VILNIUS UNIVERSITY

FACULTY OF MATHEMATICS AND INFORMATICS

SOFTWARE ENGINEERING STUDY PROGRAM

Optimization of location facilities

Paslaugas teikiančių objektų vietų parinkimas

Master's Thesis

Author: Rostyslav Borovyk _____

Supervisor: Assoc. Prof. Dr. Algirdas Lančinskas _____

Reviewer: Prof. Dr. Romas Baronas _____

Vilnius – 2024

Summary

This work describes the Tabu Search approach for the solution of Competitive Facility Location Problem. The algorithm aims to find the optimal locations for retail store chain facilities in the city of Vilnius with the data gathered from open sources. The work describes different strategies of Tabu Search with focus on the diversification procedure and measures their performance. The work introduces new Weighted Random Initial Solution strategy that helped to significantly improve the diversification capabilities of Tabu Search compared to other methods.
Keywords: optimization problem, competitive facility location, tabu search.

Santrauka

Šiame darbe aprašomas Tabu Paieškos (angl. Tabu Search) metodas sprendžia konkurencingos objektų vietos nustatymo problemą. Algoritmas siekia rasti optimalias mažmeninės prekybos tinklo objektų vietas Vilniaus mieste, naudojant duomenis surinktus iš atvirų šaltinių. Darbe aprašomos įvairios Tabu paieškos strategijos, daugiausia dėmesio skiriant diversifikavimo procedūroms ir jų veikimo vertinimui. Darbe pristatoma nauja Svertinio Atsitiktinio Pradinio Sprendimo (angl. Weighted Random Initial Solution) strategija, kuri padėjo žymiai pagerinti Tabu Paieškos diversifikavimo galimybes, lyginant su kitais metodais.
Raktiniai žodžiai: optimizavimo problema, konkurencinga objekto vieta, Tabu paieška.

Table of contents

Introduction

The Facility Location Problem (FLP) plays a crucial role for businesses and organizations aiming to achieve operational excellence. It involves optimizing the location of facilities within a supply chain network to enhance their efficiency and effectiveness.

The FLP addresses the fundamental question of determining the optimal location for new facilities or the relocation of existing ones. By strategically positioning facilities, such as manufacturing plants, distribution centers, or service outlets, companies can improve their operational performance, reduce costs, enhance customer satisfaction, and gain a competitive advantage in the marketplace.

The FLP can also have broader socio-economic effects. Placing the strategic facilities on the regional level, for instance, can give an immense influence on the development of those regions and thus can be a subject of geopolitical planning. Careful consideration of all factors, and the use of the right model to calculate the effects of the decision is paramount and in result can bring a significant return on investment.

The FLP variations can focus on the optimization of different parameters such as the greatest number of serviced customers, the lowest operational costs, the minimal furthest distance from a facility, the minimal number of allocated facilities etc., with various input parameters. Therefore, lots of models were developed to solve a particular problem formulation as best as possible.

The solution of real world FLP is often complicated by the fact that the interactions between customers and facilities usually probabilistic in nature, and can depend on the factors like socio-demographic characteristics, the proximity of the facilities to the infrastructure facilities, internal metrics of a facility, etc. Therefore, each individual problem requires a careful consideration of the data that is needed to properly model the objective function used to estimate the quality of the solution.

Over the last century, many algorithms were introduced that solve different variations of FLP. Because the majority of FLP definitions are NP-hard, significant resources are needed to optimally solve the problem. For that reason, the majority of researches analyze the application of heuristic algorithms to solve the FLP, because of their ability to find the efficient solution of FLP within the optimal amount of time. Additionally, many general and problem specific optimizations were described, helping to find the solution faster and with less computational resources.

This work is devoted to the solution of real-world competitive FLP for retail chain stores with Tabu Search (TS) algorithm. The aim of the work is to improve existing approaches of Facility Location Problem solution with Tabu Search. Therefore, the following objectives can be defined for the work:

1) Analyze the existing approaches for the solution of the real-world FLP.
2) Gather data from open sources and build the relevant model to solve the FLP.
3) Implement the TS algorithm for the solution of FLP with different strategies.
4) Measure the performance of different strategies for TS algorithm.
5) Analyze deficiencies of the TS algorithm and propose the improvements to resolve them.
6) Compare the improvements with baseline implementation, analyze the results.

Achievement of the defined objectives can indicate that the work is successful, and it provides the contribution to the research of optimization problems.

1. Literature overview

This chapter contains overview of main concepts in the FLP, and reviews work that have been caried out in the field over the last several decades.

1.1. Origins of Facility Location Problem

In 1640 French mathematician Pierre de Fermat defined a problem of finding the position of point with respect to 3 other existing points in the space such that the distances between newly located point and 3 existing points are minimized [Pla06]. The geometrical solution to the problem was then found by E. Torricelli in 1645 and the numerical solution was proposed by Tellier [Pla06] [DH02]. In 1750 Thomas Simpson formulated the problem of finding the position of a point that minimizes the sum of distances to n existing points, given that existing points can have unequal attractive forces. It was the generalization of Fermat's special case of the problem with 4 total points and equal attractive forces. It was popularized by Alfred Weber in 1909. The simplest FLP can be formulated as Weber's problem with the additional restrictions in facility locations, and the parameters being optimized. The Weber's problem was then extended into multi-source problem, where the task was not only to identify the optimal allocation of $n > 1$ facilities, but also an optimal customers' allocations to the facilities, which is NP-hard. One of the most known algorithms for such problem formulation is Cooper's location allocation heuristic [Coo64]. For different problems, many other formulations were introduced, that do a better job at modeling real world design tasks. Mehrez and Stulman [MS82] introduced the maximal covering problem, where the task is to cover the greatest number of customers with a given number of facilities. P-center and P-median problems try to minimize the maximum distance between customers and facilities and minimize the sum of distances between customers and facilities respectively [GTM21].

The general elements characterizing the FLP were proposed by Eiselt and Laporte [EL97]:

1) Space where facilities can be allocated. Usually represented as a discrete set of possible facility locations or in continuous variation facilities can be allocated anywhere in the 2-dimensional space, with often present forbidden regions where the facilities cannot be placed. In addition, there are variations of the problem where the facility positions can be described only by relations with other facilities and not by physical locations, like in computer networks. Also, some problems can have a facilities' space of more than 2 dimensions.

2) Facilities that can be allocated. Usually are defined by characteristics like exact or maximum number of facilities to be allocated, facility capacity (represents how much demand the facility can satisfy), placement cost, etc.
3) Customers (or demand points). Usually contains the data about the location of each individual point and demand.
4) Supply costs between customers and facilities. Can be represented with Euclidian distance or Manhattan distance. In more advanced problem variations, there is a given function that represents the distance between a given facility and a customer.
5) Other constraints, customer behaviors. Some problem variations can define customer preferences (e.g., to represent customer tendance to stay with existing facilities when there is the new more preferable facility etc.), maximum/minimum distances between facilities, minimum percentage of covered customers, budget restrictions, and the whole variety of others.

1.2. Facility Location Problem types and variations

In its basic form, the FLP involves selecting a set of potential facility locations from a given set of candidates and assigning customers or demand points to these facilities. The objective is to minimize the overall cost or maximize efficiency, taking into account factors such as transportation costs, facility setup costs, demand requirements, service levels, capacity constraints and others. The FLP can be formulated as a mathematical programming problem, typically as a mixed-integer programming (MIP) or nonlinear programming (NLP) model.

The FLP can be subdivided into following types:

a) Single Facility Location Problem (SFLP) or Multi-Facility Location Problem (MFLP). defines if the objective is to determine the optimal location for a single facility that serves a set of customers, or for multiple facilities.
b) Capacitated Facility Location Problem (CFLP) or Uncapacitated Facility Location Problem (UFLP). UFLP does have any restrictions in the number of customers assigned to a facility, while CFLP incorporates capacity constraints for facilities. For CFLP, in addition to facility locations, the task is often to determine the number of products or services provided from each facility to each customer.
c) Single-stage or multistage problems. In the single-stage problems the products and services are directly delivered from facilities to customers. Multistage problems include intermediate steps for servicing customers like delivering the products from factories to

warehouses in the first stage and distributing them from warehouses to customers in the second stage. For multistage problems the objective is often to define the location of facilities for one or for multiple stages.

d) Single or multiple product models. Single product models focus on analyzing the distribution for only 1 type of service or product while multiple product models consider multiple product types.
e) Static or dynamic (multi-period) parameter models. Static models assume that the parameters like capacity of a facility, demand of a customer, transportation costs etc., remain constant. Dynamic models take into account the forecasts of parameter change to yield more accurate results for the future periods.
f) Elastic or inelastic demand models. Inelastic models assume that the distance between customer does not have any impact on the demand. For some problems, the demand of customers tends to have an inverse correlation with the distance to a facility, so elastic models consider that aspect too.

The FLP can as well be subdivided into following variations, with respect to the objective function:

a) K-median problem – one of the most popular variations, the objective is to minimize the sum of the distances to the customers' assigned facilities. Such variation of FLP can be applicable to model the retail chain, where the optimal positions of distribution centers will ensure the lowest operational costs for a company.
b) K-center problem – the objective is to minimize the maximum distance between customers and facilities. This variation can be useful for modeling the hospital locations or fire station locations, where the key factor is to ensure that in case of emergency ambulance or fire engine will be able to reach citizens within the reasonable amount of time.
c) Covering problem – the objective is to select the minimum number of facilities (or spend the minimum budget for the facilities) from the set of possible facilities and ensure that all customers (or a certain threshold of customers) will be covered. This variation of FLP can be used for modeling businesses such as telecommunications companies, where the goal is to cover all clients with the signal with the least number of cell towers.
d) Competitive problem – the facilities are introduced to the existing market, where customers already assigned to competitors' facilities or to the facilities of same chain. The goal is to maximize the revenue (or other objective value) for the new facilities or

for the whole company. This variation of FLP can be used to represent the process of entering the market for a new retail chain, a fast-food franchise or other type of business.

FLP like most NP-complete problems is often defined in the form of Integer Programming models (IP) and Mixed-Integer Programming models (MIP). Such models often work with integer values or other discrete values. They can be defined with the objective function to be minimized and the set of restrictions. The example of CFLP definition for *k*-median variation as an IP model can be presented as the following

$$\text{Minimize}$$

$$\sum_{j=1}^{m} f_j y_j + \sum_{i=0}^{n} \sum_{j=0}^{m} c_{ij} x_{ij}$$

$$\text{Subject to}$$

$$\sum_{j=1}^{m} x_{ij} = d_i, \qquad \forall i \in 1..n$$

$$\sum_{i=1}^{n} x_{ij} \leq M_i y_j, \qquad \forall j \in 1..n$$

$$x_{ij} \leq d_i y_j, \qquad \forall i \in 1..n, \forall j \in 1..m$$

$$x_{ij} \geq 0, \qquad \forall i \in 1..n, \forall j \in 1..m$$

$$y_j \in \{0, 1\}, \qquad \forall j \in 1..m$$

where $i \in \{1..n\}$ – customers, d_i – demand of a customer i, $j \in \{1..m\}$ – facilities, f_j – cost to build a facility j, M_j – capacity of a facility j, c_{ij} – transportation cost from a facility j to a customer i, x_{ij} – amount serviced from a facility j to a customer i, y_j – represents whether a facility should be opened.

1.3. Competitive Facility Location Problem

This section defines the Competitive FLP optimization problem that will be solved in the subsequent chapter.

Suppose a company called A is considering placing a group of facilities or a single facility in a region where other companies already offer services. A must compete for a market share in that region and factor in the existing competition when deciding where to establish

new facilities. This type of problem is referred to as Competitive Facility Location Problem (CFLP).

If the company A is already present in the market and wants to extend its presence in the region the newly placed facilities may interfere with the preexisting and therefore the change in the profit for the prior facilities should be considered by the model.

The FLP in this study analyses the optimal locations for the retail chain stores. As the basis for the prediction of customer demand allocation, the Bayesian Spatial Interaction Model was chosen which provides the probabilistic forecast about the attractiveness of the facilities [PAT23]. The customer demand is modeled with the population distribution data and for this problem, it is assumed that the demand is fixed. The combination of customer demand with facility attractiveness is then used to estimate the revenues of the stores.

The FLP is defined as following. Suppose there are I existing locations of facilities in different facility chains and candidate facilities, $|I| = m$. $\check{I} \subseteq I$ is a subset of the chain of facilities that is being optimized, $|\check{I}| = \check{m}$. $I^* \subseteq I$ is a subset of candidate facilities. Depending on a mode $\check{I} \subseteq I^*$ or $\check{I} \cap I^* = \emptyset$. l_i represents the size of the facility $i \in I$. $f_i = q * l_i$ represents a cost of operating a facility $i \in I$, where q is a constant set in the parameters that represents the linear relation between the cost of operating a facility and facility size. $\overline{f_i}$ represents a cost of closing an existing facility $i \in I$.

J is a set of customers, $|J| = n$. Each customer $j \in J$ represents a number of citizens living in a location with population s_j. Each customer has a spending budget b_j, $b_j = c * s_j$ where c is a constant set in the parameters which represents the linear relation between a population size in a customer location s_j and customer spending budget b_j. μ_{ij} represents the attractiveness of the facility $i \in I$ for a customer $j \in J$ and calculated with BSIM

$$\mu_{ij} = \begin{cases} l_j e^{-\frac{d_{ij}^2}{2\sigma^2}}, & 0 \le d_{ij} \le d_T \\ 0, & d_{ij} > d_T \end{cases} \quad (1)$$

where d_{ij} represents the Euclidian distance between the facility $i \in I$ and customer $j \in J$, d_T represents the maximum distance that a customer is willing to travel to use a facility, σ^2 represents the variance of the distribution for a store, and in the scope of this work defined by a constant parameter. The probability p_{ij} of a single customer $i \in I$ visiting a store $j \in J$ is

defined as

$$p_{ij} = \frac{\mu_{ij}}{\sum_{j \in J} \mu_{ij}} \quad (2)$$

An amount x_{ij} that customer $i \in I$ spends in the facility $j \in J$ is defined as the following

$$x_{ij} = p_{ij} b_j \quad (3)$$

The key objective of the current problem was defined as the maximization for the revenue of a facility chain with already present facilities of that chain and existing competitor facilities. The objective function can be defined as following

$$v_1 = \sum_{i \in \check{I}} \sum_{j \in J} x_{ij} - \sum_{i \in I^*} f_i y_i \quad (4)$$

Additionally, the problem can be extended to an additional mode where it allows to close the existing facilities. The objective function can be then defined as following

$$v_2 = \sum_{i \in \check{I}} \sum_{j \in J} x_{ij} - \sum_{i \in I^*} f_i y_i - \sum_{i \in \check{I}} \bar{f}_i \bar{y}_i \quad (5)$$

The optimization problem is defined as

$$\max(v)$$

such that

$$\sum_{i \in I} x_{ij} = b_j \,,\ \forall\, j \in J \quad (6)$$

$$x_{ij} \geq 0, \forall\, i \in \check{I}, j \in J \quad (7)$$

$$y_i \in \{0, 1\} \quad (8)$$

$$\bar{y}_i \in \{0, 1\} \quad (9)$$

The constraint (5) defines that the full demand for each customer must be satisfied. Constraints (7), (8) and (9) ensure the correct values are taken by the respective variables.

1.4. Solution approaches

Solving Facility Location Problems is challenging due to their combinatorial nature of the problem that usually leads to large solution space. Various solution approaches have been developed, including exact methods (e.g., branch and bound, cutting plane algorithms, etc.) and heuristic/metaheuristic techniques (e.g., genetic algorithms, simulated annealing, tabu search, etc.). Additionally, mathematical programming solvers and commercial software packages are often used to solve FLP instances efficiently.

Since in many cases, the facility location problem is NP-hard, it is computationally infeasible to solve exactly for large instances. Therefore, approximate algorithms are heavily utilized to produce efficient and effective solutions that provide near-optimal results while significantly reducing computation time and resources.

1.5 Clustering algorithms

Clustering algorithms are widely popular for the solution of FLP. They usually work by grouping the customers into clusters, where each cluster represents a group assigned to a certain facility. Such algorithms can help in identifying optimal locations for facilities based on the proximity or demand patterns. Also, they can be used in various preparation steps such as dividing the customers into groups to decrease the combinatorial space of a problem that can improve the efficiency of other algorithms. Clustering algorithms were rigorously studied for the solution of different FLP variations, with miscellaneous restrictions and objective functions [CKM+01] [SSJ+21]. The most popular clustering algorithms for FLP are:

1) K-means – divides the datapoints into k clusters. At the first step algorithm initializes *k* points called cluster centroids (or means), and on the second step it assigns the datapoints to the nearest centroid. The process repeats for a specifies number of iterations or until the algorithm stops reassigning the datapoints.
2) Hierarchical Clustering – works by building a hierarchy of clusters by iteratively merging and splitting the clusters based on their similarity. Can be implemented with the top-down approach when all datapoints start as a one cluster, which gets split into smaller ones, or with the bottom-up approach, where points start as their own clusters and get merged based on similarity. Unlike *k*-means, doesn't require a defined number of clusters in advance.
3) Density-Based Spatial Clustering of Applications with Noise (DBSCAN) – works by grouping closely packed datapoints together and separating regions of lower density.

The algorithm identifies dense regions as clusters and sparse regions as noise. It has high performance for datasets with the constant density. For changing density variation, optimizations were proposed in [MAR+18]. Like Hierarchical Clustering algorithm, does not require the specified number of clusters in advance. Can efficiently handle the irregularly shaped clusters.

4) Spectral Clustering – transforms data into lower-dimensional space with help of eigenvectors of a similarity matrix, and then applies classical clustering algorithms.

Before choosing the appropriate clustering algorithm it is essential to estimate if it is suitable for the particular variation of FLP and consider the characteristics of the data and other problem constraints.

1.6. Branch and bound algorithm

Branch and bound algorithm is used for a variety of FLPs often to find the exact optimal solution or the near optimal solution for a problem. Akinc and Khumawala were analyzing the application of the algorithm for Capacitated Facility Location Problem variation [AK77]. The branch and bound algorithm works by recursively partitioning the search space into smaller subproblems and exploring all of the possible solutions for a problem. In the context of FLP the algorithm is usually employed in 2 ways:

1) The algorithm can be used to exhaustively search the entire solution space of a problem to find the globally optimal solution. The algorithm usually starts with an empty solution with no facilities assigned, and then produces smaller sub-problems by assigning the facilities one by one to candidate locations. Such approach can be used for smaller and simpler problems. However bigger datasets and problems with more complex restrictions the effectiveness of such approach is very limited.
2) The algorithm can be combined with heuristic improvements. Another heuristic algorithm can be used to generate the initial solution, and then branch and bound algorithm employed to refine the solution by exploring promising regions in the search space. The lower bounds obtained during the search process guide the exploration and pruning of branches, ensuring that the search focuses on the most promising areas, resulting in significantly lower number of computations.

1.7. Geometric solutions

Geometric solutions can be employed for certain types of facility location problems. Geometric approaches leverage geometric properties and concepts to find efficient solutions. There are 2 main techniques for FLP: Voronoi diagrams and the geometric median.

Ravi and Garg [RG13] used Voronoi diagrams for solving FLP. As a result they get the efficient solution of $O(n \log (n))$ complexity. However, such approach cannot account for all constraints given for a specific problem, such as transportation costs between customers and facilities, capacity of a facility (for capacitated problem variation), etc.

Geometric median approach was used by Torricelli for the Fermat problem and consisted of building equilateral triangles from the sides of initial 3-point triangle. The point of intersection of the circumcircles bounding those equilateral triangles is the solution to the Fermat problem. But such geometrical solution cannot be applied to more general Weber problem.

1.8. Linear Programming

Linear Programming (LP) is an effective method that widely used across different kinds of optimization problems. In its core it involves maximizing or minimizing a linear objective function, while satisfying the given set of constraints. The LP problem can generally be defined by:

1) Objective function – a mathematical expression that represents the quantity to be maximized or minimized. The objective function is typically a linear function of the decision variables.
2) Decision variables – the unknowns or variables that we need to determine in order to optimize the objective function. These variables are typically a subject to certain constraints.
3) Constraints – the limitations or restrictions imposed on the decision variables. They can be expressed as a system of linear equations or inequalities.

Although the FLP is NP-hard problem, LP can have the polynomial time complexity for a given error rate but might have performance issues for large scale instances. Kratica, Dugošija and Savic [KDS14] were working on efficient Mixed Integer Linear Programming solution (MILP) for Multi-Level Uncapacitated Facility Location Problem (MLUFLP) and achieved sufficiently good results.

1.9. Greedy algorithms

Greedy algorithms are often used to solve facility location problems efficiently. This group of algorithms works by starting from the initial conditions and iteratively adding more facilities or changing the location of a facility. At each step the algorithm selects an option that provides the most immediate benefit, without considering the following options. The algorithm stops when the terminal condition is met (e.g., number of facilities or number of steps). Greedy algorithms often tend to provide suboptimal solutions that are different from the globally optimal solution. Usually, such algorithms can be used for the problems with large datasets, where the speed of computation is more important than precision of the result.

Guha and Khuller [GK99] worked on improvement for the greedy algorithm in the scope of Facility Location Problem and achieved substantial improvements for approximation guarantee.

1.10. Genetic Algorithms

Genetic Algorithms (GA) are used across variety of complex optimization problems, including FLP. Classical GA is population-based search algorithm which is inspired by the process of natural selection [KCK20]. Each individual (solution) in the population is represented by the chromosome that encodes the information about the solution. The chromosome is usually a binary string, consisting of smaller bits of information (alleles). The fitness function is used to estimate the performance of each population member and usually is an objective function of a specific problem. New populations of improved solutions are produced iteratively with the use of genetic operators:

1) Selection – the performance of each solution is estimated, top performers are selected to form the new generation.
2) Crossover – the chromosome alleles of the top performers from the previous generations are mixed to produce the new chromosomes. Since the chromosomes are usually represented by static length array of values, the process is very trivial, and often done by swapping random alleles in the corresponding places, as demonstrated in Figure 1.

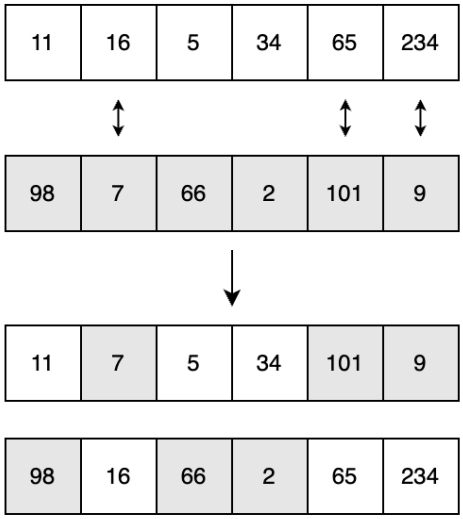


Figure 1. Simplified demonstration of crossover process.

3) Random mutation – the alleles in newly generated chromosomes are introduced with random changes. Such step often helps the further generation to escape local optimums and seek for better solutions.

GA can be decomposed into the following steps:

1) Initialization – a population of possible solutions for FLP is randomly generated, each solution gets assigned with a gene (or chromosome).
2) Fitness evaluation – each gene is evaluated by the fitness function. For FLP the fitness function is defined from the objective function of the problem.
3) Selection – the fittest genes are selected to be parents for the next generation.
4) Genetic mutations – selected genes are introduced with slight random changes, or the changes from its peers to create the new population.
5) Replacement – new generation replaces the old one, and the process continues from the step 2 of the algorithm.
6) Solution extraction – after the given number of generations the algorithm gets terminated.

The flow chart of GA can be seen in the Figure 2.

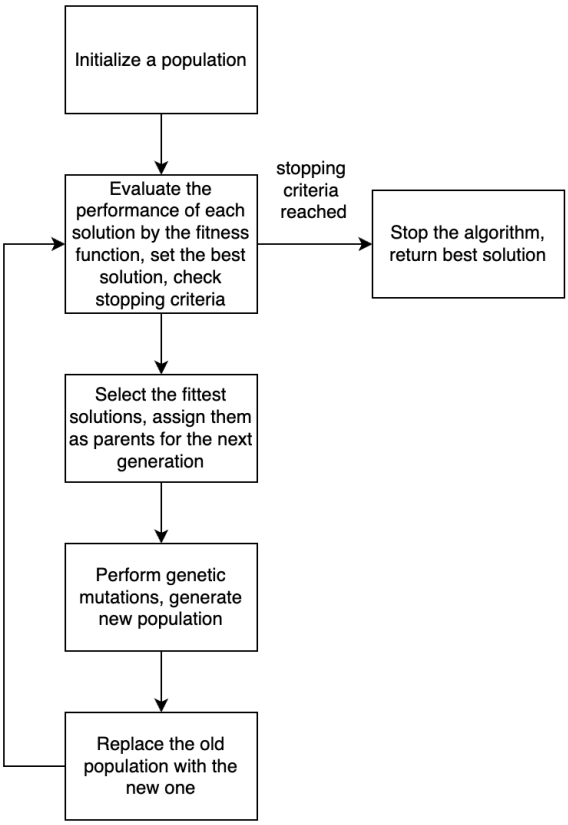


Figure 2. Flow chart of the Genetic Algorithm

Arostegui, Kadipasaoglu and Khumawala [AKK06] were comparing the performance of various heuristic algorithms in the scope of FLP, including GA. The conducted experiments for the particular types of FLP showed that the GA had less overall performance comparing to the Tabu Search (TS) and Simulated Annealing (SA) algorithm. But with certain constraints (like time limit), it showed better results than others.

1.11. Neural Networks

The application of Neural Networks (NN) has been studied for different types of FLP. NN are usually used to optimize certain parts of other algorithms that are more suitable for solving FLP. However, some researches [Har19] studied the application of NN as core algorithm for problem solution.

NN are usually used for FLP in several ways:

1) NN can be used for customer demand prediction in dynamic FLP, to predict the operational costs more accurately in the future. The advantage of the NN over the other approaches is that it can take multiple complex factors like demographics, purchasing behavior, etc. and produce accurate predictions for future changes.

2) NN can help to estimate transportation costs between facilities and customers more accurately, considering historical traffic jams data, traffic patterns over the course of a day, etc.
3) NN can help to identify the forbidden areas for placing facilities, paving the roads between customers and facilities etc., with help of geographical image data and image recognition.

1.12. Tabu Search

Tabu Search (TS) is a metaheuristic algorithm used for solving various optimization problems, including FLP. The key idea of the algorithm is to maintain the list of recently explored solutions, that should be avoided in subsequent iterations. Also, the algorithm may allow worsening moves if no improving moves are available. Unlike the local search, it allows the algorithm not to get stuck on suboptimal local solution and continue seeking for the global optimum.

There are multiple parameters of TS that can differ from implementation to implementation, such as: stopping rule, aspiration conditions (define conditions for elements to escape the tabu list), neighborhood structures [Glo90][Sun06].

TS uses flexible attribute-based memory structures designed to permit evaluation criteria and historical search information to be explored more thoroughly, than in other rigid memory structures. There can be identified 3 main memory components for TS [Glo90][Sun06]:

1) Short term memory – the core of an algorithm, based on a set of tabu and aspiration conditions.
2) Intermediate memory – restricts the search within available options, to improve the search results.
3) Long term memory – helps the TS to explore new regions in the search space, to diversify the search results.

Before starting the TS, a problem and the objective function should be defined. Following parameters should be set: maximum number of iterations, tabu tenure (the number of iterations a solution remains in the tabu list), aspiration criteria (conditions under which a solution in the tabu list can be revisited) and stopping criteria.

The TS can be decomposed into following steps:

1) Generate an initial solution to start the algorithm. This solution can be random or obtained through a heuristic or other techniques depending on the problem. Calculate the

value of objective function for the current solution and assign it as the best solution found so far.

2) Generate a set of neighboring solutions (candidates), without the tabu neighboring solutions.
3) Calculate the objective function value for each neighboring solution and keep track of the best one found among them. Choose the next solution based on a selection strategy. This can be done by considering the objective function values of neighboring solutions, the aspiration criteria, and the tabu list. The selection strategy should balance exploration (searching for new promising solutions) and exploitation (exploiting the best solutions found so far). Add the selected solution to the tabu list, making it unavailable for a certain number of iterations (tabu tenure). If the tabu list exceeds its maximum size, remove the oldest entry.
4) Check if the stopping criteria are met. This can be based on a maximum number of iterations, a time limit, or other conditions.
5) If the selected solution is better than the current best solution, update the best solution with the selected solution.
6) If stopping criteria is not met, start from step 2, else return the best solution.
7) The flow chart of TS can be seen in the Figure 3.

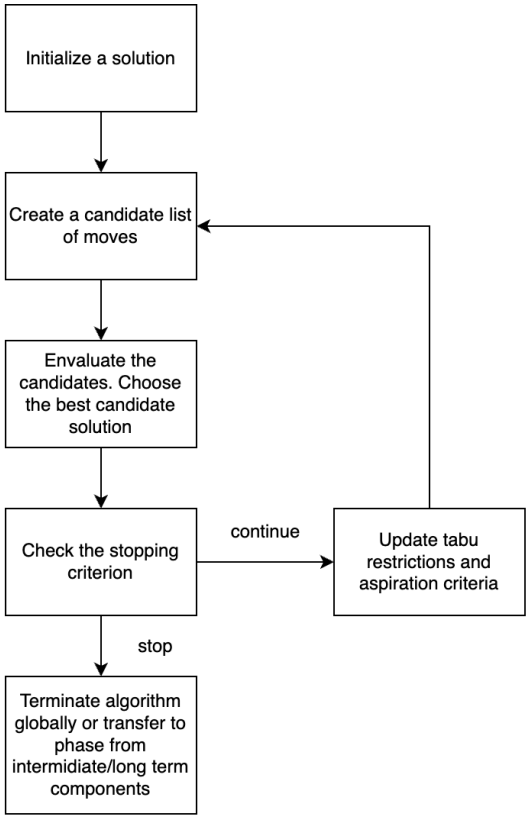


Figure 3. Tabu Search algorithm flow chart

In their research Arostegui, Kadipasaoglu and Khumawala [AKK06] concluded that TS is far superior for solving the CFLP comparing to SA and GA, and in average has better performance for most other variations.

1.13. Numerical solution

Kulin and Kuenne proposed an efficient algorithm for the numerical solution of the generalized Weber problem [KK62]. It consists of several steps:

1) Random initialization of potential facility locations.
2) Calculation of a gradient value from existing facility-customer connections.
3) Calculation of a new facility location from the gradient value, assignment of the new facility location.
4) Checking the terminal condition (number of iterations), if not met, continue from step 2.

In the Kulin's and Kuenne's research, the algorithm produced an efficient solution within the 7^{th}, 8^{th} iteration for the 15 problems consisting of no more than 24 possible facility locations.

1.14. Real world solution examples

Facility Location Problem is a fundamental optimization challenge with significant practical implications. By addressing the problem effectively, organizations can optimize their facility locations, streamline supply chain operations, reduce costs, enhance customer satisfaction, and gain a competitive edge in the market. The following sections will delve into specific variants of the FLP, exploring solution methodologies in real-world applications and case studies.

1.15. Facility Location Problem with Tabu Search

Csoke was working on application Facility Location Problem [Cso15] to determine if the Oak Lawn Police and Fire departments in Chicago, IL have optimal locations for their stations. Firstly, the general approach for solving the facility location problems was analyzed from the 1986 mathematical competition problem. The task there was to identify 2 emergency facility locations to minimize the total response time. The map was defined as 10x5 grid with 2 forbidden regions with numbers in cells representing the demand. The distances between cells were defined by travel time and took different amount in vertical and horizontal directions. The solution for the problem was implemented with Excel and Visual Basic and involved calculating sums of distances between all customers and facilities for all possible facility locations, and then picking the most optimal one. Although such approach can be used for small instances of FLP, the complexity grows exponentially with the number of possible facility locations and more facilities to add, and therefore might be not the best option for bigger inputs. In the subsequent chapters, Csoke described the preparation for the problem of analysis of the Oak Lawn Police and Fire departments' locations.

Firstly, data on police and fire calls was gathered from the department, and with the use of Google Maps plotted to determine the exact geographical location of calls. Some assumptions were made such as that all emergencies will get the vehicles from the nearest facilities, no responders will go from facility to facility, etc. to make it simpler to analyze.

To solve the problem Csoke used an FLP solver, introduced by Erdogan [ESV19], that was initially created to solve the vehicle routing problem, with a Tabu Search algorithm. The solver was then introduced with 749 possible locations for fire and police departments. Objective function was selected as a minisum. To calculate the distances between the objects, the Bing Maps service within the FLP Solver was used. As a result, program was running for 2 days, and had 252 210 iterations. Because of the substantial computation time, an

optimization approach was proposed as well. The idea is to aggregate the demand points, to reduce the total number of customers.

The research of Csoke has a good overview of a FLP solving approaches, with the use of modern (at the time of research) tools. Although the solution computation time for this problem was not extremely significant, it may have a greater impact on other FLP cases and therefore more optimizations for the proposed approach or other solution techniques should be considered.

1.16. Uncapacitated Facility Location Problem with genetic algorithm

Kumar [Kum13] was working on optimization of the storage facilities for Indian clay producing company. The aim was to re-allocate the existing storage locations to new locations in order to decrease total operating costs. The problem size can be classified as small with input data containing 2 existing plants locations, 47 customer locations and 47 candidate facility locations for 5 warehouses.

The objective function being minimized accounted for:

1) Transportation costs between plants and warehouses.
2) Transportation costs between warehouses and customers.
3) Fixed cost of opening new warehouse facility in the candidate location.
4) Variable costs of opening new warehouse facility in the candidate location.

The following restrictions were as well considered for a model:

1) The total number of products supplied from a warehouse to a customer should be equal to the customer demand.
2) Total amount of products supplied from a warehouse should not be greater than the capacity of a warehouse.
3) The candidate location can place only 1 warehouse facility.
4) The traveling costs are symmetric, and directly proportional to a distance between facilities on the map.

To solve the problem, heuristic Genetic Algorithm was utilized. The chromosomes in GA consisted of 2-dimensional binary strings and contained information about the locations of chosen candidate facilities. Problem's objective function was used as a fitness function for GA. To generate new population the crossover and mutation techniques were utilized. The program was then implemented in C++ with the following parameters:

1) Population size = 50

2) Max number of generations = 40
3) Crossover rate = 80%
4) Mutation rate = 10%

The resulting facility locations were estimated to produce the total costs of 13.7 crores (Indian unit of currency), which is 6.5% better when compared to existing costs of 14.65 crores.

In conclusion, the research contains a good summary of solution methods for UFLP, with additional restrictions, and a thorough overview of the solution approach using the GA. However, the paper doesn't contain any information on the solution speed for the problem. Although the size of examined problem might not have had a substantial running time, with the increase in input data, the solution time increases drastically. Therefore, such problem formulation can be a good basis for further investigation of GA optimizations for UFLP.

1.17. Facility Location Problem with Neural Networks

Haralampiev was working on researching the application of Neural Networks (NN) for FLP [Har19]. The idea was to design the NN model with small number of parameters and not needing much tuning. The model was tested on minmax and minisum variations of FLP. Minmax variation was reduced to solving the sequence of minisum problems. The architecture for proposed NN is represented in the Figure 4.

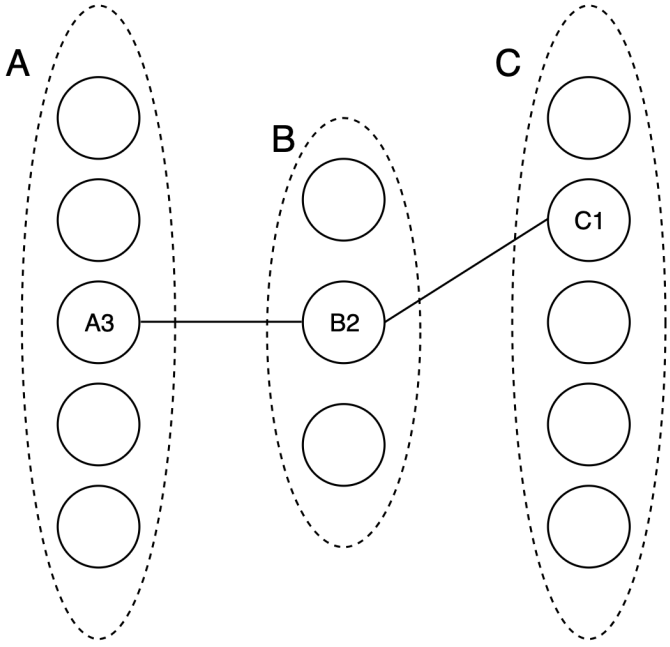


Figure 4. Neural network architecture for Facility Location Problems

Layer A represents the clients, layer B represents the facilities to be located and layer C represents the candidate facility locations. Layers A – B and B – C are fully connected, and directed from A to B and from B to C. The weight of the edge between nodes A3 and B2 represents the amount of goods serviced to customer A3 from facility B2 (values are normalized between 0 to 1). The weight of the edge between nodes B2 and C1 indicates to what degree the facility B2 is located on the location C1. For each node in A the sum of weights in outgoing edges should be equal to 1, which implies the exact satisfaction of customer's demand. The weights of edges between B and C are taking intermediate values at the start, but eventually some values start dominating. The algorithm works by randomly initializing the weights of NN. With the number of iterations, the edge values between B and C start to converge to 1. Final solution is obtained by assigning the facilities to the locations with dominating edges between B and C, and assigning the customers to facilities with respect to the edges between A and B.

Overviewed NN was tested on the multiple groups of problems, including one with the real-world data from Bulgarian road network. For such problem groups multiple instances of problems were generated. For each instance a subset of possible facility locations was introduced to the model, to obtain the final optimal answer. The performance of the NN was compared with the Local Search (LS) algorithm. As the result, for minisum both methods showed an excellent performance, and have obtained an optimal solution most of the time. LS showed slightly better performance, because of its ability to run a number of searches undependably in each iteration. For minimax problem, the results were overall worse than for minisum, but NN had significantly better performance than LS.

As a conclusion, it is stated that on the simpler instances of problems the results of proposed NN are comparable to the ones of LS, and with the harder problems, the results are significantly better for the NN.

Haralampiev have done a great overview of the NN models usage for FLP, proposing own architecture with the least number of parameters for the simple FLP. However, such model may not be the best choice for harder FLPs, with more input parameters. So, in further researches the proposed architecture can be taken as the basis for improvements to solve more complex FLPs with higher number of input parameters.

1.18. Competitive Facility Location Problem with Hierarchical Search

Perera, Aglietti, and Damoulas [PAT23] were researching new approaches for Competitive Facility Location Problem (CFLP). They introduced a new Bayesian Spatial Interaction Model that helped to estimate the location-specific revenues and new framework to find the best possible location for new facilities with maximized revenue. Also, a new Hierarchical Search Algorithm (HSA) was introduced. The mentioned approaches were tested on two large supermarket and pub markets in Greater London.

Firstly, authors developed a Bayesian Spatial Interaction Model (BSIM) that produces the predictions of the revenues for each facility, considering various characteristics of facilities and customers. The core of the model is built from the Gaussian distribution, that represents an "attraction" of each facility, where the probability distribution function is defined by the characteristics of each facility.

To solve the problem, authors propose a new heuristic algorithm called Hierarchical Search (HS) that starts from initial locations of facilities in broader region and narrows it down to the optimal solution. The algorithm works at several levels. At the first level the candidate facility locations are split into random samples. Then the optimal sites are calculated independently for each sample. The optimal locations are then used at the next level, to calculate more optimal solution. The space is represented as a grid and the neighboring locations are calculated with help of quad tree to improve the computational complexity. The algorithm continues to recursively subdivide the grid cells where the optimal solution was found and seeks for the optimal solution there until the improvement of an objective value is greater than a defined threshold. When the improvement becomes lower than that value it stops.

To obtain a better set of initial locations, that end up in more optimal solution with less steps three different sampling methods were proposed. With the first method the candidate locations are generated from the midpoints of the initial grid, which results in facilities located in the regular distance from each other. Second and third methods take into account the customer demand density. The probability density function considers the customers' spending power to estimate the available spending capacity for each location. Second method uses the Inhomogeneous Poisson Point Process (IPPP) to generate the initial set. In the IPPP the intensity function is nonconstant which allows to account for the spending power in the location to produce the optimal initial locations. The third method works by creating a grid and calculating the average demand value for each cell, and then repeating the decomposition to smaller cells for parent cells with higher demand up to q steps. The candidate facility locations

are then defined from the middle points of the cells. As a result, more candidate facilities are assigned to the places of higher demand.

The algorithm was then tested with three objective functions and three sampling algorithms. The third sampling method yielded the best results for all objective functions, producing the best optimal value approximately 60% of the time, while the IPPP performing the worst with only 2% of best the optimal values.

The research provides a great framework to solve the FLP with various parameters for facilities. The proposed Hierarchical Search is highly adjustable, with option to extend it with other sampling methods or objective functions, which can be done in further research. Unlike the previous research [ESV19] [PAT23] the proposed framework accounts for already present facilities and other factors, that help to model most of real world FLPs more accurately.

1.19. Results of the literature review

After an overview of the existing FLP solutions in the available literature, a few candidate algorithms that have great performance were identified: Tabu Search (TS), Simulated Annealing (SA), Genetic Algorithm (GA), and Hierarchical Search (HS).

TS focuses on finding optimal solutions within a reasonable amount of computation. Employing short-term memory and long-term memory allows the algorithm to efficiently propagate the best solutions (intensification procedure) and search the other regions in the solution space (diversification procedure) when the algorithm is getting stuck in the local optimum. Tabu list structure allows the algorithm to escape the duplicated checks on the same solution with the trade-off of using more memory. It can be a significant benefit considering that a calculation of the objective function can be an extensive task.

SA and GA show a great performance when solving the NP-hard problems, however, Arostegui, Kadipasaoglu, and Khumawala showed [AKK06] that TS has a superior performance for the Capacited FLP which is closely related to UFLP and good performance Multi-Period FLP and Multi-Commodity FLP variations. For CFLP the GA showed a strong initial descent but a significant slowdown before reaching the optimal solution while the SA performed with gentler initial descent but showed the same solution quality as TS after some time.

The HS proposed by Perera, Aglietti, and Damoulas [PAT23] appears to be an efficient solution method that can significantly decrease the number of computations by employing the quadtree data structure to search the neighboring facility locations. However, the significant

drawback of the mentioned algorithm that was not addressed in the study is that it does not contain any steps to guard it from getting stuck on the locally optimal solution in Multiple FLP variations.

Therefore, TS was selected for the implementation due to its proven efficiency for the particular problem type.

2. Solving the Facility Location Problem

This chapter is devoted to the solution of Competitive Uncapacitated Facility Location Problem with Tabu Search and description of different strategies to that are proposed to improve the performance of the algorithm.

2.1. Data gathering

The most realistic FLPs include dozens of parameters to accurately predict the customer behavior and facility attractiveness values. Advanced researches comprise population, socio-demographic factors etc., to determine customer demand distribution. Facility data incorporates location, design (i.e., size of a shop), customer rating, proximity to the means of public transport, tourist attractions and others [PAT23].

In the current work, the decision was made to utilize only publicly accessible data that can be easily obtained from the open data repositories to ease further applicability with other data setups.

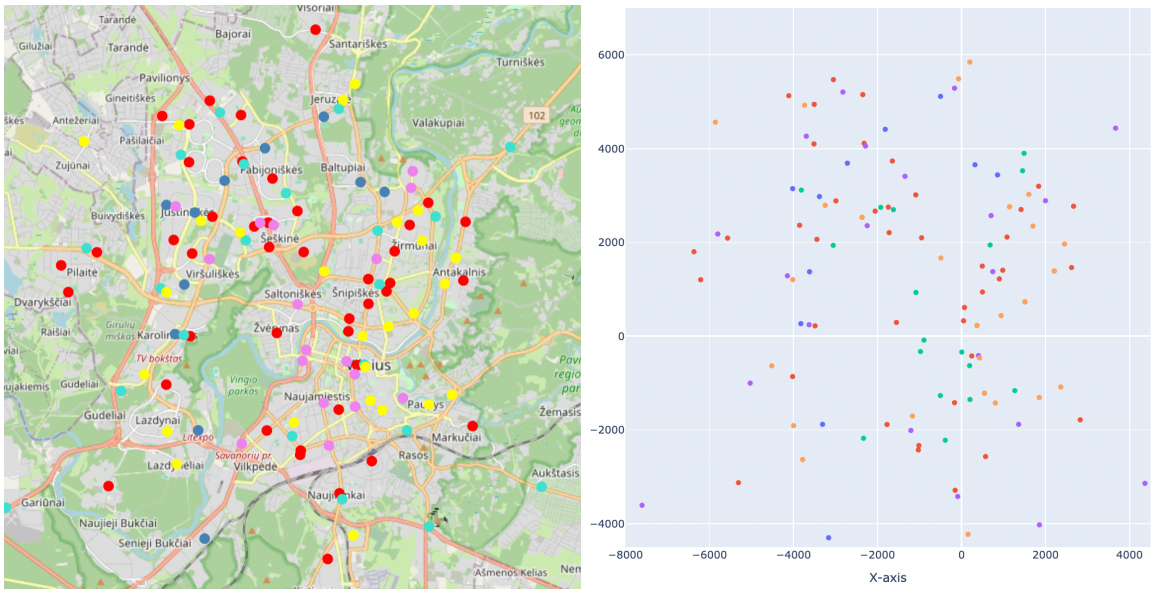
2.2. Facility locations

As a primary facility dataset, the locations of the most popular retail store chains in Vilnius, Lithuania were picked. To get the data about the existing stores, the facility locations were extracted from the Google Maps service [Goo24]. To simplify the computations, the data about shop locations was converted from spherical coordinates to cartesian coordinates. The facility data was extended with the randomly generated facility size data, to achieve a closer alignment to the real world CFLP. The facility size parameter was defined to correspond to the floor area of the shop in the m^2. The actual area sizes and their distribution were taken from the data presented in one of the retail chain websites [Nor24]. The facility dataset was defined with the following fields:

- lon – longitude of the facility location.
- lat – latitude of the facility location.
- label – name of the retail chain.
- size – floor area of the facility in m^2.

The actual shop locations of 5 most popular retail store chains in Vilnius are represented on the Figure 5a and Figure 5b. The actual facility size distribution is represented on the Figure 5c.

a) b)



c)

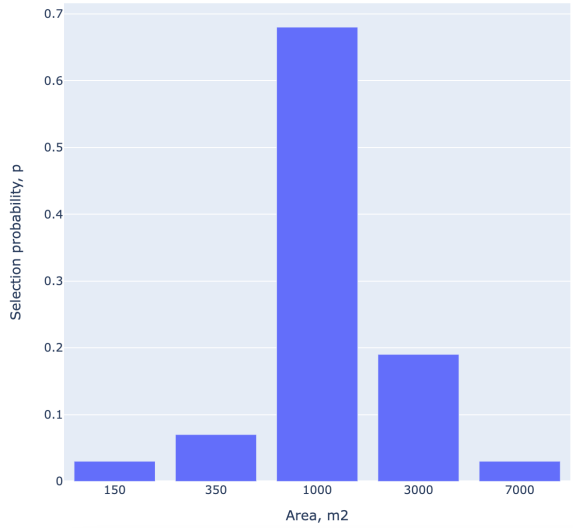


Figure 5. Shop locations of 5 most popular retail store chains in Vilnius from Google Maps. a) Shop locations on the geographical map. b) Shop locations in cartesian coordinates. c) Shop size distribution.

2.3. Customer data

The data about the clients was taken from the Lithuanian Population and Housing Census 2021 of State Data Agency service [Gyv21] and represented as a grid with 100m x 100m granularity. Each cell in the grid contains an information about the number of citizens living within this region. To simplify the computations, the data about customer locations was

converted from spherical coordinates to cartesian coordinates. The customer dataset was defined with the following fields:

- lon – longitude of the cell location.
- lat – latitude of the cell location.
- population – number of people living in the cell.

The actual population data is represented on the Figure 6.

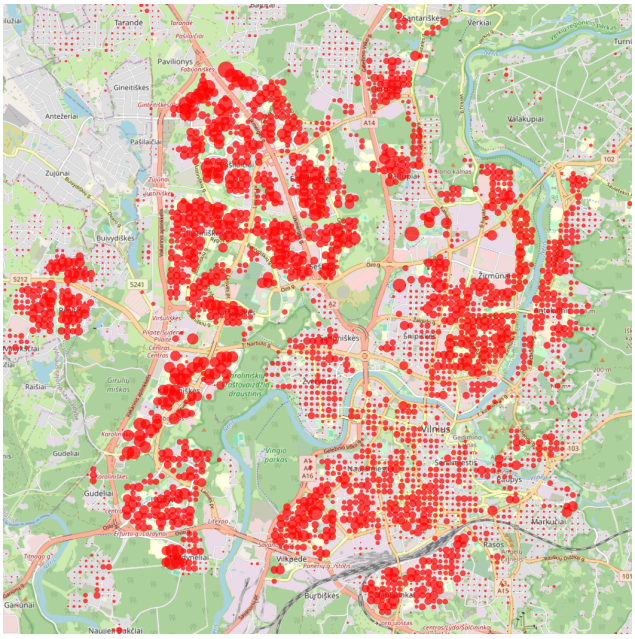


Figure 6. Population data from Lithuanian Population and Housing Census 2021.

2.4. Generating candidate locations

To generate customer demand and shop supply distributions the truncated BSIM described by Perera, Aglietti, and Damoulas [PAT23] was used. The example of the base distribution for BSIM is represented on the Figure 7a. The variance and truncation values for the customer demand and shop supply distributions were set to the constant values in the program with the possibility to later be extended by more specific demographic data and store information. To generate the shop supply distribution heatmap, the actual shop data was used represented on the Figure 5a and Figure 5b. To generate the customer demand distribution heatmap, the actual customer data was used represented on the Figure 6. The results of generated customer demand and shop supply distribution heatmaps are represented on the Figure 7b, Figure 7c. These values were then used to calculate the demand and supply difference rate, Figure 7d. To generate the candidate locations for the new facilities the inhomogeneous Poisson point process (IPPP) was used with the probability density function

that corresponds to the demand and supply difference rate, Figure 7d. The resulting candidate locations are demonstrated on the Figure 7e.

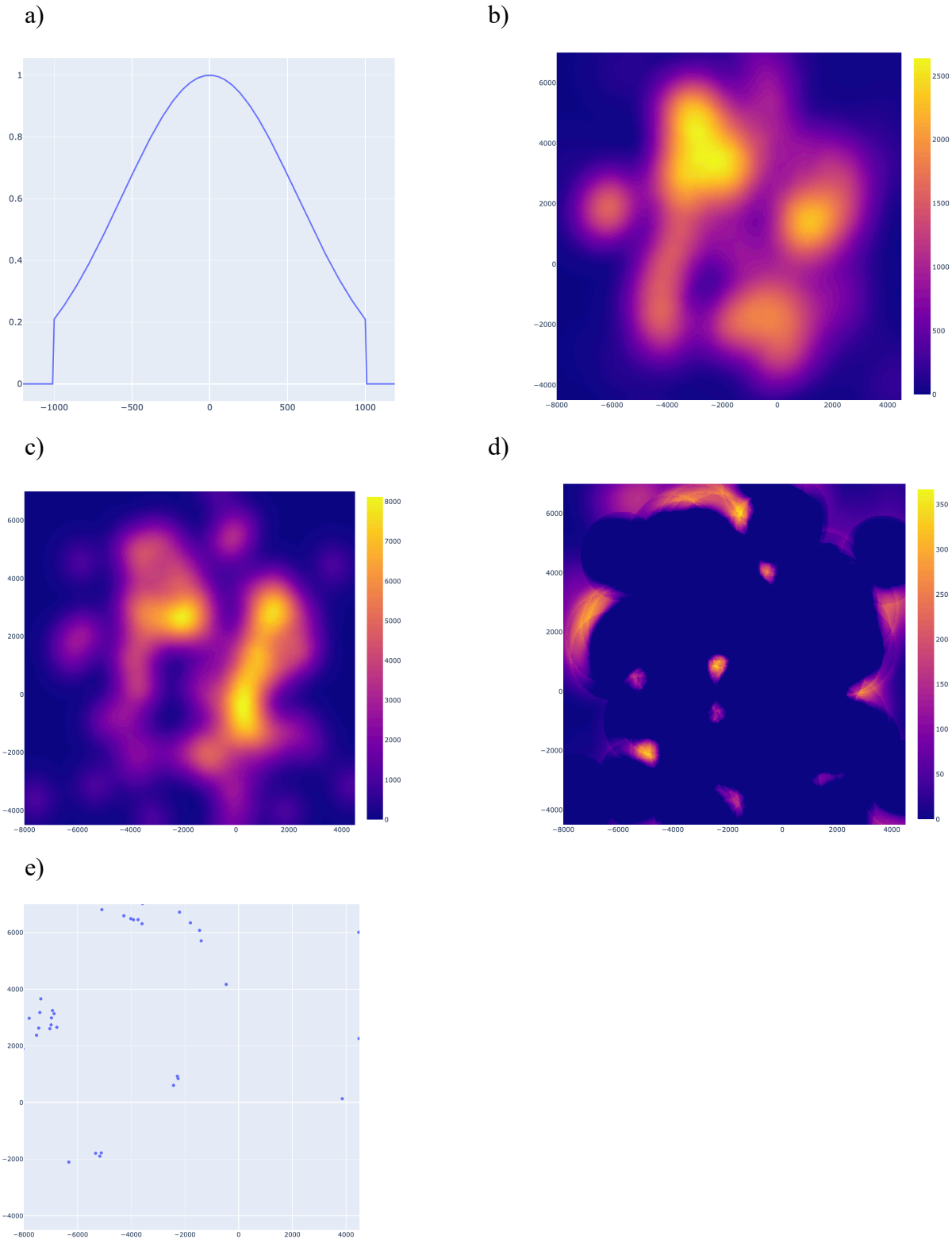


Figure 7. Process of candidate locations generation. a) 2D cut of truncated BSIM distribution. b) Customer demand generated with truncated BSIM and population data. c) Shop supply

generated with truncated BSIM and existing shop locations. d) Demand and supply difference ratio. e) Candidate locations.

2.5. Algorithm implementation

The core of Tabu Search algorithm lies in the initial solution strategies, diversification and intensification procedures and memory structures. Each of these components is focused on the algorithm performing the least number of iterations yet producing the highest value solution. An initial solution algorithm helps to find the optimal starting point for the algorithm to further improve the solution. Intensification and diversification procedures help to improve the current solution or explore the other regions in the solution space. Short-term memory and long-term memory structures were utilized to reduce the number of computations. For each step, multiple strategies were defined that will be overviewed in the following sections.

2.6. Initial solution

To find the initial solution, 2 algorithms were introduced: random and greedy heuristic. The random algorithm is focused on exploring the solution space with randomly opened facilities before giving the initial suboptimal solution while the greedy heuristic algorithm finds the suboptimal solution from the start.

The random algorithm consists of the following steps:

1) Generate n solutions where each solution has k candidate facilities opened. $k \in [k_l, k_u]$, k_l, k_u- constant values.
2) Calculate the objective function value z for each solution, the best solution becomes the initial solution.

The greedy heuristic algorithm was inspired by the *ADD*-procedure and facility placement priority rules described by Domschke and Drexl and was adapted for the UFLP to account for existing competitor facilities [DD85]. The algorithm consists of the following steps:

1) Let L_1 be a set of existing facilities, L_2 be a set of candidate facilities. $\forall\, l_i \in L_2$: $L_i = L_1 \cup l_i$. Calculate the objective function value for L_i. Put the candidate facilities into the ordered set L_3 by increasing value of the objective function.
2) Let L_4 be a set of opened candidate facilities. $L_4 = \emptyset$. $\forall\, l_i \in L_3$: if $\Delta z_i > 0$ and BSIM of l_i does not interfere with facilities L_4 then $L_4 = L_4 \cup l_i$.
3) $\forall\, l_i \in L_3$: if $\Delta z_i > 0$ and $l_i \notin L_4$ then $L_4 = L_4 \cup l_i$.

4) $L_5 = L_1 \cup L_4$. L_5 – initial solution.

The BSIM interference check in step 2 verifies that the candidate facilities are not located close to each other which may lead to the oversaturated regions with poorer results. Step 3 verifies that facilities skipped in step 2 are getting added if they increase the objective function value.

Each of the algorithms has its strengths and weaknesses. The random algorithm can explore a wider space of options and produce different results between runs however requires a lot of computations to produce good solutions. The heuristic greedy algorithm produces a good suboptimal result from the beginning with a reasonably low number of computations. However, because it relies on the determined order of facilities, it produces the same result every time, which may harm the main search algorithm to find the globally optimal result.

2.7. Memory structures

The short-term and long-term memory structures are the key components that allow the Tabu Search algorithm to show a superior performance for most of the tasks when compared with other similar meta-heuristic algorithms at the expense of using more memory. Because calculating the objective function for a solution is an expensive task, it is usually easier to store and retrieve the results of already calculated solutions than to recalculate it more than one time.

Short-term memory is used to record the change of status for a facility when producing the potential solutions during the intensification step. It is implemented in the form of a hash map where the keys are the indices of facilities, and the values are moves on which the facility status was changed for the last time. After facilities are added to the short-term memory, they are getting banned from changing their status for several iterations. Such an approach provides a couple of benefits for the algorithm:

1) If neighboring solutions that were generated during an iteration do not lead to the improvement of the result, they will not be generated and evaluated during the next iteration, resulting in fewer iterations to find the next move.
2) Because short-term memory structure only stores the facility index and a move of the last status change for a facility, it uses far less space when compared to long-term memory and is cheap to have.

The representation of short-term memory is depicted on the Figure 8a.

Long-term memory is used to record the evaluated (i.e., visited) solutions. It is implemented as a hash set where the elements are binary vectors of Boolean values. The Boolean values within the vectors indicate if the facility is open or closed while the indices of

these values correspond to the indices of the facilities. In addition to the obvious benefit of not repeating the calculations for the same solution, long-term memory promotes the algorithm to not explore the same solution region for a long time which leads to more efficient exploration of the solution space in general. The representation of long-term memory is depicted on the Figure 8b.

In addition to the advanced short- and long-term memory structures, the algorithm has a primitive counter of iterations since the last improvement was produced that is used to switch the algorithm to diversification mode when the algorithm is getting stuck at the local optimum and eventually to terminate the algorithm.

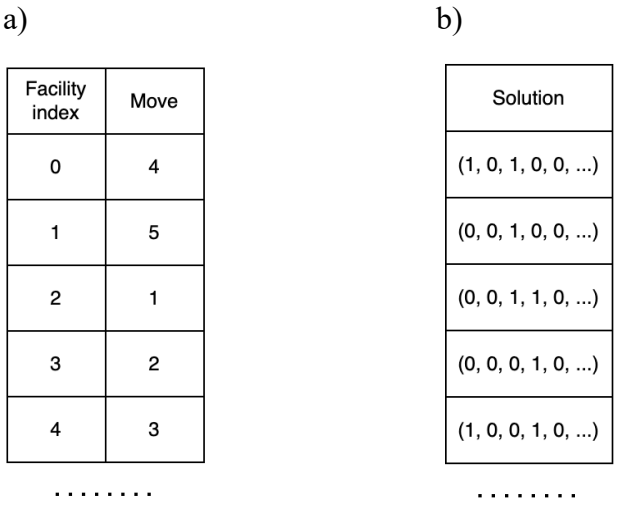


Figure 8. Representation of short- and long-term memory structures. a) Short-term memory. b) Long-term memory.

2.8. Aspiration condition

While most of the tabu search implementations have an aspiration condition [Glo90, 84] (i.e., aspiration criteria) that is designated to relax the strict tabu conditions and allow the worsening moves for the algorithm that should enable it to escape the local optima, the presented algorithm was implemented without it. Despite improving the diversification step, the downside that the aspiration condition introduces is that the candidate solution must be evaluated to obtain the objective function value before being filtered out from the long-term memory to be able to be checked for the aspiration condition. Removing the aspiration criteria allows to move the filtering step before the objective function evaluation. The difference between these 2 approaches is presented on the Figure 9.

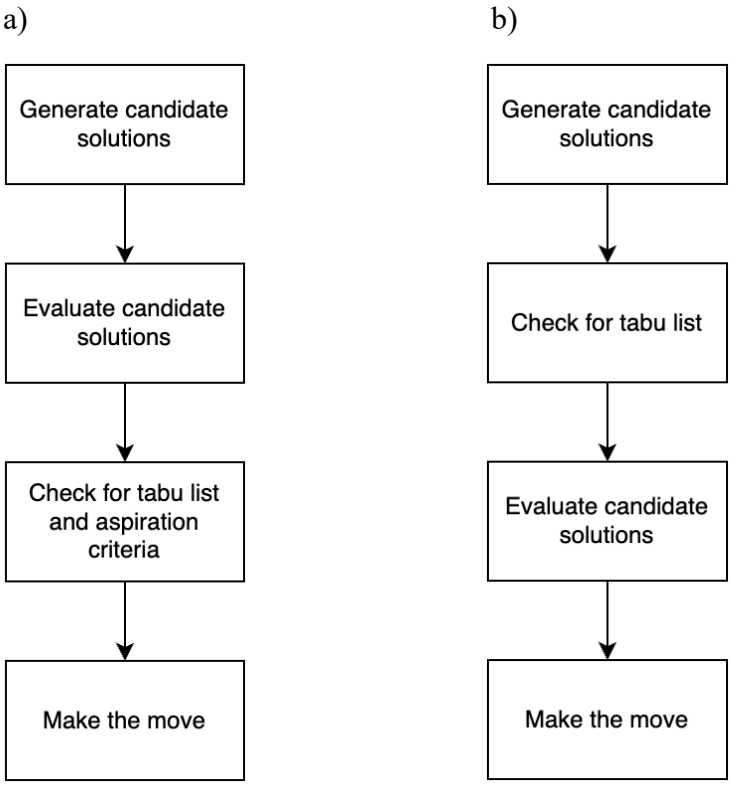


Figure 9. Tabu search aspiration condition implementation approaches. a) Tabu search with aspiration condition. b) Tabu search without aspiration condition.

The proposed implementation allows to decrease the number of duplicated solution evaluations. To reduce the consequences of the algorithm's reduced ability to overcome the local optimum, the diversification step was adjusted towards more aggressive exploration.

2.9. Intensification procedure

The intensification procedure allows the algorithm to efficiently explore the neighborhood solution space to further improve the objective function value. The conceptual example of the intensification procedure is demonstrated on the Figure 10a. Generally, intensification rules encourage one or multiple solutions that were found to have good performance to produce consequent solutions. In the current implementation, the algorithm tracks only one good solution within the iteration and produces the next neighboring solutions from it. A neighboring solution is generated by changing a random facility status in the current solution to the opposite. If a facility is present in the short-term memory, the tabu rules are checked to define if the facility is allowed to change the status. Such an approach allows the algorithm to efficiently improve existing solution incrementally, preserving the best facility combination so far.

a) b)

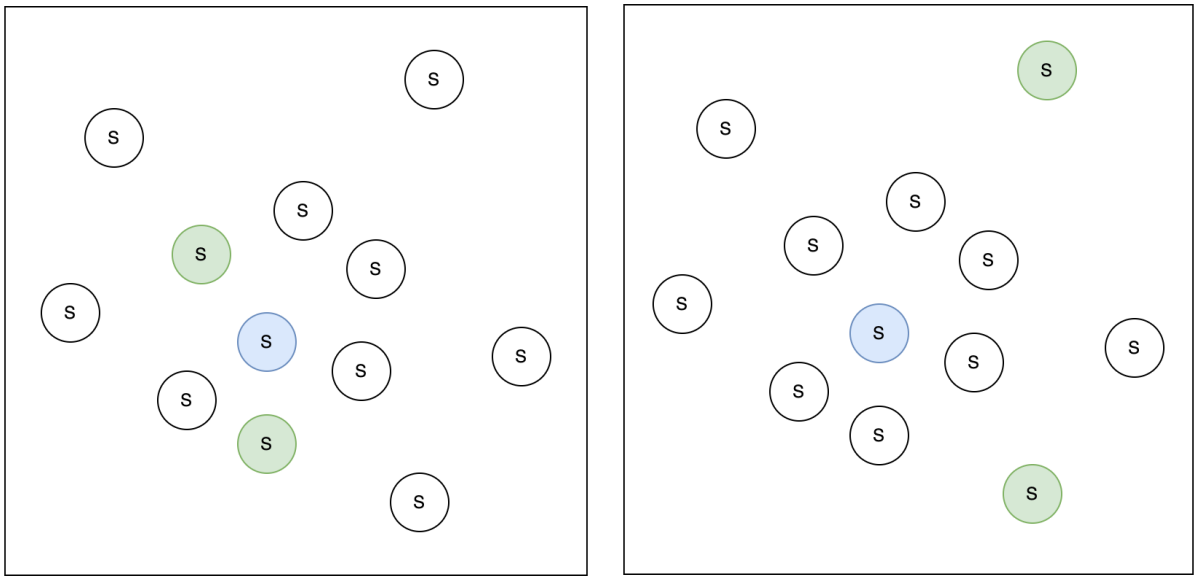


Figure 10. Diversification and intensification procedures conceptual example. Blue nodes represent the current solution, green nodes represent the candidate solutions. a) Intensification procedure, more similar candidate solutions are chosen. b) Diversification procedure, more diverse candidate solutions are chosen.

2.10. Diversification procedure

The diversification procedure allows the algorithm to escape the locally optimal solution and explore other regions. The conceptual example of the diversification procedure is demonstrated on the Figure 10b. In the scope of this work, 2 different strategies were developed for the diversification procedure: *remote solutions* strategy and *remove facility and get neighbors* strategy.

At the core of *remote solutions* strategy lies the recursive generation of neighboring solutions for a specified number of times. The main advantage of this approach is that the degree of remoteness for generated solutions can be dynamically updated by changing the number of recursive iterations. If the algorithm is getting stuck on the locally optimal solution for a number of iterations, the degree of remoteness for this diversification strategy can be increased to explore further solutions.

The *remove facility and get neighbors* strategy was developed to address the problem of early saturation of existing demand surplus when the number of candidate facilities is significantly larger than the number of opened candidate facilities. Since the *get neighbors* step randomly chooses a facility to change its status, in case there are significantly more closed candidate facilities than opened the step will be more likely to choose opening a new facility

and not closing existing. If all demand was already efficiently allocated to opened facilities, the opening of a new one will inevitably lead to worsening the objective function value. The closing of the existing shop releases the demand that can be once again allocated to the other facilities combinations. Additionally, like with the *remote solutions* strategy, the number of closed facilities and generated neighboring solutions can be dynamically adjusted to help the algorithm escape the local optimum.

2.11. Closing existing facilities

To enhance the algorithm's overall quality and effectiveness, an additional setting was introduced, to cover different scenario for closing the existing facilities. The price of removal of a facility is set in the parameters as a relation to the price of operating a facility. Removing the existing facilities allows the algorithm to additionally estimate the adequacy of their placement and try finding an improved combination of facilities' locations that covers the demand better and produces a higher revenue for a chain.

An example of more optimal facility placement with the closing of an existing facility is demonstrated on the Figure 11. Figure 11a shows a single facility with all demand of both customer clusters allocated to it. Figure 11b demonstrates the demand allocation with two additional competitor facilities that are now serving most of the demand. Figure 11c is the demonstration of the most optimal solution for the target facility chain, where new facility locations are opened in the center of customers' clusters allowing them to cover more demand. The initial facility gets removed because it is no longer capable of covering the additional demand and producing more revenue.

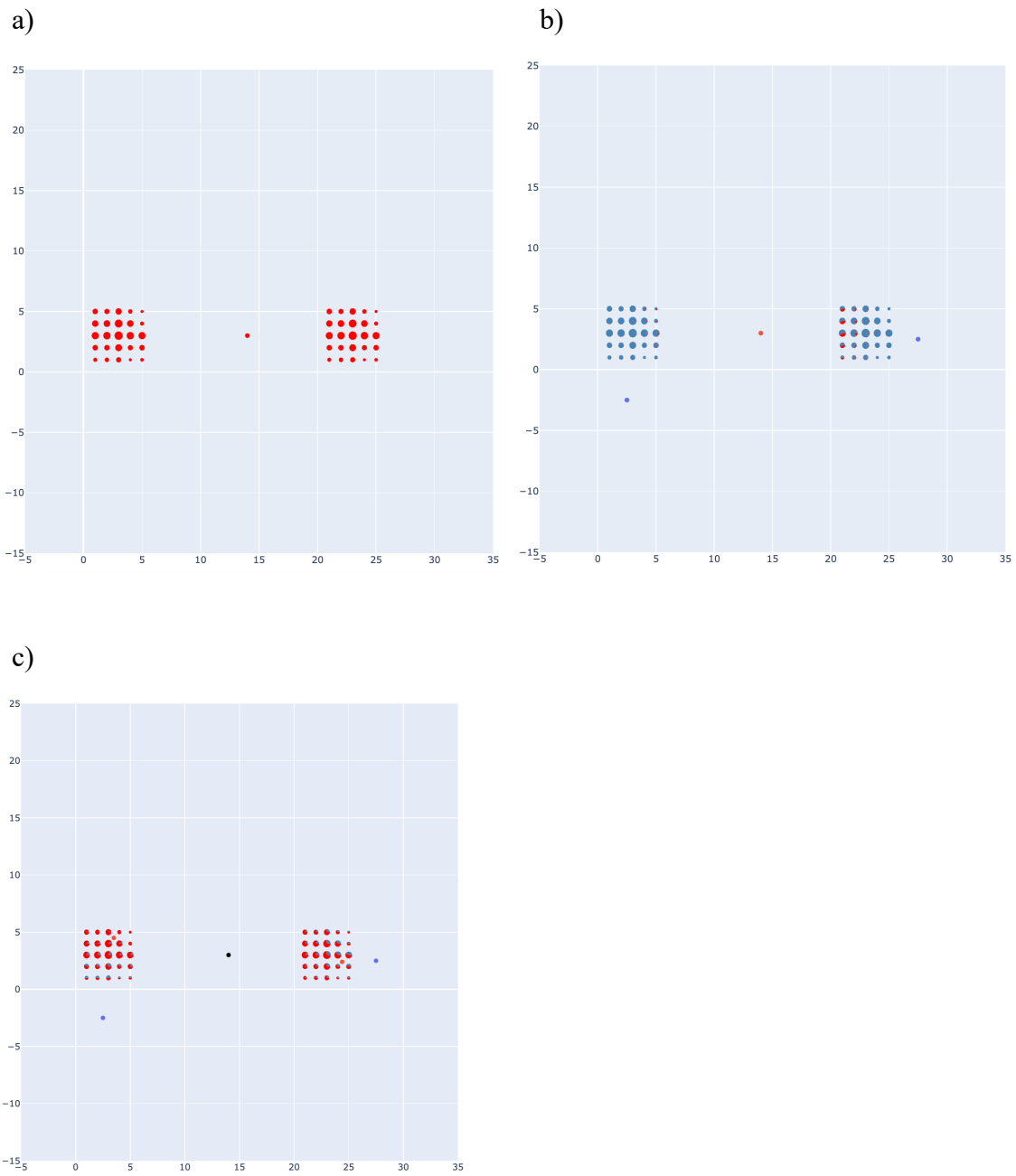


Figure 11. Example of facilities location. a) One target facility b) One target facility with two competitors. c) Two target facilities with two competitors and one closed facility

2.12. Tabu Search algorithm implementation

The main Tabu Search is composed of several parts that are described as the following.

Step 0. Data setup and preparation. Set parameters for the algorithm. Set the facility data I and customer data J. Calculate facility supply and customer demand distributions. Calculate the difference between supply and demand distributions. With IPPP generate a defined number of candidate facilities I^*. Initialize the long- and short-term memory structures.

Step 1. Initial solution. Depending on the algorithm settings generate the initial solution and objective function value with the random algorithm or the greedy heuristic algorithm. Set maximum solution value z_{max} as initial solution value.

Step 2. Running the iteration. Given iteration number q, diversification period div, diversification threshold T_1, max iterations T_2, iterations without improvement q^*, max iterations without improvement T_3: if $q \% div = 0$ or $q^* > T_1$ then go to step 4 else go to step 3. Retrieve the iteration candidate solutions from step 3 and 4. Evaluate the objective function value z_q for candidate solutions. Update the long-term memory with evaluated solutions. If $z_q > z_{max}$ then set z_{max} as z_q, set q^* as 0, else increment q^*. If $q \geq T_2$ or $q \geq T_3$ then terminate the algorithm.

Step 3. Intensification procedure. Select the candidate facilities I^*. If a candidate facility is present in short-term memory, filter out the disallowed facility index from current iteration. Generate the neighboring solutions from the current solution and allowed facility indices. Update the short-term memory with the relevant facilities indices. If a solution is present in long-term memory, filter out the generated solution.

Step 4. Diversification procedure. Select the candidate facilities I^*. Depending on the diversification strategy generate remote solutions or remove a random facility and generate neighbors. If a solution is present in long-term memory, filter out the generated solutions.

The flowchart of the algorithm is represented on the Figure 12.

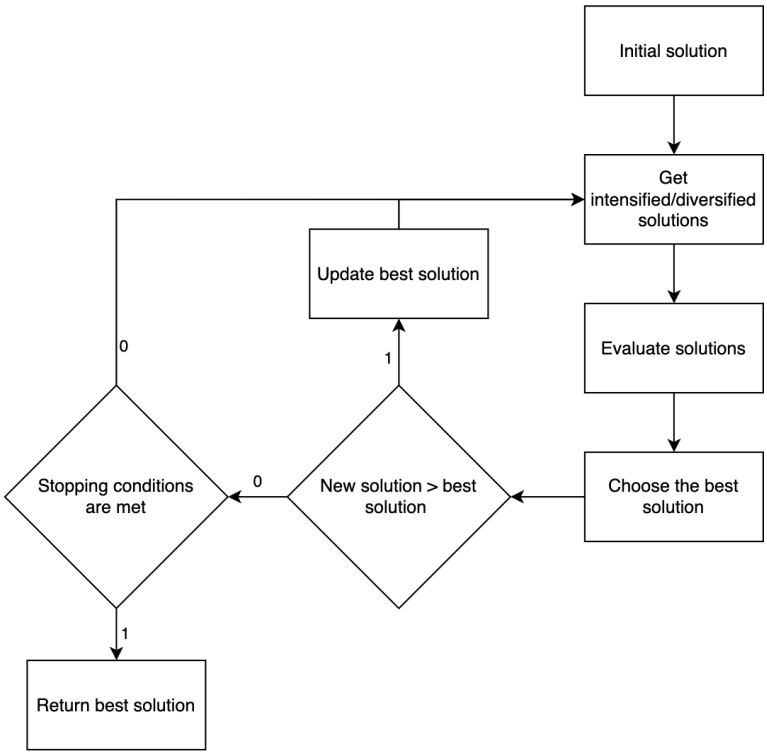


Figure 12. Flowchart of Tabu Search algorithm.

2.13. Multiple Elite Solutions

After multiple experiments it was identified that the algorithm is able to find better solutions after multiple reruns, which indicates that it is occasionally stuck at the local optimum. To escape the local optimums Tabu Search utilizes the short- and long-term memory structures, and diversification procedure which allows it not to return to the same solutions and travel further in the solution space. However, if better optimal solution is located far enough in the solution space, the algorithm is unable to reach it before the stopping rule is applied. Therefore, the experiments have shown that the current algorithm implementation lacks in the global diversification.

A better global diversification can be achieved by running algorithm multiple times with random initial solutions. However, this approach has a drawback of performing excessive computations, and therefore results in the longer solution time.

To overcome the problem of global diversification efficiently, Multiple Elite Solutions (MES) approach was introduced for the algorithm described by Glover, Laguna and Marti [GLM18]. The flowchart of improved Tabu Search algorithm is represented on the Figure 13. The approach works by saving multiple promising solutions which were evaluated with the

highest score while getting the initial solutions with random algorithm. The elite solution set is then used to generate new solutions. To induce the algorithm towards generating more potential solutions from the better performing elite solutions, the elite solution weights were introduced. The highest performers in the elite solution set are assigned with higher elite solution weights, which are used for generating the new candidate solutions.

Having MES to generate the candidate solutions from may result in the conflicts in the short-term memory. The example of a conflict is demonstrated on the Figure 14. The Figure 14a represents a conflicting state with elite solutions S_1 and S_2. The node j_2 gets banned from being selected as a candidate from both solutions S_1 and S_2, even though it was potentially added only from the single one, and therefore complicates the development of one of the solutions. To resolve the conflict, the independent short-term memory structures were introduced, that are associated with each of the elite solutions. The result of the conflict resolution is demonstrated on the Figure 14b. The usage of independent short-term memory structures introduces only a minor increase in the memory usage, since it is implemented as a constant size hash map.

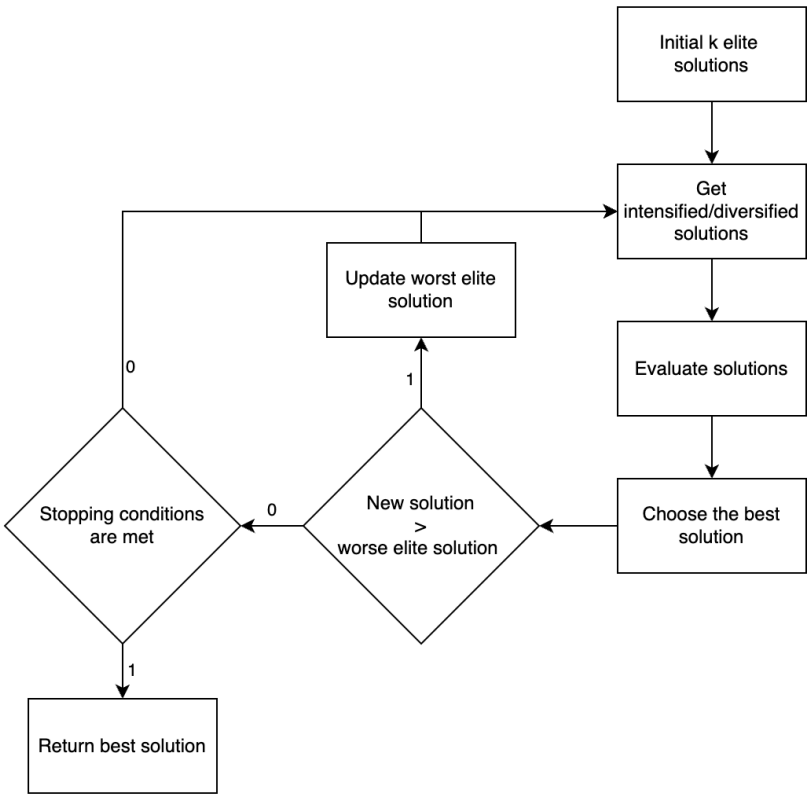


Figure 13. Flowchart of Tabu Search algorithm with Multiple Elite Solutions approach.

a) b)

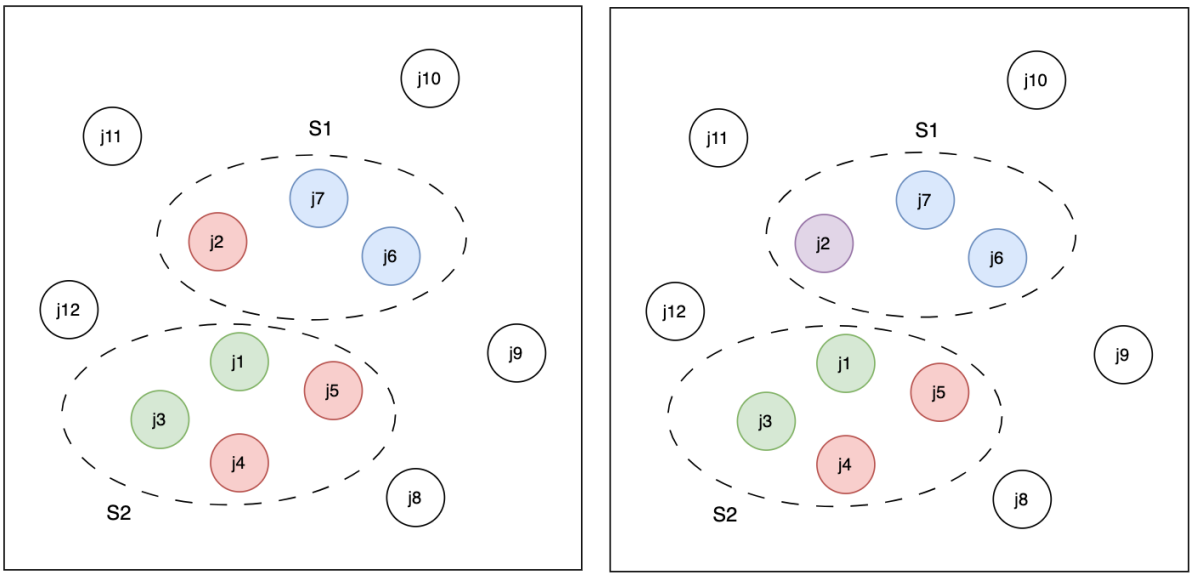


Figure 14. Short-term memory conflict with Multiple Elite Solutions approach. Green and blue nodes represent the facilities chosen for the solution. Red and purple nodes represent banned facilities. a) Conflicting situation with shared short-term-memory. b) Conflict resolution with introduction of independent short-term memory structures.

2.14. Weighted Random Initial Solution strategy

To further enhance the diversification capability of the algorithm, the Weighted Random Initial Solution (WRIS) strategy was introduced. The WRIS strategy combines the evaluation of the individual facilities to from greedy heuristic algorithm to assess the impact on the objective function of individual candidate facilities and the random generation of initial solutions after the candidate facilities are ordered by priority. To generate the weights for the ordered by priority candidate facilities the function of type $w = a * e^{-b*x} + c$ was utilized, where w is the weight of candidate facility for a random generation, x – an index of a facility in the list ordered by facility impact, a, b, c – parameters.

2.15. Testing the algorithm

The presented tabu search implementation was tested on the described data with various parameter combinations. The algorithm was implemented in Python programming language with libraries and extensions enhancing its performance. All of the experiments were executed on Apple M1 CPU with 16 GB of RAM. The number of customer datapoints is 5604 and the number of facilities (i.e., shops) is 125.

2.16. Parameter considerations

Several key parameters that define the algorithm and model behavior were set with the following considerations:

1) BSIM truncation radius, d_T – represents the maximum amount customer is willing to travel to the shop. In the simulation was set to the 1.5 km.
2) Facility operating and removal price, $f_i, \overline{f_i}$ – due to the absence of exact facility operating costs, it was assumed that the operating price is strongly correlated with the revenue from customers. Thus, for each shop $i \in \check{I}$ of a retail chain, the revenue value $\sum_{j \in J} x_{ij}$ was calculated using BSIM. The average shop revenue value was then set as the shop's operating price. This approach implies a few things. The objective function value with opened existing facilities is equal to zero. If the newly opened facility performs better than the average existing facility, the objective function value will be positive. If newly opened facility performs worse than the average existing facility, the objective function value will be negative. Therefore, if the solution was estimated with positive objective function value, it implies the new added facilities can perform better than average existing facilities and are worth investing into. The facility removal price was set as a relation to the facility operating cost and was chosen to be 75% of the operating cost.
3) BSIM variance, σ^2 – was set such that the area of the truncated BSIM relates to the area of not truncated BSIM as 0.95. The assumption is that the portion of customers that will prefer the facility within the BSIM truncation radius is at least 95%, and the choice of the other 5% does not introduce a significant difference to the model and can be omitted.

2.17. Computational experiments

The results of the tabu search algorithm with various parameters are represented in Table 1. Each configuration was tested 50 times, with 100 generated candidate facilities. To estimate the performance of the algorithm, 3 indicators were measured:

- Average time – the average time it takes, until the algorithm is stopped by the stopping condition.
- Average z – the average objective function value.
- Max z – the maximum objective function value.

Table 1. Results of running the algorithm with different parameter combinations.

Initial solution algorithm	Diversification procedure	Average time, s	Average *z*	Max *z*
Random	Remote solutions	67	**10016**	**15934**
Random	Remove facility and get neighbor	51	8018	11629
Greedy heuristic	Remote solutions	42	3471	5612
Greedy heuristic	Remove facility and get neighbor	48	5813	7938

For the MES strategy the optimal number of elite solutions was experimentally determined to be 3. The results of testing are represented in the Table 2. The comparison between 1 elite solution (i.e., no MES strategy) and best performing MES strategy with 3 elite solutions is represented on the Figure 15.

Table 2. Results of testing MES strategy.

N elite solutions	Average time, s	Average *z*	Max *z*
1	67	10016	15934
3	55	**10313**	**18718**
5	40	7367	13453
10	33	6067	11433

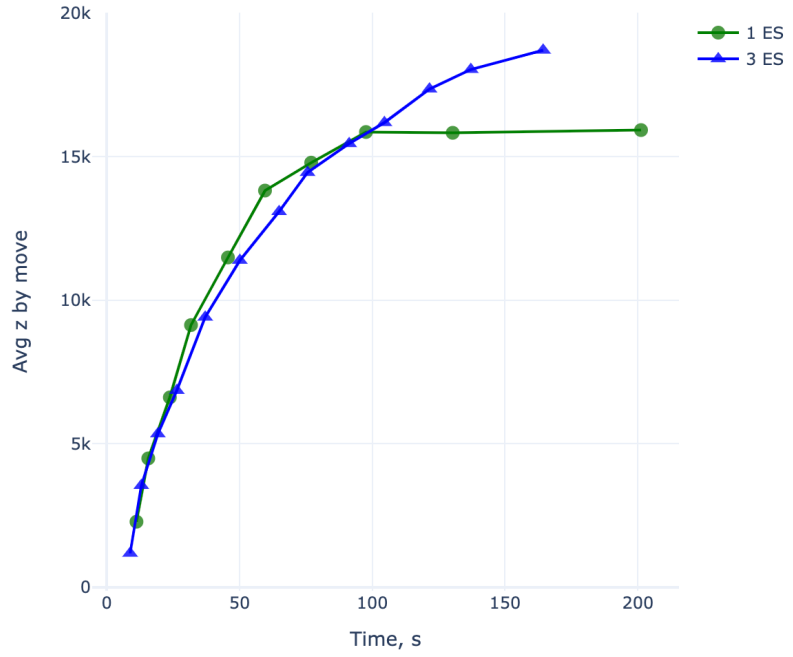


Figure 15. Results of the TS algorithm with MES strategy. Results with 1 elite solution and 3 elite solutions are marked with green and blue respectively.

The results of WRIS strategy testing are represented in the Table 3.

Table 3. Results of testing WRIS strategy.

Strategy	Average time, s	Average *z*	Max *z*
Random	67	10016	15934
Weighted random	110	**16537**	**18651**

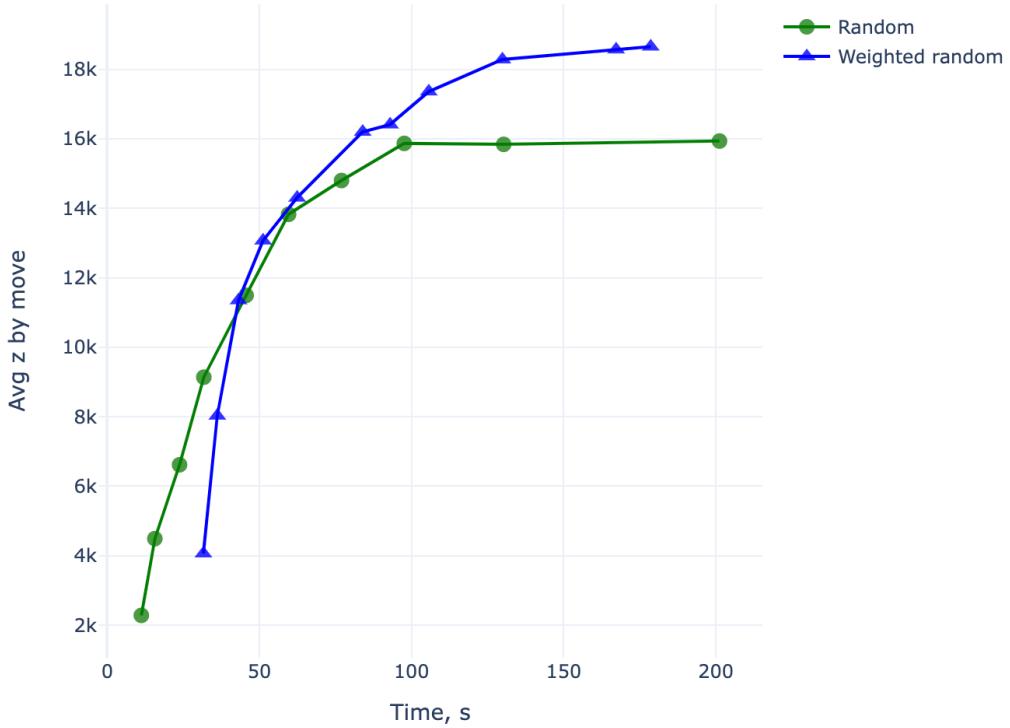


Figure 16. Results of the TS algorithm with WRIS strategy. Results with random initial solution and random weighted elite solution strategies are marked with green and blue respectively.

2.18. Analysis of results

With the MES and WRIS strategies implemented in this research, the algorithm was able to noticeably improve the quality of the solution compared to the baseline implementation of Tabu Search.

The configuration with 1 elite solution has a better opportunity for intensification and therefore during the initial moves showed to be more likely to find better solutions during the initial moves. However, on the longer distances, the diversification capability of MES strategy prevails the single elite solution and was able to find a global solution better by 18% and lower average time by 21%.

The configuration with WRIS strategy has shown to have an average slowdown of 24 seconds for the calculation of first move compared with random strategy, which is caused by calculation of priorities for the candidate facilities in the beginning, which resulted in the increase of average time for the algorithm in 68%. However, the tests have shown that the WRIS strategy was able to find global solution better by 17% and average solution better by 65% compared with initial random strategy.

The MES and WRIS strategies for TS achieved a great performance improvement on the dataset and the problem definition presented in this work. However, the introduction of additional parameters used to model the facility-customer interactions (i.e., proximity of facilities to the roads or means of public transport, customers' socio-demographic characteristics), and change of objective function (i.e., max covering problem) may change the distribution of the optimal solutions in the solution space. In this case, more aggressive diversification strategies or more aggressive intensification strategies may better suit the optimization objective. Therefore, the selection of the optimal strategies for the algorithm implementation is highly dependent on the exact problem that is being solved.

2.19. Optimal location of facilities

Figure 17 describes the optimal location of facility location found with WRIS strategy with highest objective function value of 18651. Algorithm was able to identify 3 new optimal location for the retail stores (marked in red) and 1 facility to close (marked in blue). However, the results presented by algorithm are rather approximate since the crucial data attributes like facility sizes and facility operating prices were generated artificially, due to inability to find the relevant data in the open sources. Also, some crucial aspects of the model affecting the calculation of objective function like socio-demographic factors, proximity to the roads and public transport facilities, etc., were excluded. Thus, the model presented in this work can be further extended with the mentioned factors the produce better results.

a) b)

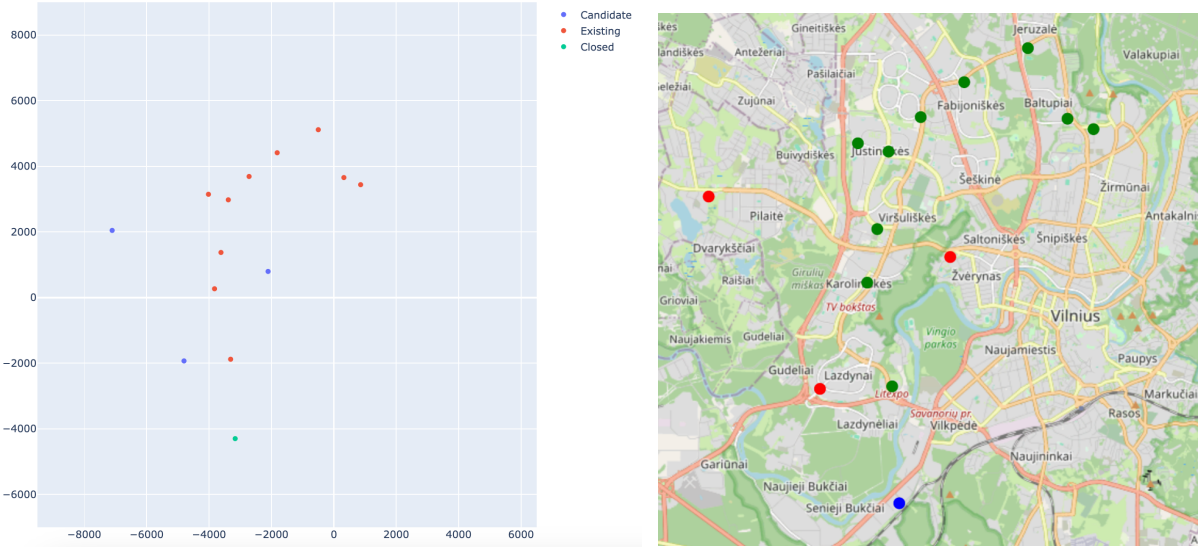


Figure 17. Best solution candidate locations. a) Facilities locations in cartesian coordinates. b) Facilities locations on geographical map.

Conclusions

This work presented a Tabu Search approach for the solution of Competitive Facility Location Problem, encountering the most crucial factors to predict the optimal location for new facilities. The mix of techniques from different studies allowed to build an efficient algorithm that combines the effective use of resources and producing valuable results.

The combination of a quadtree data structure for storing facility locations and Bayesian Spatial Interaction Model for predicting customer choices showed to be a great approach for the calculation of the objective function of the problem (i.e., store revenues) due to its ability to effectively navigate through smaller regions of the broader space when calculating the customer choices allowing to omit the unnecessary checks and perform fewer calculations overall. The use of inhomogeneous Poisson point process in combination with supply-demand density maps for the generation of potential locations for new facilities allowed to drastically improve the quality of facility candidates which resulted in more efficient use of computational resources.

Computational experiments were caried out using different diversification and initial solution strategies. 2 strategies that significantly improve the performance of Tabu Search in the setting of Facility Location Problem were identified – Multiple Elite Solution strategy and Weighted Random Initial Solution, and their performance was measured. The MES strategy was able to find the global solution better by 18% and decreased average running time by 21% compared to the single solution strategy implemented initially. The WRIS strategy increased the running time by 68%, however was able to find global solution better by 17% and average solution better by 65% compared with random initial solution strategy implemented initially.

With the results presented in the Chapter 1 and Chapter 2 the objectives of the work are accomplished and therefore the aim of the work is achieved.

The approaches and techniques presented in this work may provide value for building other frameworks for the solution of optimal placement of other facilities (i.e., cafes, restaurants, schools, kindergartens, etc.).

References

[AK77] AKINC, Umit and KHUMAWALA, BASHEER M. An efficient branch and bound algorithm for the capacitated warehouse location problem. *Management Science*. February 1977. Vol. 23, p. 585–594. DOI https://doi.org/10.1287/mnsc.23.6.585.

[AKK06] AROSTEGUI, Marvin A, KADIPASAOGLU, SUKRAN N and KHUMAWALA, BASHEER M. An empirical comparison of Tabu Search, Simulated Annealing, and Genetic Algorithms for facilities location problems. *International Journal of Production Economics.* October 2006. Vol. 103, p. 742–754. DOI https://doi.org/10.1016/j.ijpe.2005.08.010.

[Bar13] BARBATI, Maria. Models and algorithms for facility location problems with equity considerations. *University of Naples Federico II.* PhD Thesis. April 2013.

[CKM+01] CHARIKAR, Moses, SAMIR KHULLER, MOUNT, David M and NARASIMHAN, Giri. Algorithms for facility location problems with outliers. *Symposium on Discrete Algorithms.* January 2001. P. 642–651. DOI https://doi.org/10.5555/365411.365555.

[Coo64] COOPER, Leon. Heuristic methods for location-allocation problems. *SIAM Review.* January 1964. Vol. 6, p. 37–53. DOI https://doi.org/10.1137/1006005.

[Cso15] CSOKE, Meghan. The facility location problem. *All Student Theses.* Online. July 2015. [Accessed January 2024].

[DD85] DOMSCHKE, Wolfgang and DREXL, Andreas. ADD-heuristics' starting procedures for capacitated plant location models. *European Journal of Operational Research.* July 1985. Vol. 21, p. 47–53. DOI https://doi.org/10.1016/0377-2217(85)90086-4.

[DH02] DREZNER, Zvi and HAMACHER, Horst W. Facility location: applications and methods. *Springer*. 2002.

[EL97] EISELT, H.A and LAPORTE, Gilbert. Sequential location problems. *European Journal of Operational Research.* January 1997. Vol. 96, p. 217–231. DOI https://doi.org/10.1016/s0377-2217(96)00216-0.

[ESV19] ERDOĞAN, Güneş, STYLIANOU, Neophytos and VASILAKIS, Christos. An open source decision support system for facility location analysis. *Decision Support Systems.* October 2019. Vol. 125, p. 113116. DOI https://doi.org/10.1016/j.dss.2019.113116.

[GK99] GUHA, Sudipto and KHULLER, Samir. Greedy strikes back: Improved facility location algorithms. *Journal of Algorithms.* April 1999. Vol. 31, p. 228–248. DOI https://doi.org/10.1006/jagm.1998.0993.

[Glo90] GLOVER, Fred. Tabu search: A tutorial. *Interfaces*. August 1990. Vol. 20, p. 74–94. DOI https://doi.org/10.1287/inte.20.4.74.

[GLM18] GLOVER, Fred W, LAGUNA, Manuel and MARTÍ, Rafael. Principles and Strategies of Tabu Search. *Chapman and Hall/CRC eBooks.* May 2018. P. 361–377. DOI https://doi.org/10.1201/9781351236423-21.

[Goo24] Find Place. Places API. [accessed 2024-01-15] URL: <https://developers.google.com/maps/documentation/places/web-service/search-find-place>

[GTM21] GWALANI, Harsha, TIWARI, Chetan and MIKLER, Armin R. Evaluation of heuristics for the p-median problem: Scale and spatial demand distribution. *Computers, Environment and Urban Systems.* July 2021. Vol. 88, p. 101656. DOI https://doi.org/10.1016/j.compenvurbsys.2021.101656.

[Har19] VLADISLAV HARALAMPIEV. Neural networks for facility location problems. *Annual of Sofia University St Kliment Ohridski Faculty of Mathematics and Informatics.* December 2019. Vol. 106, p. 3–10. DOI https://doi.org/10.60063/gsu.fmi.106.3-10.

[KCK20] KATOCH, Sourabh, CHAUHAN, Sumit Singh and KUMAR, Vijay. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications.* October 2020. Vol. 80. DOI https://doi.org/10.1007/s11042-020-10139-6.

[KDS14] KRATICA, Jozef, DUGOŠIJA, Djordje and SAVIĆ, Aleksandar. A new mixed integer linear programming model for the multi level uncapacitated facility location problem. *Applied Mathematical Modelling.* April 2014. Vol. 38, p. 2118–2129. DOI https://doi.org/10.1016/j.apm.2013.10.012.

[KK62] KULIN, Harold W and KUENNE, Robert E. An efficient algorithm for the numerical solution of the generalized weber problem in spatial economics. *Journal of Regional Science.* December 1962. Vol. 4, p. 21–33. DOI https://doi.org/10.1111/j.1467-9787.1962.tb00902.x.

[Kum13] KUMAR S., RAJESH. Genetic algorithm for solving the uncapacitated facility location problem. *International Journal of Engineering Research & Technology.* 2013. Vol. 2.

[MAR+18] MAHMUDUR, Mohammad, ABU, Md, REZOANA BENTE ARIF and MAHJABIN RAHMAN OISHE. ADBSCAN: Adaptive density-based spatial clustering of applications with noise for identifying clusters with varying densities. *arXiv (Cornell University).* September 2018. DOI https://doi.org/10.1109/ceeict.2018.8628138.

[MS82] MEHREZ, Abraham and STULMAN, Alan. The maximal covering location problem with facility placement on the entire plane. *Journal of Regional Science.* August 1982. Vol. 22, p. 361–365. DOI https://doi.org/10.1111/j.1467-9787.1982.tb00759.x.

[Nor24] Norfa. Development. [accessed 2024-04-27] URL: <https://www.norfa.lt/en/about-us/development>

[PAT23] PERERA, Shanaka, AGLIETTI, Virginia and THEODOROS DAMOULAS. On the competitive facility location problem with a Bayesian spatial interaction model. *Applied statistics*. January 2023. Vol. 72, p. 165–187. DOI https://doi.org/10.1093/jrsssc/qlad003.

[Pla06] PLASTRIA, Frank. Four-point Fermat location problems revisited. New proofs and extensions of old results. *IMA Journal of Management Mathematics.* October 2006. Vol. 17, p. 387–396. DOI https://doi.org/10.1093/imaman/dpl007.

[RG13] RAVI, A. and GARG, D. Nearest Facility Location for Multiple Customers using Voronoi Diagram. *Semantic Scholar.* Online. 16 August 2013.

[SSJ+21] SHARMA, Ashish, SHARMA, Ashish, JALAL, A.S and KANT, Krishna. A two-step clustering method for facility location problem. *International Journal of Advanced Intelligence Paradigms*. 2021. Vol. 18, p. 337. DOI https://doi.org/10.1504/ijaip.2021.113326.

[Sun06] SUN, Minghe. Solving the uncapacitated facility location problem using tabu search. *Computers & Operations Research*. September 2006. Vol. 33, no. 9, p. 2563–2589. DOI https://doi.org/10.1016/j.cor.2005.07.014.