INFORMATICA, 2025, Vol. 36, No. 4, 875–902

DOI: https://doi.org/10.15388/25-INFOR610

Global Optimization Algorithm for the Multi-Weber Problem with Polyhedral Barriers

Mindaugas KEPALAS*, Julius ŽILINSKAS

Vilnius University, Faculty of Mathematics and Informatics,
Institute of Data Science and Digital Technologies, Akademijos st. 4, Vilnius, LT-08412, Lithuania
e-mail: mindaugas.kepalas@mif.stud.vu.lt, julius.zilinskas@mif.vu.lt

Received: August 2025; accepted: November 2025

Abstract. In this paper, we consider the multi-Weber problem with polyhedral barriers. For this problem, a set of obstacles are introduced where travelling or placement is prohibited, which makes the distance metric non-convex and requires constructing a special graph for calculating the distances between pairs of points. For obtaining the global solution of the problem, we build a branch and bound algorithm with pruning criteria based on dividing clients into groups and analysing them separately. We have managed to obtain global solutions to several multi-Weber with polyhedral barriers problem size instances which to our knowledge have not been reported before.

Key words: multi-Weber problem with barriers, clustering problems with centre location constraints, global optimization.

1. Introduction

This paper considers the problem of determining the global solution of a general continuous multi-locations problem, where a set of clients (i.e. points at given coordinates) have to be assigned to a set of facilities, for which the optimal locations have to be determined. In the paper, the distance from a client to a facility is assumed to be arbitrary in a sense that it is not necessarily measured by the standard Euclidean distance between two points; in particular, in our numerical part we consider the multi-Weber with polyhedral barriers problem. For this problem, a set of obstacles are introduced where travelling or facility placement is prohibited, which makes the distance metric non-convex. Also, we assume that the facilities could in addition be restricted to an arbitrary non-convex finite subset of the plane (e.g. this could be a union of a set of segments and triangles).

At optimality for the problems which are studied in this paper, a client always chooses the closest facility, thus the problem can be also seen as a clustering problem: the goal is to find the optimal partition of the clients into groups (clusters), and then the optimal facility can be determined for each cluster independently. Thus, for obtaining the global solution of the multi-locations problem, we employ a branch and bound with an all-possible-partitions

[*] Corresponding author.

enumeration tree and pruning criteria. The pruning criteria roughly uses the following simple idea: if we divide the clients into two groups and find the optimal solution within each group separately, summing the two terms we obtain a lower bound estimate for the partitions within a corresponding branch.

The Weber problem, i.e. the problem of determining the coordinates of the optimal point minimizing the sum of the Euclidean distances to other given points, has a long and rich history, many extensions and publications (Drezner *et al.*, 2002). The extension we concentrate on in this paper was probably firstly introduced by Aneja and Parlar (1994). In this model, obstacles (barriers) in the plane are introduced; it is not allowed to travel through these barriers or to place a solution point inside. In real life, such barriers could be considered to represent lakes or buildings on a geographical map, or some prohibited places in printed circuit boards (Hamacher and Nickel, 1995).

The Weber location with barriers problem has attracted the attention of many scientists. However, most of the work concentrates on solving the single location problem, see e.g. Aneja and Parlar (1994), Klamroth (2002), Bischoff and Klamroth (2007), Oğuz *et al.* (2016). The first reason for this is the following motivation: having a "good" algorithm for the single location problem, we can implement a location-allocation algorithm in the style of Cooper (1964). For example, such an algorithm for multilocations with polyhedral barriers problem is considered by Bischoff *et al.* (2009). The second reason why the case of multilocations has not received much scientific effort might be related to the combinatorial complexity of this problem.

The only non-theoretical work dedicated to the multi-Weber with barriers problem we have found in the scientific literature is the thesis of Krau (1997). In his work, the author extends the column generation principle of Gilmore and Gomory (1961) to the problem. The column generation approach seems to provide a benchmark solution for several clustering-like problems such as the minimum sum of squares clustering (Aloise *et al.*, 2012) or the (barrierless) multi-Weber problem (du Merle *et al.*, 1999). Using this method, Krau (1997) solves the problem presented by Aneja and Parlar (1994) which consists of 18 points and 12 barriers, and also an extended version of this problem with 30 points, for cluster sizes $K = 2, \dots, 7$.

As the results indicate, the cases with small K seem to be the hardest for a column generation based algorithm (Krau, 1997). However, with our approach we have solved the problem instances of $60/2$, $45/3$, $40/4$ and $35/5$ (N/K means "N points into K clusters") to global optimality (within ϵ-accuracy with $\epsilon = 1\%$). To our knowledge, this is the first time the solution of instances with such a size to global optimality is reported. Moreover, we think our approach is very general. We believe the ideas could be also extended to other variants of multi-Weber problems.

Our paper is structured as follows. In Section 2, we formulate the general model of the problem we seek to solve in this paper. In Section 3, we describe the building blocks of our approach and give some mathematical proofs of intuitively rather obvious facts. Using these blocks, in Section 4 we present the sketch of the enumeration algorithm. Finally, in Section 5, we clarify algorithm details for the multi-Weber with polyhedral barriers problem and give an analysis of the performance of the algorithm.

2. General Multi-Locations (Clustering) Problem

Let us formulate the following general multi-locations (clustering) problem:

$$\min_{(\mathcal{C}_1,,\mu_1),\dots,(\mathcal{C}_K,\mu_K)} \sum_{k=1}^{K} \sum_{i\in\mathcal{C}_k} d(p_i,\mu_k) \quad \text{s.t.} \quad \mu_k \in \mathcal{R}^{\cup}, \quad k=1,\dots,K. \tag{1}$$

Problem input:

- N "points", "clients" $p_1, p_2, \dots, p_N$ in the problem space $\mathcal{S}$: these are the elements we want to group into clusters. In our numerical part for the multi-Weber with barriers problem, these are points in $\mathbb{R}^2$, but the theoretical ideas could also be applied to points in an arbitrary $\mathbb{R}^D$ with dimension $D > 2$, or even other elements from some more complex space $\mathcal{S}$, e.g. some function space.
- K – the number of clusters.
- $\mathcal{R}^{\cup}$ – the constrained set for cluster centres. This set is a union of M "containers", "cells" or "region-units" $\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_M$: $\mathcal{R}^{\cup} = \bigcup_{m=1}^{M} \mathcal{R}_m$. See further comments on $\mathcal{R}^{\cup}$ in Section 2.1.
- Distance metric (function) $d : \mathcal{S}\times\mathcal{S} \mapsto \mathbb{R}^+$ to measure the distance between a "client" p_i and a "facility" μ_k. In this paper, we concentrate on a multi-location with barriers problem, where obstacles (barriers) in the plane are introduced. Any allowed path between points p_i and μ_k must not cross any of the barriers.

Problem variables:

- Cluster centres ("facilities") μ_k, constrained to be placed within the set $\mathcal{R}^{\cup}$:

$$\mu_k \in \mathcal{R}^{\cup}, \quad k=1,\dots,K.$$

- Corresponding clusters[1] $\mathcal{C}_k$, $k=1,\dots,K$. These are the sets of "clients" assigned to "facility" at μ_k. Somewhat, such variable is redundant, because in our problem formulation each "client" is assigned to the closest "facility", thus the clusters are determined given μ_k, $k=1,\dots,K$. However, this notion will be useful when considering the k-means-type (location-allocation) algorithm discussed further and also for the branch and bound enumeration algorithm presented in the paper.

Our goal is to find the global solution to problem (1). Using notation

$$\mathbb{P}_{n_1}^{n_2} = \{p_{n_1},\dots,p_{n_2}\}, \quad \text{for } n_1 \leqslant n_2,$$

we label problem (1) with $[\mathbb{P}_1^N/K]|[\mathcal{R}^{\cup}]$ for short.[2] Correspondingly, we also label the globally optimal (minimal) value of (1) with $\mathbf{Loss}^{\mathrm{opt}}([\mathbb{P}_1^N/K]|[\mathcal{R}^{\cup}])$.

[1] A partition of the set $\mathbb{N}_1^N = \{1,2,\dots,N\}$: $\bigcup_{k=1}^{K}\mathcal{C}_k = \mathbb{N}_1^N$, $\mathcal{C}_k \cap \mathcal{C}_l = \emptyset$, $k \neq l$.

[2] In words, "cluster the point set $\mathbb{P}_1^N$ into (/) K clusters given a condition (|) that cluster centres are restricted to be within $\mathcal{R}^{\cup}$".

2.1. *Constrained Set* $\mathcal{R}^{\cup}$

In the most general case, we assume that the constrained set $\mathcal{R}^{\cup}$ is a (finite) union of arbitrary closed and bounded subsets ("region-units") of the problem space $\mathcal{S}$:

$$\mathcal{R}^{\cup} = \bigcup_{m=1}^{M} \mathcal{R}_m, \qquad \mathcal{R}_m \subset \mathcal{S}, \quad m = 1, \dots, M.$$

The bounded property is not restrictive for the problem, since the set $\{p_1, \dots, p_N\}$, as a finite set of points, is always bounded, and it can be shown that this implies that all optimal centres μ_k for the problem (1) always lie within some bounded subset of $\mathcal{S}$. In addition, the closed property for each region-unit $\mathcal{R}_m$ ensures that for any $p \in \mathcal{S}$, we can always find the nearest and the furthest points $\mu_m^{\inf} \in \mathcal{R}_m$ and $\mu_m^{\sup} \in \mathcal{R}_m$, such that

$$\begin{aligned} d\big(p, \mu_m^{\inf}\big) &= \inf_{\mu \in \mathcal{R}_m} d(p, \mu) \geqslant 0, \\ d\big(p, \mu_m^{\sup}\big) &= \sup_{\mu \in \mathcal{R}_m} d(p, \mu) < \infty. \end{aligned}$$

As a less general example (for the multi-locations with polyhedral barriers problem we study in the numerical part of this paper), we define $\mathcal{R}^{\cup}$ to be a union of (arbitrary) polygons. We note that any such set $\mathcal{R}^{\cup}$ of (possibly, overlapping) polygons can be decomposed into a union of (non-overlapping) triangles by some triangulation method,[3] where by "non-overlapping" we mean that all the triangles will "touch" only along the sides. Such decomposition is also possible if some of the region-units $\mathcal{R}_m,\ m = 1, \dots, M$ are segments; then $\mathcal{R}^{\cup}$ can be decomposed into a union of "non-overlapping" segments and triangles.

Going to higher dimensions, e.g. for $\mathcal{S} = \mathbb{R}^3$, $\mathcal{R}^{\cup}$ could be decomposed into a union of segments, triangles and tetrahedrons (Paulavičius and Žilinskas, 2013), etc., we can continue to higher dimensional Euclidean spaces. This "decomposition into simplices" is a separate research topic and we do not consider this question in our paper. Therefore, in our numerical part for the multi-Weber with barriers problem we assume that $\mathcal{R}^{\cup} = \bigcup_{m=1}^{M} \mathcal{R}_m$ is provided as a union of triangles $\mathcal{R}_m$, $m = 1, \dots, M$.

2.2. *Alternative Problem Formulation and an Iterative Algorithm to Obtain a Local Solution*

For an arbitrary cluster $\mathcal{C}_k \subseteq \{1, 2, \dots, N\}$ and an arbitrary centre $\mu_k \in \mathcal{S}$, define the related loss for this cluster-centre pair $(\mathcal{C}_k, \mu_k)$:

$$\mathbf{Loss}(\mathcal{C}_k, \mu_k) = \sum_{i \in \mathcal{C}_k} d(p_i, \mu_k).$$

[3] See for example the implementation in [R] sfdtc package (Sumner, 2025).

With this definition, the loss in (1) can be written this way:

$$\sum_{k=1}^{K}\sum_{i\in\mathcal{C}_k} d(p_i,\mu_k)=\sum_{k=1}^{K}\mathbf{Loss}(\mathcal{C}_k,\mu_k). \tag{2}$$

As can be seen from the above equation, the optimal centres given a predefined partition of the clients into clusters can be found independently. Therefore, (1) can be rewritten this way:

$$\min_{\mathcal{C}_1,\dots,\mathcal{C}_K}\sum_{k=1}^{K}\min_{\mu_k\in\mathcal{R}^{\cup}}\mathbf{Loss}(\mathcal{C}_k,\mu_k). \tag{3}$$

Formulation (3) can be noticed to lead us to the classical location-allocation (Cooper, 1964) (k-means-type) algorithm:

Step 1. Given the (fixed) clusters $\mathcal{C}_1,\dots,\mathcal{C}_K$, determine (update) optimal cluster centres:

$$\mu_k=\arg\min_{\mu\in\mathcal{R}^{\cup}}\mathbf{Loss}(\mathcal{C}_k,\mu),\quad k=1,\dots,K.$$

Step 2. For the (fixed) cluster centrees $\mu_1,\dots,\mu_K$, determine (update) optimal clusters:

$$\mathcal{C}_k=\big\{i: d(p_i,\mu_k)<d(p_i,\mu_l),\ \forall l\neq k\big\},\quad k=1,\dots,K.$$

Repeating the procedure above, the loss (3) decreases with every step, and at the termination we are guaranteed to arrive at a local solution of problem (1), i.e. at a solution which cannot be improved by switching cluster memberships or changing cluster centres. For the multi-Weber with polyhedral barriers problem, see, for example, such a location-allocation algorithm by Bischoff *et al.* (2009).

3. Building Blocks of the Algorithm

3.1. *Enumeration Tree*

To find the global solution of problem (1), we could theoretically enumerate all the possible partitions of the index set $\mathbb{N}_1^N=\{1,\dots,N\}$ into K clusters $\mathcal{C}_1,\dots,\mathcal{C}_K$, find the optimal centres for each possible clustering, and pick the best solution.

Such an approach would be brutal and would allow to solve only very small problems, as there are $S_2(N,K)$ of such partitions into clusters.[4] Nevertheless, we have shown in our

[4] Here, $S_2(\bullet,\bullet)$ is Stirling's number of the 2nd kind. This number satisfies the recursive relation

$$S_2(N+1,K)=KS_2(N,K)+S_2(N,K-1), \tag{4}$$

and from this formula one can calculate that, e.g. $S_2(18,4)=2,798,806,985$ – already a very big number.

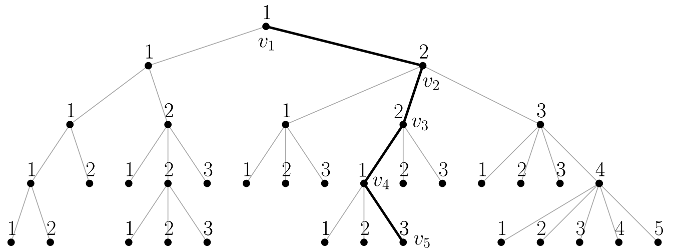


Fig. 1. Clustering enumeration tree $\mathcal{T}(N, K)$ for the algorithm.

recent article (Kepalas and Žilinskas, 2024) on a related problem that having a good pruning criteria, problems of size 50/7 (i.e. $N = 50$ points into $K = 7$ clusters) become possible to solve using this approach. We will discuss the pruning criteria in Section 3.2. For now, consider the tree $\mathcal{T}(N, K)$ in Fig. 1, which visualizes such an enumeration procedure.

In the tree, the n-th level (distance from the root v_1) of the node corresponds to the n-th element in the set $\{p_1, \dots, p_N\}$, and the label λ of the corresponding node represents the cluster (index) to which this element is assigned along the path. As an example, the highlighted path $v_1v_2v_3v_4v_5$ in Fig. 1 results in a sequence

$$\lambda(v_1) = 1,\ \lambda(v_2) = 2,\ \lambda(v_3) = 2,\ \lambda(v_4) = 1,\ \lambda(v_5) = 3,$$

which corresponds to clustering

$$\mathcal{C}_1 = \{1, 4\},\ \mathcal{C}_2 = \{2, 3\},\ \mathcal{C}_3 = \{5\}.$$

In general, the enumeration procedure is as follows:

- Start with the first element in the first cluster $\mathcal{C}_1 = \{1\}$ and the remaining clusters empty: $\mathcal{C}_k = \emptyset$, $k = 2, \dots, K$.
- When assigning the next element to a cluster (i.e. when going to the next level), you are only allowed to put the corresponding element
  1) into a non-empty cluster or
  2) into an empty cluster with the lowest index.
     - By two rules, the 2nd element can be assigned to either $\mathcal{C}_1$ or $\mathcal{C}_2$.
     - For the 3rd element, $\mathcal{C}_3$ might not be available (in case $\mathcal{C}_1 = \{1, 2\}$, $\mathcal{C}_2 = \emptyset$; see the nodes at the 3rd level of $\mathcal{T}(N, K)$).
     - For the 4th element, see the nodes at the 4th level of $\mathcal{T}(N, K)$ and etc.
- Terminating at the leaf-node $v_N \in \mathcal{T}(N, K)$, we obtain clustering

  $$\mathcal{C}_1 = \{1, \dots\},\ \mathcal{C}_2 = \{\dots\},\ \dots,\ \mathcal{C}_K = \{\dots\}$$

  by tracing the labels in the path from the root-node to the leaf-node.

Furthermore, from (4) one can see that adding any additional element to the clustering problem (and keeping the number of clusters fixed at K) increases the number of partitions more than K times, i.e. this number grows exponentially.

It can be shown by simple arguments that this way the nodes v_n at the level n enumerate all the possible partitions of the set $\{1, 2, \dots, n\}$ into no more than K groups. We notate such partition associated with any vertex $v_n \in \mathcal{T}(N, K)$ with $\mathcal{P}(v_n)$, i.e. $\mathcal{P}(v_n) = \{\mathcal{C}_1(v_n), \dots, \mathcal{C}_K(v_n)\}$.[5] When $n < N$, to emphasize that not all N points are yet assigned to some cluster, we sometimes call such a partition "partial" and add the symbol ∂ to our notation, i.e. $\partial\mathcal{P}(v_n) = \mathcal{P}(v_n)$ for $n < N$. Also, we notate the set of all the vertices at the level N with $\mathcal{L}(N, K)$ (this is the leaf-node set of the tree).

3.2. *Pruning Criteria*

The pruning criterion is based on the following simple observation.

Theorem 1 (*Lower bound by division into two groups*). Consider a division of the point set $\mathbb{P}_1^N = \{p_1, \dots, p_N\}$ into two groups $\widetilde{\mathbb{P}}_1, \widetilde{\mathbb{P}}_2$ s.t. $\widetilde{\mathbb{P}}_1 \cap \widetilde{\mathbb{P}}_2 = \emptyset$ and $\widetilde{\mathbb{P}}_1 \cup \widetilde{\mathbb{P}}_2 = \mathbb{P}_1^N$. Then,

$$\mathbf{Loss}^{\mathrm{opt}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^{\cup}]\big) \geqslant \mathbf{Loss}^{\mathrm{opt}}\big([\widetilde{\mathbb{P}}_1/K]\big|[\mathcal{R}^{\cup}]\big) + \mathbf{Loss}^{\mathrm{opt}}\big([\widetilde{\mathbb{P}}_2/K]\big|[\mathcal{R}^{\cup}]\big). \quad (5)$$

Proof. Define index sets $\widetilde{\mathbb{N}}_1 = \{i : p_i \in \widetilde{\mathbb{P}}_1\}$, $\widetilde{\mathbb{N}}_2 = \{i : p_i \in \widetilde{\mathbb{P}}_2\}$. Suppose the globally optimal value for problem $[\mathbb{P}_1^N/K]|[\mathcal{R}^{\cup}]$ is attained for clustering $\mathcal{C}_1, \dots, \mathcal{C}_K$ with centres at locations $\mu_1, \dots, \mu_K$. Also label

$$\mathcal{C}_k^{\dagger} = \mathcal{C}_k \cap \widetilde{\mathbb{N}}_1, \qquad \mathcal{C}_k^{\ddagger} = \mathcal{C}_k \cap \widetilde{\mathbb{N}}_2, \quad k = 1, \dots, K.$$

We have

$$\begin{aligned}
\mathbf{Loss}^{\mathrm{opt}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^{\cup}]\big) &= \sum_{k=1}^{K}\sum_{i \in \mathcal{C}_k} d(p_i, \mu_k) \\
&= \sum_{k=1}^{K}\sum_{i \in \mathcal{C}_k^{\dagger}} d(p_i, \mu_k) + \sum_{k=1}^{K}\sum_{i \in \mathcal{C}_k^{\ddagger}} d(p_i, \mu_k) \\
&\geqslant \sum_{k=1}^{K}\Big[\min_{\mu \in \mathcal{R}^{\cup}} \sum_{i \in \mathcal{C}_k^{\dagger}} d(p_i, \mu)\Big] + \sum_{k=1}^{K}\Big[\min_{\mu \in \mathcal{R}^{\cup}} \sum_{i \in \mathcal{C}_k^{\ddagger}} d(p_i, \mu)\Big] \\
&\geqslant \mathbf{Loss}^{\mathrm{opt}}\big([\widetilde{\mathbb{P}}_1/K]\big|[\mathcal{R}^{\cup}]\big) + \mathbf{Loss}^{\mathrm{opt}}\big([\widetilde{\mathbb{P}}_2/K]\big|[\mathcal{R}^{\cup}]\big).
\end{aligned}$$

The last inequality is justified by the reasoning that if we "relax" the partition of the set $\widetilde{\mathbb{N}}_1$ from being "fixed" to $\mathcal{C}_1^{\dagger}, \dots, \mathcal{C}_K^{\dagger}$, we can only obtain an improvement. (Same applies to $\widetilde{\mathbb{N}}_2$ and $\mathcal{C}_1^{\ddagger}, \dots, \mathcal{C}_K^{\ddagger}$.) □

Corollary 1. Take some $n < N$. Then,

$$\mathbf{Loss}^{\mathrm{opt}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^{\cup}]\big) \geqslant \mathbf{Loss}^{\mathrm{opt}}\big([\mathbb{P}_1^n/K]\big|[\mathcal{R}^{\cup}]\big) + \mathbf{Loss}^{\mathrm{opt}}\big([\mathbb{P}_{n+1}^N/K]\big|[\mathcal{R}^{\cup}]\big).$$

[5] $\mathcal{C}_k(v_n) = \mathcal{C}_k(v_1 v_2 \dots v_n) = \{i \in \{1, 2, \dots, n\} : \lambda(v_i) = k\}$.

Label

$$\mathbf{Loss}^{\mathrm{opt}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}^{\cup}\big) = \min_{\mu \in \mathcal{R}^{\cup}} \sum_{i \in \mathcal{C}_k} d(p_i, \mu).$$

Consider some $v_N \in \mathcal{L}(N, K)$ and the corresponding partition

$$\mathcal{P}_N = \big\{\mathcal{C}_1(v_N), \dots, \mathcal{C}_K(v_N)\big\}.$$

Define

$$\mathbf{Loss}^{\mathrm{opt}}\big(\big[\mathbb{P}_1^N/K\big]\big|\big[\mathcal{R}^{\cup}, \mathcal{P}_N\big]\big) = \sum_{k=1}^{K} \mathbf{Loss}^{\mathrm{opt}}\big(\big\{p_i, i \in \mathcal{C}_k(v_N)\big\}\big|\mathcal{R}^{\cup}\big). \tag{6}$$

Now suppose we are at some vertex $v_n \in \mathcal{T}(N, K)$ at the level $n < N$, i.e., we are in the path $v_1 v_2 \dots v_n$ which corresponds to a partial clustering $\partial\mathcal{C}_1(v_n), \dots, \partial\mathcal{C}_K(v_n)$. Consider the following problem, where v_N is a leaf-node of the tree $\mathcal{T}(N, K)$, i.e. $v_N \in \mathcal{L}(N, K)$:

$$\begin{aligned} &\min_{v_N \in \mathcal{L}(N,K)} \left(\sum_{k=1}^{K} \mathbf{Loss}^{\mathrm{opt}}\big(\big\{p_i, i \in \mathcal{C}_k(v_N)\big\}\big|\mathcal{R}^{\cup}\big) \right) \\ &\text{s.t.} \quad \text{vertex } v_N \text{ is reached through } v_n. \end{aligned} \tag{7}$$

Looking at the enumeration tree in Fig. 1, in problem (7) we only consider clusterings which correspond to the branch starting at the vertex v_n.

Consider the unique path $v_1 v_2 \dots v_n$ from v_1 to v_n and label the corresponding partial partition with $\partial\mathcal{P}_n = \partial\mathcal{P}(v_n)$. For short, we notate problem (7) with $[\mathbb{P}_1^N/K]|[\mathcal{R}^{\cup}, \partial\mathcal{P}_n]$, and the optimal solution with $\mathbf{Loss}^{\mathrm{opt}}([\mathbb{P}_1^N/K]|[\mathcal{R}^{\cup}, \partial\mathcal{P}_n])$.

We now state the following theorem.

Theorem 2 (*Optimal solution pruning criteria*).

$$\begin{aligned} \mathbf{Loss}^{\mathrm{opt}}\big(\big[\mathbb{P}_1^N/K\big]\big|\big[\mathcal{R}^{\cup}, \partial\mathcal{P}_n\big]\big) \geqslant{} & \mathbf{Loss}^{\mathrm{opt}}\big(\big[\mathbb{P}_1^n/K\big]\big|\big[\mathcal{R}^{\cup}, \partial\mathcal{P}_n\big]\big) \\ & + \mathbf{Loss}^{\mathrm{opt}}\big(\big[\mathbb{P}_{n+1}^N/K\big]\big|\big[\mathcal{R}^{\cup}\big]\big). \end{aligned} \tag{8}$$

Proof. The proof is rather analogous to Theorem 1. Suppose the optimal value for problem (7) is attained for clustering $\mathcal{C}_1, \dots, \mathcal{C}_K$ with centres at locations $\mu_1, \dots, \mu_K$. Label

$$\mathcal{C}_k^{\dagger} = \partial\mathcal{C}(v_n) = \mathcal{C}_k \cap \mathbb{N}_1^n, \qquad \mathcal{C}_k^{\ddagger} = \mathcal{C}_k \cap \mathbb{N}_{n+1}^N, \quad k = 1, \dots, K.$$

We have

$$\begin{aligned}\mathbf{Loss}^{\mathrm{opt}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^\cup,\partial\mathcal{P}_n]\big) &= \sum_{k=1}^{K}\sum_{i\in\mathcal{C}_k} d(p_i,\mu_k)\\ &= \sum_{k=1}^{K}\sum_{i\in\mathcal{C}_k^\dagger} d(p_i,\mu_k)+\sum_{k=1}^{K}\sum_{i\in\mathcal{C}_k^\ddagger} d(p_i,\mu_k).\end{aligned}$$

Now if we “relax” centre locations for the first term on the right-hand-side by allowing optimization, and, similarly, centre locations and partition for the second term, we can only obtain an improvement. These “relaxations” result precisely in the definitions of $\mathbf{Loss}^{\mathrm{opt}}([\mathbb{P}_1^n/K]|[\mathcal{R}^\cup,\partial\mathcal{P}_n])$ and $\mathbf{Loss}^{\mathrm{opt}}([\mathbb{P}_{n+1}^N/K]|[\mathcal{R}^\cup])$. □

An idea for pruning criteria. Suppose we have an upper bound $L^{\mathcal{UB}}$ for the problem $[\mathbb{P}_1^N/K]|[\mathcal{R}^\cup]$, i.e. suppose we know that

$$\mathbf{Loss}^{\mathrm{opt}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^\cup]\big)\leqslant L^{\mathcal{UB}}.$$

Such a bound is given by the optimal solution of any (fixed) particular clustering $\mathcal{C}_1,\mathcal{C}_2,\dots,\mathcal{C}_K$. A “good quality” lower bound can be obtained by a location-allocation algorithm as discussed in Section 2.2. Now, if we know that

$$\mathbf{Loss}^{\mathrm{opt}}\big([\mathbb{P}_1^n/K]\big|[\mathcal{R}^\cup,\partial\mathcal{P}_n]\big)+\mathbf{Loss}^{\mathrm{opt}}\big([\mathbb{P}_{n+1}^N/K]\big|[\mathcal{R}^\cup]\big)\geqslant L^{\mathcal{UB}}.$$

Theorem 2 implies that $\mathbf{Loss}^{\mathrm{opt}}([\mathbb{P}_1^N/K]|[\mathcal{R}^\cup,\partial\mathcal{P}_n])\geqslant L^{\mathcal{UB}}$ and the branch corresponding to $\partial\mathcal{P}_n$ can be cut, i.e. there is no need to explore it, as we will not find a better solution there.

3.3. *Bounds for Cluster Loss*

Take some arbitrary cluster $\mathcal{C}_k\subseteq\{1,\dots,N\}$ and consider the problem

$$\mathbf{Loss}^{\mathrm{opt}}\big(\{p_i,i\in\mathcal{C}_k\}\big|\mathcal{R}^\cup\big)=\min_{\mu\in\mathcal{R}^\cup}\sum_{i\in\mathcal{C}_k} d(p_i,\mu), \tag{9}$$

i.e. the problem of finding the optimal centre for a given set of points. For a distance metric d induced by a set of barriers, this is sufficiently complicated problem in itself (Klamroth, 2002). The book suggests solving (9) by formulating a mixed integer programming problem. Other suggested methods in the literature use heuristic algorithms (Aneja and Parlar, 1994; Bischoff and Klamroth, 2007).

Instead of trying to solve (9) directly, we will concentrate on finding a lower bound for the problem, i.e. we will define $\mathbf{Loss}^{\mathcal{LB}}(\{p_i,i\in\mathcal{C}_k\}|\mathcal{R}^\cup)$ (here, the superscript $\mathcal{LB}$ stands for the “lower bound”) which will satisfy

$$\mathbf{Loss}^{\mathcal{LB}}\big(\{p_i,i\in\mathcal{C}_k\}\big|\mathcal{R}^\cup\big)\leqslant\mathbf{Loss}^{\mathrm{opt}}\big(\{p_i,i\in\mathcal{C}_k\}\big|\mathcal{R}^\cup\big). \tag{10}$$

The main thing we seek for this definition is that given the bounds for some cluster $\mathcal{C}_k$, it would be "easy" to find the bounds for cluster $\mathcal{C}_k \cup \{j\}$, e.g. if we "increment" the cluster with "client" p_j.

3.3.1. *The Nearest and the Furthest Point in a Region-Unit*

Remember that we have defined $\mathcal{R}^{\cup}$ as a union of a finite number of "region-units":

$$\mathcal{R}^{\cup} = \bigcup_{m=1}^{M} \mathcal{R}_m.$$

Consider an arbitrary "client" p_n and an arbitrary "region-unit" $\mathcal{R}_m \in \{\mathcal{R}_1, \dots, \mathcal{R}_M\}$. Define the nearest and the furthest points $\mu_{nm}^{\inf}$ and $\mu_{nm}^{\sup}$ for p_n in cell $\mathcal{R}_m$:

$$\mu_{nm}^{\inf} = \arg\min_{\mu \in \mathcal{R}_m} d(p_n, \mu), \tag{11a}$$

$$\mu_{nm}^{\sup} = \arg\max_{\mu \in \mathcal{R}_m} d(p_n, \mu). \tag{11b}$$

The existence and finiteness of these points is ensured by the closed-and-bounded property of $\mathcal{R}_m$ as discussed in Section 2.1. Also, we notate the minimum and maximum distances attained at these points:

$$d_{nm}^{\inf} = \inf_{\mu \in \mathcal{R}_m} d(p_n, \mu) = d\big(p_n, \mu_{nm}^{\inf}\big), \tag{12a}$$

$$d_{nm}^{\sup} = \sup_{\mu \in \mathcal{R}_m} d(p_n, \mu) = d\big(p_n, \mu_{nm}^{\sup}\big). \tag{12b}$$

3.3.2. *The Lower and Upper Bounds for the Loss of a Cluster*

Take some arbitrary cluster $\mathcal{C}_k \subseteq \{1, 2, \dots, N\}$ and some arbitrary region-unit $\mathcal{R}_m \in \{\mathcal{R}_1, \dots, \mathcal{R}_M\}$. Consider the following problem:

$$\mathbf{Loss}^{\mathrm{opt}}\big(\{p_i, i \in \mathcal{C}_k\} \big| \mathcal{R}_m\big) = \min_{\mu \in \mathcal{R}_m} \sum_{i \in \mathcal{C}_k} d(p_i, \mu), \tag{13}$$

the problem of finding the optimal "facility" location for cluster $\mathcal{C}_k$ when the centre is restricted to region-unit $\mathcal{R}_m$.

Theorem 3. The following bounds are valid:

$$\sum_{n \in \mathcal{C}_k} d_{nm}^{\inf} \leqslant \mathbf{Loss}^{\mathrm{opt}}\big(\{p_i, i \in \mathcal{C}_k\} \big| \mathcal{R}_m\big) \leqslant \sum_{n \in \mathcal{C}_k} d_{nm}^{\sup}. \tag{14}$$

Proof. Define $\mu_k^{\mathrm{opt}}[m]$ – the optimal centre position for cluster $\mathcal{C}_k$ in region-unit $\mathcal{R}_m$, i.e. the solution of (13), which exists and is finite since $\mathcal{R}_m$ is closed and bounded. We have

$$\begin{aligned}\mathbf{Loss}^{\mathrm{opt}}\big(\{p_n, n \in \mathcal{C}_k\}\big|\mathcal{R}_m\big) &= \min_{\mu \in \mathcal{R}_m} \sum_{n \in \mathcal{C}_k} d(p_n, \mu) = \sum_{n \in \mathcal{C}_k} d\big(p_n, \mu_k^{\mathrm{opt}}[m]\big) \\ &\geqslant \sum_{n \in \mathcal{C}_k} \inf_{\mu \in \mathcal{R}_m} d(p_n, \mu) = \sum_{n \in \mathcal{C}_k} d_{nm}^{\inf}.\end{aligned}$$

Analogously,

$$\begin{aligned}\mathbf{Loss}^{\mathrm{opt}}\big(\{p_n, n \in \mathcal{C}_k\}\big|\mathcal{R}_m\big) &= \min_{\mu \in \mathcal{R}_m} \sum_{n \in \mathcal{C}_k} d(p_n, \mu) \\ &\leqslant \sum_{n \in \mathcal{C}_k} \sup_{\mu \in \mathcal{R}_m} d(p_n, \mu) = \sum_{n \in \mathcal{C}_k} d_{nm}^{\sup}.\end{aligned}$$ □

Now consider the optimal loss for cluster $\mathcal{C}_k$ in the "complete" constrained-set $\mathcal{R}^{\cup}$, $\mathbf{Loss}^{\mathrm{opt}}(\{p_i, i \in \mathcal{C}_k\}|\mathcal{R}^{\cup})$ – as defined in (9). Since $\mathcal{R}^{\cup} = \bigcup_{m=1}^{M} \mathcal{R}_m$, it can be seen that

$$\mathbf{Loss}^{\mathrm{opt}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}^{\cup}\big) = \min_{m=1,\dots,M} \mathbf{Loss}^{\mathrm{opt}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}_m\big). \tag{15}$$

Theorem 4. For the optimal cluster loss $\mathbf{Loss}^{\mathrm{opt}}(\{p_n, i \in \mathcal{C}_k\}|\mathcal{R}^{\cup})$, these bounds are valid:

$$\min_{m=1,\dots,M} \sum_{n \in \mathcal{C}_k} d_{nm}^{\inf} \leqslant \mathbf{Loss}^{\mathrm{opt}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}^{\cup}\big) \leqslant \min_{m=1,\dots,M} \sum_{n \in \mathcal{C}_k} d_{nm}^{\sup}. \tag{16}$$

Proof. Since the bounds

$$\sum_{n \in \mathcal{C}_k} d_{nm}^{\inf} \leqslant \mathbf{Loss}^{\mathrm{opt}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}_m\big) \leqslant \sum_{n \in \mathcal{C}_k} d_{nm}^{\sup}$$

are valid element-wise (see (14)), they must be also valid over the minimum values. □

Here we note that our sub-task of finding bounds for $\mathcal{C}_k \cup \{j\}$ given bounds for $\mathcal{C}_k$ becomes easy. For this, we have to "memorize" M sums $(\sum_{n \in \mathcal{C}_k} d_{nm}^{\inf})$ and $(\sum_{n \in \mathcal{C}_k} d_{nm}^{\sup})$, and using them we can obtain the updated bounds in $O(M)$ time:

$$\mathbf{Loss}^{\mathrm{opt}}\big(\big\{p_i, i \in \mathcal{C}_k \cup \{j\}\big\}\big|\mathcal{R}^{\cup}\big) \geqslant \min_{m=1,\dots,M} \Bigg[\bigg(\sum_{n \in \mathcal{C}_k} d_{nm}^{\inf}\bigg) + d_{jm}^{\inf}\Bigg], \tag{17a}$$

$$\mathbf{Loss}^{\mathrm{opt}}\big(\big\{p_i, i \in \mathcal{C}_k \cup \{j\}\big\}\big|\mathcal{R}^{\cup}\big) \leqslant \min_{m=1,\dots,M} \Bigg[\bigg(\sum_{n \in \mathcal{C}_k} d_{nm}^{\sup}\bigg) + d_{jm}^{\sup}\Bigg]. \tag{17b}$$

3.4. *The Global Lower Bound*

3.4.1. *Definition*

Define

$$\mathbf{Loss}^{\mathcal{LB}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}^{\cup}\big) = \min_{m=1,\dots,M} \sum_{n\in\mathcal{C}_k} d_{nm}^{\inf}. \tag{18}$$

In Theorem 4 we have shown that this definition satisfies (10):

$$\mathbf{Loss}^{\mathcal{LB}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}^{\cup}\big) \leqslant \mathbf{Loss}^{\mathrm{opt}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}^{\cup}\big). \tag{19}$$

Now take some arbitrary leaf-node $v_N \in \mathcal{L}(N, K)$ and consider the related partition

$$\mathcal{P}_N = \big\{\mathcal{C}_1(v_N), \dots, \mathcal{C}_K(v_N)\big\}.$$

For each cluster $\mathcal{C}_k(v_N)$, $k = 1, \dots, K$, (19) holds. Summing over all the clusters, we obtain

$$\sum_{k=1}^{K} \mathbf{Loss}^{\mathcal{LB}}\big(\big\{p_i, i \in \mathcal{C}_k(v_N)\big\}\big|\mathcal{R}^{\cup}\big) \leqslant \sum_{k=1}^{K} \mathbf{Loss}^{\mathrm{opt}}\big(\big\{p_i, i \in \mathcal{C}_k(v_N)\big\}\big|\mathcal{R}^{\cup}\big). \tag{20}$$

The right-hand-side of this inequality is the definition of $\mathbf{Loss}^{\mathrm{opt}}([\mathbb{P}_1^N/K]|[\mathcal{R}^{\cup}, \mathcal{P}_N])$. Notate the left-hand-side with

$$\mathbf{Loss}^{\mathcal{LB}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^{\cup}, \mathcal{P}_N]\big) = \sum_{k=1}^{K} \mathbf{Loss}^{\mathcal{LB}}\big(\big\{p_i, i \in \mathcal{C}_k(v_N)\big\}\big|\mathcal{R}^{\cup}\big). \tag{21}$$

Now define:

$$\mathbf{Loss}^{\mathcal{LB}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^{\cup}]\big) = \min_{v_N\in\mathcal{L}(N,K)} \mathbf{Loss}^{\mathcal{LB}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^{\cup}, \mathcal{P}(v_N)]\big). \tag{22}$$

Since

$$\mathbf{Loss}^{\mathrm{opt}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^{\cup}]\big) = \min_{v_N\in\mathcal{L}(N,K)} \mathbf{Loss}^{\mathrm{opt}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^{\cup}, \mathcal{P}(v_N)]\big),$$

we immediately see from definition (22) and inequality (20) that

$$\mathbf{Loss}^{\mathcal{LB}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^{\cup}]\big) \leqslant \mathbf{Loss}^{\mathrm{opt}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^{\cup}]\big).$$

3.4.2. *Determination*

Now consider the problem of finding $\mathbf{Loss}^{\mathcal{LB}}([\mathbb{P}_1^N/K]|[\mathcal{R}^{\cup}])$ as defined in (22). Suppose we want to find it using the enumeration tree $\mathcal{T}(N, K)$. We will need pruning criteria.

Take arbitrary $v_n \in \mathcal{T}(N, K)$, the related partial partition $\partial\mathcal{P}_n = \partial\mathcal{P}(v_n)$ and consider the following problem (in analogy with (7)):

$$\min_{v_N \in \mathcal{L}(N,K)} \left(\sum_{k=1}^{K} \mathbf{Loss}^{\mathcal{LB}}\big(\{p_i, i \in \mathcal{C}_k(v_N)\} \big| \mathcal{R}^{\cup}\big) \right) \tag{23}$$
$$\text{s.t.} \quad \text{vertex } v_N \text{ is reached through } v_n.$$

We label the solution to the above equation with $\mathbf{Loss}^{\mathcal{LB}}([\mathbb{P}_1^N/K] \| [\mathcal{R}^{\cup}, \partial\mathcal{P}_n])$.

Theorem 5 (*Lower bound pruning criteria*).

$$\mathbf{Loss}^{\mathcal{LB}}\big([\mathbb{P}_1^N/K] \big\| [\mathcal{R}^{\cup}, \partial\mathcal{P}_n]\big) \geqslant \mathbf{Loss}^{\mathcal{LB}}\big([\mathbb{P}_1^n/K] \big\| [\mathcal{R}^{\cup}, \partial\mathcal{P}_n]\big) + \mathbf{Loss}^{\mathcal{LB}}\big([\mathbb{P}_{n+1}^N/K] \big\| [\mathcal{R}^{\cup}]\big).$$

Proof. Again, analogous to the proof of Theorem 1 and Theorem 2. Suppose the minimal value of problem (23) is attained for clustering $\mathcal{C}_1, \dots, \mathcal{C}_K$ with optimal region-unit indexes $m_1, \dots, m_K$, e.g.

$$\mathbf{Loss}^{\mathcal{LB}}\big(\{p_i, i \in \mathcal{C}_k\} \big| \mathcal{R}^{\cup}\big) = \min_{m=1,\dots,M} \sum_{n \in \mathcal{C}_k} d_{nm}^{\inf} = \sum_{n \in \mathcal{C}_k} d_{nm_k}^{\inf}, \quad k = 1, \dots, K.$$

Define

$$\mathcal{C}_k^{\dagger} = \partial\mathcal{C}_k(v_n) = \mathcal{C}_k \cap \mathbb{N}_1^n, \qquad \mathcal{C}_k^{\ddagger} = \mathcal{C}_k \cap \mathbb{N}_{n+1}^N, \quad k = 1, \dots, K.$$

Now we enumerate the partitions of the index-set $\mathbb{N}_{n+1}^N$. Consider the enumeration tree $\mathcal{T}(N - n, K)$ and its leaf-set $\mathcal{L}(N - n, K)$. Take any $w_{N-n} \in \mathcal{L}(N - n, K)$ and consider the corresponding path $w_1 w_2 \dots w_{N-n}$. Label

$$\mathcal{C}_k^{\ddagger}(w_{N-n}) = \mathcal{C}_k^{\ddagger}(w_1 w_2 \dots w_{N-n}) = \big\{ i \in \mathbb{N}_{n+1}^N : \lambda(w_{i-n}) = k \big\}.$$

We have (in the equations below, $d_{nm} = d_{nm}^{\inf}$, as defined in (11a))

$$\begin{aligned}
\mathbf{Loss}^{\mathcal{LB}}\big([\mathbb{P}_1^N/K] \big\| [\mathcal{R}^{\cup}, \partial\mathcal{P}_n]\big) &= \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} d_{im_k} = \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k^{\dagger}} d_{im_k} + \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k^{\ddagger}} d_{im_k} \\
&\geqslant \sum_{k=1}^{K} \left(\min_{m=1,\dots,M} \sum_{i \in \mathcal{C}_k^{\dagger}} d_{im} \right) + \min_{w \in \mathcal{L}(N-n,K)} \sum_{k=1}^{K} \left(\min_{m=1\dots,M} \sum_{i \in \mathcal{C}_k^{\ddagger}(w)} d_{im} \right) \\
&= \mathbf{Loss}^{\mathcal{LB}}\big([\mathbb{P}_1^n/K] \big\| [\mathcal{R}^{\cup}, \partial\mathcal{P}_n]\big) + \min_{w \in \mathcal{L}(N-n,K)} \sum_{k=1}^{K} \mathbf{Loss}^{\mathcal{LB}}\big(\{p_i, i \in \mathcal{C}_k^{\ddagger}(w)\} \big| \mathcal{R}^{\cup}\big).
\end{aligned}$$

The last term is the definition of $\mathbf{Loss}^{\mathcal{LB}}([\mathbb{P}_{n+1}^N/K] \| [\mathcal{R}^{\cup}])$, see (21) and (22). □

3.4.3. *MIP Formulation*

Problem of finding $\mathbf{Loss}^{\mathcal{LB}}([\mathbb{P}^N_{n+1}/K]|[\mathcal{R}^\cup])$ can be seen as a discrete facility location problem. Indeed, we have to choose K "facilities" from $\{\mathcal{R}_1, \dots, \mathcal{R}_M\}$ at positions $m_1, \dots, m_K$, so that the loss

$$\sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} d^{\inf}_{i m_k}$$

is minimal. We formulate the corresponding mixed integer programming (MIP) problem. For this, introduce M binary variables $\{y_1, \dots, y_M\}$:

$$y_m = \begin{cases} 1, & \text{if "facility" at } \mathcal{R}_m \text{ is opened,} \\ 0 & \text{otherwise.} \end{cases}$$

Also, introduce $N \times M$ binary variables $\{x_{nm},\ n = 1, \dots, N,\ m = 1, \dots, M\}$:

$$x_{nm} = \begin{cases} 1, & \text{if "client" } p_n \text{ is assigned to "facility" at } \mathcal{R}_m, \\ 0 & \text{otherwise.} \end{cases}$$

MIP formulation:

$$\begin{aligned} \min_{\{y_m\},\,\{x_{nm}\}} \quad & \sum_{n=1}^{N} \sum_{m=1}^{M} d^{\inf}_{nm} x_{nm} \\ \text{s.t.} \quad & \sum_{m=1}^{M} y_m \leqslant K, \\ & x_{nm} \leqslant y_m, \quad x_{nm} \in \{0, 1\}, \quad y_m \in \{0, 1\}, \\ & n = 1, \dots, N, \quad m = 1, \dots, M. \end{aligned} \tag{24}$$

3.5. *A Comment on the Error Between the Lower Bound and the Exact Solution*

Define the "diameter" diam of an arbitrary region-unit $\mathcal{R}_m$ as

$$\operatorname{diam}(\mathcal{R}_m) = \sup_{x, y \in \mathcal{R}_m} d(x, y).$$

In addition, suppose that distance metric d satisfies the triangle inequality: given any three points $x, y, z \in \mathcal{S}$, it is always true that

$$d(x, z) \leqslant d(x, y) + d(y, z).$$

In particular, this property holds for the distance metric d induced by a set of barriers, i.e. when $d(x, y)$ is equal to the length of the shortest path between x and y (and the length of an arbitrary path from x to y is defined by the standard length of a curve in the Euclidean geometry).

Now take an arbitrary point $p_n \in \{p_1, p_2, \dots, p_N\}$. Consider the nearest and the furthest point for p_n in region-unit $\mathcal{R}_m$, as was defined in (11). By the triangle inequality, we have

$$d_{nm}^{\sup} = d\big(p_n, \mu_{nm}^{\sup}\big) \leqslant d\big(p_n, \mu_{nm}^{\inf}\big) + d\big(\mu_{nm}^{\inf}, \mu_{nm}^{\sup}\big) \leqslant d_{nm}^{\inf} + \operatorname{diam}(\mathcal{R}_m). \tag{25}$$

Thus, if region-unit $\mathcal{R}_m$ is sufficiently small ($\operatorname{diam}(\mathcal{R}_m) \approx 0$), we can expect that $d_{nm}^{\inf} \approx d_{nm}^{\sup}$. (If $\operatorname{diam}(\mathcal{R}_m)$ is not sufficiently small, we could divide $\mathcal{R}_m$ into smaller parts and redefine $\mathcal{R}^{\cup}$.)

Now consider some cluster $\mathcal{C}_k$ and suppose that $\operatorname{diam}(\mathcal{R}_m) \leqslant \varepsilon$ for any $\mathcal{R}_m \in \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_M\}$ for some $\varepsilon > 0$. By combining (16) and (25),

$$\begin{aligned}\mathbf{Loss}^{\mathrm{opt}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}_m\big) &\leqslant \sum_{n\in\mathcal{C}_k} d_{nm}^{\sup} \\ &\overset{(25)}{\leqslant} \sum_{n\in\mathcal{C}_k} \big(d_{nm}^{\inf} + \operatorname{diam}(\mathcal{R}_m)\big) \leqslant \sum_{n\in\mathcal{C}_k} d_{nm}^{\inf} + |\mathcal{C}_k|\varepsilon,\end{aligned}$$

where $|\mathcal{C}_k|$ is the number of items in cluster $\mathcal{C}_k$. Therefore, we have the following bounds:

$$\sum_{n\in\mathcal{C}_k} d_{nm}^{\inf} \leqslant \mathbf{Loss}^{\mathrm{opt}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}_m\big) \leqslant \sum_{n\in\mathcal{C}_k} d_{nm}^{\inf} + |\mathcal{C}_k|\varepsilon. \tag{26}$$

Similarly, for $\mathbf{Loss}^{\mathrm{opt}}(\{p_i, i \in \mathcal{C}_k\}|\mathcal{R}^{\cup})$ we obtain

$$\begin{aligned}\min_{m=1,\dots,M} \sum_{n\in\mathcal{C}_k} d_{nm}^{\inf} \leqslant \mathbf{Loss}^{\mathrm{opt}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}^{\cup}\big) &\leqslant \min_{m=1,\dots,M} \sum_{n\in\mathcal{C}_k} d_{nm}^{\sup} \\ &\leqslant \min_{m=1,\dots,M} \sum_{n\in\mathcal{C}_k} d_{nm}^{\inf} + |\mathcal{C}_k|\varepsilon,\end{aligned} \tag{27}$$

and noting the definition of $\mathbf{Loss}^{\mathcal{LB}}(\{p_i, i \in \mathcal{C}_k\}|\mathcal{R}^{\cup})$ in (18), inequalities (27) can be rewritten as

$$\begin{aligned}\mathbf{Loss}^{\mathcal{LB}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}^{\cup}\big) &\leqslant \mathbf{Loss}^{\mathrm{opt}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}^{\cup}\big) \\ &\leqslant \mathbf{Loss}^{\mathcal{LB}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}^{\cup}\big) + |\mathcal{C}_k|\varepsilon.\end{aligned} \tag{28}$$

Summing over all the clusters of a particular partition $\mathcal{P} = \{\mathcal{C}_1, \dots, \mathcal{C}_K\}$, we obtain (use the definitions in (6), (21) and inequality above)

$$\begin{aligned}\mathbf{Loss}^{\mathcal{LB}}\big(\big[\mathbb{P}_1^N/K\big]\big|\big[\mathcal{R}^{\cup}, \mathcal{P}\big]\big) &\leqslant \mathbf{Loss}^{\mathrm{opt}}\big(\big[\mathbb{P}_1^N/K\big]\big|\big[\mathcal{R}^{\cup}, \mathcal{P}\big]\big) \\ &\leqslant \mathbf{Loss}^{\mathcal{LB}}\big(\big[\mathbb{P}_1^N/K\big]\big|\big[\mathcal{R}^{\cup}, \mathcal{P}\big]\big) + N\varepsilon.\end{aligned}$$

In particular, the gap between the lower bound and the optimal solution for any partition $\mathcal{P}$ is at most $N\varepsilon$. Thus, we can control the level of this gap by subdividing $\mathcal{R}^{\cup} = \bigcup_{m=1}^{M} \mathcal{R}_m$ into sufficiently small region-units $\mathcal{R}_m$. These bounds also hold for the globally optimal solution:

$$\begin{aligned}\mathbf{Loss}^{\mathcal{LB}}\big(\big[\mathbb{P}_1^N/K\big]\big|\big[\mathcal{R}^{\cup}\big]\big) &\leqslant \mathbf{Loss}^{\mathrm{opt}}\big(\big[\mathbb{P}_1^N/K\big]\big|\big[\mathcal{R}^{\cup}\big]\big)\\ &\leqslant \mathbf{Loss}^{\mathcal{LB}}\big(\big[\mathbb{P}_1^N/K\big]\big|\big[\mathcal{R}^{\cup}\big]\big) + N\varepsilon.\end{aligned} \tag{29}$$

3.6. *Summary*

We have defined $\mathbf{Loss}^{\mathcal{LB}}(\{p_i, i \in \mathcal{C}_k\}|\mathcal{R}^{\cup})$ in (18) and have proven in Theorem 4 that

$$\mathbf{Loss}^{\mathcal{LB}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}^{\cup}\big) \leqslant \mathbf{Loss}^{\mathrm{opt}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}^{\cup}\big).$$

Also, we have defined $\mathbf{Loss}^{\mathcal{LB}}([\mathbb{P}_1^N/K]|[\mathcal{R}^{\cup}])$ in (22) which satisfies

$$\mathbf{Loss}^{\mathcal{LB}}\big(\big[\mathbb{P}_1^N/K\big]\big|\big[\mathcal{R}^{\cup}\big]\big) \leqslant \mathbf{Loss}^{\mathrm{opt}}\big(\big[\mathbb{P}_1^N/K\big]\big|\big[\mathcal{R}^{\cup}\big]\big).$$

We have also proven in Theorem 5 the pruning criteria:

$$\begin{aligned}\mathbf{Loss}^{\mathcal{LB}}\big(\big[\mathbb{P}_1^N/K\big]\big|\big[\mathcal{R}^{\cup}, \partial\mathcal{P}_n\big]\big) \geqslant{}& \mathbf{Loss}^{\mathcal{LB}}\big(\big[\mathbb{P}_1^n/K\big]\big|\big[\mathcal{R}^{\cup}, \partial\mathcal{P}_n\big]\big)\\ &+ \mathbf{Loss}^{\mathcal{LB}}\big(\big[\mathbb{P}_{n+1}^N/K\big]\big|\big[\mathcal{R}^{\cup}\big]\big),\end{aligned} \tag{30}$$

where $\partial\mathcal{P}_n = \{\partial\mathcal{C}_1, \dots, \partial\mathcal{C}_K\}$ is a (given) partition of index-set $\mathbb{N}_1^n$.

4. Overview of the Proposed Algorithm

Consider solving problem (1), i.e. clustering N points into K clusters. The algorithm we propose in this paper can be summarized with these steps:

Step 0 (Initialization). Calculate $d_{nm}^{\inf}$ as defined in (11a). From programming point of view, this is the most difficult part of the algorithm which is problem specific. Fortunately, for multi-Weber problem with polyhedral barriers, calculation of $d_{nm}^{\inf}$ is possible (see Section 5.2).

Step 1 (Lower bound problems). Solve lower bound problems

$$\mathbf{Loss}^{\mathcal{LB}}\big(\big[\mathbb{P}_{n+1}^N/K\big]\big|\big[\mathcal{R}^{\cup}\big]\big), \quad \text{for } n = N-1, \dots, 0,$$

using the enumeration tree $\mathcal{T}(N-n, K)$ (illustrated in Fig. 1). Use pruning criteria (30): at step n, when looking for $\mathbf{Loss}^{\mathcal{LB}}([\mathbb{P}_{n+1}^N/K]|[\mathcal{R}^{\cup}])$, for some partial partition $\partial\mathcal{P}$ of point set $\mathbb{P}_{n+1}^{\eta}$, values

$$\mathbf{Loss}^{\mathcal{LB}}\big(\big[\mathbb{P}_{n+1}^{\eta}/K\big]\big|\big[\mathcal{R}^{\cup}, \partial\mathcal{P}\big]\big),\ \mathbf{Loss}^{\mathcal{LB}}\big(\big[\mathbb{P}_{\eta+1}^N/K\big]\big|\big[\mathcal{R}^{\cup}\big]\big)$$

are known.

Step 2 (Finding an upper bound). Suppose

$$\mathbf{Loss}^{\mathcal{LB}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^{\cup}]\big)$$

is attained for clustering $\mathcal{C}_1^{\mathcal{LB}},\dots,\mathcal{C}_K^{\mathcal{LB}}$ – as found in the previous step. For this clustering, for each cluster $\mathcal{C}_k^{\mathcal{LB}}$, $k=1,\dots,K$, solve the optimal centre problem $\mathbf{Loss}^{\mathrm{opt}}(\{p_i, i\in\mathcal{C}_k^{\mathcal{LB}}\}|\mathcal{R}^{\cup})$ within ϵ-accuracy by subdividing the most promising region-units $\mathcal{R}_m$, $m=1,\dots,M$ into smaller parts (e.g. as described in Section 5.3). This way, we obtain an upper bound $L^{\mathcal{UB}}$ – as an optimal value of a particular solution corresponding to $\mathcal{C}_1^{\mathcal{LB}},\dots,\mathcal{C}_K^{\mathcal{LB}}$.

Step 3 (Find candidates to the global solution). Enumerate all $v_N\in\mathcal{L}(N,K)$ s.t.

$$\mathbf{Loss}^{\mathcal{LB}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^{\cup},\mathcal{P}(v_N)]\big)\leqslant L^{\mathcal{UB}}. \tag{31}$$

At any vertex $v_n\in\mathcal{T}(N,K)$, prune the corresponding branch if

$$\mathbf{Loss}^{\mathcal{LB}}\big([\mathbb{P}_1^n/K]\big|[\mathcal{R}^{\cup},\partial\mathcal{P}(v_n)]\big)+\mathbf{Loss}^{\mathcal{LB}}\big([\mathbb{P}_{n+1}^N/K]\big|[\mathcal{R}^{\cup}]\big)>L^{\mathcal{UB}}.$$

Label the obtained set with $\mathcal{V}=\{v_N\in\mathcal{L}(N,K): v_N \text{ satisfies } (31)\}$.

Step 4 (Determining the best solution). Starting from the most promising $v_N\in\mathcal{V}$ (e.g., with minimal values of $\mathbf{Loss}^{\mathcal{LB}}([\mathbb{P}_1^n/K]|[\mathcal{R}^{\cup},\mathcal{P}(v_N)])$), solve each problem

$$\mathbf{Loss}^{\mathrm{opt}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^{\cup},\mathcal{P}(v_N)]\big)$$

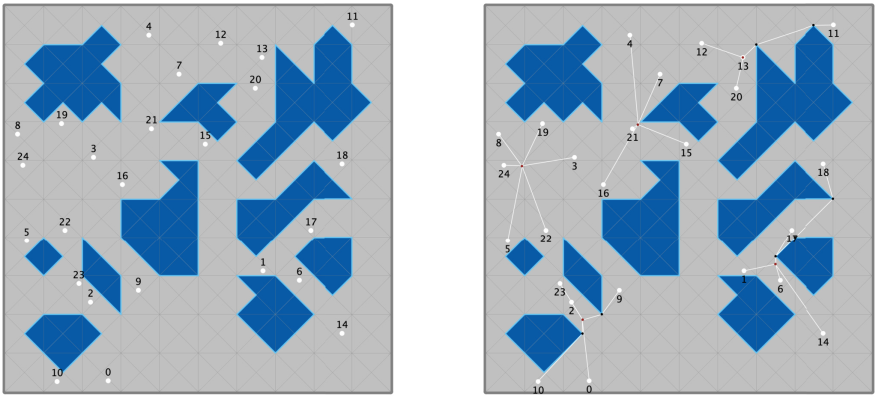
within ϵ-accuracy by partitioning region-units into smaller parts (e.g. as described in Section 5.3). Keep record of $L^{\mathcal{UB}}$ – the upper bound of the best solution. If we get that

$$\mathbf{Loss}^{\mathrm{opt}}\big([\mathbb{P}_1^N/K]\big|[\mathcal{R}^{\cup},\mathcal{P}(v_N)]\big)>L^{\mathcal{UB}},$$

remove v_N from $\mathcal{V}$.

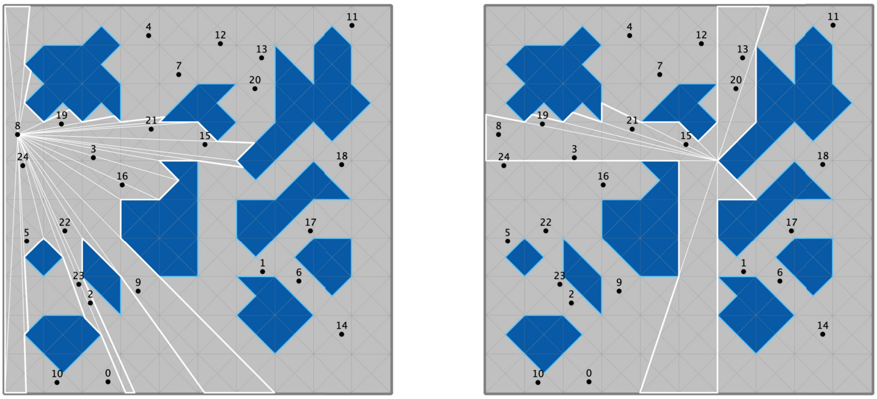
5. Numerical Example

We apply the ideas from previous sections to the multi-Weber with polyhedral barriers problem. An example of the problem and its ϵ-accuracy solution are illustrated in Fig. 2. The problem is defined within a unit-square with a 10 × 10 grid, with each square initially divided into 4 triangles. The blue polygons are the barriers; it is not allowed to travel through them – these barriers induce the distance metric $d:\mathcal{S}\times\mathcal{S}\mapsto\mathbf{R}^+$. There are 10 barriers consisting of 100 triangles in total. The region-union $\mathcal{R}^{\cup}$ consists of the remaining 300 gray-coloured triangles (region-units $\mathcal{R}_m$): $\mathcal{R}^{\cup}=\bigcup_{m=1}^{300}\mathcal{R}_m$.



(a) Input (b) Output

Fig. 2. Location with barriers problem example. We seek to cluster 25 given points into 5 clusters. The solution is proved by our algorithm to be within $\epsilon = 0.01$ accuracy to the global optimum (i.e. the globally optimal loss differs by at most 1% when compared with the reported solution).



(a) Visibility polygon for a client (b) Visibility polygon for a barrier-corner

Fig. 3. Visibility polygons for selected points.

5.1. *Visibility Graph*

To compute the distances $d(p_i, \mu_k)$ between client-points and facility-points, we need to find the so-called visibility-graph $\mathcal{G}(V, E)$ (Viegas and Hansen, 1985). Each point $p \in \mathcal{R}^{\cup}$ defines a visibility-polygon as illustrated in Fig. 3. This polygon contains the set of points "seen" from the corresponding point directly through a ray; to reach the points outside this polygon, we must bypass some barriers. It is easy to convince oneself that (see Viegas and Hansen, 1985), in case point $q \in \mathcal{R}^{\cup}$ is not seen directly from point p, the shortest path $d(p, q)$ consists of a set of segments with the end-points at the corners

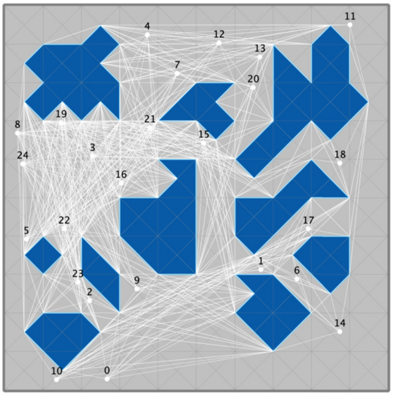


Fig. 4. Visibility graph for the example problem.

of the barrier-regions.[6] Therefore, the visibility-polygons for the barrier corner-points are particularly interesting (as in Fig. 3b). Barrier corner-points, together with client-points, define the set of vertices V for the visibility-graph $\mathcal{G}(V, E)$. The (directed) edges E are inserted as follows: if vertex $v \in V$ is "seen" from a barrier-corner-vertex w (i.e. v belongs to the visibility polygon of w), we add the edge $w \mapsto v$ to graph (with edge weight equal to the Euclidean-length from point $p(v)$ to point $p(w)$).

To compute the shortest paths from a given facility-point μ to each client-vertex $v(p_1), \dots, v(p_N)$, we could insert an additional vertex $v(\mu)$ to the graph, find the set of vertices $v \in V$ "seen" from μ, and insert the corresponding edges. Now, the shortest-paths can be found with Dijkstra's algorithm (Dijkstra, 1959).

Visibility graph for our example problem is shown in Fig. 4.

5.2. *Minimal Distances to a Triangle-Facility*

To apply the ideas from the previous sections, we have to compute the lower-bound distances $d_{nm}^{\inf}$, $n \in \mathbb{N}_1^N$, $m \in \mathbb{N}_1^M$, i.e. the shortest possible distance $d_{nm}^{\inf}$ from the client-point p_n to the region-unit (triangle-facility in our case) $\mathcal{R}_m$.

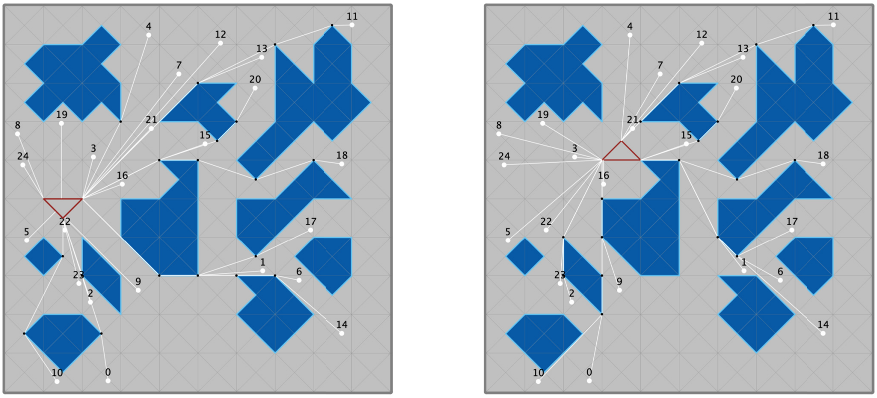
For this task, we use the visibility-graph $\mathcal{G}(V, E)$ described in the previous subsection.

To find the lower-bound distances $d_{nm}^{\inf}$, $n \in \mathbb{N}_1^N$, we firstly insert an additional vertex $v(\mathcal{R}_m)$ into the graph. Now, we have to take care of the edges. This can be done as follows: for each $v \in V$, intersect the visibility polygon $\mathcal{VP}(v)$ of v with triangle $\mathcal{R}_m$: $\mathcal{I}(v, \mathcal{R}_m) = \mathcal{VP}(v) \cap \mathcal{R}_m$.[7] If the intersection $\mathcal{I}(v, \mathcal{R}_m)$ is empty, do nothing. Otherwise, find the nearest point $\mathbf{proj}(v, \mathcal{R}_m)$ from $p(v)$ to $\mathcal{I}(v, \mathcal{R}_m)$.[8] Now, add to $\mathcal{G}(V, E)$ a directed-edge $v(\mathcal{R}_m) \mapsto v$ with weight $\|p(v) - \mathbf{proj}(v, \mathcal{R}_m)\|_2$. The distances $d_{nm}^{\inf}$, $n \in \mathbb{N}_1^N$ for the triangle-facility $\mathcal{R}_m$ can now again be found with Dijkstra's algorithm (Dijkstra, 1959).

[6] Except the two segments "starting" at p and "ending" at q.

[7] Since the polygon $\mathcal{VP}(v)$ is non-convex, it might be that $\mathcal{I}(v, \mathcal{R}_m)$ is a not a single, but a set of polygons.

[8] The nearest distance from a point to a polygon can be found by iterating the "projections" on the edges of the polygon. Iterating further over the polygons constituting $\mathcal{I}(v, \mathcal{R}_m)$, we obtain the result.



(a) Triangle-facility 1.

(b) Triangle-facility 2.
LB ≈ 8.73, UB ≈ 9.98

Fig. 5. Shortest distances to triangle-facilities. The triangle-facility in Fig. 5b has the minimal sum of distances $\sum_{n=1}^{N} d_{nm}^{\text{inf}}$ from all region-units $\mathcal{R}_m$, $m = 1, \dots, M$.

The output of this procedure is illustrated in Fig. 5.

To compute an upper bound for the sum of distances from triangle-facility $\mathcal{R}_m$ to all the clients, we could pick any point $\mu \in \mathcal{R}_m$ and calculate the distances as outlined in Section 5.1, i.e. using the visibility graph. However, in our case where all the region-units are triangles, for finding an upper-bound faster, we can use the following simple idea.

- Pick the longest edge of the triangle, compute its length l_1.
- Pick the middle point of this edge. Define l_2 as the distance from this middle-point to the vertex in front of the longest edge.
- All the points within the triangle are at maximum distance $l = \max\left(\frac{l_1}{2}, l_2\right)$. The upper bound for the triangle-facility $\mathcal{R}_m$ can be found by the formula $\left(\sum_{n=1}^{N} d_{nm}^{\text{inf}}\right) + Nl$.

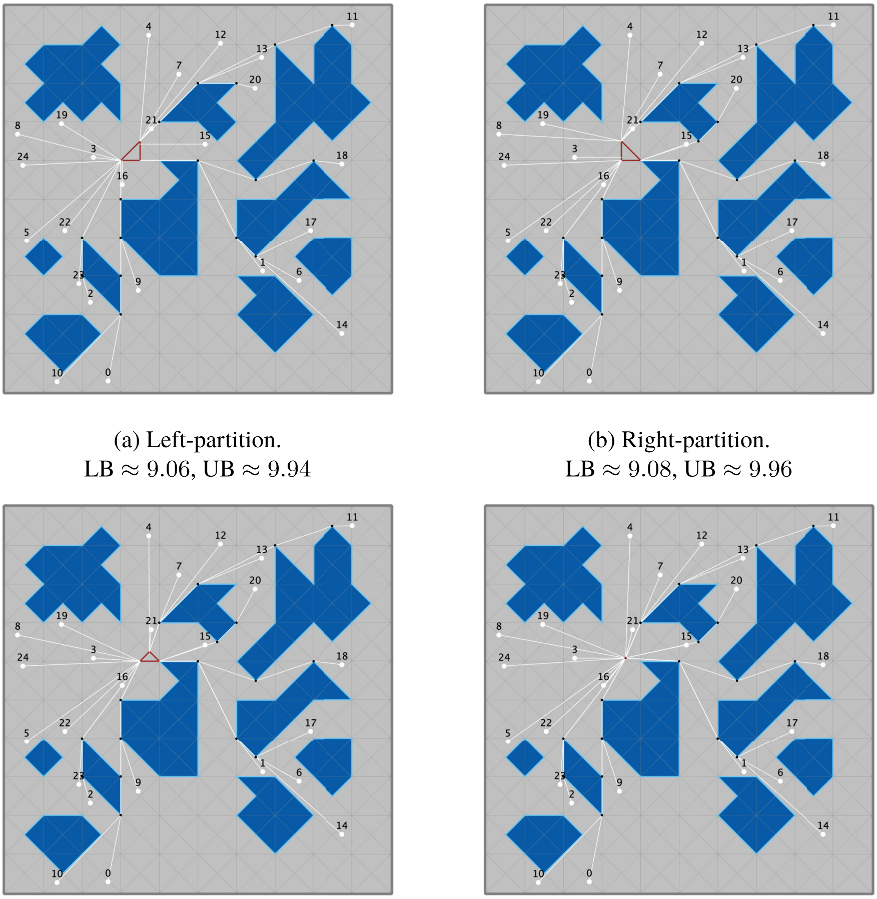
5.3. *Finding the Optimal Centre Within a Predefined Accuracy ϵ*

To find the globally-optimal centre for a set of points, i.e. optimal centree for some cluster $\mathcal{C}_k$, we use the triangle-partitioning algorithm in the style of Paulavičius and Žilinskas (2013). For a start, we place all the initial triangles (region-units $\mathcal{R}_m$) in a MinPQ[9] (Sedgewick and Wayne, 2011), ordered by the lower-bound value

$$\mathbf{Loss}^{\mathcal{LB}}\left(\{p_i, i \in \mathcal{C}_k\} | \mathcal{R}_m\right) = \sum_{i \in \mathcal{C}_k} d_{im}^{\text{inf}}, \quad m = 1, \dots, M. \tag{32}$$

Initially, the MinPQ contains M elements. Now, pick from MinPQ the triangle-facility $\mathcal{R}_{\min}$ with the smallest lower-bound value (32). Delete this triangle from MinPQ. Next,

[9] Minimum priority queue.



(a) Left-partition.
LB ≈ 9.06, UB ≈ 9.94

(b) Right-partition.
LB ≈ 9.08, UB ≈ 9.96

(c) After 20 iterations.
LB ≈ 9.175, UB ≈ 9.80

(d) Within ϵ-accuracy with $\epsilon = 0.1\%$.
LB ≈ 9.69675, UB ≈ 9.70365.
Number of partitions: 4979

Fig. 6. Illustration of the triangle-partitioning algorithm.

partition $\mathcal{R}_{\min}$ into two smaller triangles $\mathcal{R}_{\text{left}}$ and $\mathcal{R}_{\text{right}}$ along the longest edge (see the partitions of the triangle-facility from Fig. 5b in Fig. 6a, Fig. 6b). Both these triangles have an improved (increased) lower-bound estimates

$$\mathbf{Loss}^{\mathcal{LB}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}_{\text{left}}\big),\ \mathbf{Loss}^{\mathcal{LB}}\big(\{p_i, i \in \mathcal{C}_k\}\big|\mathcal{R}_{\text{right}}\big),$$

because they were obtained by subdividing $\mathcal{R}_{\min}$ into two smaller pieces. Since the triangles have been shrunk, the gap between the lower bound and the upper bound for these triangles improves, i.e. we have shown that inequality (28) holds with $\varepsilon_{\text{left}} = \text{diam}(\mathcal{R}_{\text{left}})$ and $\varepsilon_{\text{right}} = \text{diam}(\mathcal{R}_{\text{right}})$. Insert $\mathcal{R}_{\text{left}}$ and $\mathcal{R}_{\text{right}}$ into MinPQ. Continue deleting/partitioning triangles in MinPQ until the distance between the best known lower-bound and the best known upper-bound is within a predefined accuracy ϵ, i.e. until condition

$$\frac{\mathbf{Loss}^{\mathcal{LB}}(\{p_i, i \in \mathcal{C}_k\}|\mathcal{R}_{\min})}{\mathbf{Loss}^{\mathcal{LB}}(\{p_i, i \in \mathcal{C}_k\}|\mathcal{R}_{\min}) + |\mathcal{C}_k|\mathrm{diam}(\mathcal{R}_{\min})} > 1 - \epsilon$$

holds for some "small" ϵ (take for example $\epsilon = 0.001$). The procedure is illustrated in Fig. 6.

5.4. *Other Steps*

Other steps in the algorithm of Section 4 given a routine to calculate $d_{nm}^{\inf}$ and a routine to find the optimal centre for a cluster (as described in Section 5.3) are rather straightforward. We just emphasize that during enumeration procedure at any vertex $v_n \in \mathcal{T}(N, K)$, we use formula (17a) to find

$$\mathbf{Loss}^{\mathcal{LB}}\big(\{p_i, i \in \mathcal{C}_{\lambda(v_n)} \cup \{n\}\}\big|\mathcal{R}^{\cup}\big)$$

in $O(M)$ time.

5.5. *Algorithm Analysis*

5.5.1. *Improvement by Initial Subdivision of Region-Units*

Our initial analysis of the algorithm for the problem instance presented in Fig. 2 is summarized in Table 1.[10] In the experiment, we have analysed algorithm performance for the number of clients $N = 20, \dots, 25$ and the number of clusters $K = 3, 4, 5$. Table 1a shows time statistics for lower bound calculations (**Step 1** in the algorithm of Section 4), Table 1b shows time statistics for the enumeration of candidate solutions (**Step 3** in the algorithm of Section 4), and Table 1c gives the statistics for the number of candidate solutions to the global optima.

While for the lower bound step times differ only slightly, the number of possible candidates to the global solution grows exponentially; the time to enumerate all these solutions is proportional. The reason to this is that our lower bound estimates are considerably (say, within 20%-tolerance) smaller than the value of the globally optimal solution.

We try the following simple "remedy" to this issue: suppose that each triangle-unit $\mathcal{R}_m$, $m = 1, \dots, 300$ is subdivided into 2 smaller triangles $\mathcal{R}_{m1}$, $\mathcal{R}_{m2}$.[11] Such a subdivision gives an improvement of the error between the lower bound and the exact solution (as discussed in Section 3.5). The updated statistics are presented in Table 2. Comparing with Table 1, we can see time increase for lower bound calculations (as expected), though this time is not crucial in the total time of the algorithm. But for the numbering the candidate solutions, the effect is clear: the number of candidates is reduced significantly.

[10] The numbers are particular for this problem instance, i.e. the one shown in Fig. 2 (if $N < 25$, we have used only points referring to $0, \dots, N-1$). We did not seek to perform a statistical experiment for this part as we just wanted to illustrate that a denser subdivision into region-units significantly improves the algorithm.

[11] Giving $\mathcal{R}^{\cup} = \bigcup_{m=1}^{300}(\mathcal{R}_{m1} \cup \mathcal{R}_{m2})$ – a union of 600 triangle-units.

Table 1
Algorithm statistics for initial triangles (300 region-units). The question mark (?) indicates that the data is missing due to the time limit (see the last column in Table 1b and also compare with Table 2b).

(a) Time for finding lower bounds (s)

N	3	4	5
20	0.076	0.128	0.232
21	0.072	0.165	0.300
22	0.116	0.179	0.285
23	0.061	0.107	?
24	0.053	0.145	?
25	0.064	0.139	?

(b) Time for enumerating possible global solutions (s)

N	3	4	5
20	0.241	0.293	10.562
21	0.213	0.424	21.648
22	0.697	0.895	141.80
23	0.308	1.545	?
24	0.498	6.119	?
25	0.620	10.135	?

(c) Number of candidates to the global optima

N	3	4	5
20	1358	1422	31422
21	1385	1726	42111
22	4912	5709	91078
23	3468	10905	?
24	4876	23926	?
25	5965	32460	?

Table 2
Algorithm statistics when initial triangles are halved (600 region-units).

(a) Time for finding lower bounds (s)

N	3	4	5
20	0.356	0.402	0.584
21	0.292	0.401	0.613
22	0.277	0.440	0.620
23	0.333	0.305	0.654
24	0.320	0.358	0.706
25	0.237	0.375	0.706

(b) Time for enumerating possible global solutions (s)

N	3	4	5
20	0.193	0.311	2.189
21	0.183	0.259	3.975
22	0.360	0.396	7.815
23	0.208	0.631	21.358
24	0.202	0.783	106.361
25	0.275	1.038	211.205

(c) Number of candidates to the global optima

N	3	4	5
20	319	373	5333
21	286	458	9273
22	1085	955	17026
23	599	1991	38852
24	805	3830	77354
25	903	4646	95275

Table 3
Algorithm statistics when initial triangles are divided into 8 pieces (2400 region-units).

(a) Time for finding lower bounds (s)

N	3	4	5
20	1.166	1.056	1.885
21	1.197	1.000	2.158
22	1.011	1.261	2.613
23	0.994	1.235	2.609
24	1.060	1.266	2.792
25	1.051	1.042	2.744

(b) Time for enumerating possible global solutions (s)

N	3	4	5
20	0.624	0.857	2.112
21	0.583	0.532	2.613
22	0.714	0.731	4.956
23	0.574	0.786	5.111
24	0.630	0.911	7.224
25	0.641	0.792	7.607

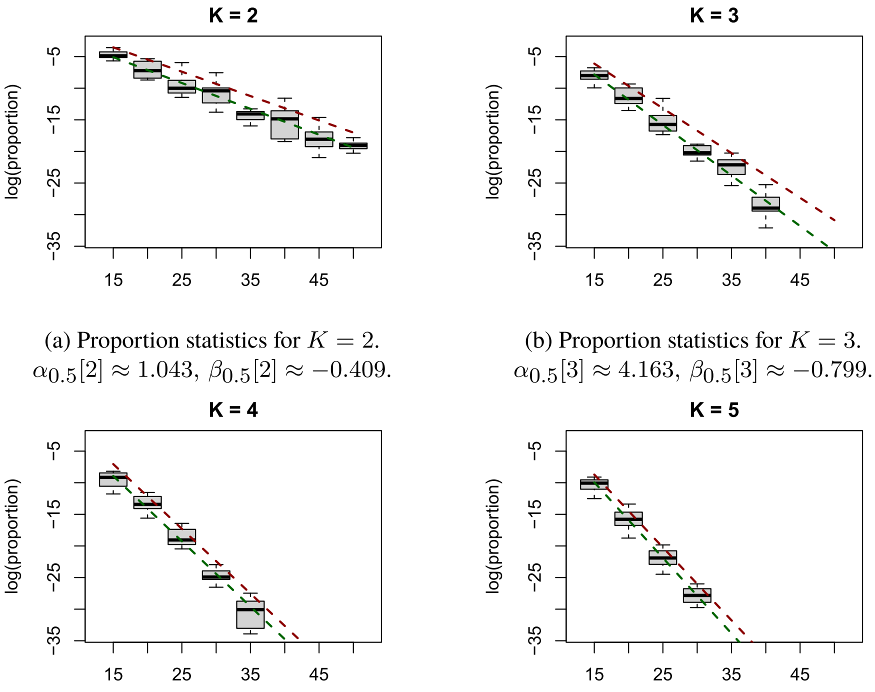
(c) Number of candidates to the global optima

N	3	4	5
20	33	33	358
21	22	35	455
22	72	58	780
23	38	108	1410
24	50	201	2688
25	43	203	2529

Continuing the experiment, we have halved the initial triangles 3 times, in total into 8 smaller triangles. The statistics are presented in Table 3. Comparing with Table 2, we can see that lower bound calculations take approximately 4 times more time, as expected – because we have 4 times more triangle-units. But as can be seen in Table 3c, the number of candidates to the global solution further decreases significantly.

5.5.2. *Predicted time*

Based on the observations of previous subsection, we decided to test our algorithm performance when initial triangle-units of $\mathcal{R}^{\cup}$ are subdivided 4 times into 16 smaller triangles. We have collected time statistics for various random problem instances. These problems



(a) Proportion statistics for $K = 2$. $\alpha_{0.5}[2] \approx 1.043$, $\beta_{0.5}[2] \approx -0.409$.

(b) Proportion statistics for $K = 3$. $\alpha_{0.5}[3] \approx 4.163$, $\beta_{0.5}[3] \approx -0.799$.

(c) Proportion statistics for $K = 4$. $\alpha_{0.5}[4] \approx 6.571$, $\beta_{0.5}[4] \approx -1.033$.

(d) Proportion statistics for $K = 5$. $\alpha_{0.5}[5] \approx 8.011$, $\beta_{0.5}[5] \approx -1.196$.

Fig. 7. Enumeration algorithm visited nodes log-proportion statistics. Number of clients N in the x-axis, number of clusters K in the title. Box-plots represent log-proportion statistics for a fixed N. Green line represents the linear model for the median log-proportion ($l_{0.5}(N, K) = \alpha_{0.5}[K] + \beta_{0.5}[K]N$), red – the 90th percentile. [12]

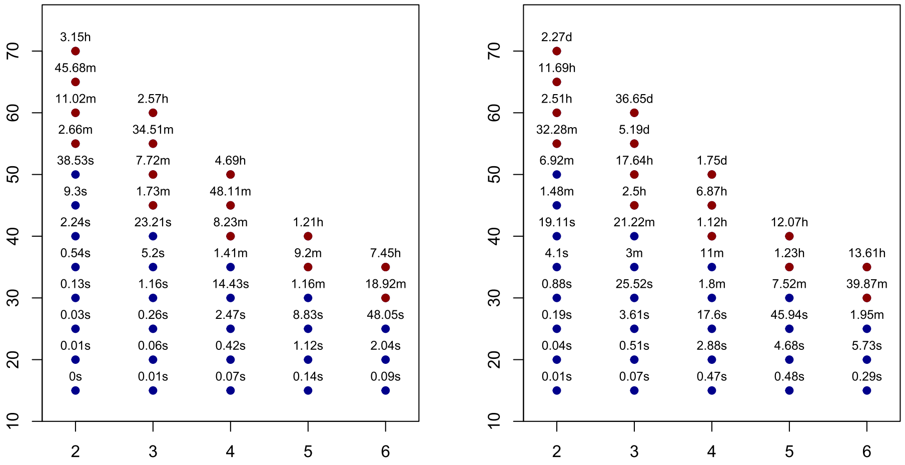
were all generated within a unit-square with a 10×10 grid, with each square initially divided into 4 pieces (in total, 400 problem-triangles). 10 barriers were then randomly generated, with a total of 100 triangles classified as barrier-triangles. The remaining 300 were subdivided into 16 smaller triangles, resulting that $\mathcal{R}^{\cup}$ consists of $M = 300 \times 16 = 4800$ triangle-units.

Algorithmic complexity of our approach is strongly related with the size of the tree $\mathcal{T}(N, K)$ of Fig. 1. Labeling with $S_2(N, K)$ the Stirling's number of the second kind, and defining $S_2(N, K) = 0$ when $K > N$, one can calculate that $\mathcal{T}(N, K)$ has

$$\mathcal{N}(N, K) = \sum_{n=1}^{N} \sum_{k=1}^{K} S_2(n, k) \tag{33}$$

nodes. Our algorithm performance depends on what proportion of the tree will be explored. During the experiment, we have recorded the number of tree-nodes visited in

[12] Models estimated using [R] quantreg package (Koenker, 2025).



(a) Predicted median time.

(b) Predicted 90th percentile time.

Fig. 8. Estimated time of the algorithm for various problems $[\mathbb{P}_1^N/K]\|[\mathcal{R}^\cup]$ with K in the x-axis and N in the y-axis. Blue points represent the pairs for which time statistics were collected, red points have no time statistics corresponding to them and are predicted. In the figure, the letter next to a number means "s"-seconds, "m"-minutes, "h"-hours, "d"-days (= 24 hours).

the enumeration of candidate solutions step (**Step 3** in the algorithm of Section 4). The proportion statistics are represented in Fig. 7. Using the obtained linear models for log-proportion, we can predict the median time and the 90th percentile time outside of experiment data. On our machine,[13] processing of a particular $v_n \in \mathcal{T}(N, K)$ using formula (17a) takes approximately $c = 9.098275 \times 10^{-6}$ seconds.[14] Thus, we can predict algorithm time using the formulas below:

$$T_{0.5}(N, K) = c \cdot \exp\big\{\alpha_{0.5}[K] + \beta_{0.5}[K]N\big\} \cdot \mathcal{N}(N, K), \tag{34a}$$

$$T_{0.9}(N, K) = c \cdot \exp\big\{\alpha_{0.9}[K] + \beta_{0.9}[K]N\big\} \cdot \mathcal{N}(N, K). \tag{34b}$$

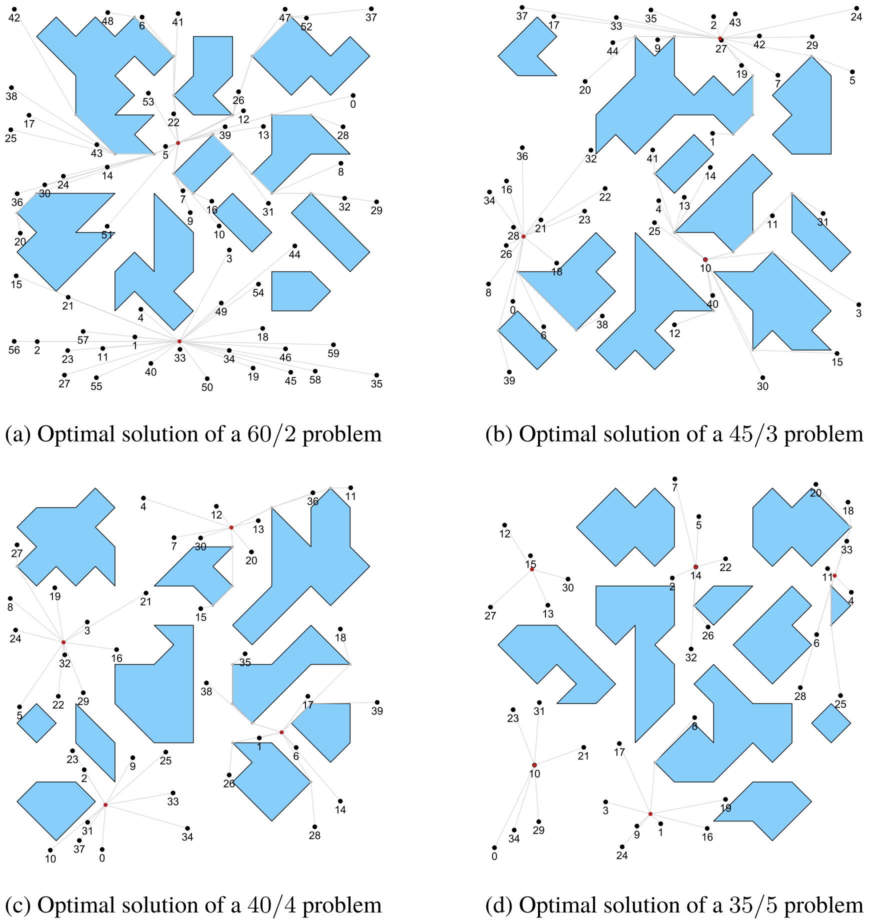
In the above equations, $\mathcal{N}(N, K)$ is the number of nodes in the tree as defined in (33). The predicted times for various combinations of (N, K) when the number of triangle-units is fixed to $M = 4800$ are visualized in Fig. 8.

5.6. *Solved Example Instances*

Some examples of problems solved within ϵ-accuracy are shown in Figs. 9 and 10. These are optimal solutions for largest problem instances we can solve in acceptable time. The results suggest themselves to be compared with the work of Rosing (1992), where the author had studied the **unconstrained** multi-Weber problem **without barriers**. This paper

[13] Processor 1.4 GHz Quad-Core Intel Core i5, Memory 8 GB 2133 MHz.

[14] The algorithmic complexity of formula (17a) is roughly M additions and M conditionals; in the experiment, M is always fixed to 4800 triangle-units.



(a) Optimal solution of a $60/2$ problem

(b) Optimal solution of a $45/3$ problem

(c) Optimal solution of a $40/4$ problem

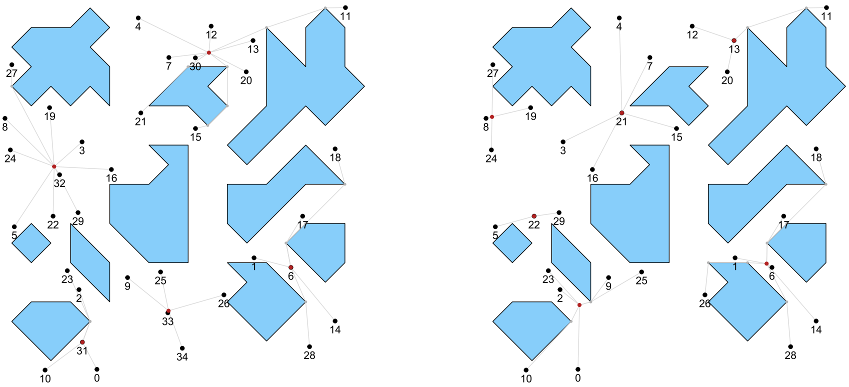
(d) Optimal solution of a $35/5$ problem

Fig. 9. Examples of within ϵ-accuracy solved problems ($\epsilon = 1\%$).

was published more than 30 years ago, but the results of their work are comparable to ours in the sense that, just like us (see Fig. 10), they reported global solutions for up to 35/5 and 30/6 problems. Our contribution is that we have managed to solve **constrained** multi-Weber problems **with barriers** of the same size, i.e. the ideas presented in this paper allow us to solve much more general class of uncapacitated multi-location problems.

6. Conclusions

In this paper, we have outlined ideas how the globally-optimal solution of a very general multi-locations problem can be approached. The problem of finding the global solution perhaps is more of theoretical and intellectual than practical interest. Also, the question is more of proving that the best found solution is the global optimum. In practice, if one



(a) Optimal solution of a $35/5$ problem

(b) Optimal solution of a $30/6$ problem

Fig. 10. More examples of within ϵ-accuracy solved problems ($\epsilon = 1\%$).

is faced with a barriers problem, a good-quality solution can be obtained by a location-allocation algorithm. The ideas of this paper can be easily extended for an implementation of such an algorithm.

We provided some numerical examples solving the multi-Weber with polyhedral barriers problem instances and predicted time for various sizes of problems. With our approach we have solved the problem instances of $60/2$, $45/3$, $40/4$, $35/5$ and $30/6$ (N/K means "N points into K clusters") to global optimality (within ϵ-accuracy with $\epsilon = 1\%$). To our knowledge, this is the first time the solution of instances with such a size to global optimality is reported.

Acknowledgements

Our solver for the multi-locations with barriers problem was implemented with java programming language on top of the algorithms and data-structures from Sedgewick and Wayne (2011), which are available at https://algs4.cs.princeton.edu/home/. In particular, most of the figures were generated using the *StdDraw.java* class, and we have reused their implementations of Dijkstra's algorithm (*DijkstraSP.java*) and various data-structures like MinPQ (*MinPQ.java*).

References

Aloise, D., Hansen, P., Liberti, L. (2012). An improved column generation algorithm for minimum sum-of-squares clustering. *Mathematical Programming*, 131, 195–220.

Aneja, Y.P., Parlar, M. (1994). Algorithms for Weber facility location in the presence of forbidden regions and/or barriers to travel. *Transportation Science*, 28(1), 70–76.

Bischoff, M., Klamroth, K. (2007). An efficient solution method for Weber problems with barriers based on genetic algorithms. *European Journal of Operational Research*, 177(1), 22–41.

Bischoff, M., Fleischmann, T., Klamroth, K. (2009). The multi-facility location–allocation problem with polyhedral barriers. *Computers & Operations Research*, 36(5), 1376–1392.

Cooper, L. (1964). Heuristic methods for location-allocation problems. *SIAM Review*, 6(1), 37–53.

Dijkstra, E.W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271.

Drezner, Z., Klamroth, K., Schöbel, A., Wesolowsky, G. (2002). The Weber Problem. *Facility Location: Applications and Theory*, 1–36.

du Merle, O., Villeneuve, D., Desrosiers, J., Hansen, P. (1999). Stabilized column generation. *Discrete Mathematics*, 194(1), 229–237.

Gilmore, P.C., Gomory, R.E. (1961). A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6), 849–859.

Hamacher, H.W., Nickel, S. (1995). Restricted planar location problems and applications. *Naval Research Logistics (NRL)*, 42(6), 967–992.

Kepalas, M., Žilinskas, J. (2024). Solving net-constrained clustering problem. *Journal of Nonlinear and Variational Analysis*, 8, 987–1012.

Klamroth, K. (2002). *Single Facility Location Problems with Barriers*. Springer, New York.

Koenker, R. (2025). quantreg: Quantile Regression.

Krau, S. (1997). *Extensions du problème de Weber*. PhD thesis, École Polytechnique de Montréal.

Oğuz, M., Bektaş, T., Bennell, J.A., Fliege, J. (2016). A modelling framework for solving restricted planar location problems using phi-objects. *The Journal of the Operational Research Society*, 67(8), 1080–1096.

Paulavičius, R., Žilinskas, J. (2013). *Simplicial Global Optimization*. Springer, New York.

Rosing, K.E. (1992). An optimal method for solving the (generalized) multi-Weber problem. *European Journal of Operational Research*, 58(3), 414–426.

Sedgewick, R., Wayne, K. (2011). *Algorithms*, 4th ed. Addison-Wesley Professional, Princeton.

Sumner, M. (2025). sfdct: Constrained Triangulation for Simple Features.

Viegas, J., Hansen, P. (1985). Finding shortest paths in the plane in the presence of barriers to travel (for any lp – norm). *European Journal of Operational Research*, 20(3), 373–381.

M. Kepalas is a PhD student at the Institute of Data Science and Digital Technologies, Vilnius University, Lithuania. His research interests are global optimization, location problems with constraints, net-constrained clustering, mathematical programming, optimization algorithms and applications.

J. Žilinskas is a research professor and a head of Global optimization group at the Institute of Data Science and Digital Technologies, Vilnius University, Lithuania. His research interests are global optimization, parallel computing, data analysis and visualization. He published around 100 international journal papers and 4 books. He is a member of editorial boards of 10 international journals. He is a vice president of the International Society of Global Optimization.