Joint Optimization of Cash Management and Routing for New-Generation Automated Teller Machine Networks

Şeyma Batı and Didem Gözüpek

***Abstract*—Cash-related costs constitute a large portion of management cost of an automated teller machine (ATM) network. Cash should be delivered to or picked-up 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, also called recycle ATMs, have a single cassette 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 paper, 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 money to deliver 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.**

***Index Terms*—Automated teller machine (ATM) network, cash management, heuristics, integer programming, joint optimization.**

I. Introduction

CASH management for automated teller machines (ATMs) 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]. Studies using actual ATM investment data [2] suggest that ATM usage has a positive impact on the cost efficiency of the banks. 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: 1) *classical* and 2) *recycle ATMs* [3]. While classical ATMs have separate cassettes for cash withdrawal and deposit, recycle ATMs, also called as new-generation ATMs, have a single cassette 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 CIT a certain amount of money for each visit of an ATM 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 [4].

In this paper, we formulate an optimization problem whose objective is to minimize the cash management cost. We consider a system consisting of CIT vehicles and cash centers as well as classical and recycle ATMs. Furthermore, we decide on the route of the CIT vehicles. Since armored vehicles of the CIT have a certain upper limit for the amount of money to carry due to reasons such as security, CIT vehicles' routes should be determined together with the amount of money to be delivered to or picked-up from the ATMs. To the best of our knowledge, this paper is one of the few studies that focus on recycle ATMs and also on the joint optimization of cash management and routing. Furthermore, to the best of our knowledge, this paper is the first study that focuses on an ATM network consisting of both classical and recycle ATMs and, even more importantly, provides the optimum decision on which classical ATMs need to be replaced with recycle ATMs in order to minimize the total cost of joint cash management and routing.

The optimization problem we formulate in this paper is an integer linear programming (ILP) problem that jointly optimizes cash decisions, i.e., when to deliver how much 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

Manuscript received March 20, 2017; accepted May 17, 2017. This paper was recommended by Associate Editor Z. Liu. *(Corresponding author: Didem Gözüpek.)*

Ş. Batı is with the Department of Computer Engineering, Bogazici University, 34342 Istanbul, Turkey (e-mail: seyma.ketenci@boun.edu.tr).

D. Gözüpek is with the Department of Computer Engineering, Gebze Technical University, 41400 Kocaeli, Turkey (e-mail: didem.gozupek@gtu.edu.tr).

Digital Object Identifier 10.1109/TSMC.2017.2710359

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 paper is organized as follows. In Section II we explain the motivation for this paper and summarize the related work in the literature as well as our contributions. We formulate our optimization problem as an ILP in Section III and describe our proposed heuristic algorithm in Section IV. We present the simulation results in Section V and then conclude this paper in Section VI.

II. Motivation and Related Work

For efficient cash management in an ATM network, a 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 CIT vehicles 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 delivery 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 paper, 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 delivered to or picked-up 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. Our model is suitable for a business environment where the banks also determine the route of the CIT vehicles; in other words, CIT vehicles operate simply as taxis to transport money. The money paid by the bank to the CIT consists of two terms: 1) a fixed amount for each CIT vehicle (which is implicit in our formulation) and 2) a certain amount for each visited ATM per day per CIT vehicle (which is explicit in our formulation). Note that our model does not contain proportional costs, i.e., the second term does not depend on the amount of cash delivered to the ATM. Furthermore, while the decision of the route of the CIT vehicles and the amount of cash to be delivered to or picked up from each ATM are operational level decisions, which typically need to be solved every week, the decision on whether to deploy a recycle ATM is a rather strategical decision. We integrate these three terms together in our model in order to serve a guideline to the banks on replacing which ATMs with recycle ATMs. For instance, if a replacement decision is made for a certain ATM is made for most weeks, then the bank can take such an action. If the bank wishes to see the performance when no ATM or a particular subset of the ATMs is replaced with a recycle ATM, then a simple modification on our optimization problem formulation (which we explain in Section III) can serve this purpose.

Inventory routing problem [5] is comprised of the integration of inventory management and vehicle routing so that inventory control and routing decisions are made simultaneously, while the pick-up and delivery problem [6] focuses on the collection and distribution of one/several commodities from/to a set of locations. The problem we formulate in this paper resembles the inventory routing problem with pick-ups and deliveries, which combines features of these two main problems. Our main distinction from the inventory routing problem with pick-ups and deliveries is that the routing cost in our problem does not depend on the route length and our model decides on what the ATM type (classical/recycle) should be for each ATM in order to minimize the total cost.

Before discussing the related work in the literature, let us point out that in the remainder of this paper, *withdraw* and *deposit* refer to the customer actions, whereas *deliver* and *pick-up* refer to the CIT vehicle actions.

Economists have long recognized the similarity between cash management and managing the inventory of some physical quantity. In this perspective, the Baumol model [7] has been dominant for analyzing the transactions demand at the micro level. Miller and Orr [8] then defined the cash balance as having an uneven fluctuation by characterizing a random variable and proposed a stochastic model. The work in [9] also focuses on a stochastic cash balance problem and formulate a linear programming model for it, whereas the paper in [10] studies the problem of minimizing the expected time average cash balance subject to the constraint that the probability that all demands are satisfied is at least some given number. Elton and Gruber [11] proposed a dynamic programming formulation for a cash balance problem in which stochastic changes in the cash level can be positive or negative. The work in [12] proposes a simple mathematical model for cash management at the bank branches, in particular both at the branch ATMs and at the cash desks. Another work in [13] presents a general model of cash management, viewed as an impulse control problem for a stochastic money flow process. This process is represented by a superposition of a Brownian motion and a compound Poisson process, controlled by two-sided target-trigger policies. The study in [14] 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 work in [15] focuses on the joint optimization for banks and CIT firms by using Pareto-improvement recontracting schemes based on a Baumol-type cash demand forecast.

Besides cash management in other areas, ATM cash management has also received significant attention. Most studies about ATM cash management in the literature focused on estimation of daily cash demands for ATMs, which is challenging due to the heteroscedasticity of such time series. For instance, the work in [1] proposed 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 delivery costs, and constant ATM-service costs, while neglecting the routing of CIT vehicles. Furthermore, unlike this paper, they do not take the recycle ATMs into account. Venkatesh *et al.* [16] used neural networks to predict cash demand for groups of ATMs with similar day-of-the week cash demand patterns. The work in [17] introduces a local learning model of the pseudo self-evolving cerebellar model articulation controller associative memory network for ATM cash demand forecasting. There are also other methods used for demand forecasting. For instance, the study in [18] uses neural networks and least square support vector machines, while the work in [19] also uses artificial neural networks and neuro-fuzzy models and the one in [20] uses a local linear wavelet neural network for time-series prediction. While most works in the literature studied forecasting and ATM cash replenishment policy separately, the work in [21] proposes an integrated approach where the ATMs in near-by locations are grouped into clusters and the optimal replenishment time interval as well as data forecasting are applied to the clusters. Another work in [22], elaborates on a model based on the combination of neural networks and multiagent technology for predicting future cash demand.

Besides demand forecasting, cash management literature focused on other aspects as well. For instance, the work in [4] focuses on cash management in ATMs and 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 ILP problem, whereas the mid-term model with fixed and staircase costs leads to a multistage stochastic problem. Unlike this paper, the work in [4] does not focus on the routing of CIT vehicles. Moraes and Nagano [23] developed 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. Again, unlike this paper, the work in [23] does not focus on the routing of CIT vehicles. The work in [24] 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. Unlike this paper, they do not consider the routing of CIT vehicles (since replenishment of the ATMs is impossible) or recycle ATMs. The study in [25] proposes a class of adaptive data-driven policies for a stochastic inventory control problem in order to provide a robust method for the cash deployment strategies of the ATMs. The paper in [26] studies an ATM network consisting of ATMs of several banks and it allocates the total transaction cost arising in the network among the participating banks by modeling the situation as a cooperative game with transferable utility.

TABLE I
INPUT VARIABLES

$\mathcal{N}$	Set of ATMs, where $\mathcal{N}$ = 1, 2,..., N
$\mathcal{M}$	Set of CIT vehicles, 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 the CIT for the visit of ATM i on day h by CIT vehicle k
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

There are few studies in the literature that focus on the routing of CIT vehicles. However, most of these studies do not take the cash management into account. For instance, the work in [27] models the routing of CIT vehicles as some type of vehicle routing problem and addresses it using a genetic algorithm. Besides, the work in [28], treat the routing of CIT vehicles and cash management as two separate problems, while focusing mainly on demand forecasting in cash management. Opasanon and Lertsanti [29] aimed at identifying logistics issues arising from relocating the main distribution center from the view of the company's policy makers. They use the analytic hierarchy process to evaluate and rank the importance of the logistics issues according to the needs and requirements of the company's policy makers. The work in [30] focuses on joint vehicle routing and inventory management in ATM networks; however, they do not take the recycle ATMs into account, and therefore, do not make a decision on which ATMs should be replaced by recycle ATMs. The study in [31] focuses on joint vehicle routing and inventory management of recirculation (recycle) ATMs. Unlike this paper, they focus merely on recycle ATMs, while we consider an ATM network consisting of both classical and recycle ATMs in addition to making the important decision of which classical ATMs should be replaced by recycle ATMs in order to minimize the total cost.

III. 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 delivered 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 delivered 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

TABLE II
DECISION VARIABLES

x_{ijhk}	$= \begin{cases} 1, & \text{if ATM } j \text{ is visited after ATM } i \text{ by CIT vehicle } k \\ & \text{on day } h \\ 0, & \text{otherwise} \end{cases}$
y_{ihk}	$= \begin{cases} 1, & \text{if ATM } i \text{ is visited by CIT vehicle } k \text{ on day } h \\ 0, & \text{otherwise} \end{cases}$
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 delivered to ATM i on day h by CIT vehicle k
a_{ihk}	Cash amount to be picked up from ATM i on day h by CIT vehicle k
z_{ih}	Cash amount to be delivered to ATM i on day h
a_{ih}	Cash amount to be picked up 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}	Total cash picked up from recycle ATM i until the end of day h, i.e., $\phi_{ih} = r_i \times \sum_{e=1}^{h} a_{ie}$
$\Upsilon_{ihh'}$	Variable showing whether ATM i is visited by a CIT vehicle from day h to day h', i.e., $\Upsilon_{ihh'} = \prod_{e=h}^{h'} \sum_{k=1}^{M} (1 - y_{iek})$
u_i	Extra variables used for subtour elimination

for each visit of an ATM are input parameters to our optimization problem. Tables I and II show the input and decision variables, respectively, of our ILP formulation.

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. The first term of (1) models the idle cash cost, which is proportional to the daily interest rate, whereas, the second term of (1) models the logistic cost, which is due to the money paid to the CIT firm for ATM visits. The last term of (1) models the cost of deploying a recycle ATM. Accordingly, the objective function of our ILP formulation is as follows:

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

Note that in the objective function, cost parameters may depend on the ATM, the day of the week, and vehicle type (see the definition of c_{ihk}). The reasons for these dependencies are the following: vehicle type may change the amount of gas consumed and can change the expenses of the CIT firm. Moreover, the more ATMs are visited, the more costly the tour will be for the CIT. Visiting some ATMs may be costlier than visiting other ATMs since, for instance, they may be in a less secure location necessitating extra security measures to be taken for the CIT vehicles, which are armored vehicles. The day of the week may also be an important factor since, for instance, working on holidays such as Saturday obligates the CIT to pay overtime to the CIT vehicle drivers and hence increases the cost charged to the bank. CIT vehicles have to start the route from a presepecified center node, which is usually the central bank, because the cash to be delivered to the ATMs must be picked-up from a cash center. Note that the second term in the objective function starts from $i = 0$, which refers to the index of the center node. c_{ihk} values for $i = 0$ are used to model the fixed money charged to the CIT firm; i.e., the fixed money per week can be divided by the number of days to calculate the c_{ihk} values for $i = 0$ for each day h. If it turns out as a result of the execution of our optimization formulation that a CIT vehicle does not visit any ATMs, in which case the route found for that CIT vehicle makes a tour without visiting any ATMs [due to Constraint (2)] and visiting only the center node, then only the fixed cost charged for that CIT vehicle contributes to the objective function. Note that this situation of CIT vehicles visiting only the center node is used only to model the fixed costs charged for that vehicle and it does not occur in reality.

We model the requirement that each CIT vehicle starts the route from the center node as follows:

$$\sum_{j=1}^{N} x_{0jhk} = 1; \ \forall k \in \mathcal{M}, h \in \mathcal{H}. \quad (2)$$

CIT vehicles 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. At each node, the incoming flow of CIT vehicles must be same as the outgoing flow. Therefore, we need a flow conservation constraint as follows:

$$\sum_{i=0}^{N} x_{iphk} - \sum_{j=0}^{N} x_{pjhk} = 0; \ \forall p \in \mathcal{N}, k \in \mathcal{M}, h \in \mathcal{H}. \quad (3)$$

Note that if a vehicle makes a tour without visiting any ATMs, then since the corresponding x_{0jhk} value equals 1, this will incur a cost in the second term of the objective function because the indices in the second term of the objective function start from 0. Since the objective function minimizes the cost and making a tour without visiting any ATM does not bring any benefit in terms of the objective function value, this situation will be avoided by the ILP. Indeed, not using all the available vehicles bring cost gains due to the same reason.

Visiting the same set of ATMs using more vehicles on a given day incurs more cost since the cost of visiting the center node also contributes to the objective function.

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

$$\sum_{k=1}^{M} y_{ihk} \leq 1;\ \ \forall i \in \mathcal{N}, h \in \mathcal{H}. \tag{4}$$

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

$$y_{ihk} = \sum_{j=0}^{N} x_{jihk};\ \ \forall i \in \mathcal{N}, h \in \mathcal{H}, k \in \mathcal{M}. \tag{5}$$

The cash capacity v_k of the CIT vehicle k stems from its physical and security requirements. The cash amount carried by CIT vehicle k on day h should not exceed its capacity

$$\sum_{i=1}^{N} z_{ihk} + \sum_{i=1}^{N} a_{ihk} \leq v_k;\ \ \forall h \in \mathcal{H}, k \in \mathcal{M}. \tag{6}$$

Equation (6) ensures that the vehicle capacity constraints are satisfied. The first term represents the total amount of cash to be delivered to all ATMs on the route, while the second term represents the total amount of cash to be picked-up from all ATMs on the route. In the worst case, the sum of these two terms should not exceed the vehicle capacity. Since cash is delivered in cassettes, even if a certain amount of cash is picked up from a particular ATM, it does not matter when delivering cash to the next ATM as to the initial amount of money to be loaded to the CIT vehicle.

The amount of cash delivered to ATM i on day h is equal to the sum of the cash delivered to ATM i on day h by all CIT vehicles

$$z_{ih} = \sum_{k=1}^{M} z_{ihk};\ \ \forall i \in \mathcal{N}, h \in \mathcal{H}. \tag{7}$$

The amount of cash picked-up from ATM i on day h is equal to the sum of cash picked-up from ATM i on day h by all CIT vehicles

$$a_{ih} = \sum_{k=1}^{M} a_{ihk};\ \ \forall i \in \mathcal{N}, h \in \mathcal{H}. \tag{8}$$

Note here that since at most one CIT vehicle can visit an ATM on a certain day due to (4), at most one CIT vehicle can contribute to the summation in (7). The same is valid also for (8).

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

$$z_{ihk} \leq C_i \times y_{ihk};\ \ \forall i \in \mathcal{N}, h \in \mathcal{H}, k \in \mathcal{M}. \tag{9}$$

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

$$a_{ihk} \leq C_i \times y_{ihk};\ \ \forall i \in \mathcal{N}, h \in \mathcal{H}, k \in \mathcal{M}. \tag{10}$$

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 delivered 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 at the end of day h is equal to the difference between the total amount of cash delivered and deposited to ATM i until the end of day h and the total amount of cash withdrawn and picked-up from ATM i until the end of day h. These constraints can be modeled as follows:

$$w_{ih} = \sum_{e=1}^{h} (z_{ie} - W_{ie}) + \left(\sum_{e=1}^{h} r_i \times D_{ie}\right) - \phi_{ih};\ \ \forall i \in \mathcal{N}, h \in \mathcal{H}. \tag{11}$$

For recycle ATMs, CIT vehicle can pick-up more than necessary amount of cash residing in the withdrawal box while visiting the ATM. The term ϕ_{ih} refers to the cash amount picked-up from ATM i until the end of day h if ATM i is a recycle ATM (equals zero otherwise). Note that the definition of the decision variable $\phi_{ih} = r_i \times \sum_{e=1}^{h} a_{ie}$ poses a nonlinear relationship between the decision variables. We can linearize this relationship through the following set of constraints:

$$\phi_{ih} \leq C_i \times h \times r_i;\ \ \forall i \in \mathcal{N}, h \in \mathcal{H} \tag{12}$$

$$\phi_{ih} \leq \sum_{e=1}^{h} a_{ie};\ \ \forall i \in \mathcal{N}, h \in \mathcal{H} \tag{13}$$

$$\phi_{ih} \geq C_i \times h \times (r_i - 1) + \sum_{e=1}^{h} a_{ie};\ \ \forall i \in \mathcal{N}, h \in \mathcal{H}. \tag{14}$$

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 vehicle 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 vehicle. For recycle ATMs, since there is no separate deposit box, the remaining amount is equal to zero

$$d_{ih} \leq (1 - r_i) \times \sum_{e=1}^{h} (\Upsilon_{ieh} \times D_{ie});\ \ \forall i \in \mathcal{N}, h \in \mathcal{H}. \tag{15}$$

Note that in both cases, if a CIT vehicle visits an ATM, it picks up all the money residing in the deposit box. This behavior is due to the practical business situation as suggested by our industrial partner Provus, who informed us that banks (in Turkey) always prefer to pick-up all the money in the deposit box since they are very eager to earn money from the interest rates. Even if our formulation had modeled the case where the CIT vehicle had the option of leaving some of the money in the deposit box, the bank would always require them to pick-up all the money. Another issue is that note also that some money can be deposited in the ATM after the CIT vehicle visits the ATM and picks up all the money in the deposit box. In this case, the money residing in the ATM at the end of the day would be nonzero. Our model neglects this situation due to the following two reasons: first, as can also be seen from our simulation results with real data, deposit amounts in Turkey are very low, and if such a case occurs, this deposited money can be treated as if it were deposited at the beginning of the next day. Second, incorporating such level of detail to our model would obligate us to make the resolution of scheduling in minutes rather than days and therefore would tremendously increase the complexity of our formulation. In short, for practical reasons, we treat such money as money deposited on the next day.

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

$$\Upsilon_{ihh'} \leq 1 - y_{iek};\ \forall i \in \mathcal{N}, h, h' \in \mathcal{H}, h \neq h' \tag{16}$$

$$\Upsilon_{ihh'} \geq h - h' + \sum_{e=h}^{h'} \sum_{k=1}^{M} (1 - y_{iek});\ \forall i \in \mathcal{N}, h, h' \in \mathcal{H}, h \neq h'. \tag{17}$$

The time spent in traveling between ATMs and during the cash delivery process has to be smaller than the total number of working hours; i.e., CIT vehicle 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:

$$\sum_{i=0}^{N} \sum_{j=0}^{N} t_{ijh} \times x_{ijhk} + \sum_{i=1}^{N} \delta \times y_{ihk} \leq B. \tag{18}$$

We have to provide subtour elimination in the routing of CIT vehicles. We use the Miller–Tucker–Zemlin formulation of the traveling salesman problem [32], which has a simple implementation that introduces three constraints as follows:

$$u_1 = 1 \tag{19}$$

$$2 \leq u_i \leq N;\ \forall i \neq 1 \tag{20}$$

$$u_i - u_j + 1 \leq (N - 1) \times \left(1 - x_{ijhk}\right);\ \forall i \neq 1, \forall j \neq 1, \forall k, \forall h. \tag{21}$$

Note that if the bank wishes to see the results when a particular set of ATMs necessarily consists of recycle ATMs or classical ATMs, the constraints $r_i = 1$ or $r_i = 0$, respectively, can be added to the formulation.

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

$$x_{ijhk}, y_{ihk}, r_i, \Upsilon_{ihh'} \in \{0, 1\} \tag{22}$$

$$w_{ih}, d_{ih}, a_{ih}, z_{ih}, a_{ihk}, z_{ihk}, \phi_{ih}, \mu_{ijhk}, u_i \in \left\{\mathbb{Z}^+ \cup \{0\}\right\}. \tag{23}$$

Note that in the above formulation, we determine the amount of cash to be delivered by taking into account the cash capacity of the CIT vehicles due to security constraints. In practice, there may also be a limited number of possible cassette sizes also due to security issues [30]. Which banknote types to use for a given total cash amount is called the change making problem in [33]. As suggested by our industrial partner Provus Inc., we have not included this level of detail in our model since the banks usually have a certain strategy for this decision and they make this decision independent of the cash management and routing model.

IV. Proposed Algorithm

We propose in this section a polynomial-time heuristic algorithm to address our problem formulated in (1)–(23). ILP problems are known to be computationally difficult in general. Thus, we propose a heuristic algorithm for our formulated ILP problem.

In our proposed algorithm, by considering a fixed planning horizon of seven days, we first calculate the ATM visit days and cash amount by using exhaustive search to find the schedule that gives minimum cost among all possible alternatives. Then, for each CIT vehicle, we construct a route with candidate ATM nodes by using a variant of a vehicle routing problem. After that, we assign the ATMs to one of the candidate routes constructed previously by forming an edge-weighted bipartite graph between ATMs and CIT vehicles and solving an optimization problem with the objective of minimizing total cost. Lastly, we check the solution in terms of vehicle capacity constraint. If there is a route that violates the constraint, we update the pick-up values for each ATM on the route of that CIT vehicle. Indeed, we basically split the original model into four submodels and propose four separate algorithms for every submodel. We explain the four consecutive stages that form our overall algorithm in more detail as follows.

A. Calculation of ATM Visit Days and Cash Amount (Stage 1)

First, which day to visit and the amount of cash to be delivered 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 CIT cost 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:

$$tc_{ihk} = (w_{ih} \times f_h) + (c_{ihk} \times y_{ih}). \tag{24}$$

Total cost depends on which days the ATMs are visited (y_{ih}). Calculation of the cash amount to be delivered 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 six or seven days. If we consider seven days ahead, there are total of 2^7 possible solutions. In Algorithm 1, line 7 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 deliver to ATMs, the remaining amount of cash in ATM (w_{ih}), the idle cash cost (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 8, b_j shows the binary form of j where bit values indicate whether ATM is visited or not on that day. In line 9, the cash amount to be deliver to ATM for that possible solution is calculated by using the withdrawal and deposit amounts. In line 10, the remaining cash value for each day is calculated. Similarly, delivery and remaining amounts for a recycle ATM is calculated in lines 11 and 12. Line 13 calculates the idle cash cost, which is proportional to the interest rate and the remaining amount of cash in the ATM

Algorithm 1 Stage 1 of the Proposed Algorithm

1: **procedure** GET POSSIBLE SOLUTIONS
2: **Require:** $W_{ih}, D_{ih}, c_{ihk}, f_h$
3: **Ensure:** y_{ih}, r_i
4: $tc_{ihk}, mc, cc, tcr_{ihk}, mcr, ccr \leftarrow 0$
5: $tc_{ihk}^{min}, tcr_{ihk}^{min} \leftarrow$ A very large number
6: **for** each ATM i **do**
7: **for** $j = 0$ `to` 2^7 **do**
8: $b_j \leftarrow binary(j)$
9: findDeliveryAmount(b_j, W_{ih}, D_{ih})
10: findRemainingAmount(w_{ih})
11: findDeliveryAmountRecycle(b_j, W_{ih}, D_{ih})
12: findRemainingAmountRecycle(wr_{ih})
13: $mc \leftarrow w_{ih} \times f_h$
14: $cc \leftarrow c_{ihk} \times y_{ih}$
15: $tc_{ihk} \leftarrow mc + cc$
16: $mcr \leftarrow wr_{ih} \times f_h$
17: $ccr \leftarrow c_{ihk} \times y_{ih}$
18: $tcr_{ihk} \leftarrow mcr + ccr$
19: **if** $tc_{ihk} \leq tc_{ihk}^{min}$ **then**
20: $tc_{ihk}^{min} \leftarrow tc_{ihk}$
21: $b_j^{min} \leftarrow b_j$
22: **end if**
23: **if** $tcr_{ihk} \leq tcr_{ihk}^{min}$ **then**
24: $tcr_{ihk}^{min} \leftarrow tcr_{ihk}$
25: **end if**
26: **for** each h **do**
27: $y_{ih} \leftarrow b_j$
28: **end for**
29: **end for**
30: **if** $tcr_{ihk}^{min} \leq tc_{ihk}^{min}$ **then**
31: $r_i \leftarrow 1$
32: **end if**
33: **end for**
34: **return** y_{ih}, r_i
35: **end procedure**

at the end of the day. Line 14 calculates the CIT cost, which is equal to zero if the ATM is not visited on that day. Line 15 calculates the total cash management cost for that day, which is stated in (24). In lines 16–18, money, CIT and total cost for a recycle ATM are calculated. In lines 19–22 the solution that gives the minimum total cost among all possible 2^7 solutions is stored. In lines 30 and 31, we compare total cost for classical and recycle ATM and we decide on ATM being recycle or not (r_i) by choosing the one that gives lower cost. The algorithm is executed for each day. As the output, we find the days (within the N days) to visit the ATM, what amount of cash to be delivered to that ATM and ATM type (classical/recycle). The same algorithm is executed for each ATM.

B. Candidate Route Construction for CIT Vehicles (Stage 2)

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 CIT vehicles that pass through these predetermined ATMs. The CIT vehicles' routes should satisfy the following criteria: 1) the distribution of the cash to ATMs must be completed within the working hours (B) and 2) CIT vehicles must start the route from a center node and return to it within the given time period.

The problem in stage 2 is actually a variant of the capacitated vehicle routing problem (CVRP). However, apart from having vehicle capacity constraints, we also have a time constraint for the vehicles. Therefore, traditional approaches to CVRP is insufficient for our problem.

In Algorithm 2, lines 12 through 31 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 32 through 44. The algorithm marks the selected ATMs as it proceeds; we keep two separate lists for the unselected ATMs: 1) $list_{all}$ keeps the list of ATMs not selected by any of the CIT vehicles and 2) $list_k$ keeps the list of ATMs not selected by CIT vehicle k. In order to prevent cycles nodes are removed from the lists as follows: for the first selection of each CIT vehicle, the node closest to the center from $list_{all}$ is picked and the selected node is removed from that list (line 15). By doing so, each CIT vehicle selects the one that is not been selected by other CIT vehicles and initialize different routes. After first node, each CIT vehicle selects the node from its own $list_k$ that is further from the center among the two closest nodes to the current node and removes the node from that list (line 23). For the second half of the working hours, the algorithm selects the node from $list_k$ that is closer to the center among the two closest nodes to the current node (line 33). For each CIT vehicle, a route with candidate ATM nodes is constructed. Stage 2 is executed just once for each CIT vehicle. For each route, we keep the number of the selected nodes in the first half of the working hours as *numCount* and at the beginning of the iteration we set the limit for that variable to a very large number, *numNodesLimit*. At the end of the iteration, if we cannot reach center node, we update *numNodesLimit* as *numCount* − 1 (line 41) and repeat the iteration until the duration of the route shrinks to below the allowed time. x_{ijk} in line 49 is the binary variable showing whether the route of CIT vehicle k includes the edge from ATM i to j.

Observe that some ATMs may not be selected at the end of stage 2. When the number of vehicles is insufficient, instead of returning an infeasible solution, our algorithm selects a subset of ATMs. In our simulations, we evaluate the performance of our algorithm by also taking the percentage of visited ATMs into account (see Fig. 1).

C. Assignment of ATMs to the CIT Vehicles (Stage 3)

As the output of the second stage, different route sets are given; i.e., for each CIT vehicle, a route with candidate ATM nodes are constructed. In other words, these ATMs are candidate for being visited on the route sets that are passing through it. If an ATM is decided to be visited on a particular day as a result of stage 1, then it needs to be assigned to one of the routes that pass through it on that day so that it can be visited

Algorithm 2 Stage 2 of the Proposed Algorithm

procedure CIT VEHICLES' ROUTES
Require: t_{ij}, B
Ensure: x_{ijk}
$list_{all} \leftarrow \mathcal{N}$
for each CIT vehicle $k \in \mathcal{M}$
    $totalTime \leftarrow 0$
    $list_k \leftarrow \mathcal{N}$
    $i \leftarrow 0$
    $numNodesLimit \leftarrow$ A very large number
    label1
    while (**true**)
        while (**true**)
            $numCount \leftarrow 0$
            if ($i = 0$ and $totalTime + t_{0j} \leq B/2$)
                Pick the node j from $list_{all}$ closest to the center node
                $x_{0jk} \leftarrow 1$
                $list_k \leftarrow list_k - j$
                $list_{all} \leftarrow list_{all} - j$
                $totalTime \leftarrow totalTime + t_{0j}$
                $numCount \leftarrow numCount + 1$
                $i \leftarrow j$
            else
                Pick the node j from $list_k$ that is further from the center among the two closest nodes to the current node i
                if ($totalTime + t_{ij} \leq B/2$ and $numCount \leq numNodesLimit$)
                    $x_{ijk} \leftarrow 1$
                    $list_k \leftarrow list_k - j$
                    $totalTime \leftarrow totalTime + t_{ij}$
                    $numCount \leftarrow numCount + 1$
                    $i \leftarrow j$
                else break
        end while
        while (**true**)
            Pick the node j from $list_k$ that is closer to the center among the two closest nodes to the current node i
            if ($totalTime + t_{ij} \leq B$)
                $x_{ijk} \leftarrow 1$
                $list_k \leftarrow list_k - j$
                $totalTime \leftarrow totalTime + t_{ij}$
                $i \leftarrow j$
            else
                if ($j \neq 0$)
                    $numNodesLimit \leftarrow numCount - 1$
                    break label1
                end if
                else break label2
        end while
    end while
    label2
end for
return x_{ijk}
end procedure

by the CIT vehicle of that route. Each such ATM must be assigned to exactly one of the routes that passes through it. The routes might intersect; however, each ATM must be visited and served by exactly one CIT vehicle; 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 $G = (\mathcal{A}, \mathcal{R}, E)$, where $\mathcal{A} = \{1, \ldots, A\}$ is the set of ATMs, $\mathcal{R} = \{1, \ldots, R\}$ 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 I denoting the money paid to the CIT. Since stage 2 determines for each CIT vehicle a route with candidate ATMs, each route here corresponds to a CIT vehicle 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 I) and the working hour limit (B in Table I). Let Y_{hk} be the total number of ATMs that are only on the route of CIT vehicle k on day h and X_{hk} be the total amount of time spent in traveling between the ATMs on the route of CIT vehicle k on day h. Note that Y_{hk} and X_{hk} can be stated as follows:

$$Y_{hk} = \sum_{i=0}^{N} y_{ihk}; \ \forall h \in \mathcal{H}, k \in \mathcal{M} \tag{25}$$

$$X_{hk} = \sum_{i=0}^{N} \sum_{j=0}^{N} x_{ijhk} \times t_{ijh}; \ \forall h \in \mathcal{H}, k \in \mathcal{M}. \tag{26}$$

Then we calculate Δ_r as follows:

$$\Delta_r = \frac{B - (\delta \times Y_{hk}) - X_{hk}}{\delta}. \tag{27}$$

Note that Δ_r gives the maximum number of ATMs that can be included into route r. Furthermore, since stage 2 ensures that the total time spent on the route cannot exceed B, Δ_r cannot be negative.

We then solve the ILP in (28)–(30). Let x_{ar} be a binary decision variable related to y_{ihk} in Table II 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 vehicle 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 (28) 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 (1) in Section III

$$\min \sum_{a=1}^{A} c_r \times x_{ar} \quad \text{s.t.} \tag{28}$$

The following constraint ensures that each ATM is assigned to only one route:

$$\sum_{r=1}^{R} x_{ar} = 1; \ \forall a \in A. \tag{29}$$

The following constraint ensures that at most Δ_r ATMs can be assigned to each route r:

$$\sum_{a=0}^{A} x_{ar} \leq \Delta_r; \ \forall r \in R. \tag{30}$$

Algorithm 3 Stage 4 of the Proposed Algorithm

1: **procedure** CHECK VEHICLE CAPACITY CONSTRAINT
2: **Require:** z_{ih}, a_{ih}, v_k
3: **Ensure:** a_{ih}
4:   **for** each CIT vehicle k **do**
5:     **for** each day h **do**
6:       **if** $\sum_{i=0}^{N}(a_{ih} - z_{ih}) \leq v_k$ **then**
7:         **for** each ATM i **do**
8:           **if** $(r_i = 0)$
9:             $a_{ih} \leftarrow \frac{(v_k - \sum_{i=0}^{N}(a_{ih} - z_{ih})) \times a_{ih}}{\sum_{i=0}^{N} a_{ih}}$
10:           **end if**
11:         **end for**
12:       **end if**
13:     **end for**
14:   **end for**
15:   **return** a_{ih}
16: **end procedure**

D. Vehicle Capacity Constraint (Stage 4)

In the last stage of the algorithm, we rearrange the pick-up amounts of cash if the security constraint is not satisfied, i.e., if the amount of cash in the CIT vehicle exceeds the vehicle capacity. We simply decrease the amount of cash picked-up from the classical ATMs so that the total amount of cash in the CIT vehicle adheres to the vehicle capacity. We decrease the cash amount for each classical ATM proportional to their pick-up values.

Note that in stage 4, for classical ATMs since the money can be picked up only from the deposit box, this action cannot cause the ATM to run out of cash. For recycle ATMs, on the other hand, this is not necessarily the case. Therefore, we execute line 9 in Algorithm 3 only for classical ATMs.

E. Computational Complexity of Our Heuristic Algorithm

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 vehicle or not; therefore, there are a total of 2^7 alternatives since we take the scheduling length as seven days. In lines 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 (seven days), this step takes linear time.

In stage 2, for each CIT vehicle, 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 vehicle 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 (24)–(29) 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 [34], [35]. 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 [34]. 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 non-negative, then the corresponding maximization problem gives our desired solution in polynomial time.

V. SIMULATION RESULTS

In this section, 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 (1)–(21) 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. As the problem size gets larger, CPLEX running times become prohibitively high. Therefore, we set an upper time limit to CPLEX. When we set an upper time limit, if CPLEX finds an optimal solution within this time limit, then CPLEX returns an optimal solution; otherwise, it returns the solution it has found up until that time as well as the resulting gap value. 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. The default value of the gap parameter (*epgap*) is 0.0001 and it can take any value between 0.0 and 1.0. In our experiments, we set the upper time limit to 3 h and observe that the maximum gap value resulting from our experiments is 0.01.

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 ATMs. We obtain the travel times between each pair of ATMs by using Google Maps Distance Matrix API. 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. Note here that in a real implementation, demand forecasting should be implemented in order to estimate the deposit and withdrawal amounts of the ATMs and the output of demand forecasting should be fed as input to our optimization problem.

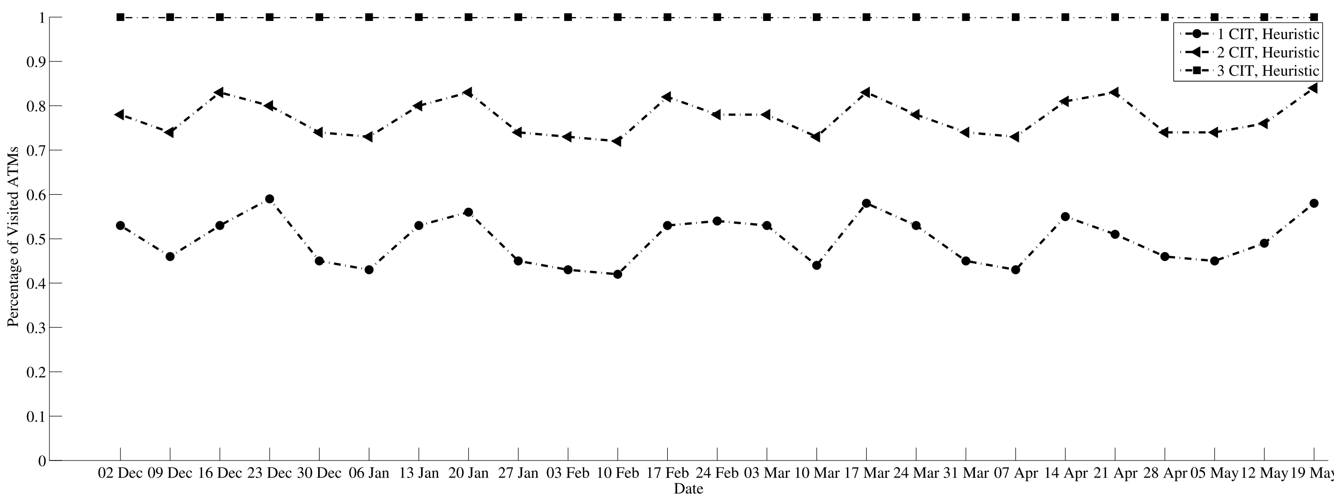

Fig. 1. Percentage of visited ATMs with 1, 2, and 3 CIT vehicles in heuristic algorithm solution for real data with 106 ATMs.

In order to better evaluate the performance of our model by isolating us from the possible errors in demand forecasting, we have implemented our simulations with the actual withdrawal and deposit data (retrospectively). Moreover, although our theoretical model enables any length for the scheduling period, we set it to 1 week (seven days) since our industrial partner Provus Inc. explicitly informed us that their scheduling is on a weekly basis.

For synthetic data, we generate three ATM networks with 25, 50, and 100 ATMs. Each network is connected and randomly generated. We set the travel times between each pair of ATMs to be uniformly random between 5 and 60 min. The reason for this setting is to ensure consistency with the real-life situation in Istanbul, which is a huge city where the travel time between two locations varies a lot depending on the traffic congestion level rather than the Euclidean distances, and the vehicles usually have high variance in their velocities due to the varying traffic congestion at different places on the road. We verified this behavior also by the results of the Google Maps Distance Matrix API in the previous experiment with real data. In all experiments, we set the CIT cost, service time for an ATM, interest rate, CIT vehicle capacity, working hours, and scheduling period to constant values, which are specified in Table III. We take the interest rate as 11.25% for a year and use the daily interest rate as 11.25%/365. Table IV shows the ranges for the amount of withdrawal and deposit for synthetic data with 25 ATMs, while Table V shows the corresponding ranges for 50 and 100 ATMs. For each of the three networks, we run ten 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 the first set of experiments, we analyze the impact of the number of CIT vehicles on feasibility. A solution is infeasible unless it satisfies all of the constraints in (2)–(21) specified in Section III. 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 vehicle. For the real data with 106 ATMs, CPLEX returns an infeasible solution (within the given time limit), 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*. Fig. 1 illustrates the performance in terms of the percentage of visited ATMs for the real data of 106 ATMs with 1, 2, and 3 CIT vehicles. Both heuristic algorithm and CPLEX visit all ATMs in the case with three CIT vehicles. For 1 and 2 CIT vehicles, CPLEX yields an infeasible solution, whereas our heuristic algorithm can generate partial solutions with the demonstrated percentage of visited ATMs. Moreover, we observe that increasing the number of CIT vehicles has an important role in increasing the percentage of visited ATMs and eventually obtaining a feasible solution.

TABLE III
PARAMETER VALUES

Parameter	Value
Scheduling length	7 days
Daily interest rate	11.25%/365
Money paid to CIT per day for visiting an ATM	100 TL
Service time for an ATM	10 min
Working hours	720 min
CIT vehicle capacity	10.000.000 TL
Cost of deploying a recycle ATM per week	500 TL

We then investigate the relation between the number of CIT vehicles and the total cost of cash management by using real data with 16 and 106 ATMs. In our experiments, we vary the number of CIT vehicles from 1 to 5. In Fig. 2, we compare the total cost in CPLEX solution and heuristic algorithm with 1 and 5 CIT vehicles for 16 ATMs. We do not show the cost of two, three, and four CIT vehicles in the figure for better visual quality; instead, we show the average cost values for these cases in Table VI. For 106 ATMs, since CPLEX gives infeasible solution for 1 and 2 CIT vehicles, we compare the results with 3, 4, and 5 CIT vehicles. In Fig. 3, we show the results for only three and five CIT vehicles, again for better visual quality. For four CIT vehicles, average costs of CPLEX and heuristic algorithm are 22.948 and 24.949, respectively. In Figs. 2 and

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE IV
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 V
WITHDRAWAL AND DEPOSIT RANGES OF SYNTHETIC DATA WITH 50 AND 100 ATMS

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

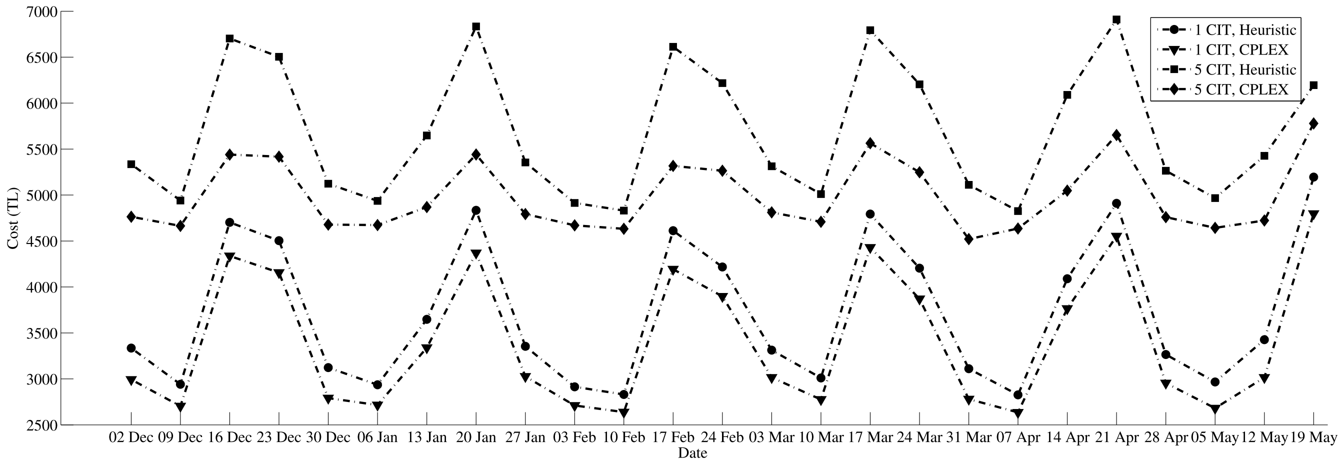

Fig. 2. Comparison of CPLEX and heuristic algorithm for real data with 16 ATMs.

TABLE VI
AVERAGE COST FOR REAL DATA OF 16 ATMS

Number of CIT vehicles	CPLEX	Heuristic
1	3405,71	3722,52
2	3713,93	4222,52
3	4075,58	4722,52
4	4492,83	5222,52
5	4988,82	5722,52

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 CIT vehicles result in higher cost values. Table VII shows the average total cost difference (in terms of percentage) between CPLEX and heuristic algorithm for real data, and again shows that the performance of our heuristic algorithm is close to the solutions obtained by CPLEX.

In order to better demonstrate the relation between the number of CIT vehicles and the total cost of cash management, in Fig. 4 we vary the number of CIT vehicles 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 CIT vehicles increases.

Decision of replacing a classical ATM with a recycle one is also an output in both CPLEX solution and heuristic algorithm. We compare them in terms of the number of classical ATMs to be replaced by recycle ones. In Fig. 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 the other cases. Referring to Table IV, 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. Fig. 5(b) and (c) show the results of the case with 25 and 50 ATMs, respectively, using synthetic data. The number of ATMs to be replaced as recycle ATMs is uniformly little in the case with 25 ATMs, whereas the results are higher in the case with 100 ATMs. When we investigate the withdrawal and deposit ranges in Table V, 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 also 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 Fig. 5 and we see that the number of ATMs to be changed to recycle ATMs

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

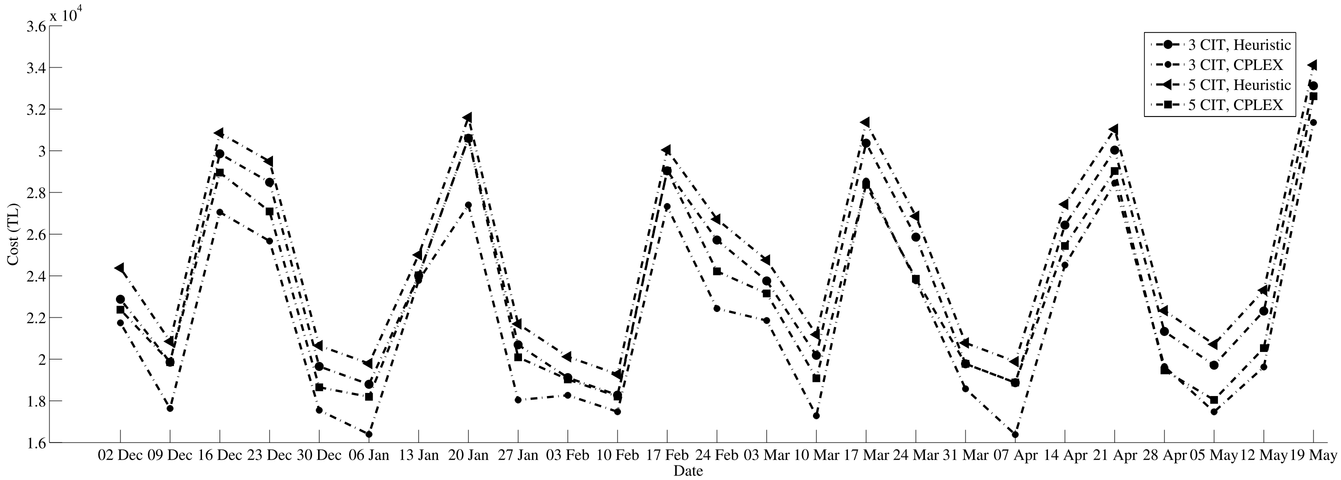

Fig. 3. Comparison of CPLEX and heuristic algorithm for real data with 106 ATMs.

TABLE VII
AVERAGE TOTAL COST DIFFERENCE (IN TERMS OF PERCENTAGE) BETWEEN CPLEX AND HEURISTIC ALGORITHM FOR REAL DATA

	1 CIT vehicle	2 CIT vehicles	3 CIT vehicles	4 CIT vehicles	5 CIT vehicles
106 ATMs	-	-	8,43%	8,02%	7,31%
16 ATMs	8,51%	12,04%	13,69%	13,97%	12,2%

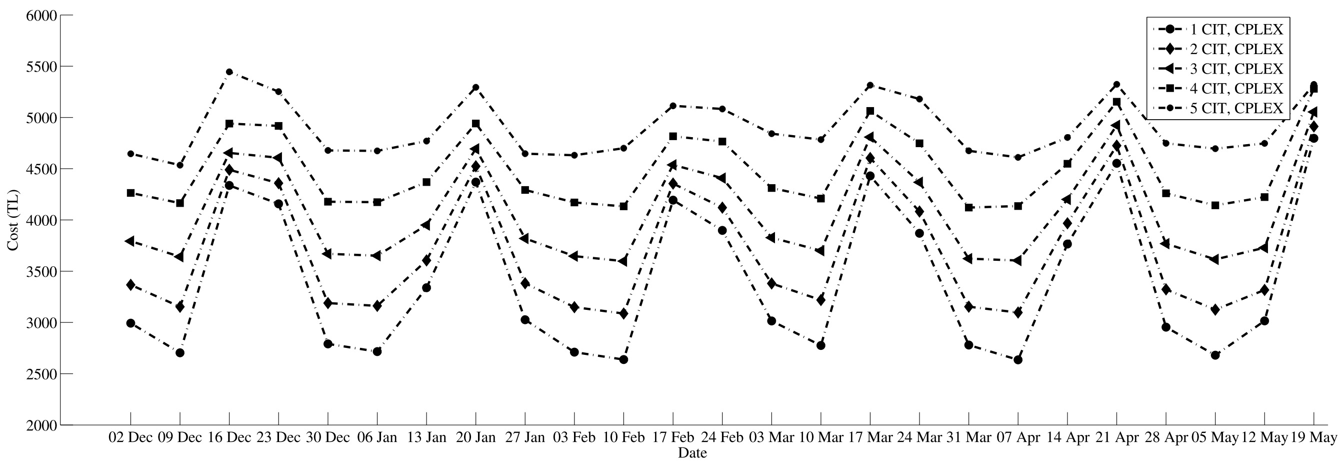

Fig. 4. Impact of the number of CIT vehicles in CPLEX solution for real data with 16 ATMs.

is little. Also by taking into account the behavior with synthetic data in Fig. 5(a)–(c), this behavior in Fig. 5(d) and (e) can be attributed to the fact that deposit amounts in real data is considerably lower than the withdrawal amounts. In Table VIII we show the difference between the average number of recycle ATMs for CPLEX and heuristic algorithm, and we observe that they are close to each other. Recall here that in our experiments with real data, we use the real withdrawal and deposit amounts provided by Provus Inc. As a consequence, this paper 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.

In Fig. 6, 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 CIT vehicles that gives feasible solution for 25, 50, and 100 ATMs are 1, 2, and 3, respectively. y-axis in Fig. 6 shows the average cost per ATM of ten samples and x-axis shows the number of ATMs. The number of CIT vehicles is set to minimum possible value that gives feasible solution in CPLEX. Fig. 6 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. In Table IX, we show the difference of average cost per ATM between CPLEX and heuristic algorithm for synthetic data, and we observe that they are very close to each other. Although, in some cases the performance of our heuristic

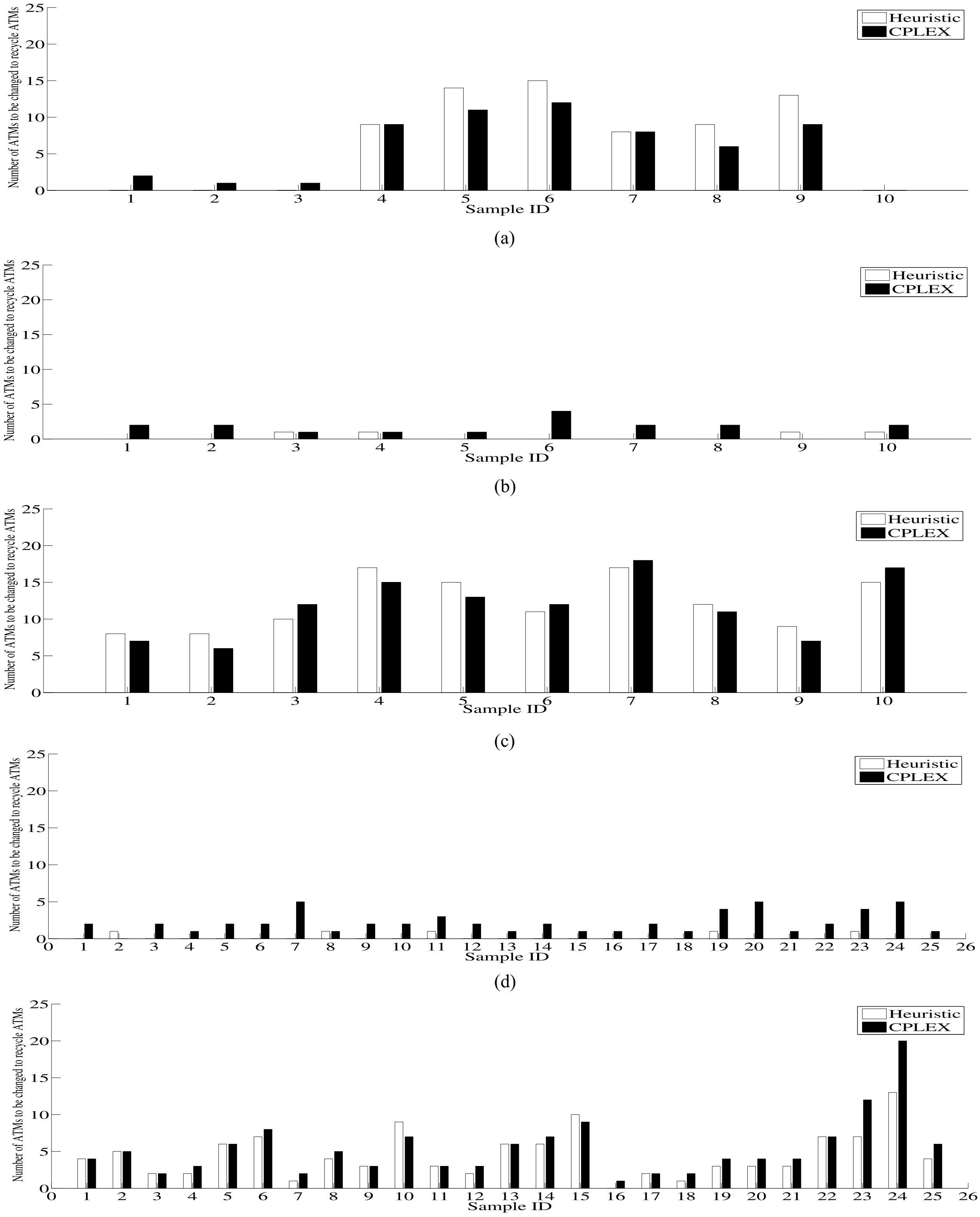

Fig. 5. Number of classical ATMs to be replaced by recycle ATM. (a) Synthetic data, 25 ATMs. (b) Synthetic data, 50 ATMs. (c) Synthetic data, 100 ATMs. (d) Real data, 16 ATMs. (e) Real data, 106 ATMs.

algorithm may not seem satisfactory at first glance (e.g., in Table VI the heuristic algorithm yields 13%–16% higher cost than the CPLEX solution), its performance was found to be very suitable for practical implementation by the employees of Provus Inc., whom we consulted at every stage of this paper, mainly because the problem, being an ILP with 22 constraints,

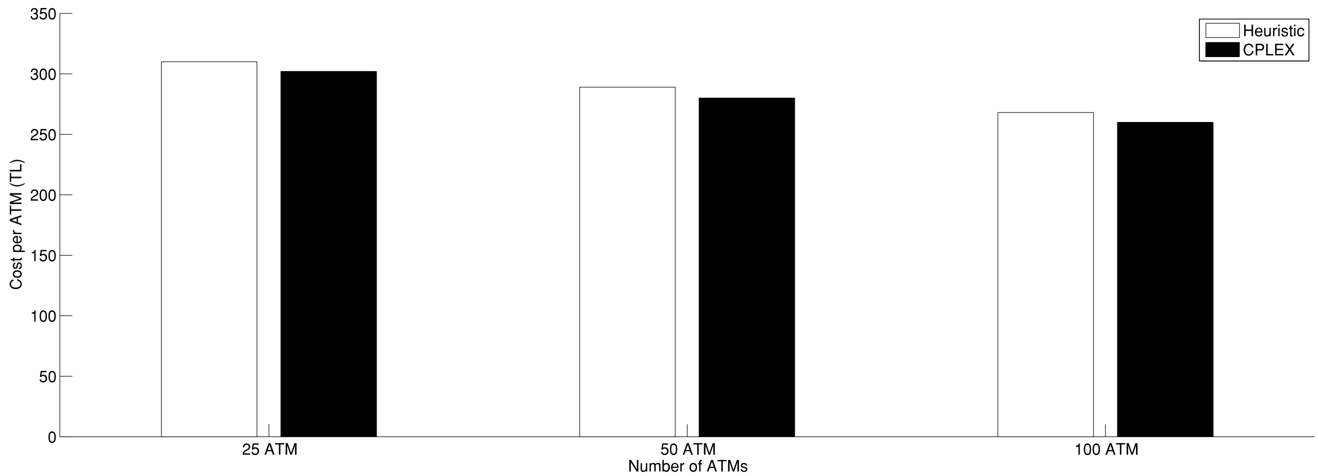

Fig. 6. Comparison of cost per ATM values in CPLEX solution and heuristic algorithm using synthetic data.

TABLE VIII
DIFFERENCE BETWEEN THE AVERAGE NUMBER OF RECYCLE ATMS FOR CPLEX AND HEURISTIC ALGORITHM

synthetic data, 25 ATMs	0,9
synthetic data, 50 ATMs	1,3
synthetic data, 100 ATMs	0,4
real data, 16 ATMs	1,96
real data, 106 ATMs	0,88

TABLE IX
DIFFERENCE OF AVERAGE COST PER ATM BETWEEN CPLEX AND HEURISTIC ALGORITHM FOR SYNTHETIC DATA

25 ATMs	2,58%
50 ATMs	3,11%
100 ATMs	2,98%

is very complex, practical implementations require simplicity in algorithm design, and 13%–16% difference from the optimal solution is quite satisfactory for all practical purposes.

VI. CONCLUSION

In this paper, 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.

ACKNOWLEDGMENT

The authors would like to thank Provus Inc. for providing them with real data collected from ATMs and K. Geçici from Provus Inc. for the invaluable information about the operations in an ATM network.

REFERENCES

[1] R. Simutis, D. Dilijonas, L. Bastina, J. Friman, and P. Drobinov, "Optimization of cash management for ATM network," *Inf. Technol. Control*, vol. 36, no. 1A, pp. 117–121, 2007.
[2] C.-S. Ou, S.-Y. Hung, D. C. Yen, and F.-C. Liu, "Impact of ATM intensity on cost efficiency: An empirical evaluation in Taiwan," *Inf. Manag.*, vol. 46, no. 8, pp. 442–447, 2009.
[3] "Recycling cash at the ATM," NCR White Paper, 2013. [Online]. Available: http://www.ncr.com/wp-content/uploads/recycling-cash-at-ATM-wp.pdf
[4] J. Castro, "A stochastic programming approach to cash management in banking," *Eur. J. Oper. Res.*, vol. 192, no. 3, pp. 963–974, 2009.
[5] L. Bertazzi and M. G. Speranza, "Inventory routing problems: An introduction," *EURO J. Transp. Logistics*, vol. 1, no. 4, pp. 307–326, 2012.
[6] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte, "Static pickup and delivery problems: A classification scheme and survey," *Top*, vol. 15, no. 1, pp. 1–31, 2007.
[7] W. J. Baumol, "The transactions demand for cash: An inventory theoretic approach," *Quart. J. Econ.*, vol. 66, no. 4, pp. 545–556, 1952.
[8] M. H. Miller and D. Orr, "A model of the demand for money by firms," *Quart. J. Econ.*, vol. 80, no. 3, pp. 413–435, 1966.
[9] G. D. Eppen and E. F. Fama, "Solutions for cash-balance and simple dynamic-portfolio problems," *J. Bus.*, vol. 41, no. 1, pp. 94–112, 1968.
[10] D. P. Heyman, "A model for cash balance management," *Manag. Sci.*, vol. 19, no. 12, pp. 1407–1413, 1973.
[11] E. J. Elton and M. J. Gruber, "On the cash balance problem," *J. Oper. Res. Soc.*, vol. 25, no. 4, pp. 553–572, 1974.
[12] J. G. Cabello, "Cash efficiency for bank branches," *SpringerPlus*, vol. 2, no. 1, p. 334, 2013.
[13] A. Bar-Ilan, D. Perry, and W. Stadje, "A generalized impulse control model of cash management," *J. Econ. Dyn. Control*, vol. 28, no. 6, pp. 1013–1033, 2004.
[14] J.-S. Yao, M.-S. Chen, and H.-F. Lu, "A fuzzy stochastic single-period model for cash management," *Eur. J. Oper. Res.*, vol. 170, no. 1, pp. 72–90, 2006.
[15] K. C. Ágoston, G. Benedek, and Z. Gilányi, "Pareto improvement and joint cash management optimisation for banks and cash-in-transit firms," *Eur. J. Oper. Res.*, vol. 254, no. 3, pp. 1074–1082, 2016.

[16] K. Venkatesh, V. Ravi, A. Prinzie, and D. Van den Poel, "Cash demand forecasting in ATMs by clustering and neural networks," *Eur. J. Oper. Res.*, vol. 232, no. 2, pp. 383–392, 2014.
[17] S. D. Teddy and S. K. Ng, "Forecasting ATM cash demands using a local learning model of cerebellar associative memory network," *Int. J. Forecasting*, vol. 27, no. 3, pp. 760–776, 2011.
[18] C. Ramírez and G. Acuña, "Forecasting cash demand in ATM using neural networks and least square support vector machine," in *Proc. Iberoamerican Congr. Pattern Recognit.*, Pucón, Chile, 2011, pp. 515–522.
[19] T. Efendigil, S. Önüt, and C. Kahraman, "A decision support system for demand forecasting with artificial neural networks and neuro-fuzzy models: A comparative analysis," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6697–6707, 2009.
[20] Y. Chen, B. Yang, and J. Dong, "Time-series prediction using a local linear wavelet neural network," *Neurocomputing*, vol. 69, nos. 4–6, pp. 449–465, 2006.
[21] Y. Ekinci, J.-C. Lu, and E. Duman, "Optimization of ATM cash replenishment with group-demand forecasts," *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3480–3490, 2015.
[22] D. Dilijonas, V. Sakalauskas, D. Kriksciuniene, and R. Simutis, "Intelligent systems for retail banking optimization—Optimization and management of ATM network system," in *Proc. 11th Int. Conf. Enterprise Inf. Syst. (ICEIS)*, Milan, Italy, 2009, pp. 321–324
[23] M. B. D. C. Moraes and M. S. Nagano, "Cash balance management: A comparison between genetic algorithms and particle swarm optimization," *Acta Scientiarum Technol.*, vol. 34, no. 4, pp. 373–379, 2012.
[24] Y. Altunoglu, "Cash inventory management at automated teller machines under incomplete information," M.S. thesis, Dept. Ind. Eng., Bilkent Univ., Ankara, Turkey, 2010. [Online]. Available: http://www.thesis.bilkent.edu.tr/0004020.pdf
[25] T. Baker, V. Jayaraman, and N. Ashley, "A data-driven inventory control policy for cash logistics operations: An exploratory case study application at a financial institution," *Decis. Sci.*, vol. 44, no. 1, pp. 205–226, 2013.
[26] E. Bjørnda, H. Hamer, and M. Koste, "Cost allocation in a bank ATM network," *Math. Methods Oper. Res.*, vol. 59, no. 3, pp. 405–418, 2004.
[27] P. Kurdel and J. Sebestyenova. *Modeling and Optimization of ATM Cash Replenishment*. Accessed on Mar. 2015. [Online]. Available: http://www.wseas.us/e-library/conferences/2012/Vienna/COMPUTERS/COMPUTERS-51.pdf
[28] D. J. du Toit, "ATM cash management for a South African retail bank," M.S. thesis, Indus. Eng., Stellenbosch Univ., Stellenbosch, South Africa, 2011. [Online]. Available: http://scholar.sun.ac.za/handle/10019.1/17977
[29] S. Opasanon and P. Lertsanti, "Impact analysis of logistics facility relocation using the analytic hierarchy process (AHP)," *Int. Trans. Oper. Res.*, vol. 20, no. 3, pp. 325–339, 2013.
[30] H. Larrain, L. C. Coelho, and A. Cataldo. (2016). *Managing the Inventory and Distribution of Cash in Automated Teller Machines Using a Variable MIP Neighborhood Search Algorithm*. [Online]. Available: https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2016-01.pdf
[31] R. G. Van Anholt, L. C. Coelho, G. Laporte, and I. F. A. Vis, "An inventory-routing problem with pickups and deliveries arising in the replenishment of automated teller machines," *Transp. Sci.*, vol. 50, no. 3, pp. 1077–1091, 2016.
[32] G. Pataki, "Teaching integer programming formulations using the traveling salesman problem," *Soc. Ind. Appl. Math.*, vol. 45, no. 1, pp. 116–123, 2003.
[33] A. Adamaszek and M. Adamaszek, "Combinatorics of the change-making problem," *Eur. J. Combinatorics*, vol. 31, no. 1, pp. 47–63, 2010.
[34] W. R. Pulleyblank, "Faces of matching polyhedra," Ph.D. thesis, Dept. Combinatorics Optim., Univ. at Waterloo, Waterloo, ON, Canada, 1973.
[35] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*. Heidelberg, Germany: Springer-Verlag, 2003.

Şeyma Batı received the B.S. and M.S. degrees in computer engineering from Bogazici University, Istanbul, Turkey, in 2011 and 2016, respectively.

From 2012 to 2014, she was a Software Developer with Provus Inc., Istanbul. Since 2015, she has been a software developer in a private bank in Gebze, Kocaeli, Turkey.

Didem Gözüpek received the B.S. (High Honors) degree in telecommunications engineering from Sabancı University, Istanbul, Turkey, in 2004, the M.S. degree in electrical engineering from the New Jersey Institute of Technology, Newark, NJ, USA, in 2005, and the Ph.D. degree in computer engineering from Boğaziçi University, Istanbul, in 2012.

She is an Associate Professor with the Computer Engineering Department, Gebze Technical University, Kocaeli, Turkey. From 2005 to 2008, she was a research and development engineer with a telecommunications company in Istanbul. Her current research interests include structural and algorithmic graph theory, approximation algorithms, and optimization problems in communication networks.

Dr. Gözüpek was a recipient of the CAREER Award from the Scientific and Technological Research Council of Turkey (TUBITAK) in 2014, the Dr. Serhat Özyar Young Scientist of the Year Honorary Award in 2013, the Boğaziçi University Ph.D. Thesis Award in 2012, and the ASELSAN Ph.D. Fellowship in 2011. She was a Finalist for the Google Anita Borg Memorial Scholarship in 2009.