







Optimizing Spatial Accessibility of Company Branches Network with Constraints

Oleg Zaikin¹ , Ivan Derevitskii¹ , Klavdiya Bochenina¹ ,
and Janusz Holyst^{1,2} 

¹ ITMO University, Saint Petersburg, Russia

zaikin.icc@gmail.com, kbochenina@mail.ifmo.ru

² Faculty of Physics, Warsaw University of Technology, Warsaw, Poland

Abstract. The ability of customer data collection in enterprise corporate information systems leads to the emergence of customer-centric algorithms and approaches. In this study, we consider the problem of choosing a candidate branch for closing based on the overall expected level of dissatisfaction of company customers with the location of remaining branches. To measure the availability of branches for individuals, we extract points of interests from the traces of visits using the clustering algorithm to find centers of interests. The following questions were further considered: (i) to which extent does spatial accessibility influence the choice of company branches by the customers? (ii) which algorithm provides better trade-off between accuracy and computational complexity? These questions were studied in application to a bank branches network. In particular, data and domain restrictions from our bank-partner (one of the largest regional banks in Russia) were used. The results show that: (i) spatial accessibility significantly influences customers' choice (65%–75% of customers choose one of the top 5 branches by accessibility after closing a branch), (ii) the proposed greedy algorithm provides an optimal solution in almost all of cases, (iii) output of the greedy algorithm may be further improved with a local search algorithm, (iv) instance of a problem with several dozens of branches and up to million customers may be solved with near-optimal quality in dozens of seconds.

Keywords: Branch network optimization · Location · Spatial accessibility · Banking · Black-box optimization

1 Introduction

Networks of company branches operating within the city arise in different domains like retail industry, banking and finance, car manufacturing and many other. Different units, e.g., shops or bank branches, may be located in different areas, serve different number of customers and may have different efficiency of functioning related to actions of line staff and managers. In this regard, modification of the network of branches (closing or opening units) is considered as

multi-criteria optimization problem with at least two criteria: (i) profitability of a branch (taking into account cost of maintenance, remuneration, rental charges etc.), (ii) Quality-of-Service (QoS) requirements (related to the business standards of customer care).

Collection of massive data sets about customer behavior makes possible to conclude about implicit factors of customer satisfaction (QoS criteria). For the set of branches distributed within the city and having identical functionality, one of the determinant factors of customer satisfaction is spatial availability of the branches. To decide if a certain branch is located conveniently for a certain customer, one needs to know the areas of his or her frequent visits. After that, we can optimize branches network to achieve better spatial availability for the population of customers or to reduce the effect of closing the branches.

This study is focused on the latter case when a decision maker is aimed to close a fixed number of branches to reduce operational costs of a network. As a baseline business case, we consider a problem of optimizing bank branches network under constraint on maximum desirable number of units after modification of the network. We assume that each customer is assigned with a number of points of interest, and available branches may be located in the neighborhood of each of these points. We formulate a problem of optimizing spatial availability of branches networks and propose several greedy algorithms to solve it. To show that spatial availability is related to actual choice of branches by customers, we make a prediction on preferable set of branches for each customer according to her points of interests, and compare the predictions with the actual places of visits after closing the branch. As the research is conducted in partnership with one of the largest regional banks in Russian Federation, for the experimental study we use the data about visits and transactions of more than 800 000 customers for 2 years.

The outline of the paper is as follows. In the next section, related works are discussed. In Sect. 3 a formulation of the problem is given. In Sect. 4 algorithms for solving the considered problem are proposed. Section 5 contains a description of the data set and the results of computational experiments. In the last section, conclusions are drawn.

2 Related Work

The problem of branch location evaluation has been widely studied. In [12], this problem in application for shopping centers is solved. The following characteristics are taken into account: diversity of the tenant inside the shopping center, retail agglomeration near the shopping center, distance to metro stations, and distance between consumers and shopping center. To calculate the last two characteristics, the Euclidean distance is used. The problem of branch network optimization was studied in [14]. In that paper, a Lagrangian relaxation optimization method is proposed to optimize location of locomotive maintenance shops. In [10], a multi-agent model for optimizing supermarkets location is proposed. Using the Particle Swarm Optimization heuristic, the model iteratively determines a place maximizing the sales volume of the new supermarket.

The problem of branch location evaluation in application to a bank network is studied in a number of papers. In [2] the problem is solved by an integrated method, based on analytic hierarchy process, geographic information system and maximal covering location problem. In [13], this problem is formulated and studied as a Markov decision process. In [6] the problem was studied using demand-covering models which determine the locations that achieve the maximum capture of the demand.

The problem of optimizing a bank branches network has been also studied recently. In [9], the problem is solved for the branch network of a German retail bank. A decision support system is used for this purpose. In [3] the problem was studied by a tabu search optimization algorithm [5] in application to a Turkish bank's branches network. In [7] the problem was solved by the following algorithm. A related linear problem is solved first, then the obtained solution is iteratively improved by a local search algorithm. In [11], three heuristic algorithms are suggested to solve the problem: greedy interchange, tabu search and Lagrangian relaxation approximation. A bank's branches network in a large-size town Amherst, USA was analyzed by these algorithms.

In this study, due to the availability of digital traces of customers of our industrial partner, we present new statement of the problem of optimizing branch location which uses zones of interests of individuals instead of city-wide information about popularity of different areas. As a result, we formulate target function as spatial availability estimated directly from points of interests of different customers. Thus, the present study is a step forward in branch network optimization and, in particular, in a bank branches network optimization.

3 Problem Statement

Suppose we have a company with branches network of size M within the city, and $b_1, \dots, b_k, \dots, b_M$ – branches represented with pairs of geo coordinates. This company has N associated customers $c_1, \dots, c_i, \dots, c_N$. Customers are described with the following attributes: (i) place of living L (geo coordinates); (ii) place of work W (geo coordinates); (iii) ordered sequence of pairs $\langle date, branch \rangle$ of visits to branches; (iv) ordered sequence of pairs $\langle date, P_{ij} \rangle$ of visits to different locations within the city where P_{ij} is j -th known location for i -th customer. If the considered company is a bank, then (iv) may be extracted, from transactional data, because transactions have the field 'Address', and one can extract from this field geo coordinates of a location where purchase was made. In this case, (iv) for a fixed date is a daily path of payments of a user within the city.

Let's assume that a company decided to close K branches of M (e.g. for optimizing the expenses). The problem is to find K of M branches which closing leads to the smallest reduction in spatial availability of branches network for the customers.

In the simplest case, all branches for customer i can be divided into two types, *accessible* and *non-accessible*. By the accessible branch for a customer i , we mean a branch that is within the δ -neighborhood of **any** of points $S = \{H, L, P_{ij}\}$.

Here δ may be chosen as a maximum walking distance still comfortable for a customer, e.g. 1 km.

Let's introduce the **accessibility coefficient** a_{ik} as a measure of accessibility of a branch k for a customer i . This coefficient may be defined in several ways as follows:

- (i) $a_{ik} = \mathbb{I}_\delta(c_i, b_k)$ where $\mathbb{I}_\delta(c_i, b_k)$ is a binary variable. If its value is 0, it denotes that k -th branch is not δ -accessible for customer i , and 1 denotes opposite situation;
- (ii) $a_{ik} = \left(\frac{\mathbb{I}_\delta(c_i, b_k)}{\mathbb{B}_\delta(c_i)} \right)^\gamma$ for $\mathbb{B}_\delta(c_i) = \sum_k \mathbb{I}_\delta(c_i, b_k) > 0$, and 0 otherwise ($\gamma > 0$). The meaning of this expression is as follows. When b_k is not accessible for customer i (or customer i does not have any accessible branch in δ -neighborhood), the accessibility coefficient is equal to zero. If b_k is accessible, a_{ik} is the greater, the smaller is the number of accessible branches for customer i , $\mathbb{B}_\delta(c_i)$. The logic behind that is that customers with a higher number of accessible branches should influence the decision about closing one of them to a smaller extent than customers for which the accessible branch is unique. Here, coefficient γ allows to tune the extent to which customers with increasing number of accessible branches will influence the results.
- (iii) $a_{ik} = e^{-\lambda \cdot d_{min}(c_i, b_k)}$, $\lambda > 0$, where $d_{min}(c_i, b_k) = \min_j d(P_{ij}, b_k)$ —minimum distance to branch b_k from any of the points of interests of customer i . In such a case, accessibility coefficient exponentially decreases with an increase of a distance of branch b_k from the nearest of points of customer i .

Further, we create an **accessibility matrix** $A = \{a_{ik}\}$, in which the sum of elements for i -th row (i -th customer) means *the potential coverage of a customer*, that is, a large value of the sum corresponds to the case when a customer has a lot of appropriate branches; and a small value of the sum denotes critical customers. For the case (ii), this value for the customers with at least one accessible branch, is equal to 1. The sum by columns for all three cases means *the potential popularity of the branch*.

Considering this matrix, we may also come to conclusion that if $\mathbb{B}_\delta(c_i) > K$ that customer c_i will be satisfied after elimination of each K columns, and then, to exclude them from consideration.

The optimization problem can be stated as follows: to find K columns of accessibility matrix A which deletion will minimize the number of customers with $\mathbb{B}_\delta(c_i) = 0$ in the case (i) or minimize the decrease of $\sum_i a_{ik}$ in the cases (ii) and (iii). In other words, we would like to reduce the number of branches keeping the number of potentially satisfied customers as large as it is possible. Further we shall focus on the case (i) and by *potentially satisfied customer*, we mean customer having at least one branch in the δ -neighborhood of any of his or her visiting points.

4 Proposed Algorithms

In order to solve the problem considered in the previous section, algorithms of three types were developed: (i) a brute-force algorithm, that guarantees to find exact solutions, but work very slowly; (ii) greedy algorithms, which are able to quickly find approximate solutions; (iii) a local search algorithm that can be used to find an approximate solution or to improve a solution found by a greedy algorithm. All these algorithms are described below. Also, in Subject. 4.4 several examples of the accessibility matrices and results obtained by the proposed algorithms on them are shown.

4.1 Brute-Force Algorithm

The optimal solution of the considered problem can be found by the brute-force approach. There are two possible variants. According to the first one, $\binom{M}{M-K}$ different combinations of choice of $M - K$ ‘good’ branches from M branches are generated, then for each of them the number of potentially satisfied customers is calculated, and the optimal combination (with the greatest number of satisfied customers) of $M - K$ branches is chosen. According to the second variant, K ‘bad’ branches should be found, so $\binom{M}{K}$ combinations are generated, then for each of them the number of potentially unsatisfied customers is calculated, and the optimal combination (with the smallest number of unsatisfied customers) of K branches is chosen.

In practice, $M \gg K$, and also K is quite small (as one wants to close just several branches), so in the following computational experiments the second variant (choosing K ‘bad’ branches) is used. Since $\binom{M}{K} = O(M^K)$, the algorithm works in polynomial time. The pseudo-code is shown in Algorithm 1.

Algorithm 1. Brute-force algorithm

Input: M bank branches, the number K of branches to close, accessibility matrix A

Output: Set S_{out} of K bank branches recommended for closing

- 1 Generate $\binom{M}{K}$ different combinations of K bank branches
 - 2 For each combination calculate the number of unsatisfied customers
 - 3 $S_{out} \leftarrow$ the combination with the lowest number of unsatisfied customers
 - 4 **return** S_{out}
-

4.2 Greedy Algorithms

Despite the fact, that the proposed brute-force algorithm works in polynomial time, on some hard instances of the considered problem it cannot be executed in reasonable time. A greedy algorithm can be used instead to quickly find an approximate solution, which can be similar to the optimal one (but it is not guaranteed).

The following greedy algorithm is based on closing K branches with the smallest number of unsatisfied customers one by one. Further this algorithm is

denoted as GreedyLP (LP stands for Lowest Popularity). The pseudo-code of this algorithm is shown in Algorithm 2.

Algorithm 2. GreedyLP algorithm

Input: M bank branches, the number K of branches to close, accessibility matrix A

Output: Set S_{out} of K bank branches recommended for closing

```

1  $S_{out} \leftarrow \emptyset$ 
2 for  $step = 1 \dots K$  do
3   Choose branch  $b^*$  with the smallest number of unsatisfied customers
4    $S_{out} \leftarrow b^*$ 
5   Remove  $b^*$  from  $A$ 
6   Recalculate the number of unsatisfied customers of remaining branches
7 return  $S_{out}$ 

```

The following greedy algorithm is based on keeping branches with the greatest number of satisfied customers. Further this algorithm is denoted as GreedyHP (HP stands for Highest Popularity). The pseudo-code of this algorithm is shown in Algorithm 3.

Algorithm 3. GreedyHP algorithm

Input: M bank branches, the number K of branches to close, accessibility matrix A

Output: Set S_{out} of K bank branches recommended for closing

```

1  $S_{out} \leftarrow$  all  $M$  branches
2 for  $step = 1 \dots M - K$  do
3   Choose branch  $b^*$  with the greatest number of satisfied customers
4   Remove  $b^*$  from  $S_{out}$ 
5   Delete from  $A$  all rows with  $a_{ib^*} = 1$  and column  $b^*$ 
6   Recalculate the number of unsatisfied customers of remaining branches
7 return  $S_{out}$ 

```

Both proposed greedy algorithms have linear time complexity. If $K < \frac{M}{2}$, GreedyLP requires less iterations than GreedyHP.

4.3 Simple Hill Climbing Algorithm

To find an approximate solution, it is also possible to use local search heuristics. A set of K branches corresponds to a Boolean vector of size M , where $i - th$ component is 1 if the set contains branch number i and 0 otherwise. Such a vector, in turn, corresponds to a point in a search space S . This search space consists of all different Boolean vectors of size M and with exactly K 1s, so its size is $\binom{M}{K}$. For an arbitrary point $\chi \in S$, a neighborhood $Nh(\chi)$ of radius R is defined as a set of such points $\chi', \chi' \in S$, that $d_H(\chi, \chi') = R$, where $d_H(\chi, \chi')$ stands for Hamming distance between χ and χ' .

Let’s consider an objective function which operates in this search space. Given a point from the search space, it calculates the number of customers that would be potentially unsatisfied after the closure of all corresponding branches. Any discrete black-box optimization algorithm can be employed to minimize it, e.g., the genetic algorithm [1] or the tabu search [5]. As an optimization algorithm, the simple hill climbing algorithm [8] was chosen.

Simple hill climbing starts from a given point, in the role of which it is possible to use a random point from the search space. Then the algorithm checks points from the neighborhood of the given point. If a better point (i.e. a point with lower value of the objective function) is found, then this new point is considered as a new best point, and checking of its neighborhood is started. If all points from a current neighborhood are worse than a current best point, then it means that a local minimum is reached. The pseudo-code of the proposed algorithm is shown in Algorithm 4. Here χ_{best} is the point with the best found value of the objective function.

Algorithm 4. Simple hill climbing algorithm

Input: Set of M bank branches, the number K of branches to close, accessibility matrix A

Output: Set S_{out} of K bank branches recommended for closing

```

1  $\chi_{best} \leftarrow$  randomly chosen point of size  $M$  and weight  $K$ 
2 repeat
3   NewOptimum  $\leftarrow$  false
4   for each  $\chi \in \text{GetNeighborhood}(\chi_{best})$  do
5     if  $\text{ObjFunction}(\chi, A) < \text{ObjFunction}(\chi_{best}, A)$  then
6        $\chi_{best} \leftarrow \chi$ 
7       NewOptimum  $\leftarrow$  true
8       Break
9 until NewOptimum = false or TimeExceeded()
10  $S_{out} \leftarrow$  branches for which  $\chi_{best}$  contains 1s
11 return  $S_{out}$ 

```

In the proposed algorithm, checking a neighborhood of a given point has a linear time complexity. The amount of such checks cannot be predicted, but in practice for the considered problem it is usually less than 10.

4.4 Examples

Let’s consider two instances of the proposed problem. In the first one, $M = 4, K = 2$, the accessibility matrix is A_1 , see Table 1. In the second one, $M = 3, K = 1$, the accessibility matrix is A_2 .

GreedyLP finds a non-optimal solution for the first instance. It removes branch 1 on step 1 and branch 2 on step 2. As a result, one customer which has only one accessible branch is unsatisfied. As for GreedyHP, it keeps branch 1 on step 1 and branch 4 on step 2, and as a result all customers are satisfied. However, GreedyHP finds a non-optimal solution for the second instance.

Table 1. Examples of the accessibility matrices

$$A_1 = \left(\begin{array}{cccc|c} 1 & 1 & 1 & 0 & 3 \\ 1 & 1 & 1 & 0 & 3 \\ 1 & 1 & 1 & 0 & 3 \\ 1 & 1 & 0 & 1 & 3 \\ \hline 0 & 0 & 0 & 1 & 1 \\ \hline 4 & 4 & 3 & 2 & \end{array} \right) \quad A_2 = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ 1 & 0 & 1 & 2 \\ 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 0 & 1 \\ \hline 4 & 4 & 5 & \end{array} \right)$$

GreedyHP keeps branch 3 at step 1, and branch 2 at step 2, leaving customer 1 unsatisfied. However, if branches 1 and 2 are kept, all customers will be satisfied.

As for simple hill climbing, it finds the optimal solution for both considered instances if it starts from a solution found by any proposed greedy algorithm.

5 Computational Experiments

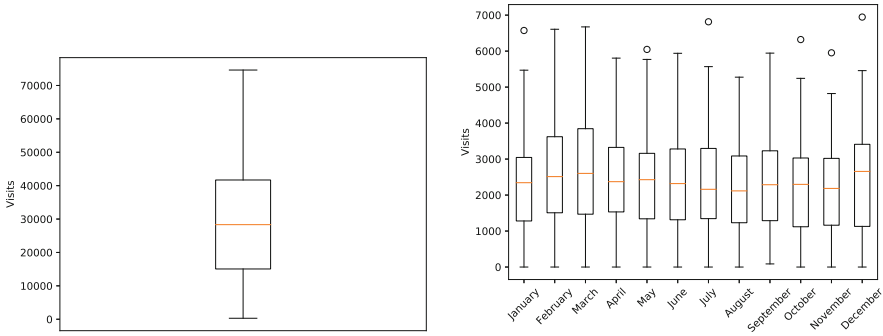
In this section, a data set used for computational experiments is described first. Then the influence of the spatial accessibility on customers choice of bank branches is studied. Finally, the experimental study of the algorithms, proposed in Sect. 4 is presented.

5.1 Data Set Description

We consider a data set from one of the largest regional banks in Russia. This data set contains data about 58 bank branches, 844 864 customers and 3 132 296 customers’ visits. Customers visits in one year (from September 2017 to September 2018) were taken into account in this study. Note, that only 51 out of 58 bank branches were active at the moment of this study. In Fig. 1a the distribution of bank branches by the amount of customers visits is presented. In Fig. 1b the distribution of bank branches by customers visits per month is shown. Seasonal effects can be seen: the activity during the summer is decreasing, while March and December are highly popular months.

According to Sect. 3, three types of points of interests (POIs) are used for each customer. The first two types (home address and work address) were provided by the industrial partner. POIs of the third type (places of visits) were taken from history of debit card transactions (the text field “Address” was used). This allows us to supplement information about home and work POIs which a customer reports to a bank because this information may become outdated.

To transform address data into pairs of geo coordinates, we use OpenStreetMap open database. OpenStreetMap is a map service, in which data are updated by its users in a collaborative manner. Using OpenStreetMap, we have



(a) Distribution of bank branches by the amount of customers visits (b) Distribution of bank branches by customers visits per month is shown

Fig. 1. Customers visits to bank branches

compiled a database of addresses of St. Petersburg. This database matches text representation of addresses to their coordinates. Then, we were able to identify geo coordinates of customers' POI.

POI extracted from transactions may contain outliers—points that were visited only once and that are off the usual routes of a customer. As our goal is to identify the frequently visited locations, we propose the following approach of getting POI from traces of visits to locations for a given customer c_i :

1. Collect a set of initial POI $\{l_{iv}\}, v = 1 \dots V_i$ where V_i is a number of distinct locations of a customer i , l_{iv} is a location of v -th point.
2. Group a set of locations $\{l_{iv}\}$ into \hat{V}_i clusters.
3. Remove clusters which contain a single point to get V_i^* clusters.
4. Find centers of V_i^* clusters and replace initial set of POI with V_i^* points which stand for the centers of payment interests of customer c_i .

We used this algorithm to get POI from transactional data. For step 2, DBSCAN clustering method was used [4].

5.2 On the Importance of Spatial Accessibility

We tested the hypothesis that the spatial accessibility of bank branches is important for customers. Two bank branches closed in Spring of 2018 were chosen for this purpose. Hereinafter they are called Branch 1 and Branch 2. For each of these branches, customers visiting it at least twice during the last 6 months before the closure were considered. It turned out, that for Branch 1 there were 964 such customers, while for Branch 2 there were 1483 of them. It was counted, how many customers visited other bank branches during 6 months after the closure. For Branch 1, there were 252 such customers, for Branch 2 – 409 customers. Then it was analyzed how many of these customers visited the most accessible active branches. The results are presented in Table 2. Here ‘Top-i’ stands for the

amount of customers visited at least one of the i most accessible active branches (in accordance with case (iii), see Sect. 3). For instance, value 109 of Top-2 for Branch 1 means, that after the closure of Branch 1, 109 out of 252 of its customers visited one of two most accessible branches. Note, that each customer has its own list of the most accessible branches.

Table 2. The comparison of bank branches spatial accessibility and real visits for Branch 1 and Branch 2 customers

Branch 1		
Top accessible branches	Customers (out of 252)	Percentage of customers
Top-1	68	27%
Top-2	109	43%
Top-3	134	53%
Top-4	149	59%
Top-5	163	65%
Branch 2		
Top accessible branches	Customers (out of 409)	Percentage of customers
Top-1	142	35%
Top-2	210	51%
Top-3	262	64%
Top-4	291	71%
Top-5	306	75%

It turned out, that 65%–75% of customers chose one of top 5 branches by accessibility (out of 51) after the closure. Thus, it is can be concluded, that the hypothesis on spatial accessibility influence was confirmed.

5.3 Experimental Study of the Proposed Algorithms

The algorithms described in Sect. 4 were implemented in a form of a sequential C++ program. While in almost all cases K was significantly smaller than M , only one greedy algorithm, GreedyLP, was used. It was done because GreedyLP suits better for small K than GreedyHP. Hereinafter ‘GreedyLP+SHC’ stands for simple hill climbing which starts from a point found by the GreedyLP algorithm. In simple hill climbing, radius R was equal to 2 (see Subsect. 4.3).

Computational experiments were held on a computer equipped with the 6-core processor Intel i7-3930K and 16 GB of RAM. Two series of experiments were conducted. In the first one, all 51 branches were available for closing (i.e. M was equal to 51), K (the amount of branches for closing) was varied from 1 to 10. The obtained results are presented in Table 3. Here ‘calc.’ stands for the number of the objective function calculations, while ‘unsat.’ stands for the number of unsatisfied customers in the solution found by the corresponding

algorithm. Hereinafter the best found solutions are marked with bold, while solutions which are worse than the best ones, are marked with *Italic*. It should be noted, that the brute-force algorithm was not launched for $K = 5, \dots, 10$ because the corresponding experiments could not be finished in reasonable time. For instance, for $K = 5$ it would take about 10 days.

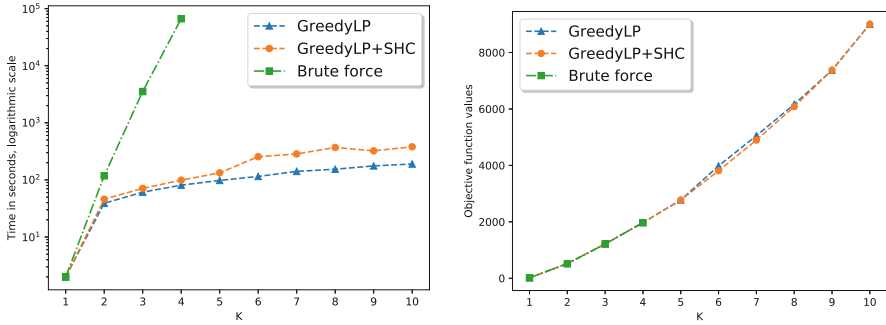
Table 3. Results for $M = 51, K = 1, \dots, 10$, time in seconds

K	Brute-force			GreedyLP			GreedyLP+SHC		
	calc.	time	unsat.	calc.	time	unsat.	calc.	time	unsat.
1	51	2	17	51	2	17	51	2	17
2	1275	118	520	101	39	520	199	46	520
3	20825	3533	1216	150	61	1216	294	71	1216
4	249900	66710	1972	198	81	1972	386	99	1972
5	2349060	-	-	245	98	2775	475	133	2775
6	18009460	-	-	291	115	<i>3984</i>	831	225	3811
7	115775100	-	-	336	141	<i>5052</i>	952	285	4898
8	636763050	-	-	380	154	<i>6166</i>	1068	370	6082
9	3042312350	-	-	423	176	7375	801	323	7375
10	12777711870	-	-	465	190	9003	875	380	9003

According to Table 3, for $K = 1, \dots, 4$ all three algorithms found optimal solutions. As for $K = 5, \dots, 10$, the best solutions are unknown, because, as it was mentioned above, the brute-force algorithm was not launched in these cases. Nevertheless, for $K = 6, 7, 8$ GreedyLP+SHC found better solutions than GreedyLP. It is not guaranteed that solutions, found by GreedyLP+SHC, are optimal. However, these solutions are local minima (in the sense described in Subsect. 4.3). It means that each of them can not be improved by replacing any branch from the corresponding set by any other possible branch. Note that GreedyLP+SHC takes a little more time compared to GreedyLP. However, it takes reasonable time even for the most difficult instances of the problem. Note, the in all cases GreedyLP+SHC required from 1 up to 3 neighborhood checks (see Subsect. 4.3) to reach a local minimum. These results show, that a solution found by GreedyLP is a very good start point for GreedyLP+SHC.

The results from Table 3 are also shown in Fig. 2. In Fig. 2a the solving time in seconds are shown on a logarithmic scale, while in Fig. 2b the objective function values are shown on a linear scale.

In the second series of experiments, a constraint was added. Only 17 out of 51 branches were available for closing (i.e. M was equal to 17). These branches were chosen because other 34 branches cannot be closed in accordance with the bank’s strategy. K was varied from 1 to 10. Note, that in the accessibility matrix all 51 branches were presented, so the whole bank branches network was taken into account during the objective function calculations. The obtained results are presented in Table 4. The brute-force algorithm managed to finish its work within



(a) Solving time in seconds (logarithmic scale) (b) Objective functions values (linear scale)

Fig. 2. Results for $M = 51, K = 1, \dots, 10$

Table 4. Results for $M = 17, K = 1, \dots, 10$, time in seconds

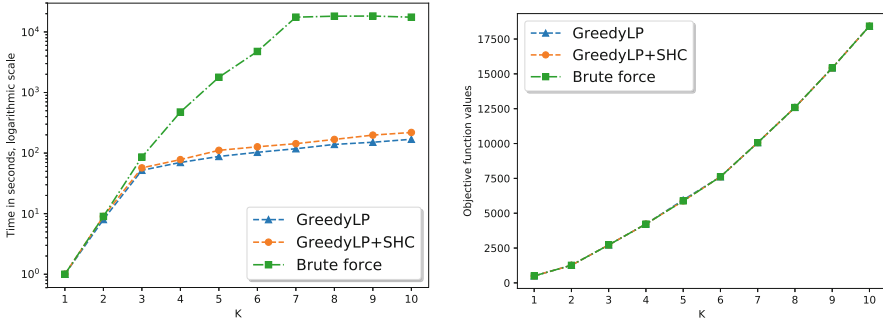
k	Brute-force			GreedyLP			GreedyLP+SHC		
	calc.	time	unsat.	calc.	time	unsat.	calc.	time	unsat.
1	17	1	498	17	1	498	17	1	498
2	136	9	1265	33	8	1265	63	9	1265
3	680	86	2722	48	52	2722	42	57	2722
4	2380	476	4224	62	70	4224	114	78	4224
5	6188	1785	5880	75	88	<i>5952</i>	195	111	5880
6	12376	4759	7614	87	103	7614	153	127	7614
7	19448	17524	10064	98	118	10064	168	143	10064
8	24310	18197	12590	108	139	12590	180	168	12590
9	24310	18287	15423	117	151	15423	189	198	15423
10	19448	17519	18429	125	169	18429	195	219	18429

the runtime limit of 1 day for all K , so it was possible to compare solutions found by GreedyLP and GreedyLP+SHC with optimal solutions in all cases.

It turned out, that the branches, recommended for closing in accordance with the first experiment (without any constraint) significantly differ from the ones recommended in accordance with the second experiment. In particular, for $K = 1$ in the first experiment the closure of the recommended branch will lead to 17 unsatisfied customers (see Table 3), while in the second experiment the closure of the recommended branch will lead to 498 unsatisfied customers. It means that the employed constraint is very strong and some branches which suit well for closing from the spatial accessibility point of view, cannot be closed due to some additional reasons.

According to Table 4, both Brute-force and GreedyLP+SHC found optimal solutions in all cases. As for GreedyLP, it found optimal solutions in all cases

except $K = 5$. The results from Table 4 are also shown in Fig. 3. In Fig. 3a the solving time are shown on a logarithmic scale, while in Fig. 3b the found objective function values are shown on a linear scale.



(a) Solving time in seconds (logarithmic scale) (b) Objective functions values (linear scale)

Fig. 3. Results for $M = 17, K = 1, \dots, 10$

The obtained results show, that GreedyLP suits very well for the considered problem. It works fast and in almost all cases finds an optimal solution. GreedyLP+SHC, being a little slower than GreedyLP, finds optimal solutions in all practical cases.

6 Conclusions

In this paper, the problem of company branches location optimization was considered. It was considered as a problem of minimizing the number of potentially dissatisfied customers according to the distances between points of individuals and branches. According to the conducted experiments, a greedy algorithm found exact solutions of the considered problem in almost all cases. A local search algorithm, which starts from a solution found by the greedy algorithm, managed to find exact solutions in all cases, where it was possible to verify by the brute-force algorithm. Meanwhile, both suggested algorithms work much faster than the brute-force algorithm. We have also shown that in the case of constraints related to the selection of branches for closing, the number of unsatisfied customers can increase significantly. In the future, we are planning to apply the suggested approach to other data sets.

Acknowledgments. This research is financially supported by the Russian Science Foundation, Agreement 17-71-30029 with co-financing of Bank Saint Petersburg.

References

1. Abualigah, L., Hanandeh, E.: Applying genetic algorithms to information retrieval using vector space model. *Int. J. Comput. Sci. Eng. Appl.* **5**, 19–28 (2015). <https://doi.org/10.5121/ijcsea.2015.5102>
2. Allahi, S., Mobin, M., Vafadarnikjoo, A., Salmon, C.: An integrated AHP-GIS-MCLP method to locate bank branches. In: *Proceedings of Industrial and Systems Engineering Research Conference*, July 2015
3. Basar, A., Kabak, O., Topcu, Y.I.: A tabu search algorithm for multi-period bank branch location problem: a case study in a Turkish bank. *Scientia Iranica* (2018). <https://doi.org/10.24200/sci.2018.20493>
4. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD 1996*, pp. 226–231. AAAI Press (1996)
5. Glover, F.: Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.* **13**(5), 533–549 (1986). [https://doi.org/10.1016/0305-0548\(86\)90048-1](https://doi.org/10.1016/0305-0548(86)90048-1)
6. Miliotis, P., Dimopoulou, M., Giannikos, I.: A hierarchical location model for locating bank branches in a competitive environment. *Int. Trans. Oper. Res.* **9**, 549–565 (2002). <https://doi.org/10.1111/1475-3995.00373>
7. Monteiro, M.S.R., Fontes, D.B.M.M.: Locating and sizing bank-branches by opening, closing or maintaining facilities. In: Haasis, H.D., Kopfer, H., Schönberger, J. (eds.) *Operations Research Proceedings 2005*, pp. 303–308. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-32539-5_48
8. Russell, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*, 3rd edn. Prentice Hall, Englewood cliffs (2009)
9. Schneider, S., Seifert, F., Sunyaev, A.: Market potential analysis and branch network planning: application in a german retail bank. In: *2014 47th Hawaii International Conference on System Sciences*, pp. 1122–1131, January 2014. <https://doi.org/10.1109/HICSS.2014.145>
10. Thiel, D., Hovelague, V., Pham, D.N.: A multi-agent model for optimizing supermarkets location in emerging countries. In: *2012 IEEE 13th International Symposium on Computational Intelligence and Informatics (CINTI)*, pp. 395–399 (2012). <https://doi.org/10.1109/CINTI.2012.6496798>
11. Wang, Q., Batta, R., Bhadury, J., Rump, C.M.: Budget constrained location problem with opening and closing of facilities. *Comput. Oper. Res.* **30**(13), 2047–2069 (2003). [https://doi.org/10.1016/S0305-0548\(02\)00123-5](https://doi.org/10.1016/S0305-0548(02)00123-5)
12. Wu, S.S., Kuang, H., Lo, S.M.: Modeling shopping center location choice: Shopper preference-based competitive location model. *J. Urban Plan. Dev.* **145**, March 2019. [https://doi.org/10.1061/\(ASCE\)UP.1943-5444.0000482](https://doi.org/10.1061/(ASCE)UP.1943-5444.0000482)
13. Xia, L., Xie, M., Yin, W., Dong, J., Shao, J.: Markov decision processes formulation for stochastic and dynamic bank branches location problems. In: *Proceedings of 2008 IEEE International Conference on Service Operations and Logistics, and Informatics, IEEE/SOLI 2008*, vol. 1, pp. 419–424, October 2008. <https://doi.org/10.1109/SOLI.2008.4686432>
14. Xie, W., Ouyang, Y., Somani, K.: Optimizing location and capacity for multiple types of locomotive maintenance shops. *Comp.-Aided Civ. Infrastruct. Eng.* **31**(3), 163–175 (2016). <https://doi.org/10.1111/mice.12114>