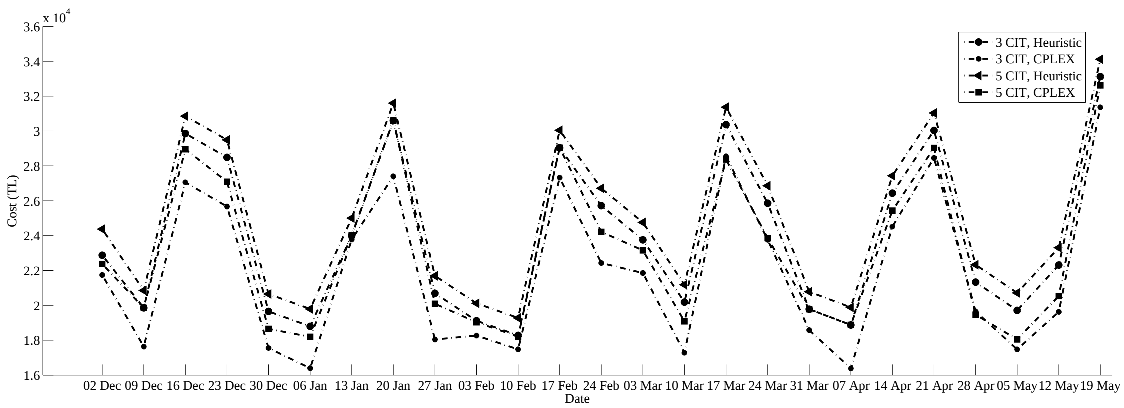


Figure 5.3. Comparison of CPLEX and heuristic algorithm for real data with 106 ATMs.

In order to better demonstrate the relation between the number of CITs and the total cost of cash management, in Figure 5.4 we vary the number of CITs from 1 to 5 and show only the CPLEX results for real data with 16 ATMs. We observe that once a feasible solution is found, the cost increases as the number of CITs increases.

Decision of replacing a classic ATM with a recycle one is also an output in both CPLEX and heuristic algorithm. We compare them in terms of the number of classic ATMs to be replaced by recycle ones. In Figure 5.5(a), we show the results of the case with 25 ATMs and synthetic data. We observe that the number of ATMs to be replaced as recycle ATMs is lower in cases with Sample ID 1, 2, and 3 compared to

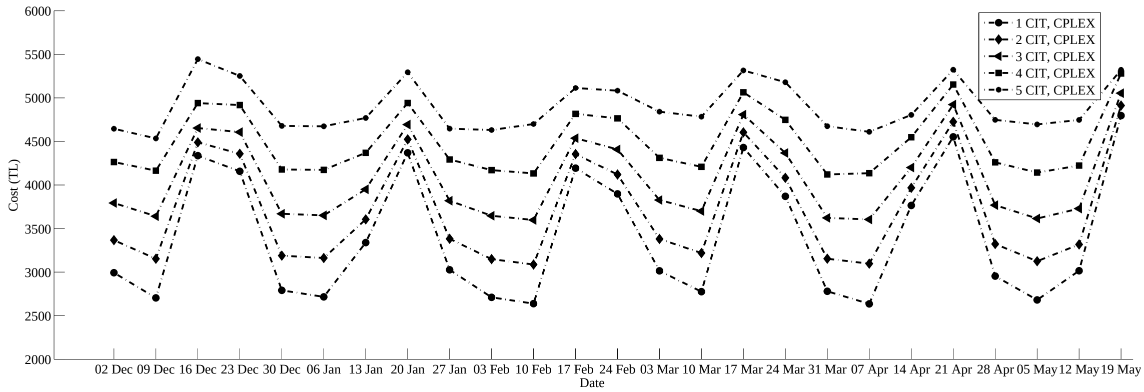


Figure 5.4. Impact of the number of CITs in CPLEX solution for real data with 16 ATMs.

the other cases. Referring to Table 5.3, we see that the deposit amounts in the cases with Sample ID 1, 2, and 3 are much lower; to be more precise, the difference between deposit and withdrawal amounts is much larger in these samples. We also observe that the deposit and withdrawal amounts are closer to each other in the cases with Sample ID 4, 5, and 6. This observation implies that deploying recycle ATMs is more suitable when the deposit amounts are closer to the withdrawal amounts since the ATM can be virtually self-operating only when the deposit amounts are large enough. Figure 5.5(b) and Figure 5.5(c) show the results of the case with 50 ATMs and 100 ATMs, respectively, using synthetic data. The number of ATMs to be replaced as recycle ATMs is uniformly little in the case with 50 ATMs, whereas the results are higher in the case with 100 ATMs. When we investigate the withdrawal and deposit ranges in Table 5.4, we see that the lower limit of deposit ranges for the case with 100 ATMs is higher than the case with 50 ATMs, whereas the upper limits of deposit ranges are very close to each other for both cases. This observation corroborates that deploying recycle ATMs is more advantageous when the deposit amounts are closer to the withdrawal amounts. Furthermore, our results demonstrate that in comparison to the total number of ATMs in the network, the difference between the withdrawal and deposit amounts has more impact on the number of ATMs to be replaced as recycle ATMs.

We also show the results of real ATM data in Figure 5.6 and we see that the number of ATMs to be changed to recycle ATMs is little. Also by taking into account the behavior with synthetic data in Figures 5.5(a), 5.5(b), and 5.5(c), this behavior in Figures 5.6(a) and 5.6(b) can be attributed to the fact that deposit amounts in real data is considerably lower than the withdrawal amounts. Recall here that in our experiments with real data, we use the real withdrawal and deposit amounts provided by Provus Inc. As a consequence, our research demonstrates that although recycle ATMs are new generation ATMs, their deployment requires careful analysis. Recycle ATMs are advantageous only in places where deposit amounts are high and real data demonstrates that this occurs rarely in practice in Turkey. If a bank or payment institution has a high motivation to deploy recycle ATMs, they should first develop business related mechanisms to increase the deposit amounts of the customers.

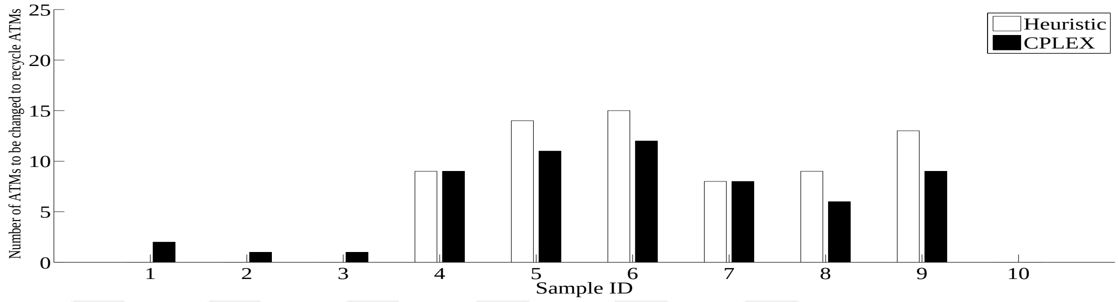
Table 5.3. Withdrawal and deposit ranges of synthetic data with 25 ATMs.

Sample ID	1	2	3	4	5	6	7	8	9	10
Withdrawal Range (x1000 TL)	[5,50]	[5,50]	[5,50]	[3,60]	[3,35]	[5,50]	[5,50]	[5,50]	[5,50]	[5,50]
Deposit Range (x 1000 TL)	[1,20]	[1,20]	[1,20]	[3,35]	[3,35]	[3,35]	[3,30]	[3,30]	[3,30]	[1,10]

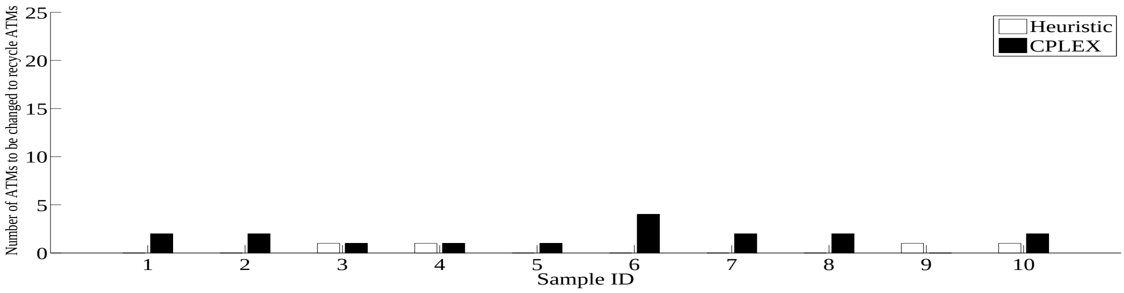
Table 5.4. Withdrawal and deposit ranges of synthetic data with 50 and 100 ATMs.

	All Samples, 50 ATM case	All Samples, 100 ATM case
Withdrawal Range (TL)	[10.000,40.000]	[10.000,40.000]
Deposit Range (TL)	[1.000,20.000]	[5000,25.000]

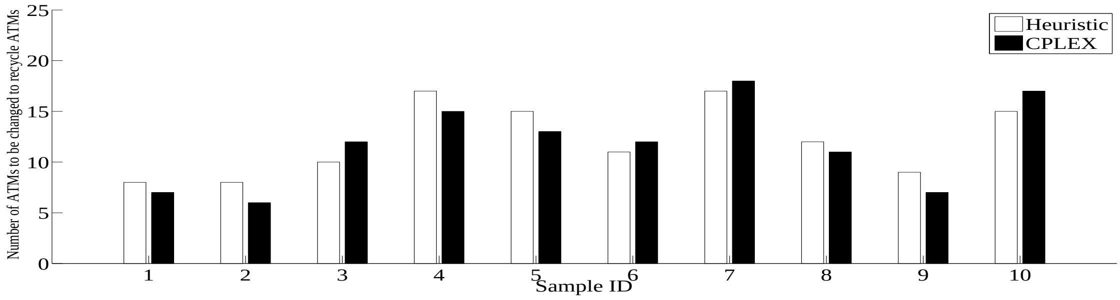
In Figure 5.7, we analyze the relation between the number of ATMs and the total cost using synthetic data. We compare the cost per ATM values in CPLEX solution and heuristic algorithm. The minimum number of CITs that gives feasible solution for 25, 50 and 100 ATMs are 1, 2 and 3 respectively. y-axis in Figure 5.7 shows the average cost per ATM of 10 samples and x-axis shows the number of ATMs. The number of CITs is set to minimum possible value that gives feasible solution in CPLEX. Figure 5.7 demonstrates that given that a feasible solution can be found, the cost per ATM decreases as the number of ATMs increases. Furthermore, the results corroborate that the performance of our proposed heuristic algorithm is close to the performance of CPLEX.



(a) 25 ATMs

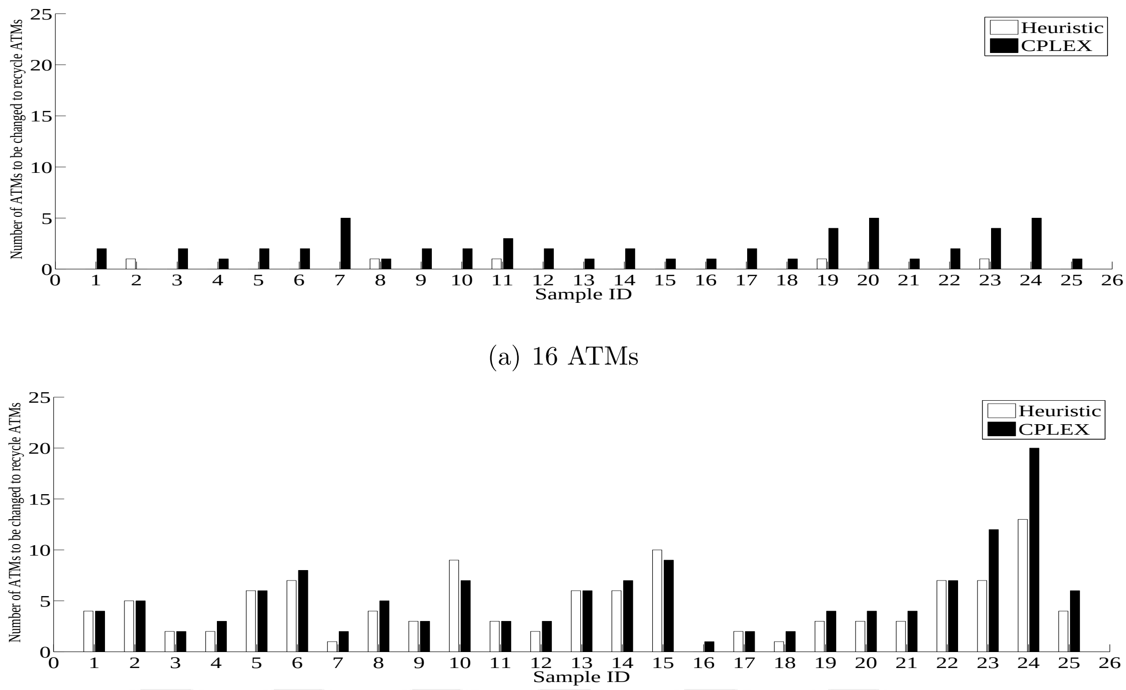


(b) 50 ATMs



(c) 100 ATMs

Figure 5.5. Number of classic ATMs to be replaced by recycle ATM for synthetic data.



(a) 16 ATMs

(b) 106 ATMs

Figure 5.6. Number of classic ATMs to be replaced by recycle ATM for real data.

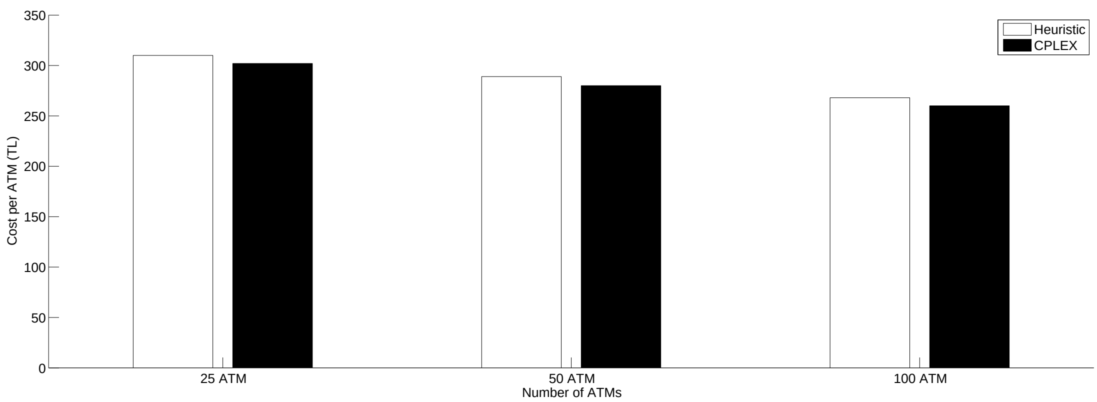


Figure 5.7. Comparison of cost per ATM values in CPLEX solution and heuristic algorithm using synthetic data.

6. CONCLUSION

In this thesis, we have formulated an integer linear program that jointly optimizes cash management and routing for new generation ATM networks. The objective of our formulated problem is to minimize the total cost of cash management in ATMs, which consists of logistic cost and idle cash cost. Our formulation also enables the decision of replacing a classical ATM with a recycle ATM. We implemented our proposed formulation by using the optimization software CPLEX. We have also proposed a polynomial-time heuristic algorithm for this problem. Via simulations using both real data obtained from Provus, a payment processing company in Turkey, and synthetically generated data, we have demonstrated that the performance of our proposed heuristic algorithm is close to the ones obtained from CPLEX. Furthermore, our results indicate that in real data, replacing a classical ATM with a recycle ATM rarely occurs in an optimal solution due to the fact that deposits occur much less frequently than withdrawals in Turkey. Therefore, if a bank or payment institution has a high motivation to deploy recycle ATMs especially in Turkey, they should first develop business related mechanisms to encourage the customers for more deposit to the ATMs.

As a future work, our proposed methods can be integrated with machine learning algorithms that predict the withdrawal and deposit amounts in ATMs using historical data.

REFERENCES

1. Simutis, R., D. Dilijonas, L. Bastina and J. Friman, “A Flexible Neural Network for ATM Cash Demand Forecasting”, *6th WSEAS Int. Conference on Computational Intelligence, Man-Machine Systems and Cybernetics*, pp. 162–165, 2007.

2. “Recycling Cash at the ATM”, NCR White Paper, 2013, `http://www.ncr.com/wp-content/uploads/recycling-cash-at-ATM-wp.pdf`, accesed at December 2014.

3. Castro, J., “A Stochastic Programming Approach to Cash Management in Banking”, *European Journal of Operational Research*, Vol. 192, No. 3, pp. 963–974, 2009.

4. Simutis, R., D. Dilijonas, L. Bastina, J. Friman and P. Drobinov, “Optimization of Cash Management for ATM Network”, *Information Technology and Control*, Vol. 36, No. 1A, pp. 117–121, 2007.

5. Arora, N. and J. K. R. Saini, “Approximating Methodology: Managing Cash in Automated Teller Machines using Fuzzy ARTMAP Network”, *International Journal of Enhanced Research in Science Technology and Engineering*, Vol. 3, pp. 318–326, 2014.

6. Venkatesh, K., V. Ravi, A. Prinzie and P. D. V. den, “Cash Demand Forecasting in ATMs by Clustering and Neural Networks”, *European Journal of Operational Research*, Vol. 232, pp. 383–392, 2014.

7. Teddy, S. and S. Ng, “Forecasting ATM Cash Demands Using a Local Learning Model of Cerebellar Associative Memory Network”, *International Journal of Forecasting*, Vol. 27, pp. 760–776, 2011.

8. Ramirez, C. and G. Acuna, “Forecasting Cash Demand in ATM Using Neural

Networks and Least Square Support Vector Machine", *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications - 16th Iberoamerican Congress*, pp. 515–522, 2011.

9. Efendigil, T., S. Önüt and C. Kahraman, "A Decision Support System for Demand Forecasting with Artificial Neural Networks and Neuro-fuzzy Models: A Comparative Analysis", *Expert Systems with Applications*, Vol. 36, pp. 6697 – 6707, 2009.

10. Chen, Y., B. Yang and J. Dong, "Time-series Prediction using a Local Linear Wavelet Neural Network", *Neurocomputing*, Vol. 69, pp. 449–465, 2006.

11. Bar Ilan, A., D. Perry and W. Stadje, "A Generalized Impulse Control Model of Cash Management", *Journal of Economic Dynamics and Control*, Vol. 28, pp. 1013–1033, 2014.

12. Yao, J. S., M. S. Chen and H. F. Lu, "A Fuzzy Stochastic Single-Period Model for Cash Management", *European Journal of Operational Research*, Vol. 170, pp. 72–90, 2006.

13. Moraes, M. B. d. C. and M. S. Nagano, "Cash balance management: A Comparison Between Genetic Algorithms and Particle Swarm Optimization", *Acta Scientiarum. Technology*, Vol. 34, No. 4, pp. 373–379, 2012.

14. Altunoglu, Y., *Cash Inventory Management at Automated Teller Machines under Incomplete Information*, Master's Thesis, Bilkent University, 2010, `http://www.thesis.bilkent.edu.tr/0004020.pdf`, accesed at September 2014.

15. Ekinci, Y., J.-C. Lu and E. Duman, "Optimization of ATM Cash Replenishment with Group-Demand Forecasts", *Expert Systems with Applications*, Vol. 42, pp. 3480–3490, 2015.

16. Kurdel, P. and J. Sebestyenova, "Modeling and Optimization of ATM Cash Replenishment", *Latest Trends in Information Technology*, pp. 322–327, 2012.

17. Zheng-yu, D., Y. Dong-yuan and W. Shang, “An Improved Genetic Algorithm for Time Dependent Vehicle Routing Problem”, *2012 2nd International Conference on Future Computer and Communication*, Vol. 1, pp. 835 – 839, 2010.

18. Toit, D. J., *ATM Cash Management for a South African Retail Bank*, Master’s Thesis, Stellenbosch University, 2011, `http://scholar.sun.ac.za/handle/10019.1/17977`, accesed at December 2014.

19. Hsieh, F. S. and H. W. Huang, “Decision Support for Vehicle Routing Problem with Arbitrary Pickup/Delivery Points”, *IEEE 12th International Conference on Networking, Sensing and Control*, pp. 597–602, 2015.

20. Wu, J. and Y. Tan, “A Particle Swarm Optimization Algorithm for Grain Logistics Vehicle Routing Problem”, *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*, Vol. 3, pp. 364 – 367, 2009.

21. Joshi, S. and S. Kaur, “Nearest Neighbor Insertion Algorithm for Solving Capacitated Vehicle Routing Problem”, *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 86–88, 2015.

22. Gregory, P., “New Planning Benchmarks: Vehicle Routing Problem Variants”, *Proceedings of PlanSIG, 28th workshop of the UK special interest group on planning and scheduling joint meeting with the 4th Italian workshop on planning and scheduling*, 2010.

23. Chen, R.-M., F.-R. Hsieh and D.-S. Wu, “Heuristics Based Ant Colony Optimization for Vehicle Routing Problem”, *2012 7th IEEE Conference on Industrial Electronics and Applications*, pp. 1039–1043, 2012.

24. Yan, X. and W. Wu, “Hysteretic Optimization for the Capacitated Vehicle Routing Problem”, *2012 9th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pp. 12 – 15, 2012.

25. Pourghaderi, A., R. Tavakkoli-Moghaddam, M. Alinaghian and B. Beheshti-Pour, “A Simple and Effective Heuristic for Periodic Vehicle Routing Problem”, *2008 IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 133–137, 2008.

26. Wang, Y. and M. Lang, “Study on the Model and Tabu Search Algorithm for Delivery and Pickup Vehicle Routing Problem with Time Windows”, *IEEE International Conference on Service Operations and Logistics, and Informatics*, Vol. 1, pp. 1464 – 1469, 2008.

27. Peng, Y. and X. Wang, “Research on a Vehicle Routing Schedule to Reduce Fuel Consumption”, *2009 International Conference on Measuring Technology and Mechatronics Automation*, Vol. 3, pp. 825 – 827, 2009.

28. Mostafa, N. A. and A. A. B., “The Production-Inventory-Distribution-Routing Problem: An Integrated Formulation and Solution Framework”, *International Conference on Industrial Engineering and Operations Management*, pp. 1–10, 2015.

29. Chandra, S., M. Christiansen and K. Fagerholt, “Combined Fleet Deployment and Inventory Management in Roll-on/Roll-off Shipping”, *Transportation Research Part E: Logistics and Transportation Review*, pp. 1–13, 2016.

30. Ye, W. and F. You, “A Computationally Efficient Simulation-Based Optimization Method with Region-wise Surrogate Modeling for Stochastic Inventory Management of Supply Chains with General Network Structures”, *Computers and Chemical Engineering*, Vol. 87, pp. 164–179, 2016.

31. Azuma, R. M., G. P. Coelho and F. J. V. Zuben, “Evolutionary Multi-Objective Optimization for the Vendor-Managed Inventory Routing Problem”, *2011 IEEE Congress of Evolutionary Computation (CEC)*, pp. 1457 – 1464, 2011.

32. Mo, D. Y., M. M. Tseng and R. K. Cheung, “Meeting Response Time Requirement in Spare Part Support Operation from Inventory Management Perspective”,

2010 8th International Conference on Supply Chain Management and Information Systems (SCMIS), pp. 1–6, 2010.

33. Armenise, R., C. Birtolo, E. Sangianantoni and L. Troiano, “A Generative Solution for ATM Cash Management”, *2010 International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, pp. 349–356, 2010.

34. Li, W. and M. J. Zuo, “Joint Optimization of Inventory Control and Maintenance Policy”, *Annual Reliability and Maintainability Symposium*, pp. 317–322, 2007.

35. Jun-jun, G. and K. Ting, “A Joint Decision Model of Inventory Control and Promotion Optimization based on Demand Forecasting”, *2009 IEEE International Conference on Automation and Logistics*, pp. 119–123, 2009.

36. Chen, L., Y. Feng and J. Ou, “Joint Management of Finished Goods Inventory and Demand Process for a Make-to-Stock Product: A Computational Approach”, *IEEE Transactions on Automatic Control*, Vol. 51, pp. 258–273, 2006.

37. Sheikh, H. F., “Efficient Heuristics for Joint Optimization of Performance, Energy, and Temperature in Allocating Tasks to Multi-core Processors”, *International Green Computing Conference (IGCC)*, pp. 1–8, 2014.

38. Wang, W., “Supply-demand Joint Inventory—A System Dynamics Model”, *2011 International Conference on Intelligent Computation Technology and Automation*, Vol. 1, pp. 1149–1151, 2011.

39. Esparcia-Alcazar, A. I., E. Alfaro-Cid, P. Garcia-Sanchez, A. Martinez-Garcia, J. Merelo and K. Sharman, “An Evolutionary Approach to Integrated Inventory and Routing Management in a Real World Case”, *IEEE Congress on Evolutionary Computation*, pp. 1–7, 2010.

40. Smith, J. and Z. C. Taşkın, *A Tutorial Guide to Mixed-Integer Programming Models and Solution Techniques*, 2007,

http://www.ie.boun.edu.tr/ taskin/pdf/IP_tutorial.pdf, accesed at June 2015.

41. Miller, C., A. Tucker and R. Zemlin, “Integer Programming Formulation of Traveling Salesman Problems”, *Journal of ACM*, Vol. 7, pp. 326–329, 1960.

42. Pataki, G., “Teaching Integer Programming Formulations using the Traveling Salesman Problem”, *2003 Society for Industrial and Applied Mathematics*, Vol. 45, No. 1, pp. 116–123, 2003.

43. Rasmussen, R., “TSP in Spreadsheets – a Guided Tour”, https://www.economicsnetwork.ac.uk/iree/v10n1/rasmussen.pdf, accesed at May 2015.

44. Kokash, N., “An Introduction to Heuristic Algorithms”, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.105.8050&rep=rep1&type=pdf, accesed at March 2016.

45. “IBM ILog CPLEX Optimizer”, http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/, accesed at December 2014.

46. “IBM ILog CPLEX Manual”, http://www.ibm.com/support/knowledgecenter/SSSA5P_12.3.0/ilog.odms.cplex.help/Content/Optimization/Documentation Optimization_Studio/_pubskel/cplex_excel_users_manual2497.html, accesed at January 2016.

47. Salvatore, J., *Bipartite Graphs and Problem Solving*, 2007, http://www.math.uchicago.edu/ may/VIGRE/VIGRE2007/REUPapers/FINALAPP/Salvatore.pdf, accesed at February 2015.

48. Pulleyblank, R., *Faces of Matching Polyhedra*, Ph.D. Thesis, 1973.

49. Schrijver, A., *Combinatorial Optimization: Polyhedra and Efficiency*, Springer,

2003.

50. “Google Maps Distance Matrix API”, https://developers.google.com/maps/documentation/distance-matrix/intro#DistanceMatrixRequests, accesed at January 2016.