Solving the Vehicle Routing Problem During the Development of ATM Cash Service Planning System

Ekaterina Kaurova
ITMO University
St. Petersburg, Russia
kaurova.ekaterina@gmail.com

Ivan Kaurov
"Delovye Konsultatsii, St. Petersburg" LLC
St. Petersburg, Russia
I.Kaurov@decosp.spb.ru

Arthur Lazdin
ITMO University
St. Petersburg, Russia
lazdin@yandex.ru

***Abstract*—the paper describes the architecture of intellectual system designed for planning of ATM cash service for commercial banks and pays attention to the routing of cash collection service. Several methods to solve Vehicle Routing Problem are considered, their pros and cons are described and proper methods are chosen to minimize calculation time and cash service costs.**

I. INTRODUCTION

The goal of this article is to review, describe and implement the module for creating cash collection service routes for commercial banks. It is a typical Vehicle Routing Problem (VRP) – to split the set of points for several routes according to given constraints and to optimize the built routes. The peculiarity of this problem is relatively many route points and also the task of integration of the module into the existing intellectual accounting and planning system.

There're plenty of similar products available at the market but most of them are proprietary and rather expensive. Also in this case it is immensely difficult to integrate the third-party product into complicated existing software.

So the decision has been made to design own route-planning module according to the requirements, listed below.

II. REQUIREMENTS FOR THE MODULE

A. Description of the ATM cash collection service planning

Modern medium-to-large commercial banks can have up to several hundreds of ATMs in a single district served by a single cash center. Many ATMs need to be filled in with cash once in several days, and ineffective planning could result in waste of money – nobody needs to refill half-empty ATM or to drive to a remote ATM to refill it every day with a relatively small amount of money.

Bank's expenses for ATM service consist of two parts: interest rate which is paid to the Central Bank of Russia (the more money is loaded into the ATM – the more money is paid to the Central Bank) and costs of cash transportation and collection service. The goal of the module is to find the optimal amount to load into the ATM to minimize both transportation and interest rate costs [1]. The graph of dependence of the ATM service price from loaded amount is shown at Fig. 1. It assumes that the ATM would be refilled only in case it is empty and the transport costs are constant. The service price is presented at *daily* basis (total price for the cycle divided by the number of days).

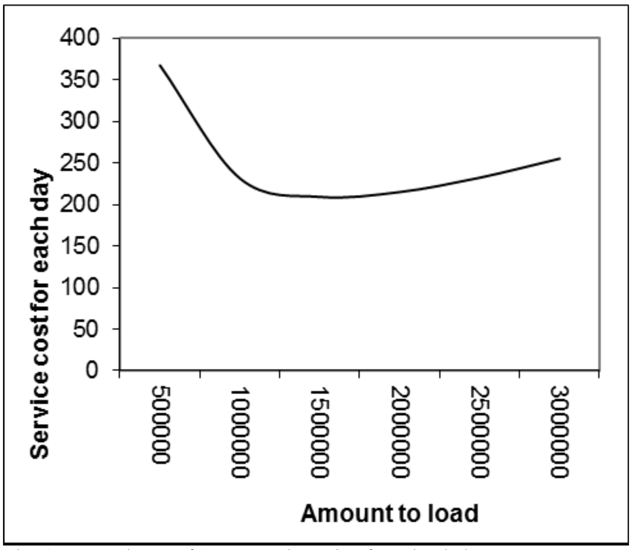

Fig. 1. Dependence of ATM service price from loaded amount

B. Why route planning is so important?

Effective route planning for several hundred points can assist the bank in saving a lot of money, because the ATM service price consists not only from the rate of interest, but also includes transport expenses, which in its turn consist of fuel price, salary for the people, car service and depreciation prices, etc.

ATM planning module that has been introduced in the previous paragraph detects the optimal cash amount and loading day only for a single ATM. Unfortunately, the neighboring ATMs are usually not considered and in some cases cash loading of two neighboring remote ATMs not on the optimal, but on one and the same day can save a lot of money on the transport costs.

ISSN 2305-7254

C. Route planning module requirements

The considered module requirements are briefly described in [8]. It is necessary to take into consideration that cash delivery service department, of course, has limited capabilities: the quantity of available cars is limited, also the staff can work for no more than acceptable time (usually from six to eight hours in Russia), every car has limited capacity, etc. So the desired route planning module should be able to split effectively the ATMs to load them during several routes according to the capabilities of the cash delivery service department. It is also important that the problem should be solved in a reasonable time (no more than several minutes).

III. VEHICLE ROUTING PROBLEM

A. Introduction

The ATM network can be presented as a full graph (a graph where all vertexes are linked to each other). Route planning module is designed to solve Vehicle Routing Problem (VRP) according to the constraints. The goal of VRP is to find the way through each vertex once and only once. Different vertexes can be served by different vehicles and all routes start and terminate at the same vertex (called depot).

Vehicle Routing Problem is a NP-hard problem and a plenty of methods to solve this problem exist. The criteria for choosing of the method to implement are:

1) Quality of the decision.

2) Computational difficulty.

3) Difficulty of the implementation.

B. Review of the existing methods

The classification graph of algorithms to solve the VRP is presented at [7] (see Fig. 2). In general, the majority of the methods can be divided into two major groups: single step algorithms which are designed explicitly for solving the VRP and two-step methods [9]. Some single step methods can yield the precise decision (methods like *full search*) whereas most of the classic algorithm can give only the approximate decision (that may be not the best). This group of methods includes both *constructive* and *improving* methods. Constructing methods are designed to build new optimal routes from scratch, and improving methods are improving the base (usually random) routes.

Two-step methods divide the solvation of the VRP in two different tasks: to split the set of vertexes to groups (routes) and to optimize the built routes. The order of these tasks is not specified: there're methods existing that at first split the initial set (clusterise) and then solve the classic Travelling Salesman Problem (TSP) on the built clusters. Also you can find the reverse order: at first the TSP is solved at the whole graph, then the route is divided into smaller routes according to the constraints and finally the smaller routes are optimized again.

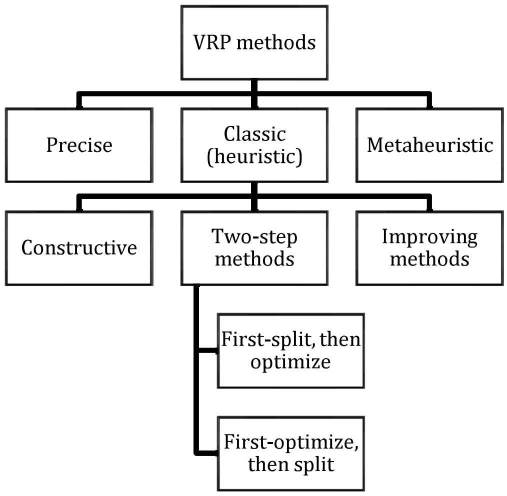

Fig. 2. Classification of the VRP methods [7]

As it has been mentioned earlier, one of the criteria for choosing of the method to solve the TSP is the difficulty of the implementation (due to development time restrictions) so we decided to choose the method of the "two-step" group. Most of the single step methods are heuristic and the time to solve the VRP can be unpredictable. Also the heuristic is hard to implement.

At this step we have to choose, and also probably to modify and implement the algorithms for two problems: the classic TSP and the splitting of the complex route in several smaller routes.

C. Travelling Salesman Problem

The task of TSP is to find the optimal route on the full graph visiting all vertexes once and only once and to return to the starting vertex (called depot). TSP is a NP-hard problem.

Plenty of methods to solve the TSP exist and all of them can be divided into two groups: the "construction" methods (algorithms are creating new routes) and "improvement" methods (designed to improve already created routes). Good reviews of the most renowned methods are presented in [2], [6], [10] and [11].

The simplest of the reviewed methods is so-called "*Greedy algorithm*" (or the method of the nearest neighbor) – at each iteration the nearest non-visited vertex is chosen. The computational difficulty of the Greedy method is O(n*log(n)) which is relatively small and the implementation is extremely easy. But the quality of the decision is not so good, especially when it comes to the last vertexes.

There're several modifications of the greedy method existing, but they don't produce much better solutions at the test graph. The computation time of the modifications is much longer when compared to the original Greedy algorithm.

One of the best among existing methods according to our criteria is Lin-Kernighan algorithm which is briefly described in [3]. With the computational difficulty of $O(n^3)$ (as stated in [2]), it is one of the best known TSP solving method. The idea of the algorithm is to heuristically improve the original route by replacing the longest legs with the shorter ones with the possibility to complete the route at each step. You can see the example of the replacing X_i legs by Y_i at the Fig. 3. This is called *3-opt move*.

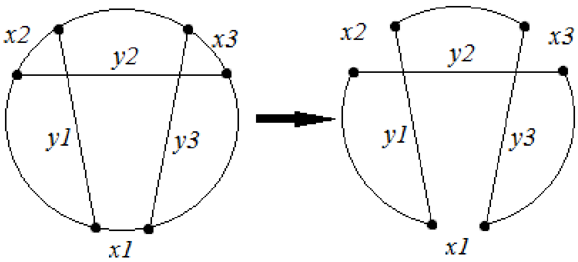

Fig. 3. Three-opt move [3]

D. Splitting the complex route

Different methods of splitting the route yield similar results so we decided to use one of the simplest but effective methods – "*Cutting the common route*" which is described in [2]. The major disadvantage of this method is that we have no ability to control the quantity of the built routes: each route is *cut and completed* as soon as it's *full* (has maximal length) according to the constraints. Therefore at the last step the constructed routes should be optimized using any TSP method.

IV. Implementation and Results

A. Choosing the application model

In our example one of the requirements, important for choosing of application model for the module, has been the graphical presentation of the built routes using any of the major geodata frameworks (such as Google, Yandex or Bing maps). This requirement leads to building the Web Application (probably the easiest way is using of Google Maps API which is free for educational use and does not require any special registration) [5].

B. Source data

We proceed from the following restrictions in a sample bank:

1) Each car can deliver cash for no more than 20 ATMs.

2) Each lap can last for no more than 6 hours.

3) The average service time for each ATM is 10 minutes.

4) About 175 ATMs located at 125 different addresses are loaded daily.

5) All ATMs are located within the bounds of St. Petersburg, Russia, and its nearest suburbs.

The main comparison criterion is quantity of built routes and their total length.

C. Impelmented methods

The following methods have been implemented and compared to each other:

1) Greedy algorithm.

2) Classic Lin-Kernighan algorithm (optimizing a random route, LK).

3) Lin-Kernighan algorithm optimizing the route built by Greedy algorithm (LK-Greedy).

The results are presented at Table I.

TABLE I. Comarison of the implemeted methods to solve the VRP on test data

	Greedy	LK	LK-greedy
Quantity of routes	10	10	10
Total length	914,310 km	726,724 km	705,576 km
Total time (to complete all routes, based on Google data)	54 h 00 min	51 h 9 min	51 h 33 min
The longest route	237,114 km	192,213 km	207,088 km
Average route length	91,4 km	72,7 km	70,5 km
The shortest route	38,5 km	30,9 km	31,7 km
Minimal time of the route	4 h 35 min	3 h 26 min	3 h 11 min
Time to solve the VRP	1-2 s	1,5 - 4 min	2 min

According to presented in Table I results, we can see that the best routes have been built by Lin-Kernighan algorithm used to optimize the routes built by Greedy algorithm. We have also found out that when a random route have been optimized by Lin-Kernighan method, the time to get the decision (computation time) varies too much according to the quality of the source route. The fastest test run was made at 90 seconds and the slowest result was 4 minutes. Total route length varies from 700 to 750 km.

Our experiment also has shown that optimizing «*Greedy*» route leads to predictable computation time of approximately 2 minutes for the test data, and the final result obtained has reasonable error compared to the classic LK method.

The shortest route and the minimal time criteria are presented in the table to compare the results of *splitting* the common route (obtained at the step of solving the TSP) to the shorter ones. In general, the shortest route can be short enough to get rid of it by moving its points to other routes (may be by violating some of the weakest restrictions, e.g. if the cash service brigade work for some extra time to serve one or two ATMs, it will cost for the bank less to pay for extra work than to keep another service brigade). The methods to solve this problem are beyond this study and are not considered in this article.

From our experiment we can see that the main advantage of the Lin-Kernighan method is very high quality of the decision that is obtained at reasonable time. The main disadvantage of the method is rather high computation time compared with the simplest Greedy algorithm. This disadvantage is not critical because the real routes are built no more than once or twice per day.

The results are presented at Fig. 4.

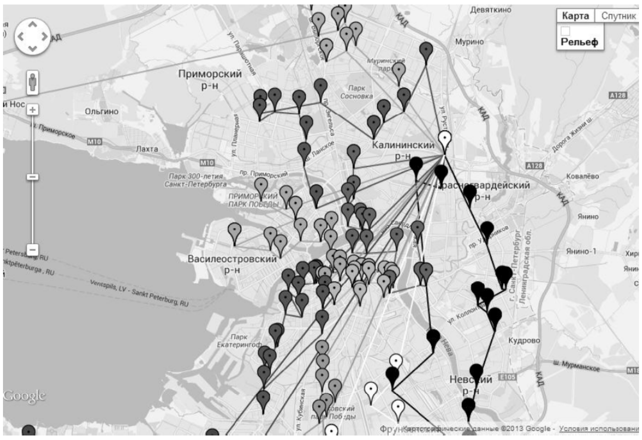

Fig. 4. Graphical representation of the proposed routes

V. Further Development

The proposed development of the whole ATM cash service planning system assumes combing two modules into a single system with a so-called *feedback*. As it has been mentioned earlier, the existing ATM cash service planning module considers each ATM as a single item and pays no attention to the state of the nearby ATMs. This leads to some unnecessary travels to neighboring remote ATMs in neighboring days. Therefore we propose to replace the current linear structure of the system (see Fig. 5) with non-linear by adding a loop between planning and routing modules (see Fig. 6) as well as to do the preliminary clusterization of the whole set of ATMs to determine the neighboring vertexes which are far enough from the depot (Fig. 7). The proposed algorithm description is presented below.

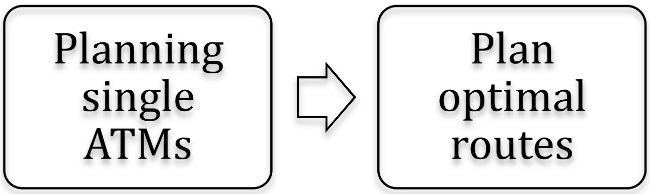

Fig. 5. Linear structure of the ATM service planning system

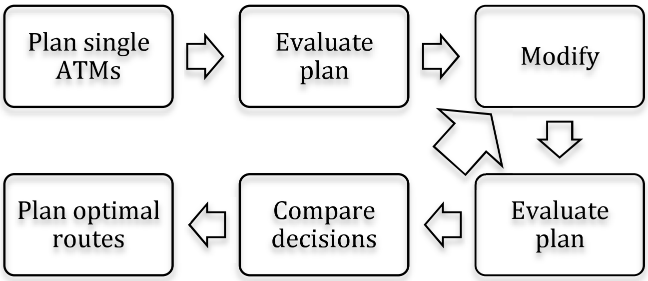

Fig. 6. Non-linear structure of the ATM service planning system

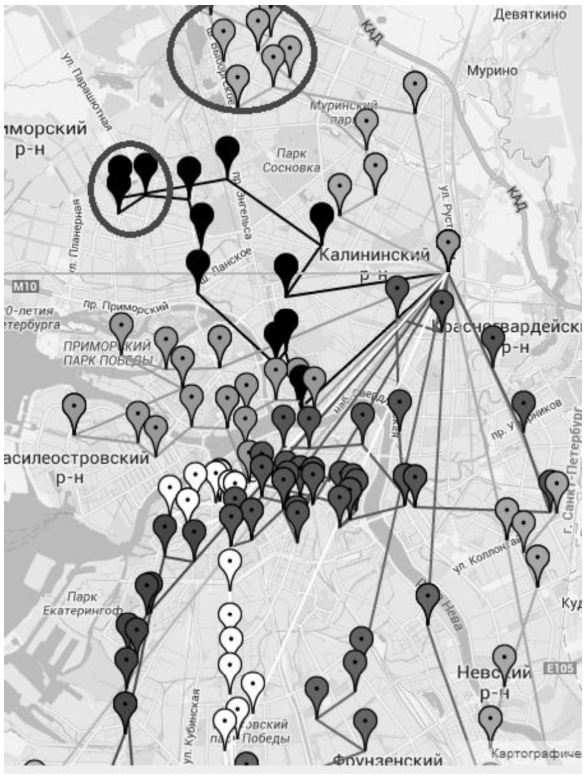

Fig. 7. Sample clusterization. The proposed remote clusters are highlited

Algorithm ATM service planning

0: **Preliminary clusterization of the whole set of ATMs.**
This will help to find the clusters of remote ATMs that are located close to each other. Only *remote* ATMs should be considered (those are located further than N km from the depot. N should be determined heuristically). This should significantly reduce the computation time. This step should be run only on modification of the set of ATMs.

1: **Plan the optimal service day for each ATM.**
This step is already implemented in the ATM planning module.

2: **Evaluate the whole price of the plan (including transport costs).**
At this step we don't need the precise decision of the VRP. We only need to get some evaluation to compare two different plans (set of ATMs to load), but we need a fast response from the routing module. For this case a simple *Greedy* algorithm can be used.

3: **Modify the plan by moving the service days for *remote* ATMs.**
Only moving the service day of the ATMs in the same remote cluster is allowed (see the description of the clusters in step 0).

4: **Evaluate the whole price of the plan.**
The same as in step 2.

5: **Compare the results and go to step 3.**
The comparison criterion is the so-called *daily* price of the plan (see the comments after the algorithm description).

6: **Precisely solve the VRP on the optimal plan.**
After getting the optimal plan the precise routes (decision of the VRP) should be obtained and presented to the user.

The optimal service day of each ATM is determined by minimizing the *daily* price (see formula (1)), where C_s is the average transport costs for this ATM, N-the number of the days to the next service, R_i – the proposed remaining amount in the ATM on day *i*, q – the interest rate that is paid to the Central Bank of Russia (price of «using» the cash) [4].

$$\frac{C_S+\sum_{i=1}^{N}(R_i*q)}{N} \tag{1}$$

At the steps of evaluating the price of the plan (steps 2 and 4) the static value C_s should be replaced by a proposed dynamically evaluated value based on the total transport costs.

This algorithm is *supposed* to significantly reduce transport costs due to consideration of the influence neighboring ATMs to each other. Unfortunately, there're no exit criteria proposed at this algorithm at the cycle of steps 3 to 5. They should be determined heuristically, and the detailed implementation and description are in the process of development by the authors now and should be proposed in the nearest future.

The main disadvantage of the proposed algorithm that it can (in theory) significantly increase the time needed to get the final plan. Also major modifications of the first module (ATM planning) are needed. The criteria of the initial clusterization (the threshold distance to decide whether the ATM is remote or not) should also be determined (see Fig.6).

Similar conclusion can be made about ATMs standing in the same place. Currently they are processed in a very simple way: during the TSP solving step they are considered as a single point. At the step of *cutting* the route, we consider these points as several different ATMs in the same place that *should* be included into the same route. According to the presented algorithm it is also possible to include such ATMs into the clusters even if they are not far enough from the depot.

VI. CONCLUSION

The common task of ATM service planning is considered at this article. Detailed attention has been paid to the routing problem and several ways to solve the common Vehicle Routing Problem have been presented. The implementation of several algorithms has been tested on rather large graph of real data and the optimal (at this step) method is found and justified.

The non-linear schema of ATM planning task is proposed at the fifth part, but the implementation and test results of this structure are beyond this article.

ACKNOWLEDGMENT

The authors would like to express their appreciation to the software developers of "Delovye Konsultatsii, St. Petersburg" LLC for providing them with the test environment and the real-world data, the staff of the chair of Information and Control Systems of the Institute of Informational Systems and Technologies of St. Petersburg State Polytechnic University for the productive teamwork on the ATM planning module and the chair of Informatics and Applied Mathematics of the ITMO University for the scientific basis of this article.

REFERENCES

[1] S.Yu. Koginov, I.V. Kaurov, D.F. Drobintsev, I.V. Butenko, "Development of the Architecture of an Analytical Module of the Cash Delivery Service Automation System", *XL Week of Science of SPbSPU. 2011. Pp. 108-109.*

[2] Yu.L. Kostyuk, M.S. Pozhidaev. "Approximate Methods to Solve the k-Travelling Salesmen Problem", *Vestnik TGU,* vol.1(2), 2008. Web: http://www.marigostra.ru/materials/ktsp-tsu-2008.pdf

[3] Keld Helsgaun. "An effective implementation of the Lin-Kernighan Travelling Salesman heuristic", *Roskiled University, Denmark.* Web:

http://www.akira.ruc.dk/~keld/research/LKH/LKH-2.0/DOC/LKH_REPORT.pdf
[4] I. Kaurov, S. Ustinov, D.Drobintsev. “Analytical Module of the Cash Delivery Service Automation System”, *CEE-SECR Conf.* Nov.2012. Web: http://2012.secr.ru/talks/analytical-module-of-commercial-bank-encashment-management-system
[5] “Google Maps API”, Web: https://developers.google.com/maps/?hl=ru
[6] A.V. Levitin, *Algorithms: An introduction to Development and Analysis.* Moscow: Williams, 2006.
[7] A.A. Smirnov. “Goods Delivery Optimization and its Mathematical Support”, *St.Petersburg State Financial and Economical Universicty.* Web: http://elibrary.unecon.ru/ver.php?matherial_id=5985
[8] S.Yu. Koginov, I.V. Kaurov, D.F. Drobintsev, I.V. Butenko, “Development of the Architecture of an Analytical Module of the Cash Delivery Service Automation System”, *XL Week of Science of SPbSPU. 2011. Pp. 108-109*
[9] D. Trofimov, A. Fedukov. “Vehicle Routing Problem”, *ITMO University.* Web: http://rain.ifmo.ru/cat/view.php/theory/unsorted/vrp-2006
[10] V.I. Mudrov. *Travelling Salesman Problem*. Moscow: Znanie, 1969.
[11] M.S. Pozhidaev. “Vehicle Routing Problem Algorithms”. *Ph.D. work,* Tomsk, 2010.