https://doi.org/10.15388/vu.thesis.898
https://orcid.org/0009-0006-2591-0838

VILNIUS UNIVERSITY

Mindaugas Kepalas

Searching for the Global Solution of Constrained Location Problems

DOCTORAL DISSERTATION

Natural Sciences,
Informatics (N 009)

VILNIUS, 2026

The dissertation was prepared between 2020 and 2025 at Vilnius University.

Academic supervisor – Prof. Dr. Julius Žilinskas (Vilnius University, Natural Sciences, Informatics – N 009).

Dissertation Defence Panel:

Chairman – Prof. Dr. Olga Kurasova (Vilnius University, Natural Sciences, Informatics – N 009).
Members:
Assoc. Prof. Dr. Algirdas Lančinskas (Vilnius University, Natural Sciences, Informatics – N 009),
Prof. Dr. Audris Mockus (University of Tennessee, USA, Natural Sciences, Informatics – N 009),
Prof. Dr. Remigijus Paulavičius (Vilnius University, Natural Sciences, Informatics – N 009),
Prof. Dr. Dmitrij Šešok (Vilnius Gediminas Technical University, Natural Sciences, Informatics – N 009).

The dissertation will be defended at a public meeting of the Dissertation Defense Panel at 12 noon on 30 March 2026 in Room 203 at the Institute of Data Science and Digital Technologies of Vilnius University.
Address: Akademijos g. 4, LT-04812, Vilnius, Lithuania.
Tel. +370 5 210 9300; e-mail: info@mii.vu.lt

https://doi.org/10.15388/vu.thesis.898
https://orcid.org/0009-0006-2591-0838

VILNIAUS UNIVERSITETAS

Mindaugas Kepalas

Globaliojo sprendinio paieška vietos parinkimo uždaviniams su apribojimais

DAKTARO DISERTACIJA

Gamtos mokslai,
Informatika (N 009)

VILNIUS, 2026

Disertacija rengta 2020–2025 metais Vilniaus universitete.

Mokslinis vadovas – prof. dr. Julius Žilinskas (Vilniaus universitetas, gamtos mokslai, informatika – N 009).

Gynimo taryba:

Pirmininkė – prof. dr. Olga Kurasova (Vilniaus universitetas, gamtos mokslai, informatika – N 009).
Nariai:
doc. dr. Algirdas Lančinskas (Vilniaus universitetas, gamtos mokslai, informatika – N 009),
prof. dr. Audris Mockus (Tenesio universitetas, JAV, gamtos mokslai, informatika – N 009),
prof. dr. Remigijus Paulavičius (Vilniaus universitetas, gamtos mokslai, informatika – N 009),
prof. dr. Dmitrij Šešok (Vilniaus Gedimino technikos universitetas, gamtos mokslai, informatika – N 009).

Disertacija ginama viešame Gynimo tarybos posėdyje 2026 m. kovo 30 d. 12 val. Vilniaus universiteto Duomenų mokslo ir skaitmeninių technologijų instituto 203 auditorijoje.
Adresas: Akademijos g. 4, LT-04812, Vilnius, Lietuva.
Tel. +370 5 210 9300; el. paštas: info@mii.vu.lt

ABSTRACT

This dissertation investigates several constrained multi-location problems.

The first model goes under the title "the constrained equal area Voronoi tessellation". The goal is to place a number of points, called sites, in such a way that the cells, resulting from the Voronoi diagram when intersected with some predefined polygon, have equal areas. Site locations are restricted to belong to a "net", i.e., a union of a set of segments, thus the name "constrained". We present a local mixed integer programming formulation and an algorithm for the problem.

The model was inspired by the load balancing problem coming from building engineering, where a building is supported on the ground by grillage foundations, which are called piles. Thus the frame (the "net") of building foundation contour defines the constraints for the pile points, and the goal is to distribute the piles in such a way so that they support equal loads.

This dissertation also investigates continuous constrained multi-facility problems. Here, we have to open a certain number of facility points within a constrained area of the plane, so that the given clients, which are assumed to choose the closest opened facility, can be serviced at the least possible cost. We study three different ways to measure the distances. In the first scenario, the distance from a client point to a facility point is measured by the squared Euclidean norm, resulting in a type of a minimum-sum-of-squares clustering problem. In the second, the distance is measured by the standard (non-squared) Euclidean norm, giving a constrained multi-Weber problem case. The third case extends the second model and additionally introduces barriers, where traveling is prohibited. For this model, even single optimal location determination becomes challenging.

The thesis addresses the considered continuous constrained multi-facility problems with the forementioned different distance metrics under a unified framework. In particular, we discuss how a k-means-type algorithm can be applied in order to find a locally optimal solution. Also, we propose a general enumeration algorithm with several pruning criteria which could be used for finding the globally optimal solution. With this algorithm, we managed to solve some problem instances which to our knowledge have not been solved elsewhere before.

LP	Linear Programming.
MIP	Mixed Integer Programming.
MIQCP	Mixed Integer Quadratically Constrained Programming.
MSS	Minimum Sum of Squares.
MSSC	Minimum Sum of Squares Clustering.
SDP	Semi-Definite Programming.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

Research Problem

This PhD thesis could be summarized under the title *constrained planar multi-location problems*. In these models, optimization variables are the coordinates of a predefined number of points in the plane, and the goal is to put these points in such a way that the cost of the placement is minimal. These locations must belong to a predefined constrained subset of the plane. In the thesis, we assume that this constrained set is a union of finite convex simplexes (i.e., in the plane, these are segments and convex polygons). The union operation on this set might (and most probably, will) result in a non-convex constrained set for point locations. This non-convexity makes the considered problems challenging (and interesting).

Research Objects

The objects of the research are constrained planar multi-location problems and algorithms to solve them. We emphasize the three main models investigated in this dissertation.

Constrained equal area Voronoi tessellation. The goal here is to place a number of points, called sites, in such a way that all the resulting Voronoi cells have equal areas. "Constrained" means that the sites are required to satisfy the location constraints.

Net-constrained Minimum Sum of Squares Clustering (MSSC) problem. Here, we seek to put the given points into clusters in such a way that the sum of weighted squared distances between them and cluster centers is minimal. Problem variables are cluster center points, which are required to belong to the constrained set.

Multi-Weber problem with polyhedral barriers. For this model, polyhedral barriers are introduced where traveling is prohibited. The goal is similar as for the problem in the previous item: we have to optimize the locations of the "facilities" so that the cost of servicing the "clients" is minimal.

Research Area

In this thesis, we use the results of and contribute to several optimization fields. First of all, our main contribution is probably to the field of global optimization, i.e., the initial aim of the research was to find the global solutions of the constrained planar multi-location problem studied in this dissertation. To achieve this goal, in our work we have used various tools of mathematical programming, e.g., solvers for Linear Programming (LP) or Mixed Integer Programming (MIP) problems. Computational geometry algorithms and tools also have played a significant role in the research.

Talking about the three problems of study listed in the previous section, we believe that the *constrained equal area Voronoi tessellation* is an interesting computational geometry problem, which could be used to model the pile balancing problem arising in building engineering. The *net-constrained Minimum Sum of Squares Clustering (MSSC) problem* – as the name suggests – can be designated to the field of cluster analysis. Lastly, our work on the *multi-Weber problem with polyhedral barriers* contributes to the field of continuous facility location optimization and, in particular, to the generalized multi-Weber problem.

Research Actuality and Practical Significance

Theoretical scientists like to say that all the theories are significant, but not for all of them practical applications have yet been discovered. Our research perhaps falls into this category, and we must admit that the practical value of our work is disputable. Nevertheless (we now use a "proof by authority"), the research fields our problems come from (like MSSC, facility location or mathematical modeling of engineering problems) have resulted in numerous scientific publications and books; thus, some value probably exists.

Referring to the particular problems studied in this research, the *constrained equal area Voronoi tessellation* seems not to have been considered in scientific literature before, but we hope that the mentioned application to the pile balancing problem in building engineering gives a sufficient motivation for its study.

The *net-constrained MSSC problem* is also newly introduced and could be claimed to be rather "exotic" in a sense that we consider the

sum of squared Euclidean distances $\|\bullet\|_2^2$ optimization, which perhaps makes little sense for clustering problems in the plane. Our initial choice of such distance measure was determined by the well-developed field of minimum sum of squares problems; perhaps our choice was not a good one. Thus, we emphasize the extension to the standard Euclidean norm $\|\bullet\|_2$ in the thesis, and under this redefinition, the problem becomes a variant of a generalized multi-Weber problem. A practical interpretation of such a model could be as follows: consider that a certain resource (say, electricity) is provided on a network. The demand of the resource is given at certain locations, which are outside this network. The resource can be delivered by some "gear" (say, a drone) to the client; this gear can get the resource only at a "station", which is constrained to be on the network. The goal is to open a predefined number of resource stations which minimize the cost of the service.

The *multi-Weber problem with polyhedral barriers* can be regarded as a variant of the standard multi-Weber problem, when the distance is influenced by some obstacles where traveling is prohibited. Such obstacles have a natural interpretation, e.g., they could represent buildings, lakes, mountains, swamps, forests, etc. In particular, introducing barriers can bring an abstract mathematical model closer to a real life, and we believe this gives a good motivation for the study of this problem.

Research Aim and Objectives

The goal of this research is to propose or to improve algorithms for solving constrained planar multi-location problems. To reach this aim the following objectives are set:

1. To investigate scientific literature to find out which problems can be seen as instances of the constrained planar multi-location problem formulation.

2. To investigate algorithms used to obtain the (global) solution of the corresponding unconstrained location problems.

3. To propose modifications and generalizations of the optimization algorithms to the constrained cases.

4. To investigate the proposed versions of the algorithms to find out their advantages and limitations.

For the author of this thesis, the search for the answer to the first objective has led to the study of the 3 already introduced problems: *the constrained equal area Voronoi tessellation*, *the net-constrained MSSC problem* and *the multi-Weber problem with polyhedral barriers*. The remaining three objectives determine the contents of this thesis.

Scientific Novelty

- *The constrained equal area Voronoi tessellation problem* is new and seems not to have been studied elsewhere before. We proposed an algorithm for the problem; this algorithm is targeted at the globally optimal solution.
- *The net-constrained MSSC problem* is new and seems not to have been studied elsewhere before. We applied the location-allocation type algorithm to obtain a locally optimal solution for the problem. Also, we have proposed an algorithm for finding the globally optimal solution.
- Our work on *the multi-Weber problem with polyhedral barriers* complements to the research on this interesting model. To our knowledge, in our work we report for the first time the global solutions of problems with sizes up to 60 points into two clusters, 45 points into three, 40 points into four, and 35 points into five clusters.

Statements to be Defended

1. The proposed algorithm based on solving a sequence of mixed-integer-programming problems obtains a high-quality solution of *the constrained equal area Voronoi tessellation problem*.
2. The proposed enumeration algorithm with the convex hull cut criterion solves *net-constrained MSSC problems* of moderate size to global optimality. The proposed k-means type local search algorithm efficiently obtains a locally optimal solution of larger instances.
3. The proposed branch-and-cut algorithm finds the globally optimal solution to *multi-Weber with polyhedral barriers problems* of larger sizes than it was previously published in the literature.

Approbation and Publications of the Research

Publications. The results of the dissertation were published in

1. Mindaugas Kepalas and Julius Žilinskas. "Solving Net-Constrained Clustering Problem". In: *Journal of Nonlinear and Variational Analysis* (2024), pp. 987–1012.

2. Mindaugas Kepalas and Julius Žilinskas. "Global Optimization Algorithm for the Multi-Weber Problem with Polyhedral Barriers". In: *Informatica* (2025), pp. 875–902.

These articles relate directly to our research problems: *the net-constrained MSSC problem* and *the multi-Weber problem with polyhedral barriers*.

Conferences. Our work was also presented at 7 conferences.
International:

- Mindaugas Kepalas, Julius Žilinskas. "Optimizing Locations on a Network". EURO 2022, 3rd – 6th of July, 2022, Helsinki, Finland.

- Mindaugas Kepalas, Julius Žilinskas. "Facility Locations on a Network". Europt 2022, 29th – 30th of July, 2022, Lisbon, Portugal.

- Mindaugas Kepalas, Julius Žilinskas. "2-Dimensional Net Constrained Clustering Problem". The 2023 World Congress on Global Optimization (WCGO 2023), 10th – 14th of July, 2023, Athens, Greece.

National:

- Mindaugas Kepalas, Julius Žilinskas. "Locations on Networks". DAMSS 2021, 2nd – 4th of December, 2021, Druskininkai, Lithuania.

- Mindaugas Kepalas, Julius Žilinskas. "Handling Two-Dimensional Net Constraint: Its Formulation in MIP Terms and Empirical Studies of Solvers". DAMSS 2022, 1st – 3rd of December, 2023, Druskininkai, Lithuania.

- Mindaugas Kepalas, Julius Žilinskas. “Minimum-Sum-Of-Squares Clustering With (Net) Constraints for Cluster-Centres”. DAMSS 2023, 30th of November – 2nd of December, 2023, Druskininkai, Lithuania.

- Mindaugas Kepalas. “Keletas optimalaus taškų paskirstymo apribotoje plokštumos dalyje uždavinių”. Lietuvos Jaunųjų Matematikų Susitikimas (LJMS). 29th of December, 2025, Vilnius, Lithuania.

Outline of the Thesis

The main text of the thesis is presented in the following chapters. Chapter 1 gives the mathematical formulations of the problems of study and presents their visual illustrations. Chapter 2 reviews the literature which was studied during the research. Chapter 3 is dedicated to the equal area Voronoi tessellation. In Chapter 4, we study single optimal facility location problems. An algorithm to obtain (prove) a globally optimal solution for the multi-location problems is presented in Chapter 5. Finally, general conclusions are given, references are listed, and the thesis is summarized in Lithuanian.

1. MATHEMATICAL FORMULATIONS AND ILLUSTRATIONS OF DISSERTATION PROBLEMS

In this thesis, the set of the problems we consider falls into the following mathematical formulation:

$$\begin{aligned} \min_{P_1,\dots,P_K} \quad & \textbf{cost}(P_1,\dots,P_K) \\ \text{s.t.} \quad & P_k \in \mathcal{R}^{\cup}, \quad k = 1,\dots,K. \end{aligned} \tag{1.1}$$

In other words, we seek to place K points in such a way that the objective function (**cost**) is optimized (minimized). Problem variables are the coordinates of $P_1,\dots,P_K$; these points are restricted to a subset $\mathcal{R}^{\cup} \subset \mathbb{R}^2$ ("$\mathcal{R}$egion $\cup$nion"), which is assumed to be a union of finite convex simplexes in the plane (i.e., segments and convex polygons):[1]

$$\mathcal{R}^{\cup} := \cup_{m=1}^{M} \mathcal{R}_m, \quad \text{here } \mathcal{R}_m \text{ is a convex simplex } \forall m.$$

In case $\mathcal{R}^{\cup}$ consists only of segments, we also use notation

$$\mathcal{N} := \cup_{m=1}^{M} \mathcal{S}_m \;\equiv\; \mathcal{R}^{\cup} := \cup_{m=1}^{M} \mathcal{R}^{\cup},$$

i.e., "$\mathcal{R}$egion $\cup$nion" is a "$\mathcal{N}$et". Under this definition, $\mathcal{R}^{\cup}$ (or $\mathcal{N}$) may not be (and, most probably, will not be) a convex set itself, i.e., taking two points $P_i, P_j \in \mathcal{R}^{\cup}$, the segment

$$\{tP_i + (1-t)P_j,\, t \in [0,1]\}$$

might not be fully contained in $\mathcal{R}^{\cup}$ (or $\mathcal{N}$). This non-convexity makes problem (1.1) challenging, since convexity is often seen as the main criterion which separates easily solvable problems (i.e., with well developed theory and efficient algorithms) from the complicated ones (Boyd and Vandenberghe [10]).

1.1. Constrained Set

Here we discuss the issue which is common for all the problems considered in this thesis: the constrained set $\mathcal{R}^{\cup}$. We concentrate on the

[1] Although circles, curves, non-convex polygons or other shapes are not considered in our definition of $\mathcal{R}^{\cup}$, this is not as restrictive as might look at first sight, because any curve can be arbitrary well approximated by a set of segments, or any area in the plane can be triangulated and approximated by a union of a set of triangles.

formulation of the constraint $P \in \mathcal{N}$ in MIP terms.[2] The formulation for $P \in \mathcal{R}^{\cup}$ is rather analogous and is skipped here for brevity.

Consider some $\mathcal{R}_m,\ m \in \{1, \ldots, M\}$ which is a segment (in this case we also use notation $\mathcal{S}_m \equiv \mathcal{R}_m$). Then it can be represented as a convex combinations of a pair of points $Q_m^A,\ Q_m^B$:

$$\mathcal{S}_m = \left\{ Q_m^A + \theta_m (Q_m^B - Q_m^A),\ 0 \le \theta_m \le 1 \right\}.$$

Now consider some point $P \in \mathcal{N}$ (with $\mathcal{N} \equiv \cup_{m=1}^{M} \mathcal{S}_m$). Clearly, P must belong to at least one of the segments $\mathcal{S}_m$, i.e., the following logical expression is true (equal to 1):

$$(P \in \mathcal{S}_1) \vee (P \in \mathcal{S}_2) \vee \cdots \vee (P \in \mathcal{S}_M). \tag{1.2}$$

If we introduce a binary indicator variable e_m for each condition $(P \in \mathcal{S}_m)$, requirement (1.2) could be written as $\sum_{m=1}^{M} e_m \ge 1$; however, it is enough to require that $\sum_{m=1}^{M} e_m = 1$ (if a point belongs to multiple $\mathcal{S}_m$, choose any of them arbitrary).

Now suppose $P \in \mathcal{S}_{m_1}$, i.e., $e_{m_1} = 1$ and $e_m = 0,\ m \ne m_1$. Then point P can be written as

$$P = Q_{m_1}^A + \theta_{m_1} (Q_{m_1}^B - Q_{m_1}^A)$$

for some $\theta_{m_1} \in [0, 1]$. Since $e_m = 0,\ m \ne m_1$, we also have

$$\begin{aligned} P &= \sum_{m=1}^{M} e_m \left[Q_m^A + \theta_m (Q_m^B - Q_m^A) \right] \\ &= \left[\sum_{m=1}^{M} e_m Q_m^A \right] + \left[\sum_{m=1}^{M} e_m \theta_m (Q_m^B - Q_m^A) \right]. \end{aligned}$$

Label $\varphi_m := e_m \theta_m$ in the above equation. If we add a requirement that $0 \le \varphi_m \le e_m$, noting that we have $e_m \in \{0, 1\}$ and $\sum_{m=1}^{M} e_m = 1$, we can see that only variable φ_{m_1} can be non-zero. Thus, the constraint $P \in \mathcal{N}$ can be parametrically represented as follows:

$$P = \left[\sum_{m=1}^{M} e_m Q_m^A \right] + \left[\sum_{m=1}^{M} \varphi_m (Q_m^B - Q_m^A) \right], \tag{1.3a}$$

$$\text{s.t.} \quad \sum_{m=1}^{M} e_m = 1, \tag{1.3b}$$

$$e_m \in \{0, 1\} \ \text{ and } \ 0 \le \varphi_m \le e_m, \quad m = 1, \ldots, M. \tag{1.3c}$$

[2] Mixed Integer Programming.

The above formulation can be seen as a MIP-type constraint: it can be represented by linear constraints with some binary variables. Thus, if in some way we can formulate our initial (unconstrained) location problem as a Linear Programming (LP) problem, adding the constraint $P \in \mathcal{N}$ via formulation (1.3) "only" changes it to a Mixed Integer Programming (MIP) problem. (The same result holds if the constraint is $P \in \mathcal{R}^{\cup}$.)

1.2. Main Problems

In this thesis, we concentrate on the following 3 problems which are instances of the formulation in (1.1).

Constrained equal area Voronoi tessellation.

$$\min_{P_1,\ldots,P_K} \quad \frac{1}{K}\sum_{k=1}^{K}\left|\mathbf{area}(\mathcal{V}_k \cap \mathcal{P}) - \frac{\mathbf{area}(\mathcal{P})}{K}\right| \tag{1.4a}$$

$$\text{s.t.} \quad P_k \in \mathcal{N}, \quad k = 1,\ldots,K. \tag{1.4b}$$

In the equation above,

- $\mathbf{area}(\bullet)$ is the area of the corresponding polygon,
- $\mathcal{N}$ is a union of segments in plane,
- $\mathcal{V}_k$ is the cell of the Voronoi tessellation corresponding to point P_k,
- $\mathcal{P}$ is some simple (non-self-intersecting) polygon.

In other words, we seek to place K points $P_1,\ldots,P_K$ in such a way, keeping to the constraint $P_k \in \mathcal{N}$, that all Voronoi cells have as equal areas as possible. See an illustration in Figure 1.1.

The presented problem (1.4) is new in a sense that introducing the net constraint (1.4b) for the sites seems to have not been considered in scientific literature before. For a method to solve the equal area Voronoi tessellation problem without the constraint (1.4b), see, e.g., Birgin et al. [6].

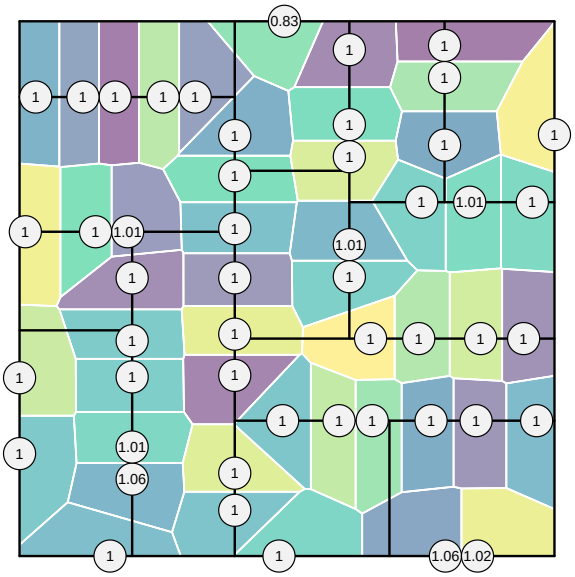


Figure 1.1. Constrained equal area Voronoi tessellation problem. The numbers within circles indicate the area of the corresponding cell.

Net-constrained Minimum Sum of Squares Clustering (MSSC).

$$\min_{P_1,\dots,P_K} \quad \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} w_i \left\| A_i - P_k \right\|_2^2 \tag{1.5a}$$

$$\text{s.t.} \quad P_k \in \mathcal{N}, \quad k = 1, \dots, K. \tag{1.5b}$$

In the equations above,

- $\mathcal{N}$ is a union of segments in the plane,
- (A_n, w_n), $n = 1, \dots, N$ is a set of weighted points in the plane ("clients" at locations A_n with "demands" w_n),
- $\mathcal{C}_k$, $k = 1, \dots, K$ is a partition of the client index set $\{1, \dots, N\}$ (satisfying $\mathcal{C}_k \cap \mathcal{C}_l = \emptyset$ for $l \neq k$ and $\cup_{k=1}^{K} \mathcal{C}_k = \{1, \dots, N\}$),
- $\|\bullet\|_2^2$ is the squared Euclidean norm.

In other words, we seek to "open" K "facilities" $P_1, \dots, P_K$ within $\mathcal{N}$ in such a way that the sum of weighted squared distances from the clients to the closest facilities is minimal. See an illustration in Figure 1.2.

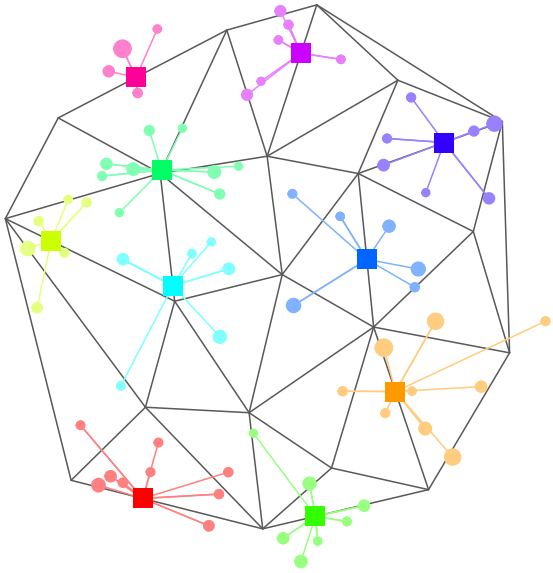


Figure 1.2. Net-constrained MSSC problem. Note that all the "facilities" P_k (represented by squares) satisfy the constraint $P_k \in \mathcal{N}$.

For the Minimum Sum of Squares Clustering (MSSC) problems, introducing the net constraint (1.5b) for cluster centers seems to have been considered for the first time in our paper [A.1]. The classical MSSC problem without the constraint (1.5b) has numerous references, see, e.g., MacQueen [36] for the famous k-means algorithm, which efficiently determines a local solution, or Aloise et al. [1] for a global algorithm employing the column generation principle of Gilmore and Gomory [24].

Multi-Weber problem with polyhedral barriers. In this problem, polyhedral barriers $\{\mathcal{B}_1, \ldots, \mathcal{B}_L\}$ are introduced where traveling is prohibited (Aneja and Parlar [2]). The mathematical formulation is as follows:

$$\min_{P_1,\ldots,P_K} \quad \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} w_i d_{\mathcal{B}}\left(A_i, P_k\right) \tag{1.6a}$$

$$\text{s.t.} \quad P_k \in \mathcal{R}^{\cup}, \quad k = 1, \ldots, K. \tag{1.6b}$$

In the equation above,

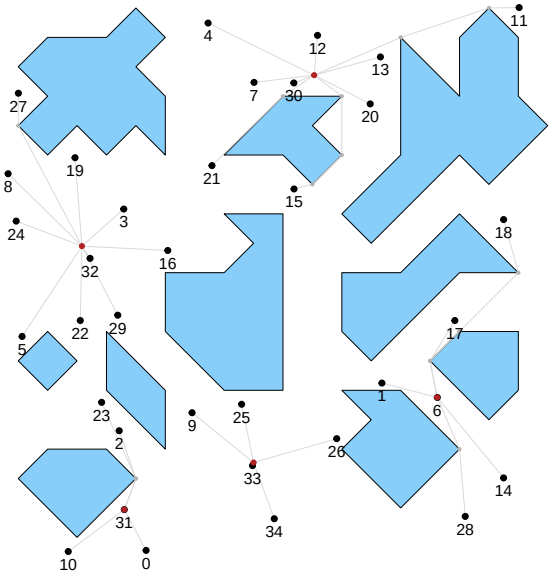


Figure 1.3. Multi-Weber problem with polyhedral barriers: "clients" (black points) assigned to "facilities" (red dots).

- $\mathcal{R}^{\cup}$ is equal to $\mathbb{R}^2 \setminus \cup_{l=1}^{L} \mathcal{B}_l$,
- (A_n, w_n), $n = 1, \ldots, N$ is a set of weighted points in $\mathcal{R}^{\cup}$ ("clients" at locations A_n with "demands" w_n),
- $\mathcal{C}_k$, $k = 1, \ldots, K$ is a partition of the client index set $\{1, \ldots, N\}$ (satisfying $\mathcal{C}_k \cap \mathcal{C}_l = \emptyset$ for $l \neq k$ and $\cup_{k=1}^{K} \mathcal{C}_k = \{1, \ldots, N\}$),
- $d_{\mathcal{B}}(\bullet, \bullet)$ is the length of the shortest path between two corresponding points taking the barriers into account.

See problem illustration in Figure 1.3.

Using our branch-and-cut enumeration algorithm in [A.2], we solved to global optimality problems of sizes[3] $60/2$, $45/3$, $40/4$, $35/5$ for the first time (to our knowledge). This algorithm is the subject of Chapter 5; for a discussion on other methods proposed in the literature for the global and the local solution of the problem, see Section 2.5.2.

[3]We use notation N/K to refer to a problem of clustering N points into K clusters.

2. LITERATURE REVIEW

In this chapter, we review some literature related to the problems studied in this dissertation, i.e., to the problems of the form

$$\begin{aligned} \min_{P_1,\dots,P_K} \quad & \textbf{cost}(P_1,\dots,P_K) \\ \text{s.t.} \quad & P_k \in \mathcal{R}^{\cup} \text{ (or } P_k \in \mathcal{N}), \quad k = 1,\dots,K. \end{aligned} \tag{2.1}$$

Here, as before, **cost** is some cost function associated with the placement of K points $P_1,\dots,P_K$, and $\mathcal{R}^{\cup}$ is a union of convex simplexes in the plane (in case all of them are segments, we use notation $\mathcal{N}$), which defines the constrained set for $P_1,\dots,P_K$.

2.1. Pile-Balancing Problem

This PhD research started with the problem which is illustrated in Figure 2.1. In the figure, you can see a sketch of a building foundation contour, which is supported on the ground by points at locations $P_1,\dots,P_K$, called piles. The objective is to find such a placement of the piles so that all of them support equal loads. The problem can be stated this way:

$$\min_{P_1,\dots,P_K} \quad \max_{k=1,\dots,K} F_k(P_1,\dots,P_K) \tag{2.2a}$$

$$\text{s.t.} \quad P_k \in \mathcal{N}, \quad k = 1,\dots,K, \tag{2.2b}$$

$$\|P_k - P_l\|_2 \geq 2\varepsilon, \quad k \neq l. \tag{2.2c}$$

In the objective function (2.2a), $F_k(P_1,\dots,P_K)$ is the load at position P_k, and the goal is to minimize the maximal load. The last condition in (2.2c) requires that the piles are sufficiently well separated, i.e., if we draw around each pile a circle of radius ε, these circles must not overlap.

From the scientific articles on the problem, [4, 46, 47, 54], one could make a conclusion that the best methods for the problem are based either on simulated annealing or on genetic algorithms (see the related materials in Kochenderfer and Wheeler [31] for an introduction of these methods). Such a conclusion would be only partly true. This result is probably due the fact that the black-box objective function implementation of the model in [4, 46, 47, 54] is discontinuous. In particular, instead of solving the problem formulated in (2.2), the authors mapped each pile onto a point $x_k := x(P_k)$ in the interval $[0, L]$, where L is the total length

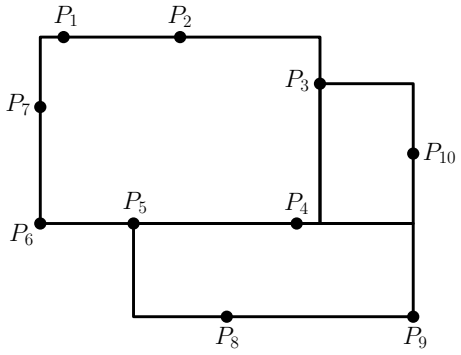


Figure 2.1. Supporting piles problem.

of the segments constituting a building foundation contour, resulting in the following problem formulation:

$$\begin{aligned} \min_{x_1,\ldots,x_K} \quad & \max_{k=1,\ldots,K} F_k(x_1,\ldots,x_K) \\ \text{s.t.} \quad & 0 \le x_k \le L, \quad k = 1,\ldots,K, \\ & x_k - x_l \ge 2\varepsilon, \quad k \ne l. \end{aligned} \tag{2.3}$$

However, stated this way, a pile represented by $y_1 - \varepsilon$ is physically far from a pile represented by $y_1 + \varepsilon$, which is not a desired feature and a significant drawback (see Figure 2.2). Intuitively, the objective function in (2.2) can be expected to behave "nicely" in a sense that moving a pile "slightly" will only "slightly" change the loads F_k. This is not the case for the positions in the neighborhood of y_1, y_2 in Figure 2.2 for the formulation in (2.3). Basically, the willingness to solve the problem in (2.2) as a continuous one has turned this research into the study of net-constrained planar location problems.

Geometrical Model. Lacking the possibility to modify the objective function in (2.3) and seeking to give the problem more geometrical interpretation, we came up with the constrained equal area Voronoi tessellation model which is the subject of Chapter 3. This model seems to be new and to our knowledge has not been studied in scientific literature before. However, the unconstrained case, i.e., the equal area Voronoi tessellation without site constraints has received some scientific attention, see, e.g., Balzer [3] or Birgin et al. [6]. However, the method presented in Chapter 3 of this thesis for the (unconstrained) equal area

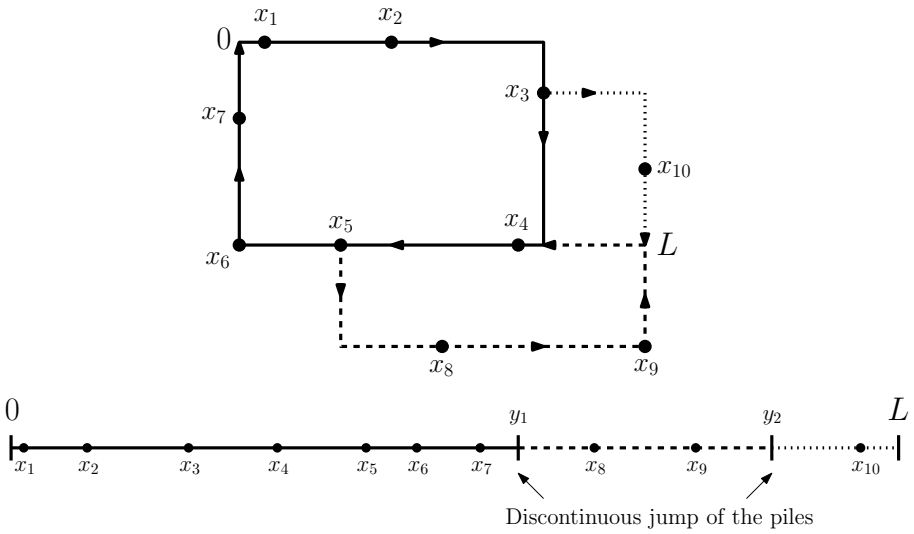


Figure 2.2. Illustration of the discontinuity in the objective function.

Voronoi tessellation was developed independently.

2.2. K-Medians Problem

Naturally, $\mathcal{N}$ can be understood as a planar embedding of a non-directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with vertex set $\mathcal{V}$ consisting of the endpoints of segments in $\mathcal{N}$, and edge set $\mathcal{E}$ consisting of the pairs of points representing each segment.

Hakimi [25] proposes the following model: for network $\mathcal{N}$ and its corresponding graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in addition define non-negative weights for vertices and edges, i.e., two functions $w_{\mathcal{V}} : \mathcal{V} \mapsto \mathbb{R}^+$, $w_{\mathcal{E}} : \mathcal{E} \mapsto \mathbb{R}^+$.[1] Consider a pair of points $x, y \in \mathcal{N}$. Distance $d(x, y)$ is defined as the minimal time required to reach $y \in \mathcal{N}$ from $x \in \mathcal{N}$ (which can be found with the shortest path algorithm of Dijkstra [16]).

Now consider some set $\mathcal{X} := \{x_1, x_2, \ldots, x_K\}$ of exactly K facilities placed on $\mathcal{N}$. For any point $y \in \mathcal{N}$, define

$$d(y, \mathcal{X}) := \min_{k=1,\ldots,K} d(y, x_k).$$

[1] The weight $w_{\mathcal{V}}(v)$, $v \in \mathcal{V}$ might be interpreted as the size of "city" v, and $w_{\mathcal{E}}(e)$, $e \in \mathcal{E}$ might be interpreted as the time required to travel along $e = (v, w)$, i.e., time it takes to reach w from v along "road" e.

The cost of the placement is defined by the equation below:

$$\mathbf{cost}(\mathcal{X}) := \sum_{v \in \mathcal{V}} w_{\mathcal{V}}(v) d(v, \mathcal{X}).$$

Our goal is to find the minimal cost:

$$\mathbf{cost}^* := \min_{\mathcal{X} \subset \mathcal{N},\, |\mathcal{X}|=K} \mathbf{cost}(\mathcal{X}).$$

Hakimi [25] proves this result: to find the optimal value $\mathbf{cost}^*$, it is sufficient to check only facility placements at the vertices, i.e.:

$$\min_{\mathcal{X} \subset \mathcal{N},\, |\mathcal{X}|=K} \mathbf{cost}(\mathcal{X}) = \min_{\mathcal{X} \subset \mathcal{V},\, |\mathcal{X}|=K} \mathbf{cost}(\mathcal{X}).$$

This result simplifies the search space to a discrete set and puts this problem in a class of discrete facility location problems. That is, precalculate the distances $d_{ij} := d(v_i, v_j)$ between pairs of vertices. Then, the problem can be stated as the following MIP problem:

$$\begin{aligned}
\min_{\{z_i\},\{x_{ij}\}} \quad & \sum_{i=1}^{|\mathcal{V}|} \sum_{j=1}^{|\mathcal{V}|} w_{\mathcal{V}}(v_j) d_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{i=1}^{|\mathcal{V}|} x_{ij} = 1, \quad j = 1, \ldots, |\mathcal{V}|, \\
& x_{ij} \le z_i, \quad i, j = 1, \ldots, |\mathcal{V}|, \\
& \sum_{i=1}^{|\mathcal{V}|} z_i = K \quad \text{and} \quad z_i \in \{0, 1\}, \quad i = 1, \ldots, |\mathcal{V}|.
\end{aligned}$$

This somewhat makes the problem well-studied; i.e., to find a viable method for the K-medians problem, one should concentrate on discrete facility location optimization algorithms. This was not in the objectives of this thesis and we mention this problem here only because it falls into the formulation in (2.1).

2.3. Net-Constrained Multi-Location Problem

What happens if we put the clients off the net, but keep the net constraint for the facilities? Such formulation results in a net-constrained multi-location problem: for a given weighted set of points

$$(A_1, w_1), \ldots, (A_N, w_N),$$

find K locations $P_1, \ldots, P_K$ restricted to the net so that the weighted sum of distances is minimal:

$$\min_{(P_1,\mathcal{C}_1),\ldots,(P_K,\mathcal{C}_K)} \quad \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} w_i d(A_i, P_k) \tag{2.5a}$$

$$\text{s.t.} \quad P_k \in \mathcal{N}, \quad k = 1, \ldots, K. \tag{2.5b}$$

2.4. Net-Constrained MSS Clustering

The problem in (2.5) is ambiguous in a sense that distance metric d is not precisely defined. Our initial decision was to measure the distance by the squared Euclidean norm, $d(A_i, P_k) = \|A_i - P_k\|_2^2$. The choice to define the metric this way was motivated by the well-developed theory of Minimum Sum of Squares (MSS) problems. For this metric, the problem in (2.5) gets the following form:

$$\min_{(P_1,\mathcal{C}_1),\ldots,(P_K,\mathcal{C}_K)} \quad \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} w_i \|A_i - P_k\|_2^2 \tag{2.6a}$$

$$\text{s.t.} \quad P_k \in \mathcal{N}, \quad k = 1, \ldots, K. \tag{2.6b}$$

We named the problem in (2.6) *the net-constrained MSS clustering problem*, or – alternatively – *the net-constrained k-means problem*, after the famous k-means algorithm, which can be used to find a locally optimal solution of the problem.

Looking for what has been done by other researchers on the problem defined in (2.6), we found that "constrained clustering" in scientific literature does not refer to constraints for cluster center locations. Instead, this keyword refers to must-link, cannot-link[2] (see Piccialli et al. [44]) or cluster size constraints (see Bradley et al. [11]). That is, to our knowledge, constraints for cluster centers have not been considered in MSS clustering problems before. Thus, our scientific literature studies concentrated on the non-constrained MSS clustering problem, with the attempts to generalize the discovered results to the location constrained case in (2.6). That is, the methods reviewed are targeted at the problem

$$\min_{(\mathcal{C}_1,P_1),\ldots,(\mathcal{C}_K,P_K)} \quad \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} \|A_i - P_k\|_2^2 \,, \tag{2.7}$$

[2]That is, such constraints define pairs of points which must or must not be in the same cluster.

with no constraints for centers P_k: these can be any points in $\mathbb{R}^D$.

2.4.1. K-Means Principle

Consider again the problem in (2.5), but now stated this way:

$$\min_{\mathcal{C}_1,\ldots,\mathcal{C}_K} \sum_{k=1}^{K} \left[\min_{P \in \mathcal{N}} \sum_{i \in \mathcal{C}_k} w_i d(A_i, P) \right]. \tag{2.8}$$

From this formulation one can immediately see that given a partition of the objects into K clusters $\mathcal{C}_1, \ldots, \mathcal{C}_k$, we can find the optimal center for each cluster independently. This property leads to the following idea for the algorithm:

Location Step. Given a (fixed) partition into clusters $\mathcal{C}_1, \ldots, \mathcal{C}_K$, determine the optimal center for each cluster, i.e., solve problems

$$\min_{P \in \mathcal{N}} \sum_{i \in \mathcal{C}_k} w_i d(A_i, P), \quad k = 1, \ldots, K.$$

Allocation Step. Reassign each point A_i to the closest center as found in the **Location Step** (i.e., redefine clusters $\mathcal{C}_1, \ldots, \mathcal{C}_K$).

Repeat the above two steps until clusters and centers stabilize.

It is probably exactly the simplicity of the algorithm above which makes it so powerful. According to Bock [9], the idea is usually credited to the work of Lloyd [34]. This paper has more than 20,000 citations on Google Scholar, indicating the numerous applications and variants of the algorithm. In the location analysis field, this principle is also known as the location-allocation paradigm, with the first application in Cooper [14, 15]. The corresponding two papers together have more than 2,000 citations on Google Scholar. Another interesting application comes from the field of computational geometry where k-means principle can be used to obtain a centroidal Voronoi tessellation (see Du et al. [20] with more than 3,000 citations of the paper).

2.4.2. Global Algorithms

Though k-means-type algorithms can be used to obtain a proper (i.e., a locally optimal) solution to the MSS clustering problem in (2.7) in

a reasonable time, it does not guarantee the best possible solution. If the goal is to find the globally optimal solution or to prove the global optimality of a particular solution, we have to use other methods. In the following, we review the literature on the main approaches which could be used to obtain the global optimum of a MSSC problem. However, since MSSC problems are known to be NP-complete even in the plane (see, e.g., the paper of Vattani [50]), the computational times and problem instances at scope for the global optimization methods are incomparable with those of the k-means algorithm.

2.4.2.1. MIP Formulation

We initially encountered the ideas presented here in Werner's thesis [53], although earlier papers containing similar ideas likely exist. In this section, we use notation $\mathbb{N} := \{1, \ldots, N\}$ (client index set) and $\mathbb{K} := \{1, \ldots, K\}$ (cluster index set).

Pick arbitrary K locations $P_1, \ldots, P_K$ for the MSSC problem in (2.7) and consider the optimization problem below:

$$\begin{aligned} \min \quad & d_n \\ \text{s.t.} \quad & \|A_n - P_1\|_2^2 \leq d_n \vee \ldots \vee \|A_n - P_K\|_2^2 \leq d_n. \end{aligned} \tag{2.9}$$

That is, for the client at location A_n we seek the minimal d_n which satisfies at least one of the conditions $\|A_n - P_k\|_2^2 \leq d_n$, $k \in \mathbb{K}$. Clearly, the optimal solution of (2.9) for fixed $P_1, \ldots, P_K$ is equal to

$$d_n := \min_{k \in \mathbb{K}} \|A_n - P_k\|_2^2 . \tag{2.10}$$

Notice that with this definition of d_n, the optimal (minimal) loss in (2.7) is equal to $\sum_{n \in \mathbb{N}} w_n d_n$. Thus, consider the following optimization problem:

$$\begin{aligned} \min_{\{P_k, k \in \mathbb{K}\}, \{d_n, n \in \mathbb{N}\}} \quad & \sum_{n=1}^{N} w_n d_n \\ \text{s.t.} \quad & \|A_1 - P_1\|_2^2 \leq d_1 \vee \ldots \vee \|A_1 - P_K\|_2^2 \leq d_1, \\ & \|A_2 - P_1\|_2^2 \leq d_2 \vee \ldots \vee \|A_2 - P_K\|_2^2 \leq d_2, \\ & \vdots \\ & \|A_N - P_1\|_2^2 \leq d_N \vee \ldots \vee \|A_N - P_K\|_2^2 \leq d_N. \end{aligned} \tag{2.11}$$

You can convince yourself that the (globally) optimal solutions of formulations in (2.7) and (2.11) correspond. Therefore, we can find the optimal solution of (2.7) by solving (2.11). But how can we handle the disjunctive constraint (2.9)?

For this purpose, we first define $N \times K$ binary (indicator) variables $\{x_{nk},\, n \in \mathbb{N}, k \in \mathbb{K}\}$:

$$x_{nk} = \begin{cases} 1, & \text{if element } n \text{ is assigned to cluster } k, \\ 0 & \text{otherwise.} \end{cases}$$

Since an element can be assigned to only a single cluster, exactly one of $\{x_{nk},\, k \in \mathbb{K}\}$ will be equal to 1, i.e., we have

$$\sum_{k=1}^{K} x_{nk} = 1, \quad \forall n \in \mathbb{N}.$$

Now, pick a "sufficiently large" constant M_n such that we are sure the distance from A_n to any possible optimal center P for the MSSC problem (2.7) satisfies $\|A_n - P\|_2^2 \leq M_n$. Consider the following equation:

$$\|A_n - P_k\|_2^2 \leq d_n + (1 - x_{nk})M_n, \quad k \in \mathbb{K}. \tag{2.12}$$

Suppose $x_{nk_1} = 1$ and $x_{nk} = 0,\ k \neq k_1$. Then, the minimal d_n satisfying (2.12) will be equal to $\|A_n - P_{k_1}\|_2^2$, since for k_1, $x_{nk_1} = 1$, and the term $(1 - x_{nk_1})M_n$ will be "switched off". The remaining terms $k \neq k_1$ will satisfy (2.12) for any value of d_n, because $x_{nk} = 0$ and the term $(1 - x_{nk})M_n$ will be "switched on" (and $\|A_n - P\|_2^2 \leq M_n, \forall P$ by the choice of M_n). Minimizing over $\{x_{nk},\, k \in \mathbb{K}\}$, we ensure that $d_n = \min_{k \in \mathbb{K}} \|A_n - P_k\|_2^2$. Thus, we can reformulate the problem in (2.9) as follows:

$$\begin{aligned} \min_{\{x_{nk},\, k \in \mathbb{K}\}} \quad & d_n \\ \text{s.t.} \quad & \|A_n - P_1\|_2^2 \leq d_n + M_n(1 - x_{n1}), \\ & \|A_n - P_2\|_2^2 \leq d_n + M_n(1 - x_{n2}), \\ & \vdots \\ & \|A_n - P_K\|_2^2 \leq d_n + M_n(1 - x_{nK}), \\ & \sum_{k=1}^{K} x_{nk} = 1, \\ & x_{nk} \in \{0, 1\}\,,\ k \in \mathbb{K}. \end{aligned} \tag{2.13}$$

Now, the weighted MSSC problem can be stated as a Mixed Integer Quadratically Constrained Programming (MIQCP) problem:

$$\begin{aligned} \min_{\substack{\{P_k,\, k\in\mathbb{K}\},\ \{d_n,\, n\in\mathbb{N}\},\\ \{x_{nk},\, k\in\mathbb{K},\, n\in\mathbb{N}\}}} \quad & \sum_{n=1}^{N} w_n d_n \\ \text{s.t.} \quad & \text{each } d_n \text{ satisfies the constraints in (2.13)}, \quad n \in \mathbb{N}. \end{aligned} \tag{2.14}$$

In scientific literature, the "trick" to shift from the formulation in (2.11) to the one in (2.14) is called the "big-M method". An idea to introduce such a constant for a disjunctive constraint as in (2.9) can be traced back at least to Nemhauser and Wolsey [37].

To complete this section, we should clarify the definition of "sufficiently large" M_n. For the MSSC problem in (2.7), we have no constraints for cluster centers, i.e., $P_k \in \mathbb{R}^D$, $k \in \mathbb{K}$. However, it can be shown by simple arguments that any optimal P_k will belong to the simplex $\mathbf{conv}(\{A_n, n \in \mathbb{N}\})$, where

$$\mathbf{conv}(\{A_n, n \in \mathbb{N}\}) = \left\{ \sum_{n=1}^{N} \theta_n A_n \ \text{ s.t. } \ \theta_n \geq 0,\ \sum_{n=1}^{N} \theta_n = 1 \right\}.$$

Indeed, for an arbitrary cluster $\mathcal{C}_k \subseteq \mathbb{N}$, the optimal center minimizing the weighted sum of squares is equal to the weighted mean point:

$$P_k = \frac{1}{w(\mathcal{C}_k)} \sum_{n\in\mathcal{C}_k} w_n A_n, \quad \text{with} \quad w(\mathcal{C}_k) = \sum_{n\in\mathcal{C}_k} w_n.$$

Since $\sum_{n\in\mathcal{C}_k} \frac{w_n}{w(\mathcal{C}_k)} = 1$, $P_k \in \mathbf{conv}(\{A_n, n \in \mathcal{C}_k\}) \subseteq \mathbf{conv}(\{A_n, n \in \mathbb{N}\})$. Now, to define the "big-M" constant M_n for point A_n, we have to find the farthest point to A_n in $\mathbf{conv}(\{A_n, n \in \mathbb{N}\})$. It can be shown that this farthest point coincides with one of the vertices, i.e., we can define

$$M_n := \max_{n'\in\mathbb{N}} \|A_n - A_{n'}\|_2^2 . \tag{2.15}$$

Indeed, consider a hypersphere $\mathbf{sphere}(A_n, M_n)$ with center at A_n and radius M_n as defined in (2.15). By the definition of M_n,

$$A_{n'} \in \mathbf{sphere}(A_n, M_n), \quad \forall n' \in \mathbb{N}.$$

Since $\mathbf{sphere}(A_n, M_n)$ is a convex set, if follows (see Boyd and Vandenberghe [10]) also that

$$\mathbf{conv}(\{A_n, n \in \mathbb{N}\}) \subseteq \mathbf{sphere}(A_n, M_n).$$

That is, any possibly optimal P_k satisfies $\|A_n - P_k\|_2^2 \leq M_n$, $\forall k \in \mathbb{K}$.

Our contribution. The formulation in (2.14) can be easily extended to the net-constrained MSSC problem (2.6), since the constraint $P_k \in \mathcal{N}$ can be formulated as a MIP constraint (see Section 1.1). Thus, this constraint does not change problem type, i.e., it remains a MIQCP problem. However, our numerical experiments with this model were rather disappointing. Assuming a uniform distribution of input points on a unit square, we found that problems of clustering 25 (or more) points into 5 (or more) clusters are already out of scope [A.1]. This motivated us to develop the banch-and-cut type algorithm which is the subject of Chapter 5 and which appeared to be superior as compared with solving the formulation in (2.14) via a general solver [A.1].

2.4.2.2. Branch-and-Bound Approach

Although there are other attempts to solve the MSSC problem in (2.7) by a branch-and-bound technique, we only mention here the work of Brusco [12] and outline their method. We use notation

$$\mathbb{A}_{n_1}^{n_2} := \{A_{n_1}, A_{n_1+1}, \ldots, A_{n_2-1}, A_{n_2}\} \quad \text{for} \quad 1 \le n_1 \le n_2 \le N.$$

Suppose that we have assigned the starting n points $\mathbb{A}_1^n$ into K arbitrary (not necessarily optimal) clusters $\mathcal{C}_1, \ldots, \mathcal{C}_K$ using some enumeration technique. Label $\mathcal{P} := \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$. Consider the minimal loss for this fixed partition $\mathcal{P}$ (i.e., find the optimal center for each cluster):

$$\textbf{Loss}(\mathbb{A}_1^n|\mathcal{P}) := \sum_{k=1}^{K} \left[\min_{P} \sum_{i \in \mathcal{C}_k} w_i \left\| A_i - P \right\|_2^2 \right].$$

Now suppose we know the globally optimal value for the problem of clustering the remaining points $\mathbb{A}_{n+1}^N$ into K clusters. Label this loss with $\textbf{Loss}(\mathbb{A}_{n+1}^N/K)$. The branch-and-bound algorithm of [12] uses the following observation:

$$\textbf{Loss}(\left[\mathbb{A}_1^N/K\right]|\mathcal{P}) \ge \textbf{Loss}(\mathbb{A}_1^n|\mathcal{P}) + \textbf{Loss}(\mathbb{A}_{n+1}^N/K). \tag{2.16}$$

In the equation above condition $|\mathcal{P}$ means that we consider all the partitions of the set $\{1, \ldots, N\}$ which extend the partition $\mathcal{P}$, i.e., when the corresponding clusters $\mathcal{C}_1, \ldots, \mathcal{C}_K$ (which "span" the set $\{1, \ldots, n\}$) are "incremented" with the remaining elements in the set $\{n+1, \ldots, N\}$.

One can notice that if we know the optimal losses

$$\textbf{Loss}(\mathbb{A}_{n+1}^N/K),\ \textbf{Loss}(\mathbb{A}_{n+2}^N/K),\ \ldots,\ \textbf{Loss}(\mathbb{A}_N^N/K),$$

we can use the pruning criteria in (2.16) when looking for

$$\textbf{Loss}(\mathbb{A}_n^N/K),\ \textbf{Loss}(\mathbb{A}_{n-1}^N/K),\ \ldots,\ \textbf{Loss}(A_1^N/K)$$

via explicit enumeration. This is precisely the idea exploited in [12].[3] With such simple branch-and-bound algorithm, [12] managed to obtain the globally optimal solution of the iris dataset (Fisher [23]).

2.4.2.3. Semi-Definite Programming Formulation

Rather recently, Piccialli et al. [45] have implemented a branch-and-cut algorithm which, according to their paper, can solve MSSC problems up to the sizes of $2,000/15$ ($N \leq 2,000$ points into $K \leq 15$ clusters). This result in our opinion should be noted to be rather optimistic; for problem cases when the data is closer to uniform distribution in the space, one should expect this result to be significantly worse. Nevertheless, the algorithm in Piccialli et al. [45] can be considered the current state-of-the-art (or at least competing for such title), and we outline here the ideas of the paper.

MSSC problem in (2.7) can be restated as follows. Define client index set $\mathbb{N} := \{1, 2, \ldots, N\}$, cluster index set $\mathbb{K} := \{1, 2, \ldots, K\}$ and binary variables $\{x_{nk},\ n \in \mathbb{N},\ k \in \mathbb{K}\}$, which represent element assignment to clusters:

$$n \in \mathcal{C}_k \iff x_{nk} = 1, \quad n \notin \mathcal{C}_k \iff x_{nk} = 0.$$

[3] It is also important to note that [12] used an initial reordering of the points, which they call "nearest-neighbor separation heuristic". This heuristic attempts to position pairs of objects that are most similar to one another at opposite ends of the ordering. Such ordering seems to improve the enumeration algorithm significantly.

Then, the MSSC problem can be formulated this way:

$$\begin{aligned} \min_{\{P_k\},\,\{x_{nk}\}} \quad & \sum_{n\in\mathbb{N}}\sum_{k\in\mathbb{K}} x_{nk}\left\|A_n - P_k\right\|_2^2 \\ \text{s.t.} \quad & \sum_{k\in\mathbb{K}} x_{nk} = 1, \quad \forall n \in \mathbb{N}, \\ & \sum_{n\in\mathbb{N}} x_{nk} \geq 1, \quad \forall k \in \mathbb{K}, \\ & x_{nk} \in \{0,1\}, \quad \forall n \in \mathbb{N}, \forall k \in \mathbb{K}, \\ & P_k \in \mathbb{R}^D, \quad \forall k \in \mathbb{K}. \end{aligned} \tag{2.17}$$

Let's rewrite this problem formulation in matrix form. For this,

1. collect all the client-points $A_1, A_2, \ldots, A_N \in \mathbb{R}^D$ as rows in an $N \times D$ matrix $\mathbf{A}$,
2. collect the centers $P_1, P_2, \ldots, P_K$ as rows in a $K \times D$ matrix $\mathbf{P}$,
3. put binary-variables x_{nk} in an $N \times K$ binary-matrix $\mathbf{X}$.

For a fixed assignment matrix $\mathbf{X}$ of full column rank[4] K, optimal center positions can be found via equality $\mathbf{P} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{A}$, and the objective in (2.17) can be computed by the formula

$$\text{tr}\left(\mathbf{A}^T\mathbf{A}\right) - \text{tr}\left(\mathbf{A}^T\mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{A}\right). \tag{2.18}$$

Now, define matrix $\mathbf{Z} := \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$. It can be shown that this matrix is symmetric, idempotent, contains only positive elements, has rank $(\mathbf{Z}) = K$ and satisfies the equality[5] $\mathbf{Z}\mathbf{1}_N = \mathbf{1}_N$ (Peng and Wei [42]).

Thus, consider the following problem:[6]

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \text{tr}\left(\mathbf{A}\mathbf{A}^T\left(\mathbf{I}_N - \mathbf{Z}\right)\right) \\ \text{s.t.} \quad & \mathbf{Z}\mathbf{1}_N = \mathbf{1}_N, \text{tr}\left(\mathbf{Z}\right) = K, \\ & \mathbf{Z} \geq 0,\ \mathbf{Z}^2 = \mathbf{Z},\ \mathbf{Z} = \mathbf{Z}^T. \end{aligned} \tag{2.19}$$

Peng and Wei [42] proves that the problem above is equivalent to the problem in (2.17): any matrix $\mathbf{Z}$, satisfying the constraints in (2.19),

[4] This property is satisfied if there are no empty clusters. If there are empty clusters, the clustering can not be optimal.

[5] We label with $\mathbf{1}_N$ an N-length vector of all-ones.

[6] We label with $\mathbf{I}_N$ an $N \times N$ identity matrix.

can be expressed as $\mathbf{Z} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$ for some binary-assignment-matrix $\mathbf{X}$, and from here the equivalence of problems in (2.17) and (2.19) follows.

The formulation in (2.19) has "an ugly" non-linear constraint $\mathbf{Z}^2 = \mathbf{Z}$ (this is the idempotent matrix property). If we relax this constraint to a requirement $\mathbf{Z} \succeq 0$, i.e., that matrix $\mathbf{Z}$ is symmetric and positive semi-definite ($\mathbf{Z} \in \mathcal{S}_+^N$), we get a Semi-Definite Programming (SDP) problem:

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \text{tr}\left(\mathbf{A}\mathbf{A}^T\left(\mathbf{I}_N - \mathbf{Z}\right)\right) \\ \text{s.t.} \quad & \mathbf{Z}\mathbf{1}_N = \mathbf{1}_N, \text{tr}\left(\mathbf{Z}\right) = K, \\ & \mathbf{Z} \geq 0,\ \mathbf{Z} \succeq 0. \end{aligned} \tag{2.20}$$

The problem above is already "nice" in a sense that it is a convex optimization problem and large problem instances can be solved efficiently (Boyd and Vandenberghe [10]). Needless to say, the solutions of (2.19) and the corresponding relaxed problem in (2.20) are almost surely not the same, i.e., the solution $\mathbf{Z}^{\succeq 0}$ of (2.20) is an "underestimator" of solution $\mathbf{Z}^{0-1}$ to (2.19), since $\mathbf{Z}^{\succeq 0}$ comes from a broader matrix class (idempotent symmetric matrices are always positive semi-definite). To make the solution of (2.20) "closer" to (2.19), Piccialli et al. [45] introduces additional constraints. From the proof in Peng and Wei [42], one can see that for the solution of (2.19) the following holds:

$$z_{ij} = z_{ji} = 0, \quad \text{if point } i \text{ and point } j \text{ are in different clusters} \tag{2.21a}$$

$$\mathbf{Z}_{i\bullet} = \mathbf{Z}_{j\bullet}, \quad \text{if point } i \text{ and point } j \text{ are in the same cluster} \tag{2.21b}$$

Based on this, after finding the solution of (2.20) (and if it is not the solution of (2.19), i.e., a symmetric idempotent matrix), Piccialli et al. [45] "splits" the problem into branches: one with "cannot-link" constraint (2.21a) and another with "must-link" constraint (2.21b). The branching pair is chosen based on the rule

$$\arg\max_{i,\,j} \left\{ \min \left\{ z_{ij},\ \|\mathbf{Z}_{i\bullet} - \mathbf{Z}_{j\bullet}\|_2^2 \right\} \right\}, \tag{2.22}$$

since this choice can be expected to give a good lower-bound improvement. Using some other nice ideas, like an efficient way get a lower-bound for problem in (2.20) (which, although can be solved efficiently,

can still be solved only approximately by iterative methods), or a "smart" algorithm to find a good assignment matrix $\mathbf{X}$ from matrix $\mathbf{Z}^{\succeq 0}$, Piccialli et al. [45] manages to obtain a solution that is within $1+\varepsilon$ tolerance to the global optimum[7] (with $\varepsilon = 10^{-4}$).

The ideas in [45] can be generalized to the weighted MSSC problem. The corresponding 0-1 SDP problem is as follows:

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \operatorname{tr}\left(\mathbf{W}^{\frac{1}{2}}\mathbf{A}\mathbf{A}^T\mathbf{W}^{\frac{1}{2}}\left(\mathbf{I}_N - \mathbf{Z}\right)\right) \\ \text{s.t.} \quad & \mathbf{Z}\mathbf{W}^{\frac{1}{2}}\mathbf{1}_N = \mathbf{W}^{\frac{1}{2}}\mathbf{1}_N, \operatorname{tr}\left(\mathbf{Z}\right) = K, \\ & \mathbf{Z} \geq 0,\, \mathbf{Z}^2 = \mathbf{Z},\, \mathbf{Z} = \mathbf{Z}^T. \end{aligned} \tag{2.23}$$

The proof that the globally optimal solution of (2.17) with weights corresponds to the optimal solution of (2.23) can be found in Chen [13]; from this proof, one can also confirm the validity of the conditions in (2.21) for the weighted case. Thus, the results of Piccialli et al. [45] can be extended to the weighted MSSC problem.

2.4.2.4. Column Generation Approach

The column generation principle was proposed in the work of Gilmore and Gomory [24]. We review this method in the context of MSSC problems.

Consider all the possible subsets of the set $\mathbb{N} := \{1, \ldots, N\}$, of which there are 2^N. Rejecting the empty subset, we can encode each subset by a binary representations of numbers $1, 2, \ldots, 2^N - 1$. Thus, label $\mathcal{T} := \left\{1, \ldots, 2^N - 1\right\}$ and consider the binary representation of t:

$$t = 2^{N-1}x_{N-1,t} + 2^{N-2}x_{N-2,t} + \cdots + 2^1 x_{1,t} + 2^0 x_{0,t}.$$

Define

$$\mathcal{C}_t := \left\{n \in \{1, \ldots, N\} : x_{n-1,t} = 1\right\}.$$

Now consider the optimal MSS loss c_t for cluster $\mathcal{C}_t$:

$$c_t := \min_{P \in \mathbb{R}^D} \sum_{n \in \mathcal{C}_t} w_n \left\|A_n - P\right\|_2^2.$$

[7] That is, they obtain a lower bound and a solution which differ at most $1+\varepsilon$ times.

Theoretically, we can consider the following problem:

$$\min_{\{z_t,\, t\in\mathcal{T}\}} \quad \sum_{t\in\mathcal{T}} c_t z_t \tag{2.24a}$$

$$\text{s.t.} \quad \sum_{t\in\mathcal{T}} x_{nt} z_t \geq 1, \quad n \in \mathbb{N}, \tag{2.24b}$$

$$\sum_{t\in\mathcal{T}} z_t \leq K, \tag{2.24c}$$

$$z_t \in \{0,1\}, \quad t \in \mathcal{T}. \tag{2.24d}$$

The condition (2.24b) requires that each point A_n must belong to at least one cluster in the solution, and condition (2.24c) allows to "open" at most K clusters.[8] Stated this way, the problem in (2.24) can be seen as a *minimum weight set cover problem* (Korte and Vygen [32]).

The optimal value of the MIP problem in (2.24) is equal to the globally optimal value of the MSSC problem of clustering N points into K clusters. Undoubtedly, in practice, solving the MSSC via the formulation in (2.24) is only possible for very small values of N and K. However, in the full set of all possible clusters enumerated by $t \in \{1,\ldots,2^N-1\}$, only a small fraction of the clusters will be "reasonably good". By taking a "small" subset $\mathcal{T}' \subset \mathcal{T}$ of such "reasonably good" clusters, we can still expect to obtain the globally optimal value for the problem in (2.24).[9]

Now consider the dual of the problem in (2.24) for some subset $\mathcal{T}' \subset \mathcal{T}$ with the relaxation $z_t \geq 0$ of $z_t \in \{0,1\}$ (see the related chapter in Luenberger and Ye [35] on LP duality for the derivation):

$$\max_{\sigma,\, \{\lambda_n,\, n\in\mathbb{N}\}} \quad -K\sigma + \sum_{n=1}^{N} \lambda_n \tag{2.25a}$$

$$\text{s.t.} \quad -\sigma + \sum_{n=1}^{N} x_{nt}\lambda_n \leq c_t, \quad t \in \mathcal{T}', \tag{2.25b}$$

$$\sigma \geq 0, \quad \lambda_n \geq 0,\ n \in \mathbb{N}. \tag{2.25c}$$

Suppose we know the optimal solution of the problem in (2.25); label the optimal values with $\sigma(\mathcal{T}')$ and $\{\lambda_n(\mathcal{T}'),\ n \in \mathbb{N}\}$. Suppose that

[8] These conditions can be changed to $\sum_{t\in\mathcal{T}} x_{tn} z_t = 1$ and $\sum_{t\in\mathcal{T}} z_t = K$ resulting in the same optimal solution.

[9] This will happen when the subset $\mathcal{T}'$ contains the optimal clusters indexed by $t_1^*, \ldots, t_K^*$.

for some $t \in \mathcal{T}$, the condition in (2.25b) is violated, i.e.,

$$c_t + \sigma(\mathcal{T}') - \sum_{n=1}^{N} \lambda_n(\mathcal{T}') x_{nt} < 0. \tag{2.26}$$

Such "column" t violates the corresponding constraint in the theoretical dual of (2.24) for the full set $\mathcal{T}$, which must not happen at global optimality. To overcome this issue, we must include t in $\mathcal{T}'$: $\mathcal{T}' := \mathcal{T}' \cup \{t\}$. Now, solve again (2.25), i.e., find $\sigma(\mathcal{T}' \cup \{t\})$ and $\{\lambda_n(\mathcal{T}' \cup \{t\}),\ n \in \mathbb{N}\}$.

On the other hand, if no $t \in \mathcal{T}$ exists for which condition (2.26) holds, then we have arrived at the globally optimal solution.

Checking all $t \in \mathcal{T}$ for the condition in (2.26) is only possible theoretically, but we can consider the problem

$$\min_{t \in \mathcal{T}} \left[c_t + \sigma(\mathcal{T}') - \sum_{n=1}^{N} \lambda_n(\mathcal{T}') x_{nt} \right],$$

which for the MSSC is (remember the definition of c_t)

$$\sigma(\mathcal{T}') + \min_{P \in \mathbb{R}^D,\, \mathbf{x} \in \{0,1\}^N} \sum_{n=1}^{N} \left(w_n \left\| A_n - P \right\|_2^2 - \lambda_n(\mathcal{T}') \right) x_n. \tag{2.27}$$

This is a complicated combinatorial optimization problem, nevertheless, it is solvable (see, e.g., the method proposed in Aloise et al. [1]). If the optimal solution $\mathbf{x}$ to (2.27) gives a negative value, this means that inequality (2.26) holds, and the corresponding constraint (2.25b) is violated in the "theoretical" dual. Thus, we must include the corresponding index $t(\mathbf{x})$ in $\mathcal{T}'$ and again solve problem (2.25) for $\mathcal{T}' \cup \{t(\mathbf{x})\}$.

Column generation seems to provide the benchmark approach for the global optimization of MSSC. The work of Du Merle et al. [21] is probably the first paper which has reported the global solution to the Fisher's iris data [23]. In this paper, (2.27) is solved as a constrained hyperbolic program in 0-1 variables. Aloise et al. [1] improves the solution of (2.27) by exploiting geometrical properties. Although the work of Piccialli et al. [45] reported some improvements on MSSC problems as compared with the work of Aloise et al. [1] using a different approach (by solving a sequence of semi-definite programming problems as was discussed), a recent paper of Sudoso and Aloise [49] seems to propose the current best solver for the MSSC. Again, it employs the column generation principle.

Our contribution. In our paper [A.1], we solve the net-constrained MSSC problem using a branch-and-bound algorithm. We also report our numerical experiment of solving MIQCP problem via formulation in (2.14), showing that the times of the branch-and-bound algorithm are superior. In the article, we also discuss that solving the net-constrained MSSC problem via semi-definite programming approach (as in the paper of Piccialli et al. [45]) seems to be impossible.

At the time of writing our paper [A.1], we thought that the column generation approach can not be extended to the net-constrained MSSC. But as can be seen from the recent discussion, the auxiliary problem we have to solve is

$$\sigma + \min_{P \in \mathcal{N},\, \mathbf{x} \in \{0,1\}^N} \sum_{n=1}^{N} \Big(w_n \left\| A_n - P \right\|_2^2 - \lambda_n \Big) x_n, \tag{2.28}$$

e.g., all we have to do is to add the constraint $P \in \mathcal{N}$ to the formulation in (2.27). How complicated the problem in (2.28) is, remains an open question; we leave the solution of it and the study of the possible corresponding column generation method for future research.

2.5. Weber Problem

In analogy to the problem in (2.6), consider the problem of minimizing the sum of Euclidean norms:

$$\min_{(P_1,\mathcal{C}_1),\ldots,(P_K,\mathcal{C}_K)} \quad \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} w_i \left\| A_i - P_k \right\|_2^1 \tag{2.29a}$$

$$\text{s.t.} \quad P_k \in \mathcal{N}, \quad k = 1, \ldots, K. \tag{2.29b}$$

This problem could be called *the net-constrained multi-Weber problem* after the famous problem of Weber who considered the non-constrained single location case, i.e., the problem

$$\min_{P \in \mathbb{R}^D} \sum_{n=1}^{N} w_n \left\| A_n - P \right\|_2 . \tag{2.30}$$

For a survey on the problems related with the one in (2.30), we refer to the first chapter in Drezner and Hamacher [17]. Many interesting issues arise when trying to solve the problem in (2.30). The first one is that the optimal solution P^* of the problem can not be expressed via

an analytical formula, contrary to the case of squared Euclidean norms (where the optimal location corresponds to the mean point). Instead, P^* has to be determined via iterative methods; the first such method is credited to Weiszfeld [52].

Many analogous papers which extend the ideas discussed for the MSSC can be found for the multi-Weber problem. For example, the location-allocation algorithm of Cooper [14] can be seen as a k-means-type algorithm application to the case of (non-squared) Euclidean distances. Early attempts to solve the multi-Weber problem to global optimality via branch-and-cut algorithm of Rosing [48] are worth mentioning. Rosing [48] exploits the following geometrical property: in the optimal solution, convex hulls of the points within the clusters must not overlap.[10] For a column generation algorithm (see for example the work of Du Merle et al. [22]), one has to solve an auxilary "oracle" problem

$$\sigma + \min_{P \in \mathbb{R}^D,\, \mathbf{x} \in \{0,1\}^N} \sum_{n=1}^{N} \left(w_n \left\| A_n - P \right\|_2^1 - \lambda_n \right) x_n.$$

For example, this problem is studied in Drezner et al. [18].

2.5.1. Constrained Single Location Problems

While for the MSSC the practical value of center location constraints might be disputable, introducing such constraints for the Weber-type problems has much more reason. Thus, there are more articles on such models as compared with the MSSC. For example, Hansen et al. [27] considers the problem of locating the optimal Weber point when its location is constrained to a finite union of convex polygons, called "allowed regions". Looking from another angle, one can introduce "forbidden zones" for facility locations (as studied in Hamacher and Nickel [26]). The two approaches are complementary.

To make the Weber model in (2.30) to resemble a real life problem better, one can introduce even more properties or complexities. Specifically, we could consider a "combination" of a K-medians problem (see Section 2.2) and a Weber problem. Suppose we have two groups of clients: one which belongs to a net, $A_n \in \mathcal{N}$, and who can use a "cheap" network travel via road, and the other, $B_n \notin \mathcal{N}$, who must travel by

[10] This property also holds for MSSC problems.

an "expensive" off-network travel via air. Such a model is considered in Drezner et al. [19], who propose a method for obtaining the globally optimal location for the problem.

Another possible extension is proposed in Pfeiffer and Klamroth [43]. The model in this paper includes the following problem as a case. Consider a set of weighted clients $\{(A_n, w_n), n \in \mathbb{N}\}$, who can travel on foot, but can also use public transport network for a faster travel. This transport network contains a set of "stops" where you can enter/leave this network. Single facility (e.g., a hospital) has to be determined so that the sum of traveling times to the facility is minimized. Pfeiffer and Klamroth [43] proposes a bilinear programming formulation for the model.[11]

2.5.2. Barriers Problem

Traveling via straight lines, as it is considered in the Weber problem (2.30), is not always possible in real-life situations. Consider that in the plane, a set of barriers $\mathcal{B}_1, \ldots, \mathcal{B}_L$ is introduced (these barriers can represent buildings, lakes, mountains, etc.). Traveling through these barriers is prohibited. Label with $d_{\mathcal{B}}$ the distance metric as induced by the barriers. Define $\mathcal{R}^{\cup} := \mathbb{R}^2 \setminus \cup_{l=1}^{L} \mathcal{B}_l$, i.e., this is the set of possible locations for the solution. The problem is to locate a point $P \in \mathcal{R}^{\cup}$ such that

$$\sum_{n=1}^{N} w_n d_{\mathcal{B}}(A_n, P) \tag{2.31}$$

is minimal.

The first paper, which considered the optimal location for the problem in (2.31), can probably be attributed to Katz and Cooper [28]. They studied the case when a single circular barrier was introduced. Many other forms of barriers can be considered, e.g., line barriers with a defined set of "bridges" where one can cross to the other side of the "river". Also, different distance metrics can be defined for the problem, e.g., Manhattan-norm-type distances, or polyhedral-gauge-type distances.[12]

[11] A bilinear function is a function where the variables can be divided into two groups such that fixing the variables in one group as known/given, the function becomes linear in the remaining variables.

[12] Where one defines a convex polygon $\mathcal{P}$ which is centered at a reference point P, and the distance to another point Q is measured by scaling this polygon by a distance

For an extensive study of single location problems with barriers we refer to the book of Klamroth [30].

2.5.2.1. Optimal Single Facility Location for Polygonal Barriers

The barriers problem we study in this thesis considers polygonal barriers and seems to be introduced for the first time in the work of Aneja and Parlar [2]. Since the loss function in (2.31) is non-convex (e.g., it may have multiple local optima), the authors choose to solve the presented problem using the simulated annealing approach (Kirkpatrick et al. [29]). Another possible "fast" way to obtain a "good quality" solution to the problem in (2.31) is to use a genetic algorithm as proposed in Bischoff and Klamroth [7]. These two methods, although constructed to not get stuck at a local optimum, can not prove the global optimality of the solution obtained. To obtain a proven global solution, theoretically, one can subdivide the problem space ($\mathcal{R}^{\cup} := \mathbb{R}^2 \setminus \cup_{l=1}^{L} \mathcal{B}_l$) into a finite set of convex polygons $\{\mathcal{R}_i,\, i \in \mathcal{I}\}$, so that optimizing the loss in (2.31) within an additional constraint $P \in \mathcal{R}_i$ becomes relatively easy, see Klamroth [30]. The number $|\mathcal{I}|$ of such polygons however seems to be exponential, not leading to a practical algorithm to determine the minimum point of (2.31).

To our knowledge, the only deterministic algorithm to obtain a proven global minimum to (2.31) was presented in the thesis of Krau [33]. He proposes a method called "Big Square – Small Square". The idea is as follows. Initially, the rectangle of all theoretically possible minimum locations for the problem is determined and subdivided into (or "covered by") a set of identical squares. For each of these squares, a lower bound and an upper bound are estimated in case the center is placed within that square. A square with the minimum lower bound is chosen and subdivided into 4 smaller squares. This results in an improvement of the gap between the lower bound and the upper bound for each of the 4 parts. Then, the squares for which the lower bound is above the best known upper bound are removed. Again, from the remaining squares the one with the minimum lower bound is chosen and subdivided. The process continues until the gap between the best known lower bound and best known upper bound is within a predefined

constant d so that $Q \in \partial(d \cdot \mathcal{P})$ (i.e., Q belongs to the border or $d \cdot \mathcal{P}$).

tolerance.

2.5.2.2. Multi-Weber with Polyhedral Barriers Problem

In the multi-Weber with polyhedral barriers problem, not a single, but multiple locations have to be determined, i.e., we want to solve

$$\min_{\mathcal{C}_1,\ldots,\mathcal{C}_K} \sum_{k=1}^{K} \left[\min_{P_k \in \mathcal{R}^{\cup}} \sum_{i \in \mathcal{C}_k} w_i d_{\mathcal{B}}\left(A_i, P_k\right) \right]. \tag{2.32}$$

To obtain a locally optimal solution of (2.32), k-means-type algorithms can be constructed (see for example the one proposed in Bischoff et al. [8]). Our main interest in this PhD thesis is however to determine (prove) the global solution of the problem. Again, to our knowledge, the only "practical" algorithm for such problem was proposed in the thesis of Krau [33]. To obtain the globally optimal solution, they use a column generation algorithm. The auxiliary problem they solve is

$$\sigma + \min_{P \in \mathcal{R}^{\cup},\, \mathbf{x} \in \{0,1\}^N} \sum_{n=1}^{N} \left(w_n d_{\mathcal{B}}\left(A_n, P\right) - \lambda_n\right) x_n. \tag{2.33}$$

Given a fixed $P \in \mathcal{R}^{\cup}$, the minimal value of (2.33) over $\mathbf{x}$ is equal to

$$F(P) := \sigma + \sum_{n=1}^{N} (w_n d_{\mathcal{B}}\left(A_n, P\right) - \lambda_n) x_n(P), \tag{2.34}$$

with

$$x_n(P) := \begin{cases} 1, & \text{if } w_n d_{\mathcal{B}}\left(A_n, P\right) \leq \lambda_n, \\ 0, & \text{otherwise,} \end{cases} \qquad n \in \mathbb{N}.$$

The problem of minimizing function F in (2.34) is solved in Krau [33] again via "Big Square – Small Square" method. If we get that

$$\min_{P \in \mathcal{R}^{\cup}} F(P) < 0,$$

we must include the corresponding "column" $\{x_n(P),\, n \in \mathbb{N}\}$ in $\mathcal{T}'$ and again solve the dual in (2.25).

Our contribution. In our paper [A.2], to solve

$$\min_{P\in\mathcal{R}^{\cup}} \sum_{n=1}^{N} w_n d_{\mathcal{B}}\left(A_n, P\right),$$

we propose a method which could be named "Big Triangle – Small Triangle". The principle is rather identical to the "Big Square – Small Square" method, but instead of squares, we partition the location domain $\mathcal{R}^{\cup}$ into a set of triangles by a triangulation method.

For the multi-location problem, we divide $\mathcal{R}^{\cup}$ into a set of "small" triangle-units $\{\mathcal{R}_m,\, m = 1, \ldots, M\}$ of diameter (i.e., longest side) at most ε. For each pair of a client A_n and a triangle-unit $\mathcal{R}_m$, we precalculate the (minimal) distance $d_{nm}^{\inf}$ from A_n to the closest point in $\mathcal{R}_m$. Given some cluster $\mathcal{C}_k$ and the values

$$\mathbf{Loss}_m := \sum_{i\in\mathcal{C}_k} d_{nm}^{\inf}, \quad m = 1, \ldots, M,$$

we can find in $\mathcal{O}(M)$ time the best triangle when $\mathcal{C}_k$ is incremented with some new element $j \in \mathbb{N} \setminus \mathcal{C}_k$ by picking the smallest value from the set

$$\left\{\mathbf{Loss}_1 + d_{j1}, \mathbf{Loss}_2 + d_{j2}, \ldots, \mathbf{Loss}_M + d_{jM}\right\}.$$

This idea is used during the enumeration procedure. Also, similarly as in the algorithm of Brusco [12], we precalculate the lower bound estimates for problems ($\mathbb{A}_n^N := \{A_n, \ldots, A_N\}$)

$$\mathbb{A}_2^N/K,\ \mathbb{A}_3^N/K,\ \ldots,\ \mathbb{A}_{N-K}^N/K,$$

which can be later used in the pruning criteria (similarly as in (2.16)). With such branch-and-bound algorithm, we have managed to solve to global optimality problems of sizes[13] $60/2$, $45/3$, $40/4$, $35/5$ [A.1]. To our knowledge, this is the first time when global solutions for barrier problems with more than 30 client points are reported (Krau [33] considered only the cases with at most 30 clients).

At the time of writing our paper [A.2], we were not aware of the thesis of Krau [33], which was written in 1997, in French. Whether column generation approach can "beat" our result in [A.2] on a modern computer, remains an open question and is left for future research.

[13] The notation N/K refers to a problem of clustering N points into K clusters.

2.6. Chapter Summary

In this chapter, we reviewed some literature on the problems related with the ones we study in this thesis. In particular, we reviewed the literature on the pile-balancing problem, which motivated us to define the constrained equal area Voronoi tessellation model. Literature on k-means-type algorithms was reviewed. These algorithms can be used to find a locally-optimal solution to clustering-like problems. However, our main interest in this research was to find and prove the globally optimal solutions of multi-location problems. Thus, we briefly reviewed the ideas in the scientific literature which could be used to obtain the global solution of the related problems: the branch-and-bound approach, the semi-definite programming approach, or the column generation approach and its applications to clustering-like problems.

3. CONSTRAINED EQUAL AREA VORONOI TESSELLATION PROBLEM

In this chapter we consider the constrained equal area Voronoi tessellation problem as summarized by the following formulation:

$$\begin{aligned} \min_{P_1,\dots,P_K} \quad & \frac{1}{K}\sum_{k=1}^{K}\left|\textbf{area}(\mathcal{V}_k \cap \mathcal{P}) - \frac{\textbf{area}(\mathcal{P})}{K}\right| \\ \text{s.t.} \quad & P_k \in \mathcal{N}, \quad k = 1,\dots,K. \end{aligned} \tag{3.1}$$

Throughout the chapter, we will assume that $\frac{\textbf{area}(\mathcal{P})}{K} \equiv 1$. This is not restrictive, since we can always scale the objective function in (3.1) by a constant $c := \frac{K}{\textbf{area}(\mathcal{P})}$ without changing the optimal solution. Also, to make the problem formulation more general, we use notation

$$F_k(P_1,\dots,P_K) := \textbf{area}(\mathcal{V}_k \cap \mathcal{P}), \quad k = 1,\dots,K, \tag{3.2}$$

having in mind that Voronoi cell $\mathcal{V}_k$ is a "function" of all site locations $P_1, P_2, \dots, P_K$. Thus, the problem we study in this chapter is as follows:

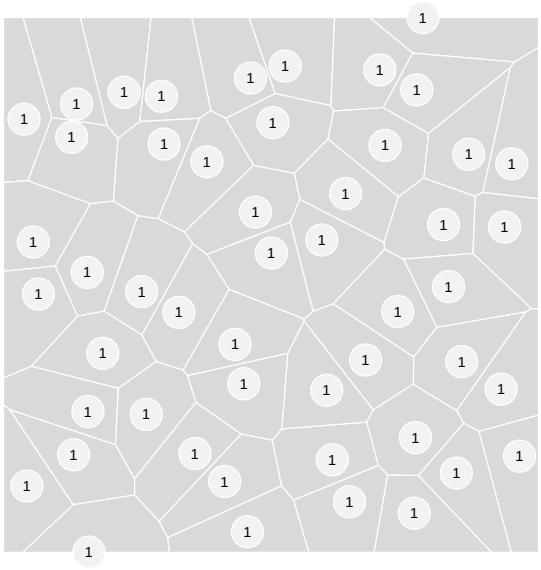
$$\min_{P_1,\dots,P_K} \quad \frac{1}{K}\sum_{k=1}^{K}|F_k(P_1,\dots,P_K) - 1| \tag{3.3a}$$

$$\text{s.t.} \quad P_k \in \mathcal{N}, \quad k = 1,\dots,K, \tag{3.3b}$$

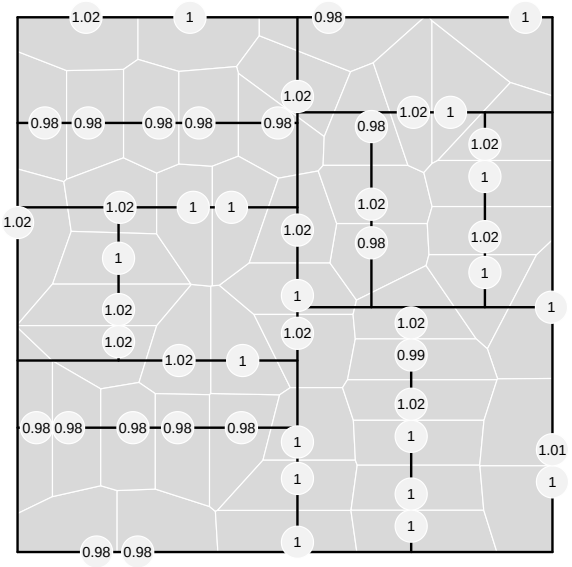
$$\|P_k - P_l\|_2 \geq 2\varepsilon, \quad k \neq l. \tag{3.3c}$$

Condition (3.3c) is introduced to make the sites sufficiently well separated. This is in parallel with the pile balancing problem formulation as discussed in Section 2.1.

The rest of the chapter is as follows. Firstly, we discuss the Jacobian approximation of the multivariate function $\vec{\mathbf{F}} := (F_1,\dots,F_K)$ at fixed locations $P_1,\dots,P_K$. Then, we show how the condition $\|P_k - P_l\|_2 \geq 2\varepsilon$ can be formulated by a set of linear inequalities in the neighborhood of each P_k, $k = 1,\dots,K$. Having these two blocks, we formulate problem (3.3) without the constraint (3.3b) as a Linear Programming (LP) problem and present an iterative procedure which converges to an output as illustrated in Figure 3.1a. To introduce the net constraint (3.3b) for the sites, we use its MIP formulation as discussed in Section 1.1. By iteratively solving the corresponding MIP problems, we present a procedure to arrive at the solution which is illustrated in Figure 3.1b.



(a) Unconstrained case.



(b) Constrained case.

Figure 3.1. Equal area Voronoi tessellation problem. The numbers within the circles indicate the area of the corresponding polygon.

3.1. Unconstrained Case

We firstly outline the building blocks for an LP formulation of the *unconstrained* equal area Voronoi tessellation problem (formulation (3.3) without the constraint (3.3b)).

3.1.1. Taylor Expansion of a Multivariate Vector Function

Every multivariate function $F : \mathbb{R}^n \mapsto \mathbb{R}$ which is twice continuously differentiable at point $\mathbf{x} \in \mathbb{R}^n$ can be approximated using Taylor expansion (see Nocedal and Wright [38], Theorem 2.1, p. 14):

$$F(\mathbf{x} + \Delta\mathbf{x}) = F(\mathbf{x}) + \nabla F(\mathbf{x})^T \Delta\mathbf{x} + \Delta\mathbf{x}^T \nabla^2 F(\mathbf{x} + t\Delta\mathbf{x})\Delta\mathbf{x}$$

for some $t \in (0, 1)$. Here $\nabla F(\mathbf{x})$ is the gradient and $\nabla^2 F(\mathbf{x})$ is the Hessian matrix of F at point $\mathbf{x}$:

$$\nabla F(\mathbf{x}) = \begin{bmatrix} \frac{\partial F}{\partial x_1}(\mathbf{x}) \\ \vdots \\ \frac{\partial F}{\partial x_n}(\mathbf{x}) \end{bmatrix}, \quad \nabla^2 F(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 F}{\partial x_1 \partial x_1}(\mathbf{x}) & \cdots & \frac{\partial^2 F}{\partial x_1 \partial x_n}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 F}{\partial x_n \partial x_1}(\mathbf{x}) & \cdots & \frac{\partial^2 F}{\partial x_n \partial x_n}(\mathbf{x}) \end{bmatrix}.$$

The above theorem means that for sufficiently small $\Delta\mathbf{x} \in \mathbb{R}^n$,

$$F(\mathbf{x} + \Delta\mathbf{x}) \approx F(\mathbf{x}) + \nabla F(\mathbf{x})^T \Delta\mathbf{x}.$$

Similarly, a sufficiently well-behaved multivariate vector function $\vec{\mathbf{F}}(\mathbf{x}) : \mathbb{R}^n \mapsto \mathbb{R}^m$ can be approximated by Taylor expansion with the Jacobian matrix:

$$\vec{\mathbf{F}}(\mathbf{x} + \Delta\mathbf{x}) = \begin{bmatrix} F_1(\mathbf{x} + \Delta\mathbf{x}) \\ \vdots \\ F_m(\mathbf{x} + \Delta\mathbf{x}) \end{bmatrix} \approx \begin{bmatrix} F_1(\mathbf{x}) \\ \vdots \\ F_m(\mathbf{x}) \end{bmatrix} + \begin{bmatrix} —\nabla F_1(\mathbf{x})^T— \\ \vdots \\ —\nabla F_m(\mathbf{x})^T— \end{bmatrix} \Delta\mathbf{x}, \tag{3.4a}$$

or, shortly,

$$\vec{\mathbf{F}}(\mathbf{x} + \Delta\mathbf{x}) \approx \vec{\mathbf{F}}(\mathbf{x}) + \mathbf{J}(\mathbf{x})\Delta\mathbf{x}. \tag{3.4b}$$

3.1.2. Numerical Approximation of the Jacobian Matrix

Our Voronoi cell area functions $F_k : \mathbb{R}^{2K} \mapsto \mathbb{R}$ as defined in (3.2) do not have an analytical form, but are well-behaved in a sense that moving

a point slightly only slightly changes the cells (and their areas). Thus, we can consider the Jacobian approximation for the multivariate load function $\vec{\mathbf{F}}$.

To make the notation more convenient, instead of

$$\vec{\mathbf{F}}(\mathbf{P}) := \vec{\mathbf{F}}(P_1, \ldots, P_K)$$

we also write $\vec{\mathbf{F}}(\mathbf{x}, \mathbf{y})$ with $\mathbf{x} := (x_1, \ldots, x_K)$ and $\mathbf{y} := (y_1, \ldots, y_K)$, e.g., the vectors of x and y coordinates of the points. Also, we consider the Jacobian matrix to be a stack of two matrices $\mathbf{J_x}(\mathbf{P})$, $\mathbf{J_y}(\mathbf{P})$:

$$\mathbf{J}(\mathbf{P}) := [\mathbf{J_x}\ \mathbf{J_y}]\,(\mathbf{P}) = [\mathbf{J_x}\ \mathbf{J_y}]\,(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{K \times 2K},$$

with

$$\mathbf{J_x}(\mathbf{P}) := \begin{bmatrix} \frac{\partial F_1}{\partial x_1}(\mathbf{P}) & \cdots & \frac{\partial F_1}{\partial x_K}(\mathbf{P}) \\ \frac{\partial F_2}{\partial x_1}(\mathbf{P}) & \cdots & \frac{\partial F_2}{\partial x_K}(\mathbf{P}) \\ \vdots & \ddots & \vdots \\ \frac{\partial F_K}{\partial x_1}(\mathbf{P}) & \cdots & \frac{\partial F_K}{\partial x_K}(\mathbf{P}) \end{bmatrix}, \quad \mathbf{J_y}(\mathbf{P}) := \begin{bmatrix} \frac{\partial F_1}{\partial y_1}(\mathbf{P}) & \cdots & \frac{\partial F_1}{\partial y_K}(\mathbf{P}) \\ \frac{\partial F_2}{\partial y_1}(\mathbf{P}) & \cdots & \frac{\partial F_2}{\partial y_K}(\mathbf{P}) \\ \vdots & \ddots & \vdots \\ \frac{\partial F_K}{\partial y_1}(\mathbf{P}) & \cdots & \frac{\partial F_K}{\partial y_K}(\mathbf{P}) \end{bmatrix}.$$

Now, using the Jacobian approximation as reminded in Section 3.1.1, we have that

$$\vec{\mathbf{F}}(\mathbf{x} + \Delta\mathbf{x}, \mathbf{y} + \Delta\mathbf{y}) \approx \vec{\mathbf{F}}(\mathbf{x}, \mathbf{y}) + [\mathbf{J_x}\ \mathbf{J_y}]\,(\mathbf{x}, \mathbf{y}) \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \end{pmatrix}, \tag{3.5}$$

if both column-vectors $\Delta\mathbf{x}$ and $\Delta\mathbf{y}$ are sufficiently small (e.g., their norms are close to 0).

Since our load function $\vec{\mathbf{F}}$ does not have an analytical form, we can not compute the Jacobian analytically. But we can approximate the Jacobian numerically (see Kochenderfer and Wheeler [31], Chapter 2): for sufficiently small ε, we have

$$\begin{bmatrix} \frac{\partial F_1}{\partial x_k}(\mathbf{x}, \mathbf{y}) \\ \frac{\partial F_2}{\partial x_k}(\mathbf{x}, \mathbf{y}) \\ \vdots \\ \frac{\partial F_K}{\partial x_k}(\mathbf{x}, \mathbf{y}) \end{bmatrix} \approx \begin{bmatrix} \frac{F_1(\mathbf{x}+\varepsilon\mathbf{e}_k, \mathbf{y}) - F_1(\mathbf{x}, \mathbf{y})}{\varepsilon} \\ \frac{F_2(\mathbf{x}+\varepsilon\mathbf{e}_k, \mathbf{y}) - F_2(\mathbf{x}, \mathbf{y})}{\varepsilon} \\ \cdots \\ \frac{F_K(\mathbf{x}+\varepsilon\mathbf{e}_k, \mathbf{y}) - F_K(\mathbf{x}, \mathbf{y})}{\varepsilon} \end{bmatrix} = \frac{\vec{\mathbf{F}}(\mathbf{x} + \varepsilon\mathbf{e}_k,\ \mathbf{y}) - \vec{\mathbf{F}}(\mathbf{x},\ \mathbf{y})}{\varepsilon},$$

here $\mathbf{e}_k$ is a vector of length K with 1 at position k and 0 elsewhere. For example, define matrices

$$\widetilde{\mathbf{J}}_{\mathbf{x}}(\mathbf{x},\mathbf{y}) := \begin{bmatrix} | & & | \\ \frac{\vec{\mathbf{F}}(\mathbf{x}+\varepsilon\mathbf{e}_1,\mathbf{y})-\vec{\mathbf{F}}(\mathbf{x},\mathbf{y})}{\varepsilon} & \dots & \frac{\vec{\mathbf{F}}(\mathbf{x}+\varepsilon\mathbf{e}_K,\mathbf{y})-\vec{\mathbf{F}}(\mathbf{x},\mathbf{y})}{\varepsilon} \\ | & & | \end{bmatrix}, \tag{3.6a}$$

and, similarly,

$$\widetilde{\mathbf{J}}_{\mathbf{y}}(\mathbf{x},\mathbf{y}) := \begin{bmatrix} | & & | \\ \frac{\vec{\mathbf{F}}(\mathbf{x},\mathbf{y}+\varepsilon\mathbf{e}_1)-\vec{\mathbf{F}}(\mathbf{x},\mathbf{y})}{\varepsilon} & \dots & \frac{\vec{\mathbf{F}}(\mathbf{x},\mathbf{y}+\varepsilon\mathbf{e}_K)-\vec{\mathbf{F}}(\mathbf{x},\mathbf{y})}{\varepsilon} \\ | & & | \end{bmatrix}. \tag{3.6b}$$

We have $\mathbf{J}_{\mathbf{x}}(\mathbf{x},\mathbf{y}) \approx \widetilde{\mathbf{J}}_{\mathbf{x}}(\mathbf{x},\mathbf{y})$ and $\mathbf{J}_{\mathbf{y}}(\mathbf{x},\mathbf{y}) \approx \widetilde{\mathbf{J}}_{\mathbf{y}}(\mathbf{x},\mathbf{y})$.
In summary, for sufficiently small $\Delta\mathbf{x}$ and $\Delta\mathbf{y}$ we get that

$$\vec{\mathbf{F}}(\mathbf{x}+\Delta\mathbf{x},\mathbf{y}+\Delta\mathbf{y}) \approx \vec{\mathbf{F}}(\mathbf{x},\mathbf{y}) + \left[\widetilde{\mathbf{J}}_{\mathbf{x}}\ \widetilde{\mathbf{J}}_{\mathbf{y}}\right](\mathbf{x},\mathbf{y}) \begin{pmatrix}\Delta\mathbf{x}\\ \Delta\mathbf{y}\end{pmatrix}. \tag{3.7}$$

3.1.3. Handling the Non-Overlapping Circles Constraint

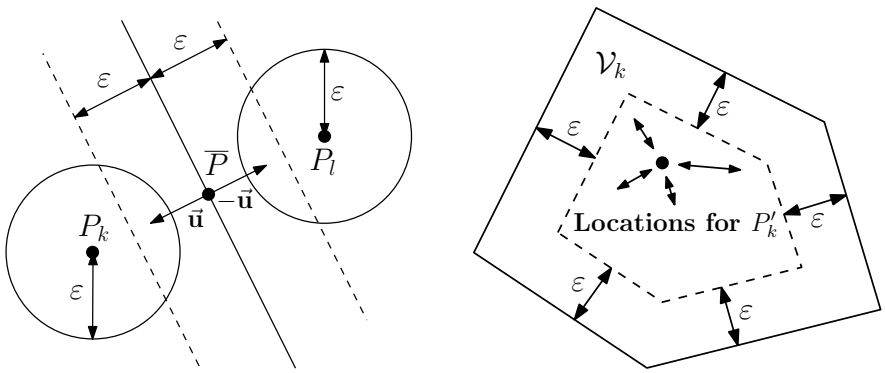
Constraint (3.3c) which requires that $\|P_k - P_l\|_2 \geq 2\varepsilon$ is non-convex, making problem (3.3) unapproachable with convex optimization tools directly. Thus, we will formulate this constraint differently.

Suppose we are given locations $P_1, P_2, \dots, P_K$ which satisfy the constraints $\|P_k - P_l\|_2 \geq 2\varepsilon$ pairwise for $k \neq l$. Take an arbitrary pair P_k, P_l and consider a vector $\vec{\mathbf{v}}_{l\to k} := P_k - P_l$ joining those two points. Now, normalize $\vec{\mathbf{v}}_{l\to k}$ and define a unit vector $\vec{\mathbf{u}}_{l\to k} := \frac{\vec{\mathbf{v}}_{l\to k}}{\|\vec{\mathbf{v}}_{l\to k}\|}$. Also, label the middle point $\overline{P}_{kl} := \frac{P_k+P_l}{2}$. We have (see Figure 3.2a):

$$\|P_k - P_l\|_2 \geq 2\varepsilon \iff \langle P_k - \overline{P}_{kl},\, \vec{\mathbf{u}}_{l\to k}\rangle \geq \varepsilon \ \wedge\ \langle P_l - \overline{P}_{kl},\, -\vec{\mathbf{u}}_{l\to k}\rangle \geq \varepsilon, \tag{3.8}$$

here $\langle \bullet, \bullet \rangle$ is the dot product. Equation (3.8) means that if points P_k and P_l can be separated by an "infinite wall" (hyperplane) of thickness 2ε, then the distance condition $\|P_k - P_l\|_2 \geq 2\varepsilon$ also holds. Thus, if we require that during an update of positions $P_1, \dots, P_K$ to positions $P'_1, \dots, P'_K$, conditions

$$\langle P'_k - \overline{P}_{kl},\, \vec{\mathbf{u}}_{l\to k}\rangle \geq \varepsilon, \quad l \neq k, \tag{3.9}$$



(a) Illustration of condition (3.8).

(b) Updating the location for P_k; $\mathcal{V}_k$ is the Voronoi cell of P_k.

Figure 3.2. Illustration of handling the constraint $\|P'_k - P'_l\| \geq 2\epsilon$.

are kept for all $k = 1, \ldots, K$, we ensure that conditions $\|P'_k - P'_l\| \geq 2\varepsilon$ will also hold. Geometrically, this means that we allow the updated positions P'_k to be placed only within Voronoi cell $\mathcal{V}_k$ with a "wall" of thickness ε (see Figure 3.2b).

3.1.4. LP Formulation

In this section, we give the Linear Programming (LP) formulation of the problem in (3.3) without the net constraint (3.3b) for cluster centers.

Suppose we are given initial locations $P_1^{(0)}, \ldots, P_K^{(0)}$ which satisfy distance conditions

$$\left\| P_k^{(0)} - P_l^{(0)} \right\| \geq 2\varepsilon, \quad k \neq l.$$

We seek to solve the l_1-norm minimization problem

$$\min_{\mathbf{x},\mathbf{y}} \left\| \vec{\mathbf{F}}(\mathbf{x},\, \mathbf{y}) - \mathbf{1} \right\|_1 = \sum_{k=1}^{K} \left| F_k(\mathbf{x},\, \mathbf{y}) - 1 \right|.$$

Consider the Jacobian approximation of the load function $\vec{\mathbf{F}}$ at the initial locations, as discussed in Section 3.1.2:

$$\vec{\mathbf{F}}\left(\mathbf{x}^{(0)} + \Delta\mathbf{x},\ \mathbf{y}^{(0)} + \Delta\mathbf{y}\right) \approx \vec{\mathbf{F}}^{(0)} + \mathbf{J}_{\mathbf{x}}^{(0)}\Delta\mathbf{x} + \mathbf{J}_{\mathbf{y}}^{(0)}\Delta\mathbf{y}, \tag{3.10}$$

with

$$\vec{\mathbf{F}}^{(0)} := \vec{\mathbf{F}}\left(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}\right),\ \mathbf{J}_{\mathbf{x}}^{(0)} := \widetilde{\mathbf{J}}_{\mathbf{x}}\left(\mathbf{x}^{(0)},\, \mathbf{y}^{(0)}\right) \text{ and } \mathbf{J}_{\mathbf{y}}^{(0)} := \widetilde{\mathbf{J}}_{\mathbf{y}}\left(\mathbf{x}^{(0)},\, \mathbf{y}^{(0)}\right).$$

The right hand side of (3.10) is linear in terms of the variables $\Delta\mathbf{x}, \Delta\mathbf{y}$. If we do not take the constraints $\|P_k - P_l\| \geq 2\varepsilon$ into account, we can formulate the minimum sum of absolute residuals problem

$$\min_{\Delta\mathbf{x},\,\Delta\mathbf{y}} \left\| \left(\vec{\mathbf{F}}^{(0)} + \mathbf{J}_{\mathbf{x}}^{(0)} \Delta\mathbf{x} + \mathbf{J}_{\mathbf{y}}^{(0)} \Delta\mathbf{y} \right) - \mathbf{1} \right\|_1$$

as an LP problem (Boyd and Vandenberghe [10], p. 294):

$$\begin{aligned} \min_{\mathbf{r},\,\Delta\mathbf{x},\,\Delta\mathbf{y}} \quad & \mathbf{1}^T \mathbf{r} \\ \text{s.t.} \quad & -\mathbf{r} \preceq \left(\vec{\mathbf{F}}^{(0)} + \mathbf{J}_{\mathbf{x}}^{(0)} \Delta\mathbf{x} + \mathbf{J}_{\mathbf{y}}^{(0)} \Delta\mathbf{y} \right) - \mathbf{1} \preceq \mathbf{r}, \\ & \mathbf{0} \preceq \mathbf{x} + \Delta\mathbf{x},\, \mathbf{y} + \Delta\mathbf{y} \preceq \mathbf{1}, \end{aligned}$$

with variables $\mathbf{r}, \Delta\mathbf{x}, \Delta\mathbf{y} \in \mathbb{R}^K$.

Now we deal with constraints $\|P_k - P_l\|_2 \geq 2\varepsilon$. We can ensure that conditions

$$\left\| P'_k - P'_l \right\| = \left\| \begin{pmatrix} x_k + \Delta x_k \\ y_k + \Delta y_k \end{pmatrix} - \begin{pmatrix} x_l + \Delta x_l \\ y_l + \Delta y_l \end{pmatrix} \right\|_2 \geq 2\varepsilon$$

will hold after a location update using the ideas from Section 3.1.3. E.g., (3.9) is equivalent to the following equation:

$$\begin{aligned} \langle P'_k - \overline{P}_{kl},\, \vec{\mathbf{u}}_{l\to k} \rangle &= \left\langle P_k + \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} - \overline{P}_{kl},\, \vec{\mathbf{u}}_{l\to k} \right\rangle \\ &= \left\langle \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix},\, \vec{\mathbf{u}}_{l\to k} \right\rangle + \left\langle P_k - \overline{P}_{kl},\, \vec{\mathbf{u}}_{l\to k} \right\rangle \geq \varepsilon. \quad (3.11) \end{aligned}$$

The above equation is linear in the variables $\Delta x_k, \Delta y_k$ and must hold for any $l \neq k$. To make the notation shorter, we stack the vectors $\vec{\mathbf{u}}_{l\to k}$ as rows in matrix $\mathbf{A}_k^{(0)}$ and constants $\left\langle P_k - \overline{P}_{kl}, \vec{\mathbf{u}}_{l\to k} \right\rangle$ as coordinates in vector $\mathbf{b}_k^{(0)}$:

$$\mathbf{A}_k^{(0)} := \begin{bmatrix} \vec{\mathbf{u}}_{1\to k}^T \\ \vdots \\ \vec{\mathbf{u}}_{(k-1)\to k}^T \\ \vec{\mathbf{u}}_{(k+1)\to k}^T \\ \vdots \\ \vec{\mathbf{u}}_{K\to k}^T \end{bmatrix}, \quad \mathbf{b}_k^{(0)} := \begin{bmatrix} \langle P_k - \overline{P}_{k1}, \vec{\mathbf{u}}_{1\to k} \rangle \\ \vdots \\ \langle P_k - \overline{P}_{k(k-1)}, \vec{\mathbf{u}}_{(k-1)\to k} \rangle \\ \langle P_k - \overline{P}_{k(k+1)}, \vec{\mathbf{u}}_{(k+1)\to k} \rangle \\ \vdots \\ \langle P_k - \overline{P}_{Kl}, \vec{\mathbf{u}}_{K\to k} \rangle \end{bmatrix}.$$

Now, conditions (3.11) can be written in matrix form:

$$\mathbf{A}_k^{(0)} \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} \succeq \varepsilon \mathbf{1} - \mathbf{b}_k^{(0)}.$$

To shrink the number of inequalities in the above expression and in the LP program, one can keep in matrix $\mathbf{A}_k^{(0)}$ and vector $\mathbf{b}_k^{(0)}$ only these rows l for which cells $\mathcal{V}_k$ and $\mathcal{V}_l$ have common edges in the Voronoi diagram. The remaining ones will be satisfied automatically (as can be seen from Figure 3.2).

We now state the LP problem:

$$\min_{\mathbf{r},\, \Delta\mathbf{x},\, \Delta\mathbf{y}} \quad \mathbf{1}^T \mathbf{r} \tag{3.12a}$$

$$\text{s.t.} \quad -\mathbf{r} \preceq \left(\vec{\mathbf{F}}^{(0)} + \mathbf{J}_{\mathbf{x}}^{(0)} \Delta\mathbf{x} + \mathbf{J}_{\mathbf{y}}^{(0)} \Delta\mathbf{y} \right) - \mathbf{1} \preceq \mathbf{r}, \tag{3.12b}$$

$$\mathbf{0} \preceq \mathbf{x} + \Delta\mathbf{x},\, \mathbf{y} + \Delta\mathbf{y} \preceq \mathbf{1}, \tag{3.12c}$$

$$\mathbf{A}_k^{(0)} \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} \succeq \varepsilon \mathbf{1} - \mathbf{b}_k^{(0)}, \quad k = 1, \ldots, K. \tag{3.12d}$$

To make sure that vectors $\Delta\mathbf{x}$ and $\Delta\mathbf{y}$ are small enough so that Jacobian approximation (3.10) remains sufficiently precise, one can also add to the LP problem (3.12) constraints

$$-\delta \le \Delta x_k, \Delta y_k \le \delta, \quad k = 1, \ldots, K \quad \text{(for some small } \delta > 0). \tag{3.12e}$$

3.1.5. Alternatives for the Loss Function

Instead of minimizing the sum of absolute deviations, i.e., instead of the loss

$$\sum_{k=1}^{K} \left| F_k(P_1, \ldots, P_K) - 1 \right|, \tag{3.13}$$

one can consider other loss functions, e.g.

$$\max_{k=1,\ldots,K} \left| F_k(P_1, \ldots, P_K) - 1 \right| \quad \text{or} \tag{3.14a}$$

$$\max_{k=1,\ldots,K} \left(F_k(P_1, \ldots, P_K) - 1 \right). \tag{3.14b}$$

In case we can obtain positions $P_1, \ldots, P_K$ such that

$$F_k(P_1, \ldots, P_K) \equiv 1, \quad k = 1, \ldots, K,$$

all these losses would give the same optimal value equal to 0. Compared with the minimal sum of absolute deviations loss (3.13), losses in (3.14a) and (3.14b) could be also formulated as LP problems, e.g., for the maximum absolute loss formulation (3.14a), we would have in LP formulation (3.12) to change equations (3.12a) and (3.12b) to

$$\min_{r_{\max},\,\Delta\mathbf{x},\,\Delta\mathbf{y}} \quad r_{\max} \tag{3.15a}$$

$$\text{s.t.} \quad -\mathbf{1}r_{\max} \preceq \left(\vec{\mathbf{F}}^{(0)} + \mathbf{J}_{\mathbf{x}}^{(0)}\Delta\mathbf{x} + \mathbf{J}_{\mathbf{y}}^{(0)}\Delta\mathbf{y}\right) - \mathbf{1} \preceq \mathbf{1}r_{\max}, \tag{3.15b}$$

where $r_{\max}$ is now a single variable. For the maximum loss (3.14b), we can simply skip the left inequality in (3.15b).

These alternative LP formulations have different optimal solution properties which we will exploit when describing our final algorithm for the constrained equal area Voronoi tessellation problem.

3.1.6. Algorithm Illustration

The unconstrained equal area Voronoi tessellation algorithm is illustrated in Figure 3.3. Starting at initial positions $P_1^{(0)}, \ldots, P_K^{(0)}$, the algorithm improves the positions to

$$P_1^{(1)} := P_1^{(0)} + \begin{pmatrix}\Delta x_1\\ \Delta y_1\end{pmatrix}, \quad \ldots, \quad P_K^{(1)} := P_K^{(0)} + \begin{pmatrix}\Delta x_K\\ \Delta y_K\end{pmatrix}$$

by solving problem (3.12). Then, the procedure is repeated for the new positions with corresponding $\vec{\mathbf{F}}^{(1)}$, $\mathbf{J}_{\mathbf{x}}^{(1)}$, $\mathbf{J}_{\mathbf{y}}^{(1)}$, updated matrices $\mathbf{A}_k^{(1)}$ and vectors $\mathbf{b}_k^{(1)}$ which will assure the constraints

$$\left\| P_k^{(2)} - P_l^{(2)} \right\| \geq 2\varepsilon, \quad k \neq l$$

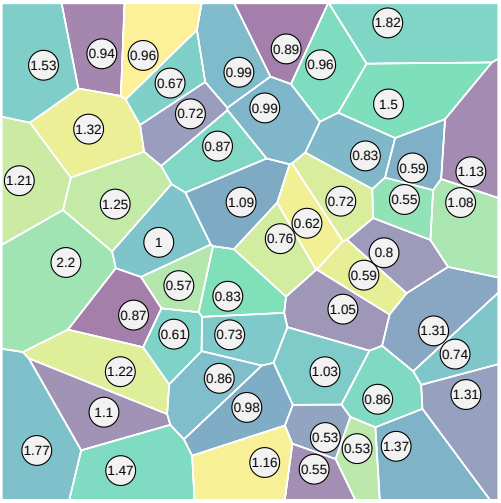
will hold.

The algorithm is illustrated in Figure 3.3. For the given initial random positions $P_1^{(0)}, \ldots, P_K^{(0)}$ shown in Figure 3.3a, it converges in 5 steps.

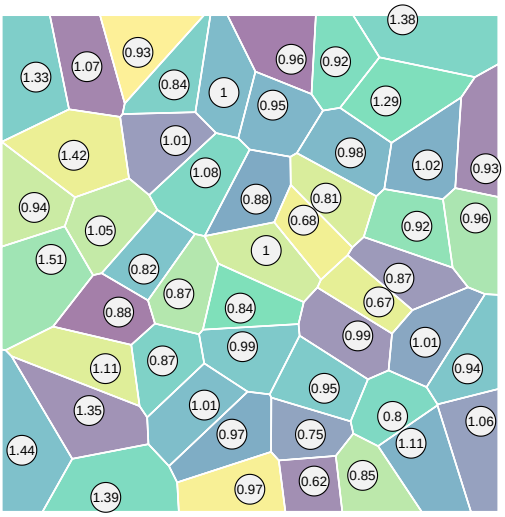
3.2. Constrained Case

Now suppose we are given initial locations $P_1, \ldots, P_K$ which satisfy

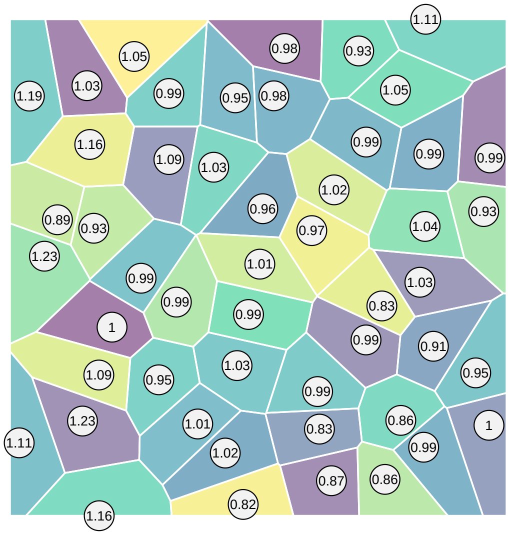
$$P_k \in \mathcal{N},\ k = 1, \ldots, K.$$



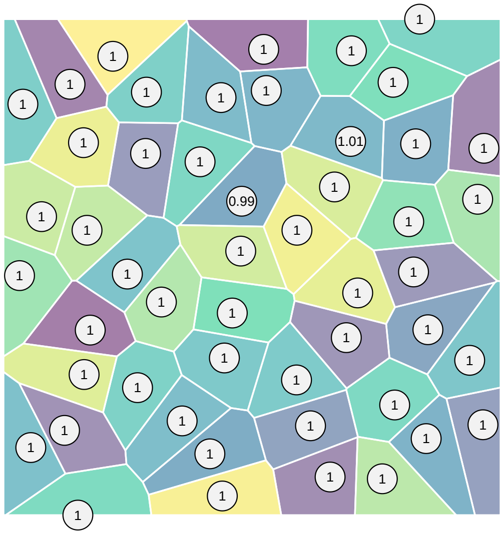
(a) Initial (**random**) positions $P_1^{(0)}, \ldots, P_K^{(0)}$.



(b) **Step 1**: positions $P_1^{(1)}, \ldots, P_K^{(1)}$.



(c) **Step 2**: positions $P_1^{(2)}, \ldots, P_K^{(2)}$.



(d) **Step 4**: positions $P_1^{(4)}, \ldots, P_K^{(4)}$.

Figure 3.3. Illustration of the unconstrained equal area Voronoi tessellation algorithm (**Steps 3 & 5** omitted). Colors for the cells are chosen randomly but kept constant throughout the figures.

Our goal in this section will be to improve these locations to new positions $P'_1, \ldots, P'_K$ while keeping to the constraint

$$P'_k \in \mathcal{N},\ k = 1, \ldots, K.$$

3.2.1. LP Formulation for Fixed Segments

By our definition, $\mathcal{N}$ is a union of segments:

$$\mathcal{N} := \cup_{m=1}^{M} \mathcal{S}_m.$$

Label the segments to which the initial positions $P_1, \ldots, P_K$ belong with $\mathcal{S}^k$:

$$P_k \in \mathcal{S}^k,\ k = 1, \ldots, K.$$

If we require that the positions belong to the same segments after the update,

$$P'_k \in \mathcal{S}^k,\ k = 1, \ldots, K,$$

due to the fact that all $\mathcal{S}_m,\ m = 1, \ldots, M$ are convex, we can still formulate our optimization problem as an LP problem.

Suppose that $\mathcal{S}_m$ has endpoints A_m, B_m and a parametric representation

$$\mathcal{S}_m = \{(1-\varphi)A_m + \varphi B_m, \} = \{A_m + \varphi(B_m - A_m)\} \text{ with } 0 \le \varphi \le 1.$$

Thus, the requirement $P'_k \in \mathcal{S}^k$ can be stated this way:

$$P'_k \in \mathcal{S}^k \iff P_k + \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} = A^k + \varphi(B^k - A^k) \text{ for some } 0 \le \varphi \le 1.$$

These are two linear equality constraints (one for x and one for y coordinate), which we can simply add to formulation (3.12), to obtain the LP program below:

$$\min_{\mathbf{r}, \Delta\mathbf{x}, \Delta\mathbf{y}, \{\varphi_k\}} \quad \mathbf{1}^T \mathbf{r} \tag{3.16a}$$

$$\text{s.t.} \quad -\mathbf{r} \preceq \left(\vec{\mathbf{F}}^{(i)} + \mathbf{J}_{\mathbf{x}}^{(i)} \Delta\mathbf{x} + \mathbf{J}_{\mathbf{y}}^{(i)} \Delta\mathbf{y}\right) - \mathbf{1} \preceq \mathbf{r}, \tag{3.16b}$$

$$\mathbf{0} \preceq \mathbf{x}^{(i)} + \Delta\mathbf{x},\ \mathbf{y}^{(i)} + \Delta\mathbf{y} \preceq \mathbf{1}, \tag{3.16c}$$

$$\mathbf{A}_k^{(i)} \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} \succeq R\mathbf{1} - \mathbf{b}_k^{(i)}, \quad k = 1, \ldots, K, \tag{3.16d}$$

$$P_k^{(i)} + \begin{pmatrix} \Delta x_k \\ \Delta x_k \end{pmatrix} = A^k + \varphi_k(B^k - A^k), \quad k = 1, \ldots, K, \tag{3.16e}$$

$$0 \le \varphi_k \le 1, \qquad k = 1, \ldots, K. \tag{3.16f}$$

Now, starting at positions $P_1^{(i)} \in \mathcal{S}^1, \ldots, P_K^{(i)} \in \mathcal{S}^K$, we can iteratively solve problem (3.16) to improve the positions to

$$P_1^{(i+1)} \in \mathcal{S}^1, \ \ldots, \ P_K^{(i+1)} \in \mathcal{S}^K,$$

similarly as was described in Section 3.1.6. The procedure is illustrated in Figure 3.4.

The segment-constrained equal area Voronoi tessellation algorithm does not work as well as the unconstrained one: improvement is slower. Furthermore, often the algorithm does not converge: after some iterations, the algorithm starts to jump back an forth without improving, or even obtaining worse positions. The main reason for this is probably related with the non-overlapping circles constraint $\|P_k - P_l\|_2 \geq 2\varepsilon, \ k \neq l$. For convergence, one could try adjusting δ as discussed in (3.12e).

3.2.2. Techniques for Escaping a Suboptimal Solution

3.2.2.1. MIP Formulation

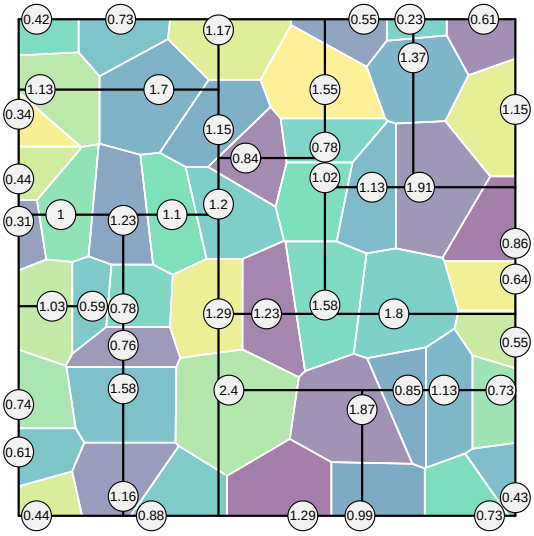
To escape a non-optimal solution we are stuck with by iteratively solving fixed segment LP problem (3.16), we extend our formulation to a MIP problem by using the ideas described in Section 1.1. We repeat the mathematical formulation of the constrained equal area Voronoi tessellation problem:

$$\min_{P_1, \ldots, P_K} \quad \sum_{k=1}^{K} |F_k(P_1, \ \ldots, \ P_K) - 1| \tag{3.17a}$$

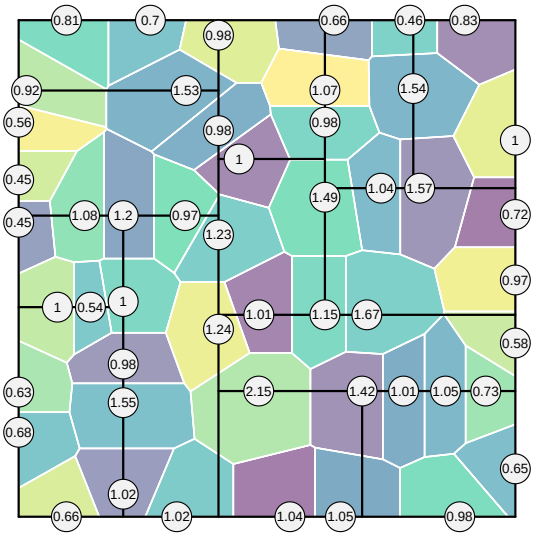
$$\text{s.t.} \quad \|P_k - P_l\|_2 \geq 2\varepsilon, \quad k \neq l, \tag{3.17b}$$

$$P_k \in \mathcal{N}, \quad k = 1, \ldots, K. \tag{3.17c}$$

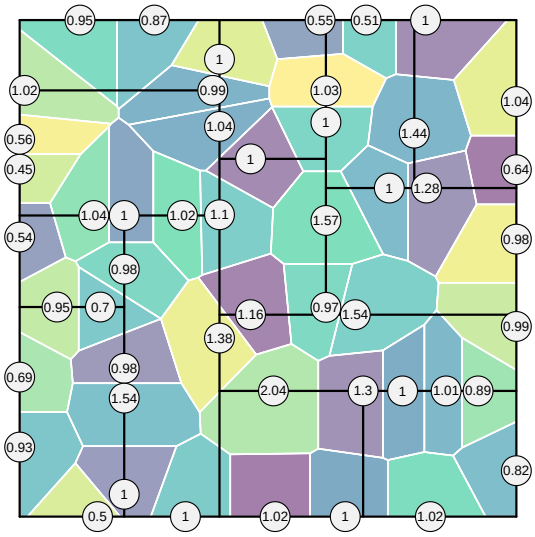
The formulation of the first 2 equations as an LP program was discussed in detail in Section 3.1.4 and summarized in (3.12), and the formulation of the constraint $P_k \in \mathcal{N}$ was given in (1.3). Together, these equations result in a MIP program, which we could already give to a MIP solver; but here we note one important improvement. To handle the constraint $\|P'_k - P'_l\|_2 \geq 2\varepsilon$, we introduced a requirement that the updated position P'_k would belong to the Voronoi cell $\mathcal{V}_k$ of the initial position P_k with a "wall" of thickness ε (recall Figure 3.2a). Thus, we



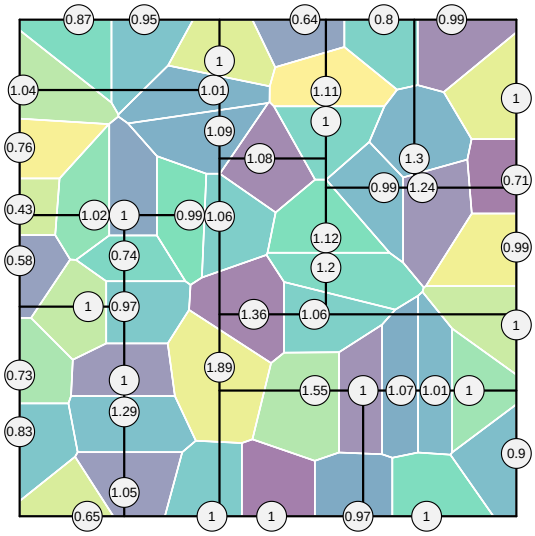
(a) Positions $\left\{P_k^{(0)}\right\}$. Loss ≈ 0.3675.



(b) Positions $\left\{P_k^{(2)}\right\}$. Loss ≈ 0.2451.



(c) Positions $\left\{P_k^{(5)}\right\}$. Loss ≈ 0.1836.



(d) Positions $\left\{P_k^{(10)}\right\}$. Loss ≈ 0.1418.

Figure 3.4. Illustration of the segment-constrained equal area Voronoi tessellation algorithm. Loss is equal to $\frac{1}{K}\sum_{k=1}^{K}|F_k(P_1,\ldots,P_K)-1|$.

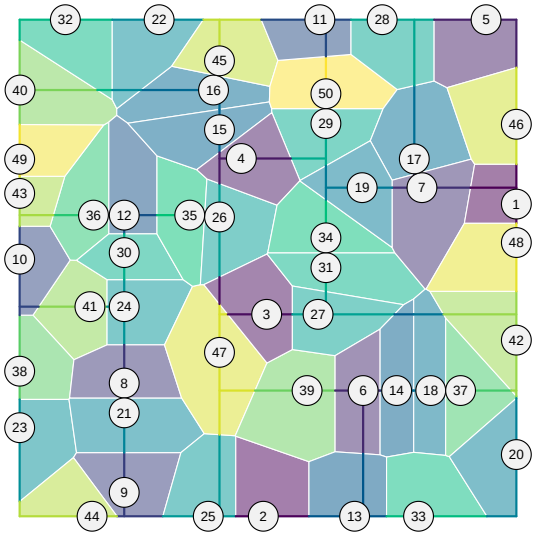


Figure 3.5. Illustration of the constraint $P'_k \in \mathcal{N} \cap \mathcal{V}_k$. Segments are colored according to the Voronoi cell they belong to. The numbers within circles show the index of the point.

can change (3.17c) to a requirement

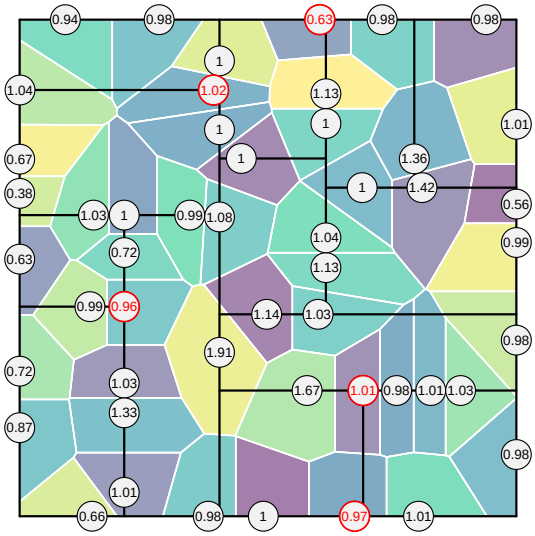
$$P'_k \in \mathcal{N} \cap \mathcal{V}_k, \tag{3.18}$$

e.g., in our case, for the MIP formulation, it is enough to consider only the part of $\mathcal{N}$ which is within Voronoi cell $\mathcal{V}_k$, as illustrated in Figure 3.5. This significantly shrinks the number of binary variables we need to introduce; in fact, for many of the constraints $P'_k \in \mathcal{N} \cap \mathcal{V}_k$ – in case Voronoi cell $\mathcal{V}_k$ contains only a part of a single segment – we don't need any binary variables.

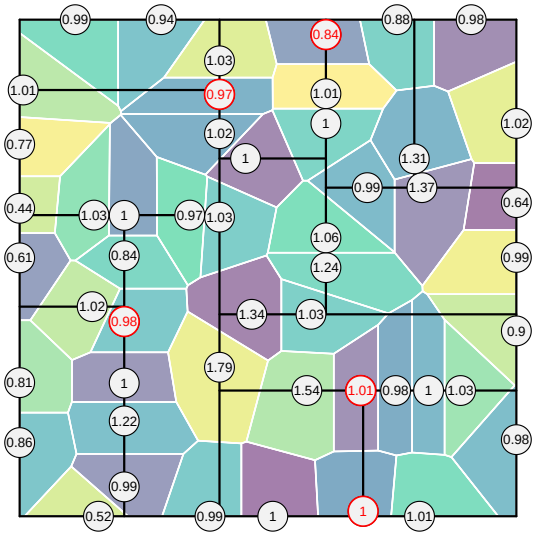
The MIP formulation allows us to escape (at the expense of a more complicated problem) a suboptimal solution obtained by the fixed segments algorithm by switching segments for some locations, as illustrated in Figure 3.6.

3.2.2.2. Replacing the Location of the Point with the Minimum Load

Another technique we use to improve a solution is driven by the following simple intuition: if for some point the load is low, why not move it



(a) Suboptimal positions obtained by iteratively solving an LP problem (3.16). Loss ≈ 0.1379.



(b) Switching to neighboring segments by solving a MIP problem. Loss ≈ 0.1251.

Figure 3.6. Escaping a suboptimal solution by solving a MIP problem; red-colored points switch the segment.

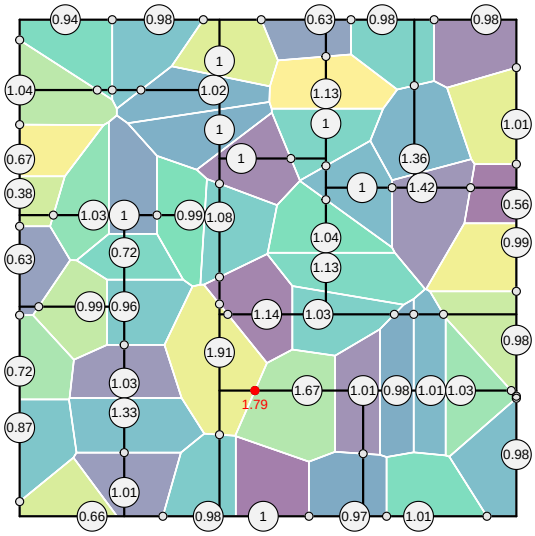


Figure 3.7. Picking the location for point replacement.

somewhere to a place where neighboring points bear high load? This idea is illustrated in Figure 3.8. The new location of the point is chosen by the following procedure (see Figure 3.7):

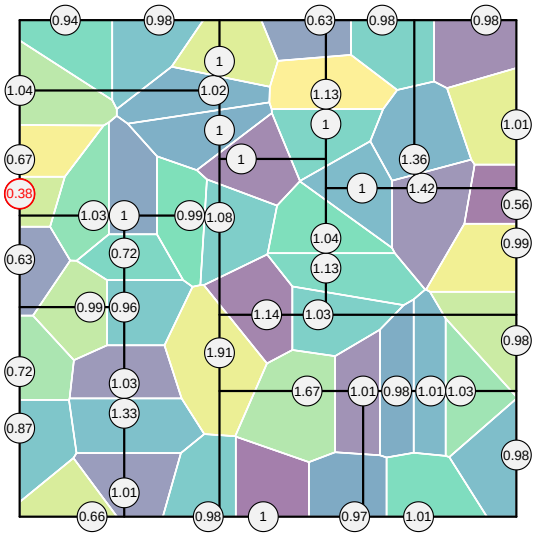
- Compute the set $\{Q_l\}$ of points where the edges of the Voronoi tessellation intersect with the net $\mathcal{N}$.
- Keep only those which satisfy the inequalities

$$\|P_k - Q_l\|_2 \geq 2\varepsilon, \quad k = 1, \ldots, K.$$

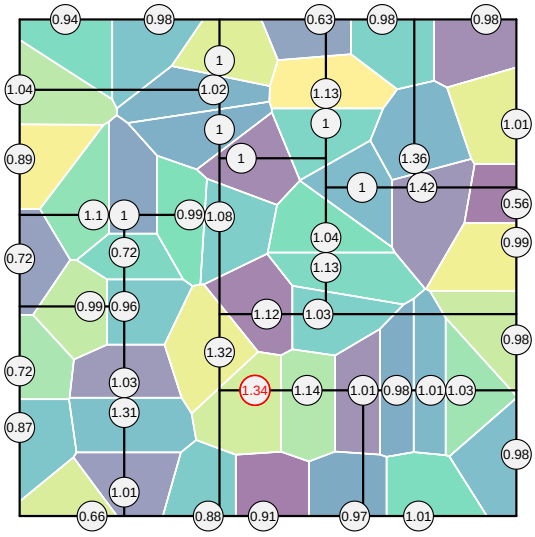
- Each intersection point belongs to an edge shared by two Voronoi cells $\mathcal{V}_{k_1}, \mathcal{V}_{k_2}$. Define the score for each point in the set $\{Q_l\}$ to be equal to the average of the two loads $\frac{F_{k_1}+F_{k_2}}{2}$. Now, pick the point with the highest score.

3.2.3. Summary of the Algorithm

We now summarize the final algorithm we have used for the site-constrained equal area Voronoi tessellation problem. Since to our knowledge, the problem has not been considered in scientific literature before, this is the first method proposed.



(a) Initial position.



(b) Updated position.

Figure 3.8. Replacement of the point with the minimum load.

The algorithm simultaneously seeks to find the optimal solutions for two objective functions. The first one is the mean absolute loss:

$$l_{\text{abs}}(\mathbf{P}) := \frac{1}{K}\sum_{k=1}^{K} |F_k(P_1, \ldots, P_K) - 1| \,. \tag{3.19a}$$

The second objective function is the maximum absolute loss:

$$l_{\max}(\mathbf{P}) := \max_{k=1,\ldots,K} |F_k(P_1, \ldots, P_K) - 1| \,. \tag{3.19b}$$

Label with $\mathbf{P}^{\text{abs}}$ the best found solution for loss l_{abs} and with $\mathbf{P}^{\max}$ the best solution for loss $l_{\max}$. During all the program, keep record of the best known solution in l_{abs} or $l_{\max}$ sense by comparing all the obtained solutions with the optimal ones.

Step 0. Random initialization. Start by randomly sampling points $P_1, \ldots, P_K$ on the constrained set $\mathcal{N}$, while keeping to the constraint $\|P_k - P_l\|_2 \geq 2\varepsilon$. At this step, to arrive at a satisfactory solution faster, one can repeatedly replace the location of the point with the minimum load as described in Section 3.2.2.2, until such replacement improves the solution. However, our experiments imply that this procedure does not seem to improve the final result; in fact, it looks like this makes the algorithm more likely to be stuck in a neighborhood of a worse suboptimal solution.

Step 1. Optimize $l_{\max}$ for fixed segments. Solve the optimization on segments LP program (3.16) with the objective function given by equations (3.15). Label the new solution obtained with $\mathbf{P}'$. If $\mathbf{P}'$ is significantly better than current solution $\mathbf{P}$ in $l_{\max}$ sense, i.e., $l_{\max}(\mathbf{P}') < (1-\epsilon)l_{\max}(\mathbf{P})$ for some small $\epsilon > 0$, then $\mathbf{P} := \mathbf{P}'$ and repeat this step. Otherwise, go to **Step 2**.

Step 2. Escape a suboptimal $l_{\max}$ solution by allowing segment switching. In case we cannot further improve locations by solving fixed segments LP program in **Step 1**, formulate and solve a MIP problem as described in Section 3.2.2.1. Label the solution obtained with $\mathbf{P}'$. If $l_{\max}(\mathbf{P}') < (1-\epsilon)l_{\max}(\mathbf{P})$, then $\mathbf{P} := \mathbf{P}'$ and return to **Step 1** – restart fixed segment optimization. Otherwise, go to **Step 3**.

Step 3. Optimize l_{abs} for fixed segments. Now, similarly as in **Step 1**, iteratively solve LP problem (3.16) for loss l_{abs}. Repeat this step until

updated locations $\mathbf{P}'$ improve current locations $\mathbf{P}$, i.e., while equation $l_{\text{abs}}(\mathbf{P}') < (1-\epsilon) l_{\text{abs}}(\mathbf{P})$ is satisfied.

Step 4. Escape a suboptimal l_{abs} solution by allowing segment switching. Similarly as in **Step 2**, formulate and solve a MIP problem which allows to escape a suboptimal solution. If obtained positions $\mathbf{P}'$ give an improvement, return to **Step 3** – restart fixed segments optimization for l_{abs} loss.

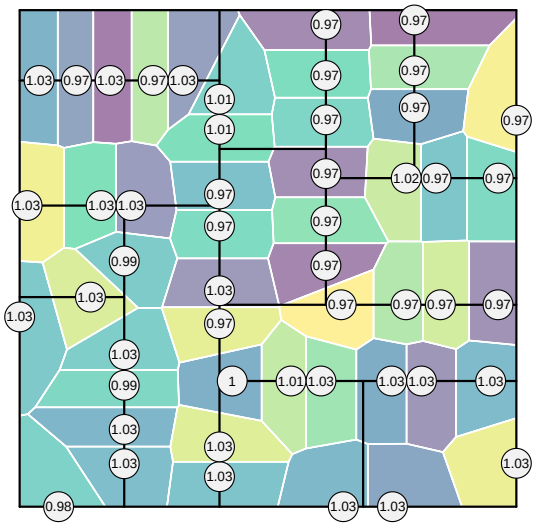
Step 5. Minimum location replacement. Replace the location of the point with the minimum load as described in section section 3.2.2.2, *even if this procedure does not improve the solution*. This step allows to escape a suboptimal or a locally optimal solution.

In the above procedure, **Steps 1 – 5** are repeated a predefined number of times or until time limit is reached.

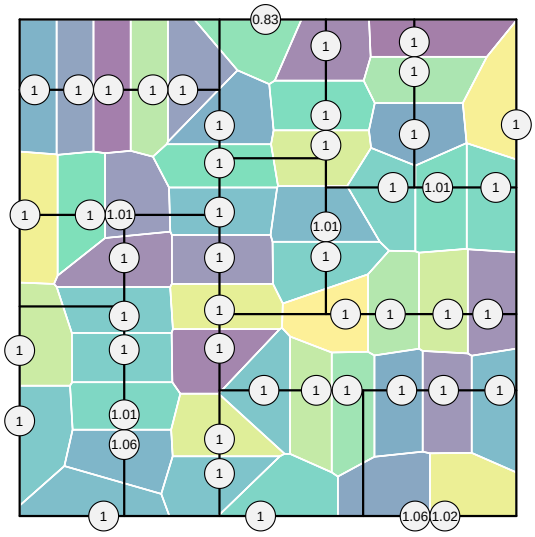
We give an important comment on our algorithm. Our choice to minimize $l_{\max}$ after point replacement (**Step 1** follows after **Steps 4 – 5**) or to replace a point with the minimum load after l_{abs} optimization (**Step 5** after **Steps 3 – 4**) is not accidental. Note that $l_{\max}$ optimization minimizes the loss for the extreme points; see Figure 3.9a, where many of the points have the load of $1 \pm l_{\max}(P_1, \ldots, P_K)$. That is, especially at the later iteration phases in the solution obtained in **Step 5**, the replaced point position is very likely to become the new minimum extreme. Thus, subsequent $l_{\max}$ optimization in **Step 1** somewhat equalizes it; this allows this point to avoid a replacement in a subsequent iteration. On the other hand, l_{abs} loss optimization seeks to obtain as many equal areas as possible, allowing the extreme loads to deviate more (see Figure 3.9b). Thus, for point replacement at **Step 5**, the choice of the minimum and the replacement locations can be made more reasonably.

3.2.4. Numerical Experiment

We ran the algorithm of Section 3.2.3 for several net-constrained Voronoi tessellation problem instances. The net $\mathcal{N}$ in all these instances consisted of the boundary of the unit square, with additional segments generated within the square, similarly as can be observed in the figures throughout this chapter. The goal was to optimize the locations of 50 sites of radius $\varepsilon = 0.03$ for various random instances of $\mathcal{N}$, i.e., our ultimate target



(a) $l_{\max}$ solution.



(b) l_{abs} solution.

Figure 3.9. The best found $l_{\max}$ and l_{abs} solutions.

Table 3.1. Optimization results for various problem instances. l_{abs} defined in (3.19a) and $l_{\max}$ defined in (3.19b).

Problem	l_{abs} loss	$l_{\max}$ loss
1	0.00815	0.03243
2	0.01383	0.03203
3	0.01202	0.05185
4	0.01443	0.05278
5	0.01286	0.04924
6	0.00980	0.03858
7	0.01337	0.08245
8	0.01298	0.10718
9	0.01287	0.07491
10	0.01166	0.05316

was to distribute 50 sites on the randomly generated nets so that the resulting Voronoi cells have equal areas.

The results are summarized in Table 3.1. The theoretical global solution is equal to 0 (in case all the resulting Voronoi cells have equal areas), however, the proposed algorithm could not reach this theoretically possible value for any problem instance considered. This indicates the limitations of our algorithm and leaves space for the research on future improvements. Whether the obtained losses as reported in Table 3.1 are sufficient depends on the application.

Our choice to run our experiments on such net-constrained Voronoi tessellation problems was motivated by the similarity of the generated frame with the sketch of the walls of the building. For future research, we expect to extend our work to real pile optimization problems and compare the results of the algorithm in Section 3.2.3 with simulated annealing and genetic evolution based methods, which, according to the literature (Belevičius et al. [4], Ramanauskas et al. [46]), give the best results for pile optimization problems.

3.3. Chapter Summary

In this chapter, we presented an iterative way to solve the constrained equal area Voronoi tessellation problem where intermediate steps re-

quire solving an LP or a MIP problem. We presented the formulations where the sites are constrained to belong to a union of a finite set of segments (e.g., a net), but the proposed method could be also extended to the case when the constrained set is a union of a finite set of arbitrary convex simplexes (i.e., segments and convex polygons) in the plane.

A note on the tools used. The results of this chapter were mainly built upon **[R]** packages **sf**[1] (for geometrical algorithms like Voronoi tessellation, polygon or segment intersection and etc.) and **lpSolve** (for linear and mixed-integer programming problems).

[1] Pebesma [40], Pebesma and Bivand [41].

Consider the following general clustering problem:

$$\min_{(\mathcal{C}_1,P_1),\ldots,(\mathcal{C}_K,P_K)} \quad \sum_{k=1}^{K}\sum_{i\in\mathcal{C}_k} w_i d(A_i,P_k) \tag{4.1a}$$

$$\text{s.t.} \quad P_k \in \mathcal{R}^{\cup}, \quad k=1,\ldots,K. \tag{4.1b}$$

Here, pairs $\{(A_n,w_n),\ n=1,\ldots,N\}$ represent client locations and their weights (demands), which are given as problem input. Optimization variables are clusters $\{\mathcal{C}_k\}$ and cluster centers $\{P_k\}$.

As was already discussed in Section 2.4.1, given **fixed** clusters $\mathcal{C}_1,\ldots,\mathcal{C}_K$, optimization of (4.1) results in K single location problems

$$\min_{P_k\in\mathcal{R}^{\cup}} \sum_{i\in\mathcal{C}_k} w_i d(A_i,P_k), \quad k=1,\ldots,K, \tag{4.2}$$

which can be solved independently. Alternatively, given **fixed** cluster centers $P_1,\ \ldots,\ P_K$, we can easily determine optimal clusters by assigning each element to the closest center. This "fixed clusters/fixed centers" idea is the core of any k-means-type algorithm (Cooper [14, 15], Lloyd [34]). To be more precise, for a start suppose we are given initial (random) cluster centers $P_1,\ \ldots,\ P_K$. Any k-means-type algorithm can be summarized by these two steps:

Assignment step. Fix cluster centers $P_1,\ \ldots,\ P_K$. Update clusters $\mathcal{C}_1,\ \ldots,\ \mathcal{C}_K$ by the following rule (assign each client point A_i to the closest cluster-center point P_k):

$$\mathcal{C}_k = \left\{ i : d(A_i,P_k) = \min_{k'=1,\ldots,K} d(A_i,P_{k'}) \right\}.$$

Location step. Fix clusters $\mathcal{C}_1,\ \ldots,\ \mathcal{C}_K$ as known, i.e., as was obtained in the assignment step. Optimize each cluster center by solving problem (4.2). With these new locations, return to the allocation step.

It is easy to notice that either after assignment step or after location step, the loss defined in (4.1a) decreases. Indeed, in the assignment step, we improve element assignments, and in the location step, we improve cluster centers; therefore, in both cases the loss must decrease.

The above two steps are repeated until convergence, i.e., until both the assignment step and the location step do not further improve the

solution. We call the final output obtained at the termination a *locally optimal* solution.

The k-means-type algorithm for the constrained MSSC problem (with $d(A_n, P_k) := \|A_n - P_k\|_2^2$) is illustrated in Figure 4.1. In the example, the constrained set $\mathcal{R}^\cup \equiv \mathcal{N}$, i.e., $\mathcal{R}^\cup$ is a net.

In the rest of this chapter, we concentrate on solving the single location problem (4.2), as the assignment step in any location-allocation algorithm is straightforward given the implementation of the distance function d (which in itself can be not so straightforward, e.g., in the presence of barriers as we will see).

4.1. Constrained Mean Point

Consider that the loss for cluster $\mathcal{C}_k$ is defined by the following formula:

$$\mathbf{Loss}_2(P) := \sum_{i \in \mathcal{C}_k} w_i \, \|A_i - P\|_2^2 \,. \tag{4.3}$$

Center position P is a variable in (4.3). We want to solve problem

$$\min_{P \in \mathcal{R}^\cup} \mathbf{Loss}_2(P). \tag{4.4}$$

It can be easily shown[1] that the optimal center location in the unconstrained case (minimizer of (4.3) in the plane, $P \in \mathbb{R}^2$) is given by the weighted average formula:

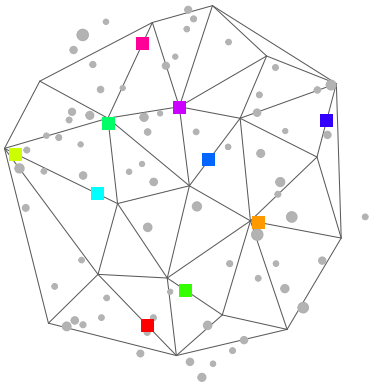
$$\overline{P} = \frac{1}{w(\mathcal{C}_k)} \sum_{i \in \mathcal{C}_k} w_i A_i, \quad \text{here} \quad w(\mathcal{C}_k) = \sum_{i \in \mathcal{C}_k} w_i.$$

Also, it can be shown that for any $P \in \mathbb{R}^2$ (see [A.1]),

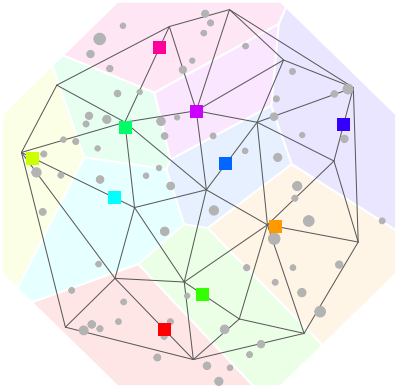
$$\mathbf{Loss}_2(P) = \mathbf{Loss}_2(\overline{P}) + w(\mathcal{C}_k) \left\|P - \overline{P}\right\|_2^2 .$$

From the above equation it is easy to notice that the optimal solution to (4.4) is simply the "projection" $\mathbf{proj}(\overline{P}|\mathcal{R}^\cup)$; i.e., we have to find the closest point to $\overline{P}$ in $\mathcal{R}^\cup$. Since $\mathcal{R}^\cup$ is a union of convex simplexes, a straightforward procedure to find $\mathbf{proj}(\overline{P}|\mathcal{R}^\cup)$ is as follows:

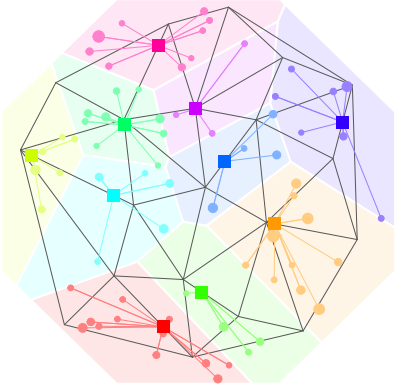
[1] By solving the linear equations resulting from first-order derivative conditions for each coordinate.



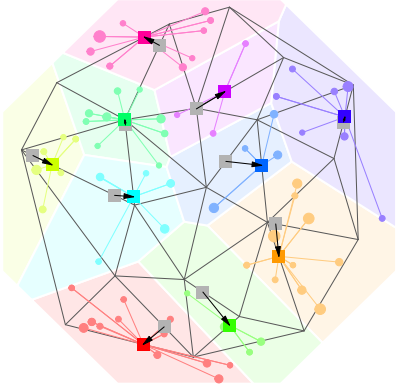
(a) **Initialize**: sample random centers on $\mathcal{R}^{\cup}$.



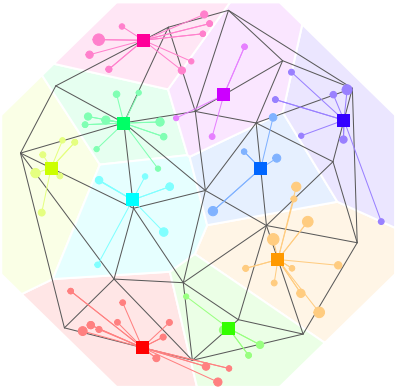
(b) Voronoi cells of random centers.



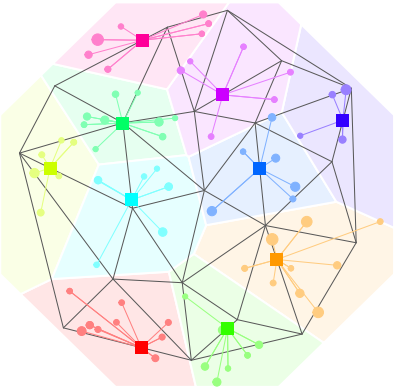
(c) **Assignment Step**: assign each point to the closest center.



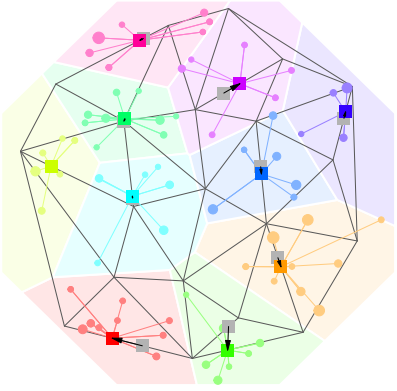
(d) **Location Step**: optimize the center for each cluster keeping to the constraint $P_k \in \mathcal{R}^{\cup}$.



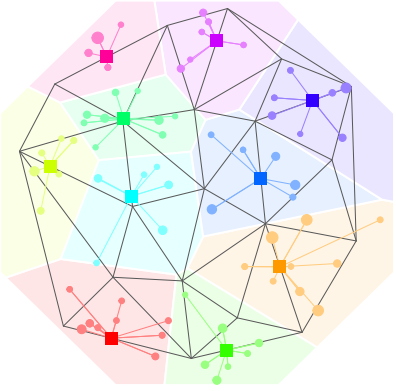
(e) Optimized centers generate new Voronoi cells, some points "switch" their cluster.



(f) **Assignment Step**: update point assignment.



(g) **Location Step**: optimize centers.



(h) **Converged**: algorithm output.

Figure 4.1. **Net-constrained k-means algorithm visualization** [A.1]. The Voronoi cells of cluster centers are shown here just for illustration purposes, they are not computed by the actual algorithm. Note that in all the figures, cluster centers (represented by little squares) satisfy $P_k \in \mathcal{N}$. Note also that there are more steps after Figure 4.1g and before the final algorithm output in Figure 4.1h (not shown here).

• If any region-unit $\mathcal{R}_m$ contains $\overline{P}$, then $\mathbf{proj}(\overline{P}|\mathcal{R}^{\cup}) = \overline{P}$. (Checking whether a convex polygon contains a point is a well known problem in computational geometry.)

• In case none of $\mathcal{R}_m$ contains $\overline{P}$, the closest point is on the boundary of some region-unit $\mathcal{R}_m$. Label the boundary[2] of $\mathcal{R}_m$ with $\partial\mathcal{R}_m$ and define $\partial\mathcal{R}^{\cup} := \cup_{m=1}^{M} \partial\mathcal{R}_m$. In case each region-unit $\mathcal{R}_m$ is a polygon, $\partial\mathcal{R}^{\cup}$ is a union of a finite number of segments:

$$\partial\mathcal{R}^{\cup} = \cup_{l=1}^{L} \mathcal{E}_l.$$

Now, to find $\mathbf{proj}(\overline{P}|\mathcal{R}^{\cup})$, we can simply find the projections of $\overline{P}$ on each of the segments $\mathcal{E}_l$, $l = 1, \ldots, L$, and pick the best point. (Finding the projection of a point on a segment is a simple geometrical problem, for details, see [A.1]).

Following the two items above, given $\overline{P}$, the solution to (4.4) can be found in $\mathcal{O}(L)$ number of steps. We note however that in case $\mathbf{proj}(\overline{P}|\mathcal{R}^{\cup})$ has to be determined for different $\overline{P}$ many times (e.g., for many different clusters $\mathcal{C}_k \subseteq \{1, \ldots, N\}$), this is not the most efficient way possible. For an improvement in algorithmic complexity, one can, e.g., consider using the trapezoidal map data structure which gives a method to locate $\overline{P}$ in $\mathcal{O}(\log(L))$ time (see the related chapter in Berg et al. [5]).

4.2. Constrained Weber Point

Now we turn to the problem

$$\min_{P \in \mathcal{R}^{\cup}} \mathbf{Loss}_1(P), \tag{4.5}$$

with

$$\mathbf{Loss}_1(P) := \sum_{i \in \mathcal{C}_k} w_i \left\| A_i - P \right\|_2^1 = \sum_{i \in \mathcal{C}_k} w_i \left\| A_i - P \right\|_2. \tag{4.6}$$

Consider the unconstrained optimal solution P^*,

$$P^* := \arg\min_{P \in \mathbb{R}^2} \mathbf{Loss}_1(P). \tag{4.7}$$

[2]This is the set of points $P \in \mathcal{R}_m$ such that for any $\varepsilon > 0$, the "ball" of radius ε with a center at P contains some point $P' \notin \mathcal{R}_m$.

Contrary to the loss $\mathbf{Loss}_2$, the constrained solution of (4.5) is not necessarily $\mathbf{proj}(P^*|\mathcal{R}^\cup)$. Furthermore, the point P^* itself can only be found in a limit via some iterative procedure, e.g., via the algorithm of Weiszfeld [52]. These issues make problem (4.5) subtle. Thus, we consider only the case when $\mathcal{R}^\cup \equiv \mathcal{N}$, i.e., when the constrained set is a union of segments,[3]

$$\mathcal{N} = \cup_{l=1}^{L} \mathcal{E}_l.$$

4.2.1. Optimal Solution on a Segment

Now consider the problem

$$\min_{P \in \mathcal{E}_l} \mathbf{Loss}_1(P),$$

i.e., when P is restricted to some segment of $\mathcal{N}$. Suppose this segment has endpoints Q_l^A and Q_l^B. Label the length of this segment with $L := \left\|Q_l^B - Q_l^A\right\|_2$ and define a unit vector $\vec{\mathbf{u}}_l := \frac{Q_l^B - Q_l^A}{L}$. Then, segment $\mathcal{E}_l$ has a parametric representation

$$\mathcal{E}_l = \left\{Q_l^A + t\vec{\mathbf{u}}_l,\ 0 \le t \le L\right\}.$$

Thus, on this segment, $\mathbf{Loss}_1$ can be optimized as a single variable function $f : [0, L] \mapsto \mathbb{R}$, with

$$f(t) = \mathbf{Loss}_1(Q_l^A + t\vec{\mathbf{u}}_l).$$

Now consider the derivative of f at $t = 0$ and at $t = L$. Using the results of multivariate calculus,[4]

$$f'(0) = \langle \nabla \mathbf{Loss}_1(Q_l^A),\, \vec{\mathbf{u}}_l \rangle, \quad f'(L) = \langle \nabla \mathbf{Loss}_1(Q_l^B),\, \vec{\mathbf{u}}_l \rangle. \tag{4.8}$$

Since $\mathbf{Loss}_1$ is convex (as a weighted sum of cones $\|A_i - P\|_2,\ i \in \mathcal{C}_k$), f is also a convex. This means that $f'(t_A) \le f'(t_B)$ for $t_A \le t_B$. Convexity also implies that tangent lines

$$f_A(t) := f(0) + f'(0)t, \quad f_B(t) := f(L) + f'(L)(t - L), \quad t \in [0, L], \tag{4.9}$$

are "underestimators" of f: $f(t) \ge \max\{f_A(t), f_B(t)\}$ (see Figure 4.2).

[3] Actually, Hansen et al. [27] presents an algorithm to find the solution of (4.5) when $\mathcal{R}^\cup$ is a union of convex polygons. They, however, ground their algorithm on an assumption that P^* is known exactly. Furthermore, in Chapter 5, we will also need a "tight" lower bound for the loss in (4.5). It is exactly this issue which makes the problem subtle; considering only $\mathcal{R}^\cup \equiv \mathcal{N}$ simplifies the estimation of such lower bound.

[4] The derivative of a multivariate function F at point $\mathbf{x}$ in a unit direction $\vec{\mathbf{u}}$ is equal to $\langle \nabla F(\mathbf{x}),\, \vec{\mathbf{u}} \rangle$ (see, e.g., Boyd and Vandenberghe [10]).

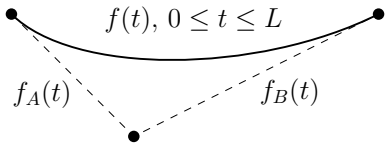


Figure 4.2. Convex function "underestimating" tangent lines.

Consider three possible scenarios:

1. $0 \le f'(0)$. We have

$$f(t) \ge f_A(t) = f(0) + f'(0)t \ge f(0), \quad t \in [0, L].$$

   That is, the optimal solution of $\min_t f(t)$ on interval $[0, L]$ is at $t = 0$.

2. $f'(L) \le 0$. We have

$$f(t) \ge f_B(t) = f(L) + f'(L)(t - L) \ge f(L), \quad t \in [0, L].$$

   The optimal solution of $\min_t f(t)$ on interval $[0, L]$ is at $t = L$.

3. $f'(0) < 0 < f'(L)$. In this case solution of $\min_t f(t)$ is somewhere in the interval $(0, L)$.

It is only the third case when we do not know the optimal value of $\min_t f(t)$. To describe the algorithm solving the problem, we introduce two variables t_A and t_B which are initially defined as

$$t_A := 0, \quad t_B := L.$$

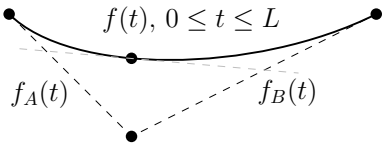
With this notation, tangent lines in (4.9) can be written as

$$f_A(t) = f(t_A) + f'(t_A)(t - t_A), \quad f_B(t) = f(t_B) + f'(t_B)(t - t_B).$$

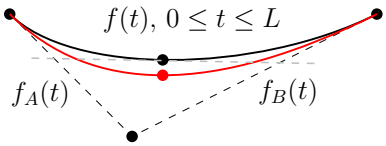
Consider the value of t where tangent lines f_A and f_B cross (i.e., $f_A(t_C) = f_B(t_C)$):

$$t_C := \frac{f(t_B) - f(t_A) - f'(t_B)t_B + f'(t_A)t_A}{f'(t_A) - f'(t_B)}. \tag{4.10}$$

The value $f^{LB} := f_A(t_C)\,(= f_B(t_C))$ is a lower bound for $\min_t f(t)$. Based on this, one can consider the following two approaches for solving problem $\min_t f(t)$.



(a) Illustration of the Lower Bound Method.



(b) Illustration of the Bezier Method.

Figure 4.3. Possible approaches to convex function optimization. For the Bezier Method, the red curve is a quadratic approximation of f.

Lower Bound Method. Calculate t_C via (4.10). Define $Q_l^C := Q_l^A + t_C \vec{\mathbf{u}}_l$. Calculate $f(t_C) = \mathbf{Loss}_1(Q_l^C)$ and the derivative of f at t_C:

$$f'(t_C) = \langle \nabla \mathbf{Loss}_1(Q_l^C),\ \vec{\mathbf{u}}_l \rangle.$$

If $f(t_C) - f^{LB} < \varepsilon$, we have achieved the desired accuracy. Report t_C as the solution to $\min_t f(t)$. Otherwise, there are three possible scenarios:

1. If $f'(t_C) > 0$, this means that the solution of $\min_t f(t)$ lies within interval $[t_A, t_C]$. Redefine $t_B := t_C$ and consider tangent lines

$$f_A(t) := f(t_A) + f'(t_A)(t - t_A), \quad f_B(t) := f(t_C) + f'(t_C)(t - t_C)$$

in the next iteration.

2. If $f'(t_C) < 0$, this means that the solution of $\min_t f(t)$ lies within interval $[t_C, t_B]$. Redefine $t_A := t_C$ and consider tangent lines

$$f_A(t) := f(t_C) + f'(t_C)(t - t_C), \quad f_B(t) := f(t_B) + f'(t_B)(t - t_B)$$

in the next iteration.

3. Otherwise (if $f'(t_C) = 0$), t_C is the solution of $\min_t f(t)$.

The method is illustrated in Figure 4.3a.

Bezier Method. Calculate t_C via (4.10). The three points

$$A = (t_A, f(t_A)),\ C = (t_C, f^{LB}),\ B = (t_A, f(t_B))$$

can be seen to define a quadratic Bezier curve (i.e., a portion of a parabola in the plane, see Figure 4.3b). This curve is given by the following parametric representation:

$$\mathcal{B}(\varphi) = (1-\varphi)^2 A + 2\varphi(1-\varphi)C + \varphi^2 B, \quad 0 \le \varphi \le 1.$$

In particular, f is "approximated" by function

$$\widetilde{f}(\varphi) = (1-\varphi)^2 f(t_A) + 2\varphi(1-\varphi) f^{LB} + \varphi^2 f(t_B), \quad 0 \le \varphi \le 1.$$

To find the minimizer of $\widetilde{f}$, we solve for the first-order condition

$$\widetilde{f}'(\varphi) = -2(1-\varphi) f(t_A) - (4\varphi - 2) f^{LB} + 2\varphi f(t_B) = 0.$$

This gives

$$\widetilde{\varphi} := \frac{f(t_A) - f^{LB}}{f(t_A) - 2f^{LB} + f(t_B)}, \tag{4.11}$$

which corresponds to

$$t^* = (1-\widetilde{\varphi})^2 t_A + 2\widetilde{\varphi}(1-\widetilde{\varphi}) t_C + \widetilde{\varphi}^2 t_B. \tag{4.12}$$

Define $Q_l^* := Q_l^A + t^* \vec{\mathbf{u}}_l$. Calculate $f(t^*) = \mathbf{Loss}_1(Q_l^*)$ and the derivative of f at t^*:

$$f'(t^*) = \langle \nabla \mathbf{Loss}_1(Q_l^*),\, \vec{\mathbf{u}}_l \rangle.$$

If $f(t^*) - f^{LB} < \varepsilon$, we have achieved the desired accuracy. Report t^* as the solution to $\min_t f(t)$. Otherwise, there are three possible scenarios:

1. If $f'(t^*) > 0$, this means that the solution of $\min_t f(t)$ lies within interval $[t_A, t^*]$. Redefine $t_B := t^*$ and consider tangent lines

$$f_A(t) := f(t_A) + f'(t_A)(t - t_A), \quad f_B(t) := f(t^*) + f'(t^*)(t - t^*)$$

in the next iteration.

2. If $f'(t^*) < 0$, this means that the solution of $\min_t f(t)$ lies within interval $[t^*, t_B]$. Redefine $t_A := t^*$ and consider tangent lines

$$f_A(t) := f(t^*) + f'(t^*)(t - t^*), \quad f_B(t) := f(t_B) + f'(t_B)(t - t_B)$$

in the next iteration.

3. Otherwise (if $f'(t^*) = 0$), t^* is the solution of $\min_t f(t)$.

The method is illustrated in Figure 4.3b.

Using either the Lower Bound Method or the Bezier Method, one obtains a sequence $t_1^*, t_2^*, \ldots$, which converges to the optimal solution of $\min_t f(t)$. At every iteration i, the estimates f_i^{LB} of the lower bound are available and are improving (increasing). However, the Bezier Method can be expected to have better convergence properties, since we solve a quadratic function approximation at every step.

4.2.2. Solution on a Union of Segments

To define an algorithm for the solution of problem

$$\min_{P\in\mathcal{N}} \mathbf{Loss}_1(P) \quad \equiv \quad \min_{P\in\mathcal{N}} \sum_{i\in\mathcal{C}_k} w_i \left\|A_i - P\right\|_2,$$

we can proceed as follows.

Initialization. Firstly, for each segment $\mathcal{E}_l$ in $\mathcal{N} \equiv \cup_{l=1}^{L}\mathcal{E}_l$ with endpoints Q_l^A and Q_l^B, define $L_l := \left\|Q_l^B - Q_l^A\right\|_2$ and a unit direction $\vec{\mathbf{u}}_l := \frac{Q_l^B - Q_l^A}{L_l}$. On each segment, consider function

$$f_l : [0, L_l] \mapsto \mathbb{R}, \quad \text{here} \quad f_l(t) := \mathbf{Loss}_1(Q_l^A + t\vec{\mathbf{u}}_l), \quad l = 1, \ldots, L.$$

Define/calculate

$$t_l^A := 0,\ t_l^B := L_l,\ f_l(t_l^A),\ f_l(t_l^B),\ f_l'(t_l^A),\ f_l'(t_l^B)$$

as outlined in Section 4.2.1. For each segment, calculate t_l^C via formula (4.10) and find the corresponding lower bound f_l^{LB} (i.e., determine the point where "underestimating" tangent lines cross).

Pick the best segment. Order segments $\mathcal{E}_l$ by increasing values of f_l^{LB}. Pick l with the smallest f_l^{LB}. For the corresponding $\mathcal{E}_l$, check the value at the best known point t_l^* on this segment: if $f_l(t_l^*) - f_l^{LB} < \varepsilon$, we have a solution and a lower bound within ε-accuracy, and the task is complete.

Improve the lower bound for the segment. Otherwise, perform a single iteration of the Lower Bound Method (or the Bezier Method) as described in Section 4.2.1. That is, calculate $f_l(t_l^C)$ (or $f_l(t_l^*)$ where t_l^* is defined via (4.12)), update t_l^A, t_l^B, equations for the corresponding tangent lines at these points, find an improved lower bound estimate for this segment using tangent lines at updated t_l^A, t_l^B, and return to **Pick the best segment** step.

4.3. Weber Point in the Presence of Polyhedral Barriers

We now consider the problem of finding the best location in the presence of barriers,

$$\min_{P\in\mathcal{R}^{\cup}} \mathbf{Loss}_{\mathcal{B}}(P) \quad \equiv \quad \min_{P\in\mathcal{R}^{\cup}} \sum_{i\in\mathcal{C}_k} w_i d_{\mathcal{B}}\left(A_i, P\right). \tag{4.13}$$

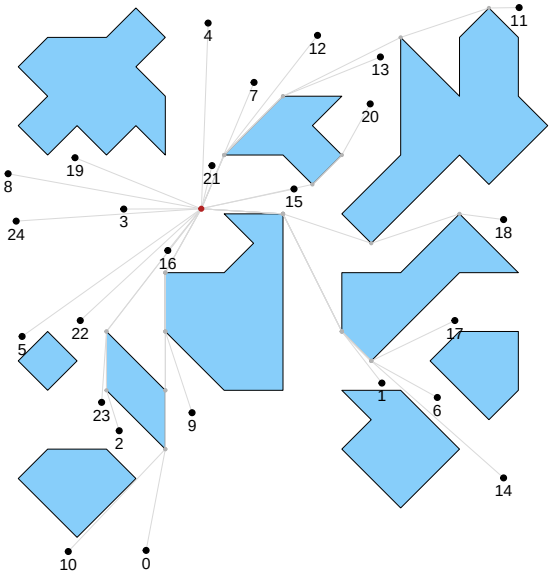


Figure 4.4. Optimal center for an example barriers problem. The blue polygons represent barriers where traveling is prohibited.

A solution to an example problem is illustrated in Figure 4.4.

One may get an impression that constrained set $\mathcal{R}^{\cup}$ is absent in Figure 4.4. Actually, we allow the center P to be anywhere within $\mathbb{R}^2 \setminus \cup_{l=1}^{L} \mathcal{B}_l$, but it can be shown that any optimal location for a barriers problem is always withing a convex hull of client points and barrier-corner points (Klamroth [30]). Since in our examples the barriers and the clients are always generated within a unit square, we can define

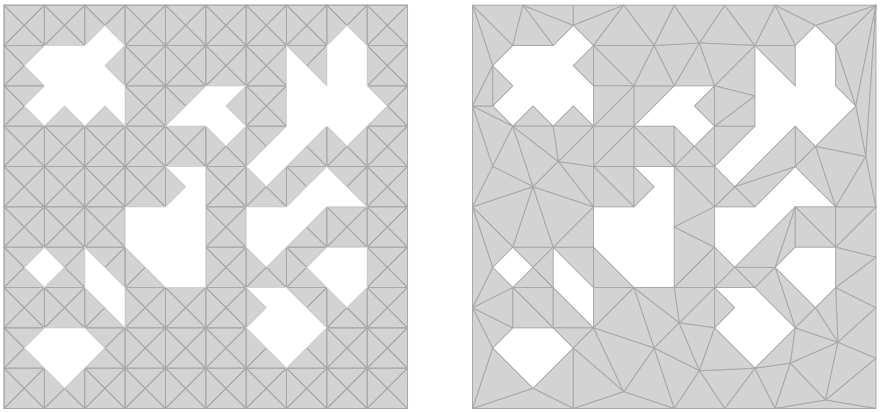
$$\mathcal{R}^{\cup} := [0,1]^2 \setminus \cup_{l=1}^{L} \mathcal{B}_l.$$

To put this in the form $\mathcal{R}^{\cup} = \cup_{m=1}^{M} \mathcal{R}_m$ with every $\mathcal{R}_m$ convex, we can, e.g., triangulate $\mathcal{R}^{\cup}$, as illustrated in Figure 4.5. The convex regions $\mathcal{R}_m$ are then simply the triangles obtained by the triangulation method.

Thus, we further assume that $\mathcal{R}^{\cup}$ is a union of **arbitrary small non-overlapping** segments and triangles,

$$\mathcal{R}^{\cup} \equiv \left(\cup_{i=1}^{I_T} \mathcal{T}_i \right) \cup \left(\cup_{j=1}^{I_S} \mathcal{S}_j \right).$$

By **non-overlapping**, we mean that the interiors of any pair of region units do not have common points. This assumption is not restrictive. For



(a) Original triangulation as compiled during problem generation.

(b) Another possible triangulation (generated using **[R] sfdtc** package).

Figure 4.5. Decomposing $\mathcal{R}^{\cup}$ into a union of triangles.

example, consider that $\mathcal{R}^{\cup}$ is a union of (possibly, overlapping) convex polygons and segments,

$$\mathcal{R}^{\cup} = \left(\cup_{i=1}^{I_P} \mathcal{P}_i\right) \cup \left(\cup_{i=1}^{\widetilde{I}_S} \widetilde{\mathcal{S}}_s\right).$$

Firstly, enumerate the segments in

$$\cup_{i=1}^{I_S} \mathcal{S}_i := \left(\cup_{i=1}^{\widetilde{I}_S} \widetilde{\mathcal{S}}_s\right) \setminus \left(\cup_{i=1}^{I_P} \mathcal{P}_i\right).$$

Then, triangulate $\cup_{i=1}^{I_P} \mathcal{P}_i$.

By **arbitrary small**, we mean that for a predefined $\delta > 0$, we have $\text{diam}(\mathcal{T}_i) \leq \delta$ and $\text{diam}(\mathcal{S}_j) \leq \delta$, where

$$\begin{aligned} &\text{diam}(\mathcal{T}_i) \quad \text{is the length of the longest side of triangle } \mathcal{T}_i, \\ &\text{diam}(\mathcal{S}_j) \quad \text{is the length of segment } \mathcal{S}_j. \end{aligned} \tag{4.14}$$

This is also not restrictive, because we can subdivide $\mathcal{T}_i$ into two triangles along its longest side, and repeat the procedure until the condition is satisfied for each subdivided part of $\mathcal{T}_i$. (The same applies to $\mathcal{S}_j$.)

In the rest of the section, we still use notation $\mathcal{R}^{\cup} = \cup_{m=1}^{M} \mathcal{R}_m$, but make an assumption that each $\mathcal{R}_m$ is either a triangle or a segment which pairwise do not overlap and are sufficiently small as previously discussed. By the term **subdivide** $\mathcal{R}_m$, we mean a division into two triangles along the longest edge for a triangle, and a division into two segments of equal lengths for a segment.

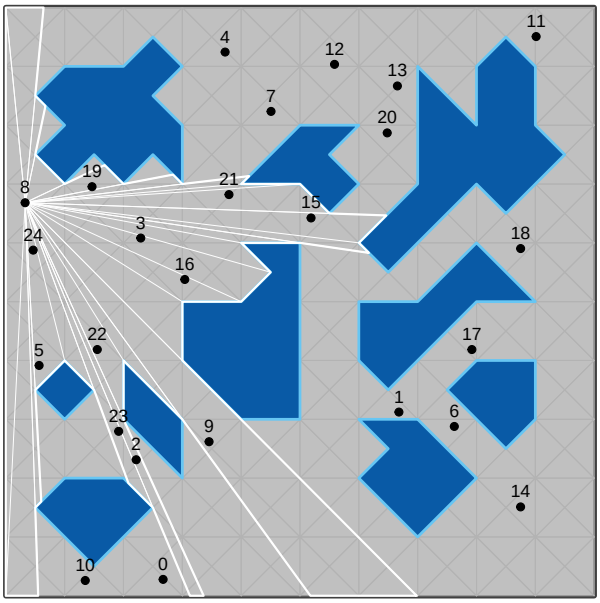


Figure 4.6. Visibility polygon for a client.

4.3.1. Visibility Graph

First of all, we consider the distance metric $d_{\mathcal{B}}$. To compute the distance $d_{\mathcal{B}}(P_1, P_2)$ between two points P_1 and P_2, we need to find the so-called visibility graph $\mathcal{G}(V, E)$ (Viegas and Hansen [51]). Any point $P \in \mathcal{R}^{\cup}$ defines a visibility polygon as illustrated in Figure 4.6 and Figure 4.7. Such visibility polygons contain the set of points seen from P directly through a ray; to reach the points outside this polygon, we must bypass some barriers. It is easy to convince oneself that (see [51]), in case point $P_2 \in \mathcal{R}^{\cup}$ is not seen directly from point $P_1 \in \mathcal{R}^{\cup}$, the shortest path between these points consists of a set of segments with the end-points at the corners of the barrier polygons. These barrier-corner points, together with client points $\{A_n, n = 1, \ldots, N\}$, define the set of vertices V for the visibility graph $\mathcal{G}(V, E)$. The (directed) edges E are inserted as follows:[5] if point $P(v)$ for $v \in V$ is seen from a barrier-corner point $P(w)$, i.e., $P(v)$ belongs to the visibility polygon of $P(w)$, we add an edge $w \mapsto v$

[5]We use notation $v(P)$ to refer to a vertex of $\mathcal{G}$ corresponding to point P, and $P(v)$ to refer to a point in the plane corresponding to vertex $v \in V$ ($P(v) \in \mathcal{R}^{\cup}$ is either a barrier-corner point or a client point).

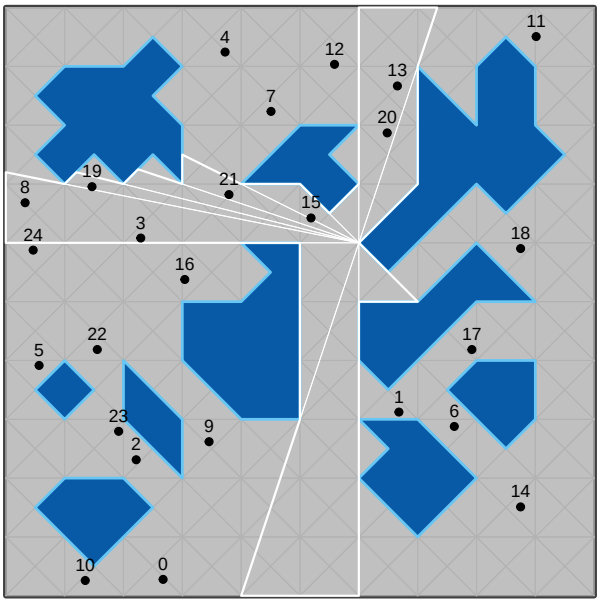


Figure 4.7. Visibility polygon for a barrier corner.

to the graph, with edge weight equal to the Euclidean length from point $P(v)$ to point $P(w)$.

To compute the shortest paths and their lengths (i.e., barrier-induced distances)

$$d_{\mathcal{B}}(A_1, P),\ d_{\mathcal{B}}(A_2, P),\ \ldots,\ d_{\mathcal{B}}(A_N, P)$$

from a given facility point P to each client point $A_1, \ldots, A_N$, we could insert an additional vertex $v(P)$ to the graph, compute the visibility polygon of P, find the set of vertices $v \in V$ seen from P, and insert the corresponding edges. The shortest paths can then be found with the algorithm of Dijkstra [16].

Visibility graph for an example barriers problem is shown in Figure 4.8.

4.3.2. Region-Unit Facilities

Knowing the distances from a facility point P to client points A_n does not yet help to find the optimal location, i.e., solve problem (4.13), as we can not check all the points $P \in \mathcal{R}^{\cup}$. Instead, we will consider the closest distance from each client A_n, $n = 1, \ldots, N$ to each region-unit

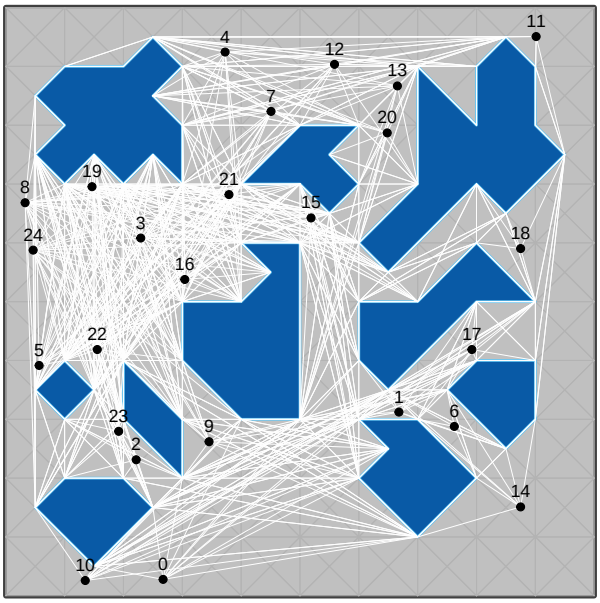


Figure 4.8. Visibility graph for the example problem.

$\mathcal{R}_m$, $m = 1, \ldots, M$. That is, define

$$d_{nm}^{\inf} := \min_{P \in \mathcal{R}_m} \; d_{\mathcal{B}}\left(A_n, P\right), \tag{4.15}$$

and label the solution to (4.15) with

$$\textbf{proj}(A_n|\mathcal{R}_m) := \arg\min_{P \in \mathcal{R}_m} \; d_{\mathcal{B}}\left(A_n, P\right). \tag{4.16}$$

We make the following observations:

- **proj**$(A_n|\mathcal{R}_m)$ belongs to the boundary $\partial\mathcal{R}_m$, for the shortest path from A_n to any $P \in \mathcal{R}_m \setminus \partial\mathcal{R}_m$ must cross boundary $\partial\mathcal{R}_m$ at some point P', indicating that $P' \in \partial\mathcal{R}_m$ is a closer point.
- Since the shortest path between any two points in $\mathcal{R}^{\cup}$ is defined by a set of segments (Viegas and Hansen [51]), the same holds for the shortest path between a point and a region unit.

From the last item it follows that the shortest path from A_n to $\mathcal{R}_m$ is either

1) via a direct travel through a ray if this ray does not cross any of the barriers,

2) via some intermediate $P(w)$ where w corresponds to a barrier-corner point.

Thus, we can find shortest paths from $\mathcal{R}_m$ to the client points $A_1, \ldots, A_N$ using an extension of the visibility graph $\mathcal{G} = (V, E)$ which was discussed in previous subsection. For this, firstly insert an additional vertex $v(\mathcal{R}_m)$ into the graph. To take care of the edges, for each $v \in V$, intersect the visibility polygon $\mathcal{VP}(v)$ of v with $\mathcal{R}_m$, i.e., compute $\mathcal{VP}(v) \cap \mathcal{R}_m$.

- In case $\mathcal{R}_m$ is a triangle, compute $\mathcal{VP}(v) \cap \mathcal{R}_m$ using polygon – polygon intersection algorithm. Since $\mathcal{VP}(v)$ in general is non-convex, it might be that $\mathcal{VP}(v) \cap \mathcal{R}_m$ is not a single, but a set of polygons.
- In case $\mathcal{R}_m$ is a segment, compute $\mathcal{VP}(v) \cap \mathcal{R}_m$ using segment – polygon intersection algorithm. Since $\mathcal{VP}(v)$ in general is non-convex, it might be that $\mathcal{VP}(v) \cap \mathcal{R}_m$ is not a single, but a set of segments.

Then:

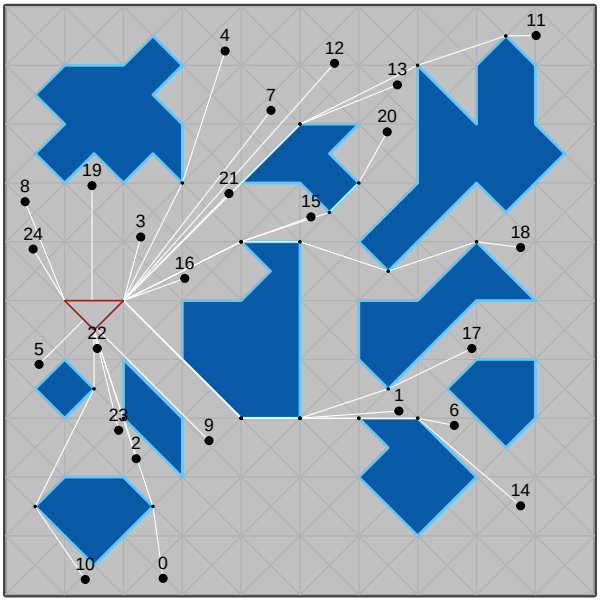
- If $\mathcal{VP}(v) \cap \mathcal{R}_m = \emptyset$, do nothing – this means region-unit $\mathcal{R}_m$ is not seen from $P(v)$;
- Otherwise, find the nearest point $Q_m(v)$ from $P(v)$ to $\mathcal{VP}(v) \cap \mathcal{R}_m$. This is done as follows.

  Case 1: $\mathcal{R}_m$ is a triangle. If $P(v) \in \mathcal{R}_m$, $Q_m(v) := P(v)$. Otherwise, projection of a point onto a polygon can be found by iterating the edges of the polygon. Thus, iterating over the edges of the polygons constituting $\mathcal{VP}(v) \cap \mathcal{R}_m$, we obtain the result.

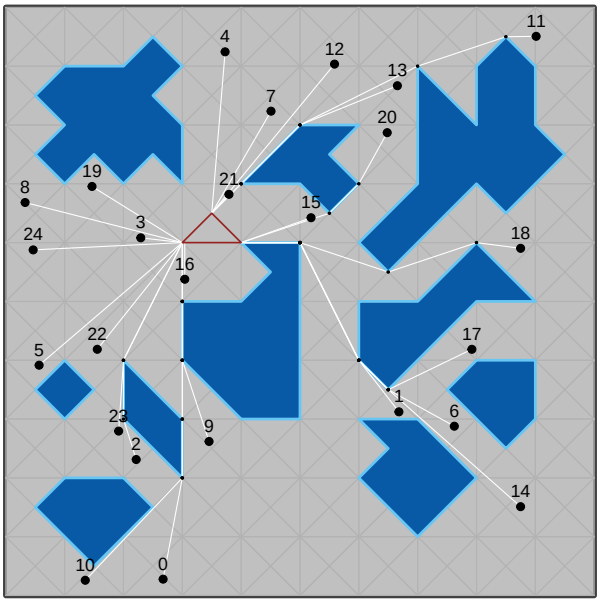
  Case 2: $\mathcal{R}_m$ is a segment. In this case, we simply have to iterate over the segments in $\mathcal{VP}(v) \cap \mathcal{R}_m$.

- Now, add to $\mathcal{G}(V \cup v(\mathcal{R}_m), E)$ a directed-edge $v(\mathcal{R}_m) \mapsto v$ with weight $\|P(v) - Q_m(v)\|_2$.

The distances $d_{nm}^{\inf}$ for region-unit $\mathcal{R}_m$ can now be found with the algorithm of Dijkstra [16]. The output of this procedure is illustrated in Figure 4.9.



(a) Triangle unit 1.



(b) Triangle unit 2.

Figure 4.9. Shortest paths to selected triangle units.

4.3.3. Bounds for the Loss in a Region Unit

By the definition of $d_{nm}^{\inf}$ in (4.15), we have

$$d_{nm}^{\inf} = \min_{P \in \mathcal{R}_m} \; d_{\mathcal{B}}\left(A_n, P\right) \leq d_{\mathcal{B}}\left(A_n, P'\right)$$

for any $P' \in \mathcal{R}_m$. Thus, it follows that

$$\sum_{i \in \mathcal{C}_k} w_i d_{im}^{\inf} = \sum_{i \in \mathcal{C}_k} w_i \min_{P \in \mathcal{R}_m} d_{\mathcal{B}}\left(A_n, P\right) \leq \min_{P \in \mathcal{R}_m} \sum_{i \in \mathcal{C}_k} w_i d_{\mathcal{B}}\left(A_i, P\right). \quad (4.17)$$

Define $\mathbf{Loss}_{\mathcal{B}}^{\mathrm{LB}}\left(\{A_i, i \in \mathcal{C}_k\} | \mathcal{R}_m\right)$ and $\mathbf{Loss}_{\mathcal{B}}^{\mathrm{opt}}\left(\{A_i, i \in \mathcal{C}_k\} | \mathcal{R}_m\right)$ as the left-hand side and the right-hand side in inequality (4.17):

$$\mathbf{Loss}_{\mathcal{B}}^{\mathrm{LB}}\left(\{A_i, i \in \mathcal{C}_k\} | \mathcal{R}_m\right) := \sum_{i \in \mathcal{C}_k} w_i d_{im}^{\inf}, \quad (4.18a)$$

$$\mathbf{Loss}_{\mathcal{B}}^{\mathrm{opt}}\left(\{A_i, i \in \mathcal{C}_k\} | \mathcal{R}_m\right) := \min_{P \in \mathcal{R}_m} \sum_{i \in \mathcal{C}_k} w_i d_{\mathcal{B}}\left(A_i, P\right). \quad (4.18b)$$

Now suppose $\mathrm{diam}(\mathcal{R}_m) = \delta_m$, see (4.14). From the triangle inequality, for any two points $P, P' \in \mathcal{R}_m$, we have

$$d_{\mathcal{B}}\left(A_n, P\right) \leq d_{\mathcal{B}}\left(A_n, P'\right) + d_{\mathcal{B}}\left(P', P\right) \leq d_{\mathcal{B}}\left(A_n, P'\right) + \delta_m.$$

Since this inequality holds for any $P' \in \mathcal{R}_m$, it also holds for the closest point $P^* := \mathbf{proj}(A_n | \mathcal{R}_m)$, see the definition in (4.16). Thus,

$$d_{\mathcal{B}}\left(A_n, P\right) \leq d_{\mathcal{B}}\left(A_n, P^*\right) + \delta_m = \min_{P \in \mathcal{R}_m} d_{\mathcal{B}}\left(A_n, P\right) + \delta_m \leq d_{nm}^{\inf} + \delta_m$$

for any $P \in \mathcal{R}_m$. Therefore:

$$\begin{aligned} \mathbf{Loss}_{\mathcal{B}}^{\mathrm{opt}}\left(\{A_i, i \in \mathcal{C}_k\} | \mathcal{R}_m\right) &= \min_{P \in \mathcal{R}_m} \sum_{i \in \mathcal{C}_k} w_i d_{\mathcal{B}}\left(A_n, P\right) \\ &\leq \sum_{i \in \mathcal{C}_k} w_i \left[\max_{P \in \mathcal{R}_m} d_{\mathcal{B}}\left(A_n, P\right)\right] \leq \sum_{i \in \mathcal{C}_k} w_i (d_{im}^{\inf} + \delta_m). \end{aligned} \quad (4.19)$$

Define

$$\begin{aligned} \mathbf{Loss}_{\mathcal{B}}^{\mathrm{UB}}\left(\{A_i, i \in \mathcal{C}_k\} | \mathcal{R}_m\right) &:= \sum_{i \in \mathcal{C}_k} w_i (d_{im}^{\inf} + \delta_m) \\ &= \mathbf{Loss}_{\mathcal{B}}^{\mathrm{LB}}\left(\{A_i, i \in \mathcal{C}_k\} | \mathcal{R}_m\right) + \delta_m \sum_{i \in \mathcal{C}_k} w_i. \end{aligned} \quad (4.20)$$

Combining the definitions in (4.18), (4.20) and inequalities (4.17), (4.19), we get

$$\begin{aligned} \mathbf{Loss}_{\mathcal{B}}^{\mathrm{LB}}\left(\{A_i, i \in \mathcal{C}_k\} | \mathcal{R}_m\right) &\leq \mathbf{Loss}_{\mathcal{B}}^{\mathrm{opt}}\left(\{A_i, i \in \mathcal{C}_k\} | \mathcal{R}_m\right) \\ &\leq \mathbf{Loss}_{\mathcal{B}}^{\mathrm{UB}}\left(\{A_i, i \in \mathcal{C}_k\} | \mathcal{R}_m\right) \end{aligned} \quad (4.21)$$

4.3.4. Finding the Optimal Center

Our goal next is to find an ε-accuracy solution to $\mathbf{Loss}_{\mathcal{B}}^{\text{opt}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}^{\cup}\right)$ where

$$\mathbf{Loss}_{\mathcal{B}}^{\text{opt}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}^{\cup}\right) := \min_{P \in \mathcal{R}^{\cup}} \sum_{i \in \mathcal{C}_k} w_i d_{\mathcal{B}}\left(A_i, P\right). \tag{4.22}$$

We firstly note that since inequalities in (4.21) hold for any $\mathcal{R}_m$, we have

$$\begin{aligned}\min_{m=1,\ldots,M} \mathbf{Loss}_{\mathcal{B}}^{\text{LB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}_m\right) \leq \min_{m=1,\ldots,M} \mathbf{Loss}_{\mathcal{B}}^{\text{opt}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}_m\right) \\ = \mathbf{Loss}_{\mathcal{B}}^{\text{opt}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}^{\cup}\right) \leq \min_{m=1,\ldots,M} \mathbf{Loss}_{\mathcal{B}}^{\text{UB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}_m\right).\end{aligned} \tag{4.23}$$

Suppose

$$\min_{m=1,\ldots,M} \mathbf{Loss}_{\mathcal{B}}^{\text{LB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}_m\right) \tag{4.24}$$

is attained at region-unit $\mathcal{R}^{\min}$ and suppose

$$\min_{m=1,\ldots,M} \mathbf{Loss}_{\mathcal{B}}^{\text{UB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}_m\right)$$

is attained at region-unit $\mathcal{R}^{\text{best}}$. We have

$$\begin{aligned}\mathbf{Loss}_{\mathcal{B}}^{\text{LB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}^{\min}\right) \leq \mathbf{Loss}_{\mathcal{B}}^{\text{opt}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}^{\cup}\right) \\ \leq \mathbf{Loss}_{\mathcal{B}}^{\text{UB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}^{\text{best}}\right).\end{aligned}$$

Consider subdividing $\mathcal{R}^{\min}$ into two smaller parts $\mathcal{R}^{\text{left}}$ and $\mathcal{R}^{\text{right}}$ (for a triangle, divide it into two smaller triangles along the middle of the longest side, for a segment, divide it into two segments of equal lengths at the middle point). We have

$$\begin{aligned}\mathbf{Loss}_{\mathcal{B}}^{\text{LB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}^{\min}\right) = \sum_{i \in \mathcal{C}_k} w_i \min_{P \in \mathcal{R}^{\min}} d_{\mathcal{B}}\left(A_i, P\right) \\ \leq \sum_{i \in \mathcal{C}_k} w_i \min_{P \in \mathcal{R}^{\text{left}}} d_{\mathcal{B}}\left(A_i, P\right) = \mathbf{Loss}_{\mathcal{B}}^{\text{LB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}^{\text{left}}\right),\end{aligned}$$

since $\mathcal{R}^{\text{left}} \subset \mathcal{R}^{\min}$. Analogous inequality holds for $\mathcal{R}^{\text{right}}$. Thus, if instead of $\mathcal{R}^{\min}$ we insert two region-units $\mathcal{R}^{\text{left}}$ and $\mathcal{R}^{\text{left}}$, we can expect an improvement of the lower bound for $\mathbf{Loss}_{\mathcal{B}}^{\text{opt}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}^{\cup}\right)$ as defined via (4.24). This observation leads to the following algorithm to obtain an ε-accuracy solution.

Algorithm initialization. For each $\mathcal{R}_m$ constituting $\mathcal{R}^{\cup}$, calculate

$$\mathbf{Loss}_{\mathcal{B}}^{\mathrm{LB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}_m\right), \quad \mathbf{Loss}_{\mathcal{B}}^{\mathrm{UB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}_m\right).$$

Store $\mathbf{Loss}_{\mathcal{B}}^{\mathrm{LB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}_m\right)$ in a queue $\mathbb{Q}$ ordered by an increasing value. Memorize $\mathcal{R}^{\mathrm{best}}$. Delete from $\mathbb{Q}$ all $\mathcal{R}_m$ which do not satisfy

$$\mathbf{Loss}_{\mathcal{B}}^{\mathrm{LB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}_m\right) < \mathbf{Loss}_{\mathcal{B}}^{\mathrm{UB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}^{\mathrm{best}}\right).$$

While part. Pick $\mathcal{R}^{\min}$ with the smallest lower bound value and delete it from $\mathbb{Q}$. If

$$\mathbf{Loss}_{\mathcal{B}}^{\mathrm{UB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}^{\mathrm{best}}\right) - \mathbf{Loss}_{\mathcal{B}}^{\mathrm{LB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}^{\min}\right) \leq \varepsilon,$$

report any point $P \in \mathcal{R}^{\mathrm{best}}$ as a solution. Otherwise, subdivide $\mathcal{R}^{\min}$ into two parts $\mathcal{R}^{\mathrm{left}}$ and $\mathcal{R}^{\mathrm{right}}$. Update $\mathcal{R}^{\mathrm{best}}$ to $\mathcal{R}^{\mathrm{left}}$ or $\mathcal{R}^{\mathrm{right}}$ if any of them has a better (i.e., smaller) upper bound. Insert $\mathcal{R}^{\mathrm{left}}$ and $\mathcal{R}^{\mathrm{right}}$ into $\mathbb{Q}$. Delete region-units $\mathcal{R}_m$ in $\mathbb{Q}$ which do not satisfy

$$\mathbf{Loss}_{\mathcal{B}}^{\mathrm{LB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}_m\right) < \mathbf{Loss}_{\mathcal{B}}^{\mathrm{UB}}\left(\{A_i, i \in \mathcal{C}_k\} \,|\mathcal{R}^{\mathrm{best}}\right)$$

and repeat the **While part**.

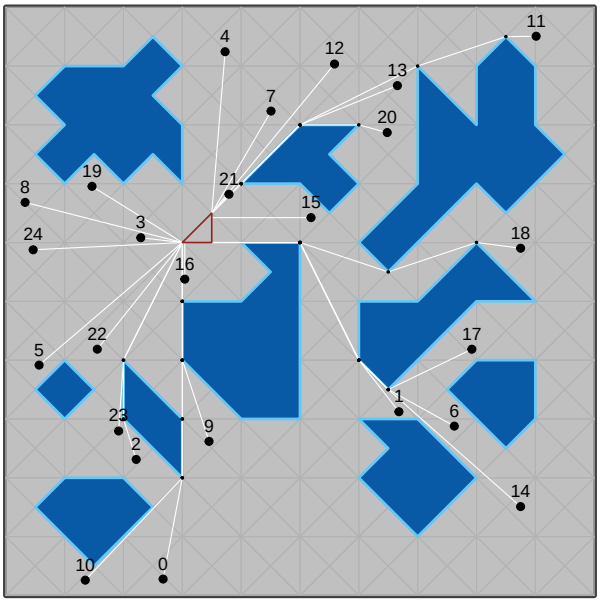
The procedure is illustrated in Figure 4.10.

4.3.5. On an Alternative Approach to Optimal Center Determination

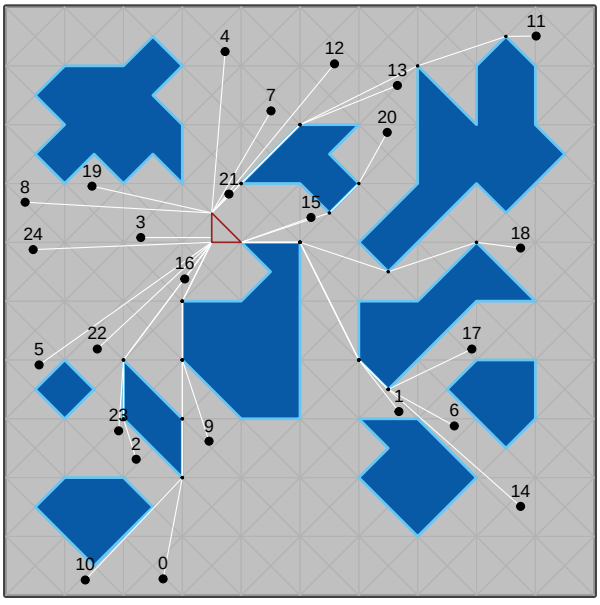
To determine the optimal center, one can think of alternative methods. For example, the global optimization method of Paulavičius and Žilinskas [39] can be used. The algorithm is called "a simplicial branch-and-bound algorithm with Lipschitz bounds" (see Algorithm 4, p. 42 in [39]). For this method, for each simplex constituting $\mathcal{R}^{\cup}$, a lower bound on the objective function within a simplex can be calculated using a Lipschitz constant. For the Weber point with polyhedral barriers problem, from the triangle inequality, we know that

$$d_{\mathcal{B}}\left(A_i, Q\right) \leq d_{\mathcal{B}}\left(A_i, P\right) + d_{\mathcal{B}}\left(P, Q\right),$$

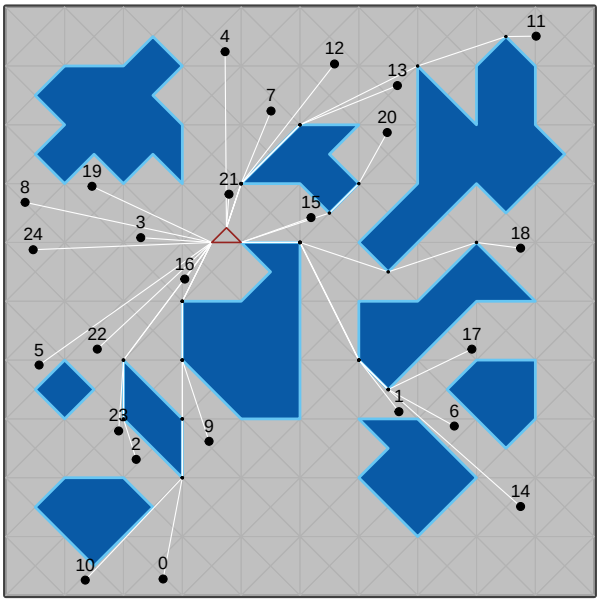
and since for a fixed simplex $\mathcal{R}_m$ (which we assumed to be either a segment or a triangle) and any two points $P, Q \in \mathcal{R}_m$ we have



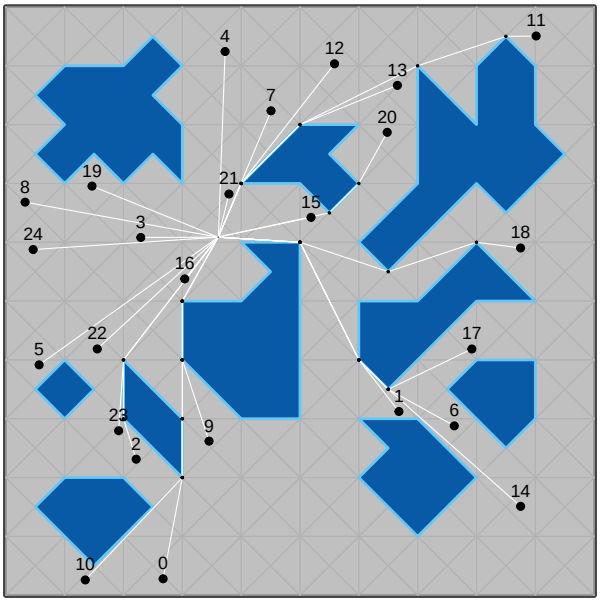
(a) Left partition. LB ≈ 9.06, UB ≈ 9.94.



(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 iterations: 4979.

Figure 4.10. Illustration of the region-unit partitioning algorithm.

$d_{\mathcal{B}}(P,Q) = \|P-Q\|_2$, we get that

$$f(Q) := \sum_{i\in\mathcal{C}_k} w_i d_{\mathcal{B}}(A_i,Q) \leq \sum_{i\in\mathcal{C}_k} w_i \left[d_{\mathcal{B}}(A_i,P) + d_{\mathcal{B}}(P,Q)\right]$$
$$= \sum_{i\in\mathcal{C}_k} w_i d_{\mathcal{B}}(A_i,P) + \Big(\sum_{i\in\mathcal{C}_k} w_i\Big) \|P-Q\|_2 .$$

Thus, we could use a Lipschitz constant $L := \sum_{i\in\mathcal{C}_k} w_i$, e.g., for this constant, we have

$$f(P) - L\|P-Q\|_2 \leq f(Q) \leq f(P) + L\|P-Q\|_2 \quad \text{for} \quad \forall P,Q \in \mathcal{R}_m.$$

That is, if we know the loss at point $P \in \mathcal{R}_m$, any other point $Q \in \mathcal{R}_m$ will satisfy $f(Q) \geq f(P) - L\|P-Q\|_2$. This gives a Lipschitz bound for all the points within $\mathcal{R}_m$, provides a method to select the most promising region unit for the next partition, or to reasonably pick the next iterate point. We refer to Paulavičius and Žilinskas [39] for further details on the method.

4.4. Chapter Summary

In this chapter, we have considered 3 single location problems.

Constrained mean point. The constrained optimal location minimizing the sum of **squared** Euclidean lengths is equal to the projection of the (unconstrained) mean point, i.e., the optimal solution is the closest point on the constrained set.

Net-constrained Weber point. This is the point within the constrained set which minimizes the sum of **non-squared** Euclidean lengths. Firstly, we described two methods for finding an optimal segment-constrained Weber point within a desired accuracy. The proposed methods converge to the optimal location by iteratively shrinking the gap between the lower bound and the upper bound (which is equal to the value at the best sample point), terminating when a desired accuracy is achieved. Algorithm extension to the net-constrained (i.e., segment-union-constrained) location problem having these segment-constrained solvers is rather straightforward.

Weber point with barriers. Here, obstacles which must be bypassed when traveling from location to location are introduced. To determine the optimal location minimizing the sum of distances, firstly, we partitioned the constrained set (e.g., the allowed set for the location) into a union of non-overlapping triangles and segments. Then we described how a special graph can be constructed and how using Dijkstra's algorithm we can find the shortest path from a fixed region unit to each client. The sum of such a region-unit to client-point distances gives a lower bound value for all the possible locations within the corresponding region unit (i.e., the best location within the region unit gives loss which is greater than the defined lower bound).

To find the optimal location within the constrained set, i.e., in the union of segments and triangles, we presented a region-unit partitioning algorithm. This algorithm picks the minimal lower bound region unit and divides it into two parts: along the middle for a segment and along the longest side for a triangle. This procedure results in two new region units with increased lower bound values. Tracking the best upper bound region unit, the algorithm terminates when the difference between the upper bound and the lower bound is sufficiently small.

Consider the following general multi-locations (clustering) problem:

$$\min_{(\mathcal{C}_1,\,P_1),\ldots,(\mathcal{C}_K,\,P_K)} \sum_{k=1}^{K}\sum_{i\in\mathcal{C}_k} w_i d(A_i,P_k) \quad \text{s.t.} \quad P_k\in\mathcal{R}^{\cup},\ k=1,\ldots,K. \tag{5.1}$$

We use the same notation as before: pairs $\{(A_n,w_n),\ n=1,\ldots,N\}$ represent client locations and their weights, which are given as problem input. Optimization variables are clusters $\{\mathcal{C}_k\}$ and cluster centers $\{P_k\}$. The distance metric is arbitrary; for example, the reader can assume that $d=d_{\mathcal{B}}$, i.e., that barriers are present, but the ideas also hold for $d_2(A,P):=\|A-P\|_2^2$ and $d_1(A,P):=\|A-P\|_2$. However, we will later consider an improvement of the algorithm for distance metrics d_2 and d_1, i.e., when there are no barriers.

Our goal in this chapter is to find the global solution of (5.1). Using notation $\mathbb{A}_{n_1}^{n_2}:=\{A_{n_1},\ldots,A_{n_2}\}$ for $n_1\le n_2$, we label this solution with $\textbf{Loss}^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right]\,|\,[\mathcal{R}^{\cup}]\right)$ for short. The choice of such notation is motivated by the following "translation" in words: "cluster the weighted point set $\mathbb{A}_1^N$ into (/) K clusters given a condition (|) that cluster centers are restricted to be within $\mathcal{R}^{\cup}$". We also add a subscript of the distance metric when referring to a particular d for the problem; e.g., we use notation $\textbf{Loss}_2^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right]\,|\,[\mathcal{R}^{\cup}]\right)$ for d_2, $\textbf{Loss}_1^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right]\,|\,[\mathcal{R}^{\cup}]\right)$ for d_1, and $\textbf{Loss}_{\mathcal{B}}^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right]\,|\,[\mathcal{R}^{\cup}]\right)$ for $d_{\mathcal{B}}$.

5.1. K-Means Principle Revisited

Consider arbitrary locations $P_k\in\mathcal{R}^{\cup},\ k=1,\ldots,K$, and define

$$\mathcal{C}_k:=\left\{i: d(A_i,P_k)=\min_{l\in\mathbb{K}} d(A_i,P_l),\ l\in\mathbb{K}\right\},$$

i.e., assign each point to the closest center. To improve this solution via k-means-type algorithm, we iterate the two steps below:

Location Step. Solve K single location problems to optimality:

$$\min_{P\in\mathcal{R}^{\cup}} \sum_{i\in\mathcal{C}_k} w_i d(A_i,P), \quad k=1,\ldots,K. \tag{5.2}$$

This step improves the loss in (5.1) by improving locations; label these new locations with $P_1',\ldots,P_K'$.

Assignment Step. Redefine the clusters by assigning each client A_n to the closest P'_k:

$$\mathcal{C}'_k := \left\{ i : d(A_i, P'_k) = \min_{l \in \mathbb{K}} d(A_i, P'_l),\ l \in \mathbb{K} \right\}.$$

This step improves the loss in (5.1) by improving element assignments.

Although solving single location problem (5.2) is not always straightforward as we saw in Chapter 4, theoretically, we will assume that we can find the **exact** optimal location for a given cluster $\mathcal{C}_k$. Thus, given some arbitrary cluster – center pairs

$$(\mathcal{C}_1,\ P_1), \ldots, (\mathcal{C}_K,\ P_K),$$

we can only get an improvement if we apply **Location Step** to obtain updated locations $P'_1, \ldots, P'_K$:

$$\sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} w_i d(A_i, P_k) \geq \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} w_i d(A_i, P'_k). \tag{5.3}$$

Analogous result holds if we apply **Assignment Step** to obtain updated assignments $\mathcal{C}'_1, \ldots, \mathcal{C}'_K$:

$$\sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} w_i d(A_i, P_k) \geq \sum_{k=1}^{K} \sum_{i \in \mathcal{C}'_k} w_i d(A_i, P_k). \tag{5.4}$$

We will use this observation in the proofs.

5.2. Enumeration Tree

To find the global solution of problem (5.1), we can theoretically enumerate all the possible partitions of the index set $\mathbb{N} := \{1, \ldots, N\}$ into K clusters $\mathcal{C}_1, \ldots, \mathcal{C}_K$, find the optimal centers for each possible clustering, and pick the best solution.

Such an approach would be brutal and would allow us to solve only very small problems, as there are $S_2(N, K)$ of such partitions into clusters.[1] Nevertheless, we showed in our article [A.1] for problem $\mathbf{Loss}_2^{\text{opt}}\left(\left[\mathbb{A}_1^N / K\right] \mid [\mathcal{N}]\right)$ 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 the

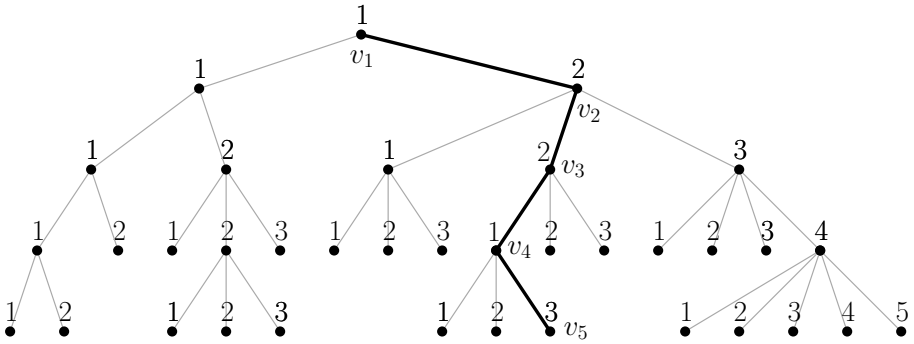


Figure 5.1. Enumeration tree $\mathcal{T}(N, K)$.

next section. For now, consider the tree $\mathcal{T}(N, K)$ in Figure 5.1, which visualizes such 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 $\{A_1, \ldots, A_N\}$, 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 Figure 5.1 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: $\mathcal{C}_k = \emptyset,\ k = 2, \ldots, K$.

- 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]Here, $S_2(\bullet, \bullet)$ is Stirling's number of the second kind. This number satisfies the recursive relation

$$S_2(N + 1, K) = KS_2(N, K) + S_2(N, K - 1), \tag{5.5}$$

and based on this one can calculate that, e.g., $S_2(18, 4) = 2,798,806,985$ – already a very big number. Also, from (5.5) one can see that adding any succeeding 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.

1) **into a non-empty cluster**,

2) **into an empty cluster with the lowest index**.

– By two rules, the second element can be assigned to either $\mathcal{C}_1$ or $\mathcal{C}_2$.

– For the third element, $\mathcal{C}_3$ might not be available (in case $\mathcal{C}_1 = \{1, 2\}$, $\mathcal{C}_2 = \emptyset$; see the nodes at the third level of $\mathcal{T}(N, K)$).

– For the fourth element, see the nodes at the fourth 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.

It can be shown by simple arguments that this way, nodes v_n at 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.,

$$\begin{aligned} &\mathcal{P}(v_n) := \{\mathcal{C}_1(v_n), \dots, \mathcal{C}_K(v_n)\}\,, \\ &\qquad \text{with} \quad \mathcal{C}_k(v_n) := \{i \in \{1, 2, \dots, n\} : \lambda(v_i) = k\}\,. \end{aligned} \tag{5.6}$$

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., $\partial\mathcal{P}(v_n) := \mathcal{P}(v_n)$, $\partial\mathcal{C}_k(v_n) := \mathcal{C}_k(v_n)$ for $n < N$. Also, we notate the set of all the vertices at level N with $\mathcal{L}(N, K)$ (this is the leaf node set of the tree).

5.3. Pruning Criteria

Label

$$\mathbf{Loss}^{\mathrm{opt}}(\{A_i, i \in \mathcal{C}_k\}\,|\mathcal{R}^{\cup}) := \min_{P \in \mathcal{R}^{\cup}} \sum_{i \in \mathcal{C}_k} w_i d(A_i, P).$$

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

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

Label

$$\mathbf{Loss}^{\mathrm{opt}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathcal{R}^{\cup}, \mathcal{P}(v_N)\right]\right) := \sum_{k=1}^{K} \mathbf{Loss}^{\mathrm{opt}}(\{A_i, i \in \mathcal{C}_k(v_N)\}\,|\mathcal{R}^{\cup}), \tag{5.7}$$

i.e., this is the problem of finding optimal cluster centers for a clustering as determined by vertex v_N and the corresponding (complete!) partition $\mathcal{P}(v_N)$.

Now suppose we are at some vertex $v_n \in \mathcal{T}(N,K)$ at level $n < N$. Consider the following problem:

$$\begin{aligned} \min_{v_N \in \mathcal{L}(N,K)} \ & \mathbf{Loss}^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathcal{R}^{\cup}, \mathcal{P}(v_N)\right]\right) \\ \text{s.t.} \quad & \text{leaf vertex } v_N \text{ is reached through } v_n. \end{aligned} \tag{5.8}$$

Looking at the enumeration tree in Figure 5.1, in problem (5.8) we only consider partitions which are in the branch starting at vertex v_n. Label

$$\partial\mathcal{P}(v_n) := \{\partial\mathcal{C}_1(v_n), \ldots, \partial\mathcal{C}_K(v_n)\}.$$

We notate problem (5.8) with $\mathbf{Loss}^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right] \mid [\mathcal{R}^{\cup}, \partial\mathcal{P}(v_n)]\right)$ for short.

Theorem 5.3.1 (Lower bound for the loss of the problem in (5.8)).

$$\begin{aligned} \mathbf{Loss}^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathcal{R}^{\cup}, \partial\mathcal{P}(v_n)\right]\right) \ \geq \ & \mathbf{Loss}^{\text{opt}}\left(\left[\mathbb{A}_1^n/K\right] \mid \left[\mathcal{R}^{\cup}, \partial\mathcal{P}(v_n)\right]\right) \\ & + \mathbf{Loss}^{\text{opt}}\left(\left[\mathbb{A}_{n+1}^N/K\right] \mid \left[\mathcal{R}^{\cup}\right]\right). \end{aligned} \tag{5.9}$$

Proof. Suppose the optimal value for problem (5.8) is attained at a leaf node v_N for clusters $\mathcal{C}_1(v_N), \ldots, \mathcal{C}_K(v_N)$ with centers at locations $P_1, \ldots, P_K$. Label[2]

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

We have

$$\begin{aligned} \mathbf{Loss}^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathcal{R}^{\cup}, \partial\mathcal{P}(v_N)\right]\right) &= \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k(v_N)} w_i d(A_i, P_k) \\ &= \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k^{\dagger}} w_i d(A_i, P_k) + \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k^{\ddagger}} w_i d(A_i, P_k). \end{aligned}$$

The two terms on the right-hand side in the equation above can be recognized to correspond to two clustering problems $\mathbb{A}_1^n/K$ and $\mathbb{A}_{n+1}^N/K$. Now if we run **Location Step** (5.3) for the first term, we can only obtain an improvement (i.e., a smaller value). Such procedure results in

[2] $\mathbb{N}_{n_1}^{n_2} := \{n_1, \ldots, n_2\}$.

$\mathbf{Loss}^{\text{opt}}\left([\mathbb{A}_1^n/K] \,|\, [\mathcal{R}^{\cup}, \partial\mathcal{P}(v_n)]\right)$. Similarly, if we run the k-means algorithm for the second term until convergence, we arrive at a local solution of clustering problem $\mathbb{A}_{n+1}^N/K$. This local solution can not have a better (i.e., a smaller) loss value than the globally optimal solution, which is equal to $\mathbf{Loss}^{\text{opt}}\left(\left[\mathbb{A}_{n+1}^N/K\right] \,|\, [\mathcal{R}^{\cup}]\right)$. $\square$

Pruning criteria. Suppose we have an upper bound L^{UB} for our clustering problem, i.e., suppose we know that

$$\mathbf{Loss}^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right] \,|\, \left[\mathcal{R}^{\cup}\right]\right) \leq L^{\text{UB}}.$$

Such bound is given by an optimal solution of any (fixed) particular clustering $\mathcal{C}_1, \ldots, \mathcal{C}_K$; a "good quality" lower bound can be obtained by k-means-type algorithm. Now, if we know that

$$\mathbf{Loss}^{\text{opt}}\left([\mathbb{A}_1^n/K] \,|\, \left[\mathcal{R}^{\cup}, \partial\mathcal{P}(v_n)\right]\right) + \mathbf{Loss}^{\text{opt}}\left(\left[\mathbb{A}_{n+1}^N/K\right] \,|\, \left[\mathcal{R}^{\cup}\right]\right) \geq L^{\text{UB}},$$

Theorem 5.3.1 implies that

$$\mathbf{Loss}^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right] \,|\, \left[\mathcal{R}^{\cup}, \partial\mathcal{P}(v_n)\right]\right) \geq L^{\text{UB}},$$

and the branch corresponding to v_n can be cut, i.e., there is no need to explore it, as we will not find a better solution there.

5.4. Lower Bound Problem

5.4.1. Lower Bound for the Optimal Cluster Loss

Consider again the single optimal location problem for cluster $\mathcal{C}_k$:

$$\mathbf{Loss}^{\text{opt}}\left(\mathcal{C}_k|\mathcal{R}^{\cup}\right) := \min_{P \in \mathcal{R}^{\cup}} \sum_{i \in \mathcal{C}_k} w_i d(A_i, P). \tag{5.10}$$

In Chapter 4, we showed that solving problem (5.10) exactly is possible for $d_2(A_i, P) \equiv \|A_i - P\|_2^2$ with $\mathcal{R}^{\cup} \equiv \mathcal{N}$. For this case, we first had to determine the optimal unconstrained center

$$\overline{P}(\mathcal{C}_k) := \frac{1}{w(\mathcal{C}_k)} \sum_{i \in \mathcal{C}_k} w_i A_i, \quad \text{with} \quad w(\mathcal{C}_k) := \sum_{i \in \mathcal{C}_k} w_i,$$

then project $\overline{P}(\mathcal{C}_k)$ on each segment of $\mathcal{N}$, and pick the best projection. This procedure can be accomplished in $\mathcal{O}(M)$ time (where M is the

number of segments in $\mathcal{N}$); the idea can be extended to an arbitrary union of convex simplexes $\mathcal{R}^{\cup}$.

However, the algorithms of Chapter 4 can determine the optimal solution to problems $\mathbf{Loss}_1^{\text{opt}}(\mathcal{C}_k|\mathcal{N})$ or $\mathbf{Loss}_{\mathcal{B}}^{\text{opt}}(\mathcal{C}_k|\mathcal{R}^{\cup})$ only in the limit, i.e., at a predefined accuracy $\varepsilon > 0$. This is unfavorable in two aspects. Firstly, since we can not solve (5.10) exactly when $d = d_1$ or $d = d_{\mathcal{B}}$, we are not able to use the pruning criteria in Theorem 5.3.1 straightforwardly. Secondly, solving problem $\mathbf{Loss}^{\text{opt}}(\mathcal{C}_k|\mathcal{R}^{\cup})$ within ε-accuracy might require much more than $\mathcal{O}(M)$ steps (this depends on how small is $\varepsilon > 0$). Thus, instead of solving (5.10) exactly, we will focus on the lower bound for $\mathbf{Loss}^{\text{opt}}(\mathcal{C}_k|\mathcal{R}^{\cup})$.

In Chapter 4, we presented a method to calculate the shortest distances $d_{nm}^{\inf}$ from each client A_n to each region-unit $\mathcal{R}_m$ for a location with barriers problem $\mathbf{Loss}_{\mathcal{B}}^{\text{opt}}(\mathcal{C}_k|\mathcal{R}^{\cup})$; problem $\mathbf{Loss}_1^{\text{opt}}(\mathcal{C}_k|\mathcal{R}^{\cup})$ can be seen as an instance of a barriers problem when there are no barriers present. Suppose the optimal solution P_k of (5.10) is contained within region $\mathcal{R}_{m_k}$ ($P_k \in \mathcal{R}_{m_k}$). Then, by the definition of $d_{nm_k}^{\inf}$ in (4.15):

$$d(A_n, P_k) \geq d_{nm_k}^{\inf}, \quad n = 1, \ldots, N, \quad \forall P_k \in \mathcal{R}_{m_k},$$

because the shortest distance to a region can not be greater than the distance to some particular point in that region. This implies that

$$\sum_{i \in \mathcal{C}_k} w_i d(A_i, P_k) \geq \sum_{i \in \mathcal{C}_k} w_i d_{im_k}^{\inf} \geq \min_{m=1,\ldots,M} \sum_{i \in \mathcal{C}_k} w_i d_{im}^{\inf}. \tag{5.11}$$

Label the right-hand side of the inequality above with

$$\mathbf{Loss}^{\text{LB}}\left(\mathcal{C}_k|\mathcal{R}^{\cup}\right) := \min_{m=1,\ldots,M} \sum_{i \in \mathcal{C}_k} w_i d_{im}^{\inf}. \tag{5.12}$$

Using (5.11) and the definitions in (5.10), (5.12), we see that

$$\mathbf{Loss}^{\text{opt}}\left(\mathcal{C}_k|\mathcal{R}^{\cup}\right) \geq \mathbf{Loss}^{\text{LB}}\left(\mathcal{C}_k|\mathcal{R}^{\cup}\right). \tag{5.13}$$

Finding (5.12) is straightforward when the shortest distances $d_{nm}^{\inf}$ are known. It is also favorable in the following aspect. Suppose that in the enumeration tree, we are currently at vertex v_n, and proceed further to vertex v_{n+1}. Suppose also that $\lambda(v_{n+1}) = k$, i.e., during this step, we "increment" cluster $\mathcal{C}_k(v_n)$ with client A_{n+1}. Then

$$\begin{aligned}\mathbf{Loss}^{\text{LB}}\left(\mathcal{C}_k \cup \{n+1\}|\mathcal{R}^{\cup}\right) &= \min_{m=1,\ldots,M} \sum_{i \in \mathcal{C}_k \cup \{n+1\}} w_i d_{im}^{\inf} \\ &= \min_{m=1,\ldots,M} \left(\mathbf{Loss}^{\text{LB}}\left(\mathcal{C}_k|\mathcal{R}_m\right) + w_{n+1} d_{(n+1)m}^{\inf}\right),\end{aligned}$$

with

$$\mathbf{Loss}^{\mathrm{LB}}\left(\mathcal{C}_k|\mathcal{R}_m\right) := \sum_{i\in\mathcal{C}_k} w_i d_{im}^{\inf}, \quad m = 1,\ldots,M.$$

In particular, the task of finding $\mathbf{Loss}^{\mathrm{LB}}\left(\mathcal{C}_k \cup \{n+1\}\,|\mathcal{R}^{\cup}\right)$ can be accomplished in $\mathcal{O}(M)$ steps if the values of $\mathbf{Loss}^{\mathrm{LB}}\left(\mathcal{C}_k|\mathcal{R}_m\right)$ have been stored.

5.4.2. Lower Bound for a Fixed Clustering

Now consider the problem

$$\mathbf{Loss}^{\mathrm{opt}}\left(\left[\mathbb{A}_1^n/K\right]\,|\,\left[\mathcal{R}^{\cup},\partial\mathcal{P}(v_n)\right]\right) \;\equiv\; \sum_{k=1}^{K}\mathbf{Loss}^{\mathrm{opt}}\left(\partial\mathcal{C}_k(v_n)|\mathcal{R}^{\cup}\right), \quad (5.14)$$

i.e., suppose we want to find the optimal loss of partial clustering $\partial\mathcal{P}(v_n)$ defined by enumeration vertex $v_n \in \mathcal{T}(N,K)$ (recall the definition in (5.6)) at level $n \le N$ for the set $\mathbb{A}_1^n \subseteq \mathbb{A}_1^N$. Using inequality (5.13), we obtain

$$\sum_{k=1}^{K}\mathbf{Loss}^{\mathrm{opt}}\left(\partial\mathcal{C}_k(v_n)|\mathcal{R}^{\cup}\right) \ge \sum_{k=1}^{K}\mathbf{Loss}^{\mathrm{LB}}\left(\partial\mathcal{C}_k(v_n)|\mathcal{R}^{\cup}\right). \quad (5.15)$$

Label the right-hand side in the inequality above with

$$\mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_1^n/K\right]\,|\,\left[\mathcal{R}^{\cup},\partial\mathcal{P}(v_n)\right]\right) := \sum_{k=1}^{K}\mathbf{Loss}^{\mathrm{LB}}\left(\partial\mathcal{C}_k(v_n)|\mathcal{R}^{\cup}\right). \quad (5.16)$$

We have (see the definitions in (5.14), (5.16) and use (5.15))

$$\mathbf{Loss}^{\mathrm{opt}}\left(\left[\mathbb{A}_1^n/K\right]\,|\,\left[\mathcal{R}^{\cup},\partial\mathcal{P}(v_n)\right]\right) \ge \mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_1^n/K\right]\,|\,\left[\mathcal{R}^{\cup},\partial\mathcal{P}(v_n)\right]\right). \quad (5.17)$$

5.4.3. Pruning Criteria for the Lower Bound Problem

In this subsection, our goal is to redefine the pruning criteria for problem

$$\mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_1^N/K\right]\,|\,\left[\mathcal{R}^{\cup}\right]\right) := \min_{v_N\in\mathcal{L}(N,K)}\mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_1^N/K\right]\,|\,\left[\mathcal{R}^{\cup},\mathcal{P}(v_N)\right]\right). \quad (5.18)$$

We note that

$$\mathbf{Loss}^{\mathrm{opt}}\left(\left[\mathbb{A}_1^N/K\right]\,|\,\left[\mathcal{R}^{\cup}\right]\right) \ge \mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_1^N/K\right]\,|\,\left[\mathcal{R}^{\cup}\right]\right), \quad (5.19)$$

since

$$\mathbf{Loss}^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathcal{R}^{\cup}\right]\right) = \min_{v_N \in \mathcal{L}(N,K)} \mathbf{Loss}^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathcal{R}^{\cup}, \mathcal{P}(v_N)\right]\right),$$

and inequality (5.17) holds for any $v_N \in \mathcal{L}(N,K)$. Thus, it also holds for the minimum values. Also, (5.19) is true for any subset of clients $\mathbb{A} \subseteq \mathbb{A}_1^N$ (not only for $\mathbb{A}_1^N$).

We now consider the pruning criteria for the solution of (5.18). Consider the following problem (in analogy with (5.8)):

$$\begin{aligned} \min_{v_N \in \mathcal{L}(N,K)} \ & \mathbf{Loss}^{\text{LB}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathcal{R}^{\cup}, \mathcal{P}(v_N)\right]\right) \\ \text{s.t.} \quad & \text{leaf vertex } v_N \text{ is reached through } v_n, \end{aligned} \tag{5.20}$$

which we label with $\mathbf{Loss}^{\text{LB}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathcal{R}^{\cup}, \partial\mathcal{P}(v_n)\right]\right)$ for short.

We now state a theorem in analogy with Theorem 5.3.1.

Theorem 5.4.1 (Lower bound for problem (5.20)).

$$\begin{aligned} \mathbf{Loss}^{\text{LB}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathcal{R}^{\cup}, \partial\mathcal{P}(v_n)\right]\right) \ \geq \ & \mathbf{Loss}^{\text{LB}}\left(\left[\mathbb{A}_1^n/K\right] \mid \left[\mathcal{R}^{\cup}, \partial\mathcal{P}(v_n)\right]\right) \\ & + \mathbf{Loss}^{\text{LB}}\left(\left[\mathbb{A}_{n+1}^N/K\right] \mid \left[\mathcal{R}^{\cup}\right]\right). \end{aligned} \tag{5.21}$$

Proof. Suppose the optimal value for problem (5.20) is attained at a leaf node v_N for clusters $\mathcal{C}_1(v_N), \ldots, \mathcal{C}_K(v_N)$ and optimal region-units $\mathcal{R}_{m_1}, \ldots, \mathcal{R}_{m_K}$. Label

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

We have

$$\begin{aligned} \mathbf{Loss}^{\text{LB}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathcal{R}^{\cup}, \partial\mathcal{P}(v_N)\right]\right) &= \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k(v_N)} w_i d_{im_k}^{\text{inf}} \\ &= \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k^{\dagger}} w_i d_{im_k}^{\text{inf}} + \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k^{\ddagger}} w_i d_{im_k}^{\text{inf}}. \end{aligned}$$

The two terms on the right-hand side in the equation above can be recognized to correspond to two lower bound problems for $\mathbb{A}_1^n/K$ and $\mathbb{A}_{n+1}^N/K$. For the first term, "optimize locations":

$$\sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k^{\dagger}} w_i d_{im_k}^{\text{inf}} \geq \sum_{k=1}^{K} \left(\min_{m=1,\ldots,M} \sum_{i \in \mathcal{C}_k^{\dagger}} w_i d_{im}^{\text{inf}} \right) = \sum_{k=1}^{K} \mathbf{Loss}^{\text{LB}}\left(\mathcal{C}_k^{\dagger} | \mathcal{R}^{\cup}\right),$$

which is the definition of $\mathbf{Loss}^{\mathrm{LB}}\left([\mathbb{A}_1^n/K]\,|\,[\mathcal{R}^{\cup}, \partial\mathcal{P}(v_n)]\right)$, see (5.16). For the second term, "optimize locations, then optimize partition":

$$\begin{aligned}\sum_{k=1}^{K}\sum_{i\in\mathcal{C}_k^{\ddagger}} w_i d_{im_k}^{\inf} &\geq \sum_{k=1}^{K}\Big(\min_{m=1,\ldots,M}\sum_{i\in\mathcal{C}_k^{\ddagger}} w_i d_{im}^{\inf}\Big) = \sum_{k=1}^{K}\mathbf{Loss}^{\mathrm{LB}}\left(\mathcal{C}_k^{\ddagger}|\mathcal{R}^{\cup}\right)\\ &= \mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_{n+1}^N/K\right]\,|\,\left[\mathcal{R}^{\cup}, \partial\mathcal{P}(v_N)\right]\right)\\ &\geq \min_{w_N\in\mathcal{L}(N,K)}\mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_{n+1}^N/K\right]\,|\,\left[\mathcal{R}^{\cup}, \partial\mathcal{P}(w_N)\right]\right),\end{aligned}$$

which is the definition of $\mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_{n+1}^N/K\right]\,|\,[\mathcal{R}^{\cup}]\right)$, compare with (5.18). □

Pruning criteria. When looking for the solution to (5.18) using enumeration tree, we can exploit the pruning criteria in Theorem 5.4.1 at any vertex $v_n \in \mathcal{T}(N,K)$. E.g., suppose an upper bound value L^{UB} is known for (5.18):

$$\mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_1^N/K\right]\,|\,\left[\mathcal{R}^{\cup}\right]\right) \leq L^{\mathrm{UB}}.$$

L^{UB} can be obtained by considering the optimal value for some particular $v_N \in \mathcal{L}(N,K)$:

$$L^{\mathrm{UB}} := \mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_1^N/K\right]\,|\,\left[\mathcal{R}^{\cup}, \mathcal{P}(v_N)\right]\right).$$

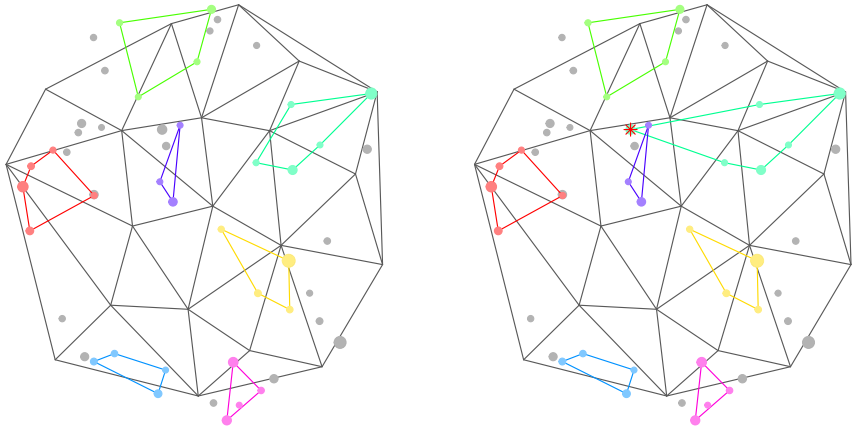
Then, if

$$\mathbf{Loss}^{\mathrm{LB}}\left([\mathbb{A}_1^n/K]\,|\,\left[\mathcal{R}^{\cup}, \partial\mathcal{P}(v_n)\right]\right) + \mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_{n+1}^N/K\right]\,|\,[\mathcal{R}]\right) \geq L^{\mathrm{UB}},$$

Theorem 5.4.1 implies that the branch at v_n can be "cut".

5.5. Convex Hull Criteria for Problems without Barriers

Consider a locally optimal solution of a clustering problem without barriers, $\mathbf{Loss}_2^{\mathrm{opt}}\left(\left[\mathbb{A}_1^N/K\right]\,|\,[\mathcal{R}^{\cup}]\right)$ or $\mathbf{Loss}_1^{\mathrm{opt}}\left(\left[\mathbb{A}_1^N/K\right]\,|\,[\mathcal{R}^{\cup}]\right)$, as obtained with the k-means algorithm at convergence. Since the solution is locally optimal, it can not be improved by an **Assignment Step** as discussed in Section 5.1. This means that each point $\{A_n, n\in\mathbb{N}\}$ is assigned to the closest center. In particular, each to P_k assigned client belongs to a Voronoi cell $\mathcal{V}_k$ of this point. Since each $\{A_i : i\in\mathcal{C}_k\}$ belongs to $\mathcal{V}_k$, so does the convex hull of these points (Boyd and Vandenberghe [10]). This leads to the following observation: **if at vertex** $v_n \in \mathcal{T}(N,K)$ **the**



(a) Clustering at $v_{n-1} \in \mathcal{T}(N, K)$. (b) Clustering at $v_n \in \mathcal{T}(N, K)$.

Figure 5.2. Illustration of convex hull cut criteria. After inclusion of the "starred" client A_n, convex hulls of clusters overlap, and we can prune the corresponding branch at v_n.

convex hulls of clusters start to overlap, we can prune this branch. This property is exploited for example in Rosing [48] when looking for the globally optimal solution of a $\mathbf{Loss}_1^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathbb{R}^2\right]\right)$ problem. The cut criteria is illustrated in Figure 5.2.

5.6. Proposed Algorithm

In this section, we summarize the proposed algorithm for the general problem

$$\mathbf{Loss}^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathcal{R}^{\cup}\right]\right).$$

Step 0 (Initialization). For problems

$$\mathbf{Loss}_1^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathcal{R}^{\cup}\right]\right) \quad \text{and} \quad \mathbf{Loss}_{\mathcal{B}}^{\text{opt}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathcal{R}^{\cup}\right]\right),$$

calculate d_{nm}^{inf}: the minimal distances from each client A_n to each region-unit $\mathcal{R}_m$ (as described in Section 4.3.2).

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

$$\mathbf{Loss}^{\text{LB}}\left(\left[\mathbb{A}_n^N/K\right] \mid \left[\mathcal{R}^{\cup}\right]\right) \quad \text{for} \quad n = N, \ldots, 1,$$

using the enumeration tree $\mathcal{T}(N, K)$ in Figure 5.1. Use pruning criteria (5.21): when looking for $\mathbf{Loss}^{\text{LB}}\left(\left[\mathbb{A}_n^N/K\right] \mid \left[\mathcal{R}^{\cup}\right]\right)$, given some partial

partition $\partial\mathcal{P}$ of point set $\mathbb{A}_n^{\tilde{n}}$, values

$$\mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_n^{\tilde{n}}/K\right] \mid \left[\mathcal{R}^{\cup}, \partial\mathcal{P}\right]\right),\ \mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_{\tilde{n}+1}^{N}/K\right] \mid \left[\mathcal{R}^{\cup}\right]\right) \tag{5.22}$$

are known. For problems without barriers, use in addition the convex hull cut criteria in Section 5.5 for pruning the tree. For the squared Euclidean distances problem, define

$$\begin{aligned}\mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_n^{\tilde{n}}/K\right] \mid \left[\mathcal{R}^{\cup}, \partial\mathcal{P}\right]\right) &:= \mathbf{Loss}_2^{\mathrm{opt}}\left(\left[\mathbb{A}_n^{\tilde{n}}/K\right] \mid \left[\mathcal{R}^{\cup}, \partial\mathcal{P}\right]\right),\\ \mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_{\tilde{n}+1}^{N}/K\right] \mid \left[\mathcal{R}^{\cup}\right]\right) &:= \mathbf{Loss}_2^{\mathrm{opt}}\left(\left[\mathbb{A}_{\tilde{n}+1}^{N}/K\right] \mid \left[\mathcal{R}^{\cup}\right]\right).\end{aligned}$$

That is, in this case, the optimal loss is determined exactly.

Step 2 (Finding an upper bound). Suppose

$$\mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_1^{N}/K\right] \mid \left[\mathcal{R}^{\cup}\right]\right)$$

is attained for clustering $\mathcal{C}_1^{\mathrm{LB}}, \ldots, \mathcal{C}_K^{\mathrm{LB}}$ – as found in the previous step.

For $\mathbf{Loss}_2^{\mathrm{opt}}\left(\left[\mathbb{A}_1^{N}/K\right] \mid \left[\mathcal{R}^{\cup}\right]\right)$, this is the globally optimal clustering, because the lower bounds are exact. The algorithm terminates.

For $\mathbf{Loss}_1^{\mathrm{opt}}\left(\left[\mathbb{A}_1^{N}/K\right] \mid \left[\mathcal{R}^{\cup}\right]\right)$ and $\mathbf{Loss}_{\mathcal{B}}^{\mathrm{opt}}\left(\left[\mathbb{A}_1^{N}/K\right] \mid \left[\mathcal{R}^{\cup}\right]\right)$, for each cluster, solve the optimal center problem

$$\mathbf{Loss}^{\mathrm{opt}}\left(\left\{A_i, i \in \mathcal{C}_k^{\mathrm{LB}}\right\} | \mathcal{R}^{\cup}\right)$$

within ε-accuracy by subdividing the most promising region units into smaller parts (as described in Section 4.3.4). This way, we obtain an upper bound L^{UB} – as an optimal value of a particular solution corresponding to $\mathcal{C}_1^{\mathrm{LB}}, \ldots, \mathcal{C}_K^{\mathrm{LB}}$.

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

$$\mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_1^{N}/K\right] \mid \left[\mathcal{R}^{\cup}, \mathcal{P}(v_N)\right]\right) \leq L^{\mathrm{UB}}. \tag{5.23}$$

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

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

Also, in case no barriers are present, check the convex hull cut criteria at v_n. Label the obtained candidate vertex set with

$$\mathbb{V} := \{v_N \in \mathcal{L}(N, K) : v_N \text{ satisfies (5.23)}\}.$$

Step 4 (Determining the best solution). Starting from $v_N \in \mathbb{V}$ with minimal values of

$$\mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathcal{R}^{\cup}, \mathcal{P}(v_N)\right]\right),$$

solve each problem

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

within ε-accuracy by partitioning region units into smaller parts (as described in Section 4.3.4). Keep record of L^{UB} – the upper bound of the best solution. If we get that

$$\mathbf{Loss}^{\mathrm{LB}}\left(\left[\mathbb{A}_1^N/K\right] \mid \left[\mathcal{R}^{\cup}, \mathcal{P}(v_N)\right]\right) > L^{\mathrm{UB}},$$

remove v_N from $\mathbb{V}$.

The output of the algorithm for an instance of a net-constrained MSSC problem is reported in Figure 5.3. This is the globally optimal solution. The output of the algorithm for several barrier problems is illustrated in Figure 5.4. The ratio between the reported solution and the exact (theoretical) global solution is at most $1 + \epsilon$ as proven by our algorithm.

5.7. Algorithm Time Analysis

We report the predicted median times of the proposed algorithm in Figure 5.5 for the net-constrained MSSC problem and in Figure 5.7 for the multi-Weber problem with polyhedral barriers. For comparison, we also present in Figure 5.6 the results on our attempts to solve the mixed-integer programming formulation of the net-constrained MSSC problem (as described in Section 2.4.2.1). For more details on the numerical experiments and the estimated time models, we refer to our papers [A.1] and [A.2].

5.8. Chapter Summary

In this chapter, we proposed a branch-and-bound algorithm for finding the globally optimal solution for a general multi-locations (clustering) problem. For this purpose, we employed the explicit enumeration tree,

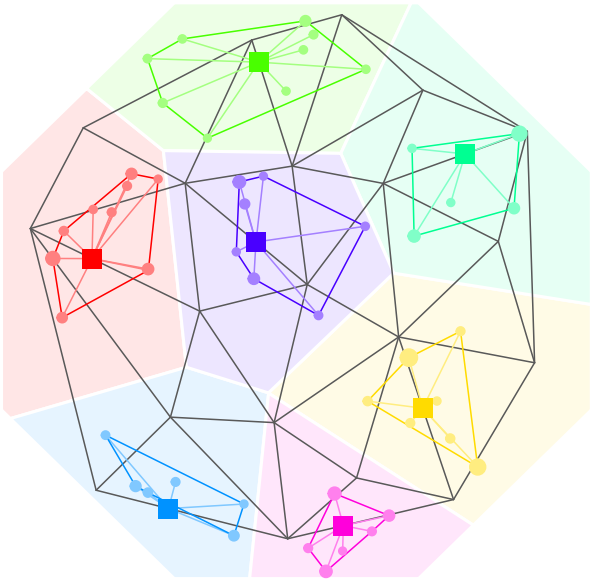
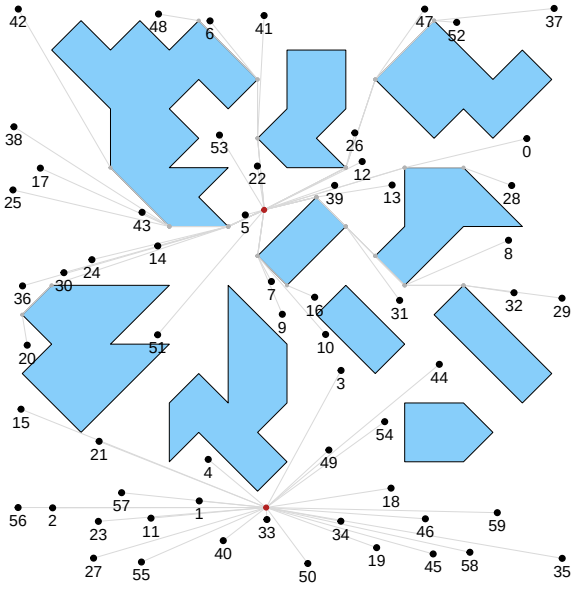
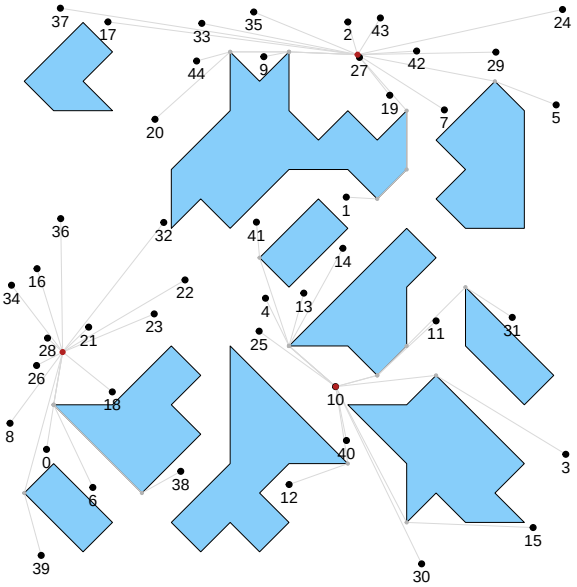


Figure 5.3. Optimal solution of a $50/7$ $\mathbf{Loss}_2^{\mathrm{opt}}\left(\left[\mathbb{A}_1^N/K\right]\,|\,[\mathcal{N}]\right)$ problem.

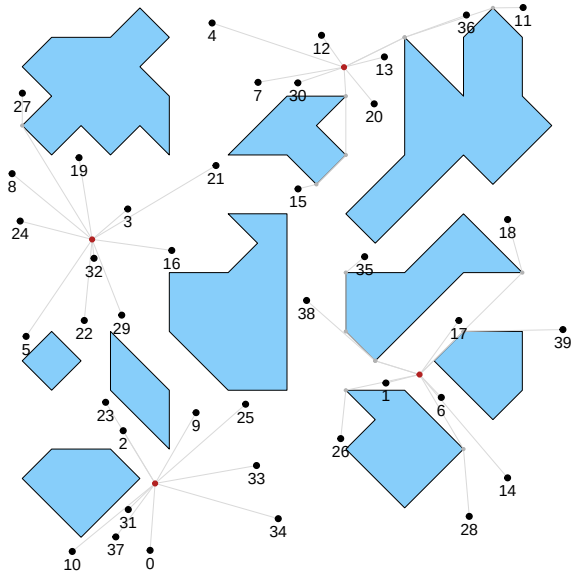
which theoretically enumerates all the possible clusterings. Such a brute force enumeration would limit this approach to only very small problem instances. Thus, we described two pruning criteria. The first one is based on the observation that if we divide the points into two sets, determine the optimal cluster centers for the first set and find the (globally) optimal clustering for the second, the sum of the two loss terms gives a lower bound on all the possible solutions which extend the clustering in the first set. The second criterion is applied for problems without barriers and is based on the observation that in the optimal clustering, convex hulls of the clusters must not overlap. The proposed algorithm allowed us to solve some problem instances of sizes which to our knowledge have not been solved before.



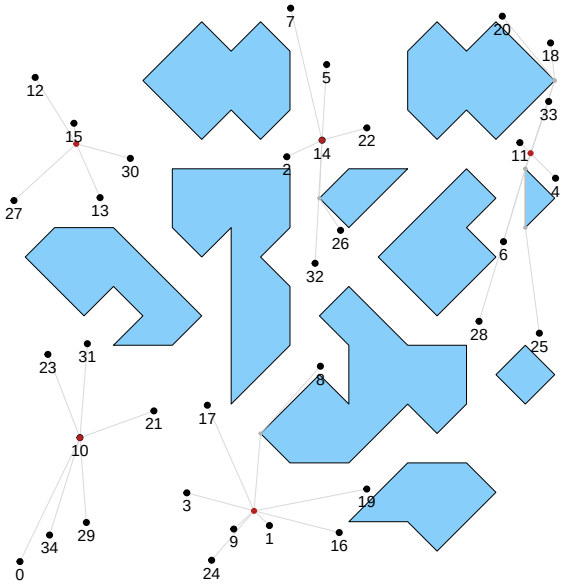
(a) Optimal solution of a $60/2$ problem.



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



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



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

Figure 5.4. Examples of within ϵ-accuracy solved problems ($\epsilon = 1\%$).

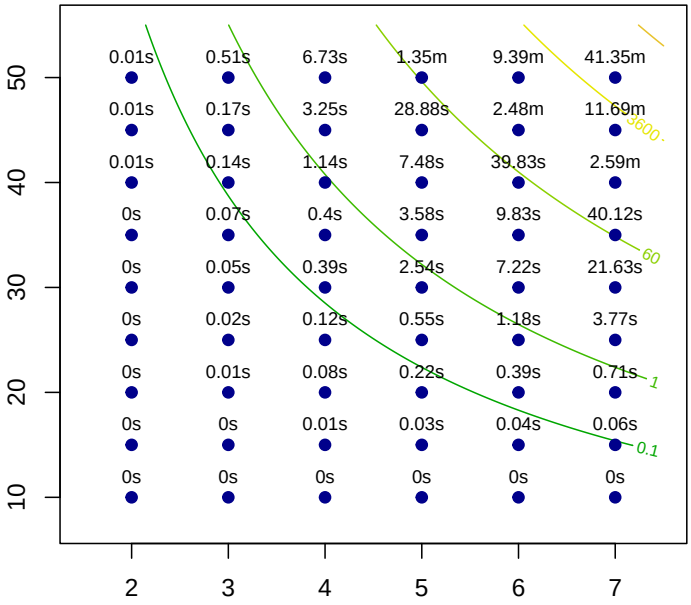


Figure 5.5. Predicted median times of the proposed algorithm for the net-constrained MSSC problem. (A letter next to a number means s=seconds, m=minutes.)

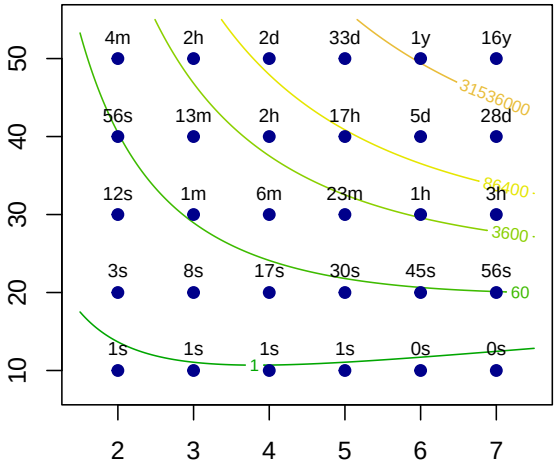


Figure 5.6. Predicted median times when solving the MIQCP problem formulation of the net-constrained MSSC problem with Gurobi. (A letter next to a number means h=hours, d=days, y=years.)

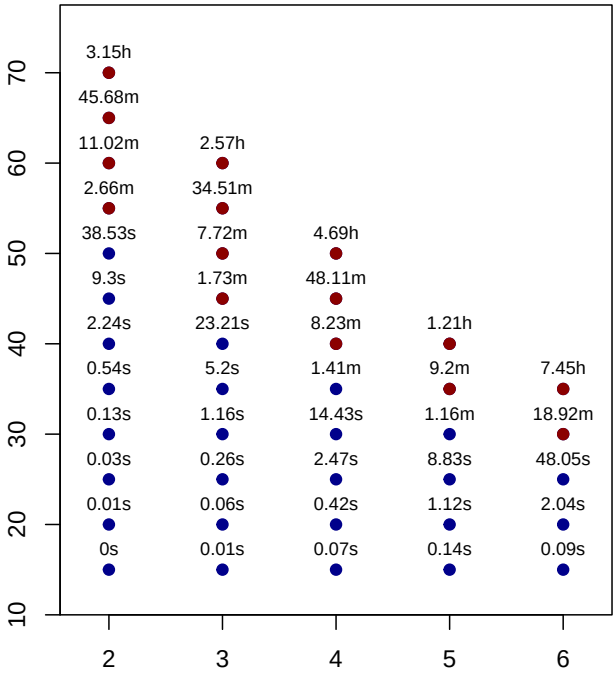


Figure 5.7. Predicted median times of the proposed algorithm for the multi-Weber with polyhedral barriers problem. (A letter next to a number means s=seconds, m=minutes, h=hours.)

GENERAL CONCLUSIONS

In this dissertation, we study constrained planar location problems. We emphasize the key conclusions for the three main problems considered.

1. *Net-constrained equal area Voronoi tessellation problem.* The proposed iterative algorithm (where on every iteration, the problem is locally linearly approximated, and the corresponding linearized problem is formulated as a linear-programming or a mixed-integer-programming problem) provides a high-quality solution for the problem. In our numerical experiment, we successfully obtained high-quality solutions to problem instances with 50 sites using the proposed method.

2. *Net-constrained minimum sum of squares clustering problem.* The proposed enumeration algorithm with convex-hull based shrinking of the enumeration tree performs better than solving the mixed-integer-programming formulation of the problem using a commercial mathematical optimization solver. The proposed algorithm solves problems with sizes[3] up to $50/7$ to global optimality within an hour – for comparison, a mixed-integer-programming formulation would take years to solve. In case of problems larger than $50/7$, a locally optimal solution can be efficiently obtained using the proposed location-allocation (k-means-type) algorithm.

3. *Multi-Weber with polyhedral barriers problem.* The proposed branch and bound algorithm solves problems up to the sizes[3] of $60/2$, $45/3$, $40/4$ and $35/5$ within a desired accuracy to global optimality. Our work presents the first reported global solutions for problems involving more than 30 client points.

[3]We use notation N/K to refer to the problem where K optimal facility points have to be determined for N client points.

[1] D. Aloise, P. Hansen, and L. Liberti. An improved column generation algorithm for minimum sum-of-squares clustering. *Mathematical Programming*, 131(1):195–220, 2012.

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

[3] M. Balzer. Capacity-constrained voronoi diagrams in continuous spaces. In *2009 Sixth International Symposium on Voronoi Diagrams*, pages 79–88, 2009.

[4] R. Belevičius, S. Ivanikovas, D. Šešok, S. Valentinavičius, and J. Žilinskas. Optimal placement of piles in real grillages: Experimental comparison of optimization algorithms. *Information Technology and Control*, 40(2):123–132, 2011.

[5] M. d. Berg, O. Cheong, M. v. Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, 3rd ed. edition, 2008.

[6] E. Birgin, A. Laurain, and T. Menezes. Sensitivity analysis and tailored design of minimization diagrams. *Mathematics of Computation*, 92(344):2715–2768, 2023.

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

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

[9] H.-H. Bock. Origins and extensions of the k-means algorithm in cluster analysis. *Journal Électronique d'Histoire des Probabilités et de la Statistique*, 4, 2008.

[10] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[11] P. S. Bradley, K. P. Bennett, and A. Demiriz. Constrained k-means clustering. *Microsoft Research*, 2000.

[12] M. J. Brusco. A repetitive branch-and-bound procedure for minimum within-cluster sums of squares partitioning. *Psychometrika*, 71 (2):347–363, 2006.

[13] H. Chen. *0-1 Semidefine Programming for Cluster Analysis*. McMaster

University, 2006.

[14] L. Cooper. Location-allocation problems. *Operations Research*, 11(3): 331–343, 1963.

[15] L. Cooper. Heuristic methods for location-allocation problems. *SIAM Review*, pages 37–53, 1964.

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

[17] Z. Drezner and H. W. Hamacher. *Facility location: applications and theory*. Springer Science & Business Media, 2004.

[18] Z. Drezner, A. Mehrez, and G. O. Wesolowsky. The facility location problem with limited distances. *Transportation Science*, 25(3):183–187, 1991.

[19] Z. Drezner, C. H. Scott, and J. G. Turner. Mixed planar and network single-facility location problems. *Networks*, 68:271–282, 2016.

[20] Q. Du, V. Faber, and M. Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999.

[21] O. Du Merle, P. Hansen, B. Jaumard, and N. Mladenovic. An interior point algorithm for minimum sum-of-squares clustering. *SIAM Journal on Scientific Computing*, 21(4):1485–1505, 1999.

[22] O. Du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized column generation. *Discrete Mathematics*, 194(1-3):229–237, 1999.

[23] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

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

[25] S. L. Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13(3):462–475, 1965.

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

[27] P. Hansen, D. Peeters, and J.-F. Thisse. An algorithm for a constrained weber problem. *Management Science*, 28(11):1285–1295, 1982.

[28] I. N. Katz and L. Cooper. Facility location in the presence of forbidden regions, i: Formulation and the case of euclidean distance with

one forbidden circle. *European Journal of Operational Research*, 6(2): 166–173, 1981.

[29] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[30] K. Klamroth. *Single Facility Location Problems with Barriers*. Springer, 2002.

[31] M. J. Kochenderfer and T. A. Wheeler. *Algorithms for Optimization*. The MIT Press, 2019.

[32] B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer Publishing Company, Incorporated, 6th edition, 2018.

[33] S. Krau. *Extensions du problème de Weber*. PhD thesis, École Polytechnique de Montréal, 1997.

[34] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[35] D. G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. Springer, 4th edition, 2016.

[36] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[37] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, 1988.

[38] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Nature, 2006.

[39] R. Paulavičius and J. Žilinskas. *Simplicial Global Optimization*. Springer, 10 2013.

[40] E. Pebesma. Simple Features for R: Standardized Support for Spatial Vector Data. *The R Journal*, 10(1):439–446, 2018.

[41] E. Pebesma and R. Bivand. *Spatial Data Science: With applications in R*. Chapman and Hall/CRC, 2023.

[42] J. Peng and Y. Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM Journal on Optimization*, 18:186–205, 2007.

[43] B. Pfeiffer and K. Klamroth. Bilinear programming formulations for weber problems with continuous and network distances. *Journal of the Operations Research Society of Japan*, pages 123–134, 2005.

[44] V. Piccialli, A. Russo Russo, and A. M. Sudoso. An exact algo-

rithm for semi-supervised minimum sum-of-squares clustering. *Computers & Operations Research*, 147:105958, 2022.

[45] V. Piccialli, A. M. Sudoso, and A. Wiegele. Sos-sdp: An exact solver for minimum sum-of-squares clustering. *INFORMS Journal on Computing*, 34(4):2144–2162, 2022.

[46] M. Ramanauskas, D. Šešok, R. Belevičius, E. Kurilovas, and S. Valentinavičius. Genetic algorithm with modified crossover for grillage optimization. *International Journal of Computers Communications & Control*, 12(3):393–401, 2017.

[47] M. Ramanauskas, D. Šešok, J. Žilinskas, V. Starikovičius, A. Kačeniauskas, and R. Belevičius. Global optimization of grillage-type foundations using a distributed genetic algorithm. *Journal of Global Optimization*, 77(1):157–173, 2020.

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

[49] A. M. Sudoso and D. Aloise. A column generation algorithm with dynamic constraint aggregation for minimum sum-of-squares clustering. *INFORMS Journal on Computing*, 2025.

[50] A. Vattani. The hardness of k-means clustering in the plane. *Manuscript*, 617, 2009.

[51] J. Viegas and P. Hansen. 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, 1985.

[52] E. Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal, First Series*, 43:355–386, 1937.

[53] H. Werner. *k-means Clustering as a Mixed Integer Programming Problem*. KTH Royal Institute of Tehcnology, 2022.

[54] D. Šešok, J. Mockus, R. Belevičius, and A. Kačeniauskas. Global optimization of grillages using simulated annealing and high performance computing. *Journal of Civil Engineering and Management*, 16(1):95–101, 2010.

Articles published in international research journals with a citation index in the Clarivate Analytics Web of Science (CA WoS) database.

Articles in CA WoS journals:

[A.1] M. Kepalas and J. Žilinskas. Solving Net-Constrained Clustering Problem. *Journal of Nonlinear and Variational Analysis*, 8(6):987–1012, 2024.

[A.2] M. Kepalas and J. Žilinskas. Global Optimization Algorithm for the Multi-Weber Problem with Polyhedral Barriers. *Informatica*, 36(4):875–902, 2025.

CURRICULUM VITAE

Mindaugas Kepalas grew up in the city of Panevėžys, where he graduated from Juozas Balčikonis Gymnasium in 2006. In the same year, he started his scientific road as a student of the Financial and Actuarial Mathematics bachelor's program at the Faculty of Mathematics and Informatics (MIF) of Vilnius University. In 2010, he completed the program and joined the ACD/Labs as a statistician, where he worked on prediction models of the properties of chemical compounds. In 2011, he started the Econometrics master's studies at the MIF of the University, which he completed in 2013. After the studies he worked as an algorithm engineer at Neurotechnology, a company that specialises in biometrics. His research at the company was primarily focused on fingerprint recognition algorithms. In 2020, he became a full-time PhD student at the Institute of Data Science and Digital Technologies, the global optimization group, under the supervision of Prof. Julius Žilinskas. Kepalas' scientific interests include not only various optimization problems which are directly related to his research, but also other fields such as discrete mathematics, probability theory and numerical mathematics.

Tyrimo problema

Disertacijoje nagrinėjamus uždavinius trumpai būtų galima pavadinti „vietų parinkimo uždaviniais su apribojimais“. Šiuose uždaviniuose optimizuojamieji kintamieji yra šių „vietų“ koordinatės, o tikslas yra taip išdėlioti tam tikrą skaičių taškų, kad būtų minimizuoti tam tikri kaštai. Papildomai keliamas reikalavimas, kad šie taškai privalo priklausyti tam tikram plokštumos poaibiui. Šis poaibis darbe yra apibrėžiamas kaip iškilųjų simpleksų sąjunga (pvz., atkarpų ir iškilųjų daugiakampių). Taip apibrėžta ribojimų aibė greičiausiai nebus iškila; ši savybė padaro uždavinį nelengvą (tačiau tuo pat metu ir įdomų).

Tyrimo objektas

Šios disertacijos tyrimo objektas yra vietos parinkimo uždaviniai su apribojimais ir algoritmai jiems spręsti. Disertacijoje daugiausia nagrinėjami trys pagrindiniai modeliai.

Vienodo ploto Voronojaus mozaikos uždavinys su ribojimu centrams. Šio uždavinio tikslas yra tam tikrame daugiakampyje taip išdėlioti nurodytą skaičių taškų, kad jų Voronojaus mozaikos ląstelės turėtų vienodus plotus. Šie taškai privalo priklausyti nurodytai ribojimų aibei, pavyzdžiui, „tinklui“ (atkarpų sąjungai).

Mažiausių kvadratų klasterizavimo uždavinys su tinklo ribojimu. Šiame uždavinyje siekiama rasti optimalias „centrų – taškų“ koordinates duotiems „klientams – taškams“, kad Euklido atstumų kvadratų suma nuo „centrų“ iki „klientų“ būtų minimali. Rasti „centrai“ privalo priklausyti uždavinyje nurodytam plokštumos poaibiui, pavyzdžiui, „tinklui“.

Multi-Weber'io uždavinys su barjerais. Šiame uždavinyje plokštumoje yra papildomai apibrėžti barjerai, kuriuose keliavimas yra draudžiamas. Tikslas panašus kaip ir prieš tai buvusiame uždavinyje: reikia rasti tokias „centrų“ koordinates, kad atstumų suma nuo duotų „klientų“ iki „centrų“ būtų minimali.

Tyrimų sritis

Ši disertacija galėtų būti priskiriama toliau nurodytoms tyrimų sritims. Pirmiausia, pagrindinis mūsų indėlis būtų globalaus optimizavimo srityje, nes pirminis tyrimų tikslas buvo nagrinėjamiems uždaviniams pasiūlyti globalius sprendinius randančius algoritmus. Šiam tikslui pasiekti naudojomės įvairiais matematinio optimizavimo įrankiais, pvz., įrankiais tiesinio programavimo arba sveikaskaitinio programavimo uždaviniams spręsti. Taip pat mūsų darbas galėtų būti įdomus besidomintiems skaičiuojamąja geometrija, nes disertacijoje plačiai naudojomės šios informatikos mokslo šakos algoritmais ir rezultatais.

Kalbant apie mūsų tyrimo objektus, apibrėžtus prieš tai buvusiame skyrelyje, šiame darbe pristatytas *vienodo ploto Voronojaus mozaikos uždavinys su ribojimu centrams* galėtų būti priskirtas skaičiuojamosios geometrijos ir matematinio optimizavimo mokslo šakų sankirtai. *Mažiausių kvadratų klasterizavimo uždavinys su tinklo ribojimu* – kaip galima spręsti iš pavadinimo – galėtų būti priskirtas mažiausių kvadratų klasterizavimo uždavinių tyrimų sričiai. Galiausiai, mūsų darbas ties multi-Weber'io uždaviniu su barjerais prisideda prie „objektų vietų“ (angl. „facility location“) uždavinių tyrimų srities.

Darbo aktualumas ir praktinė darbo vertė

Mokslininkai teoretikai mėgsta sakyti, kad visos teorijos yra vertingos, tik kai kurioms iš jų taikymai dar nėra sugalvoti. Šis darbas galbūt pakliūva į tokią kategoriją, ir disertacijos autorius pripažįsta, kad praktinė darbo vertė gali būti diskutuotina. Vis dėlto (dabar pasinaudosime „įrodymu nukreipiant į autoritetą“) tyrimų srityse, kurioms šis darbas gali būti priskirtas (tokioms kaip mažiausių kvadratų klasterizavimo uždavinių tyrimai, objektų vietų optimizavimo uždavinių tyrimai ar matematinis inžinerinių uždavinių modeliavimas), yra parašyta daugybė mokslinių straipsnių ir knygų; taigi šiokia tokia vertė galbūt yra.

Kalbant konkrečiai apie uždavinius, kurie yra nagrinėjami šioje disertacijoje, *vienodo ploto Voronojaus mozaikos uždavinys su ribojimu centrams* anksčiau, regis, nebuvo nagrinėtas mokslinėje literatūroje. Visgi tikimės, kad minėtas taikymas polių balansavimo uždaviniui statybos inžinerijoje modeliuoti suteikia pakankamai motyvacijos studijuoti šį uždavinį.

Mažiausių kvadratų klasterizavimo uždavinys su tinklo ribojimu taip pat

yra naujai pasiūlytas ir galėtų būti pavadintas „netipiniu“ tuo požiūriu, kad atstumas matuojamas Euklido atstumo kvadratu, o tai galbūt nelabai pagrįsta objektų vietų uždaviniams plokštumoje. Toks pirminis metrikos pasirinkimas buvo nulemtas gerai išvystytos mažiausių kvadratų (klasterizavimo) teorijos; bet galbūt pasirinkimas nebuvo geras. Todėl disertacijoje taip pat nagrinėjamas analogiškas uždavinio variantas su standartiniu Euklido atstumu; šiuo atveju uždavinys tampa apibendrintu multi-Weber'io uždaviniu. Reali tokio matematinio modelio interpretacija galėtų būti tokia: tarkime, jog tam tikras „resursas“ (pavyzdžiui, elektra) yra pasiekiamas tinkle. Resurso paklausa yra duota tam tikrose „klientų“ lokacijose, kurios nepriklauso tinklui. Minimas resursas gali būti „nugabentas“ iš tinklo naudojantis tam tikru „prietaisu“ (pavyzdžiui, dronu); šis prietaisas resursą gali „paimti“ tik tam tikrame tinklo taške, kurį galima būtų pavadinti „įkrovimo stotele“. Tikslas yra taip išdėlioti įkrovimo stoteles, kad klientai būtų aptarnauti minimaliais kaštais.

Siekiant geriau realų gyvenimą atitinkačio modelio, multi-Weber'io uždaviniui (kur daroma prielaida, kad visada galima keliauti „tiesia linija“) buvo sugalvotas plėtinys su barjerais (kliūtimis); kirsti šiuos barjerus keliaujant iš vieno taško į kitą šiame modelyje yra draudžiama. Tokie barjerai turi natūralią interpretaciją: pavyzdžiui, jie gali reprezentuoti pastatus, ežerus, kalnus, pelkes, miškus ir t. t. Tikimės, kad toks matematinio modelio priartinimas prie realios gyvenimo problemos skaitytojui suteikia pakankamai motyvacijos jį studijuoti.

Tyrimo tikslas ir uždaviniai

Šios disertacijos tikslas buvo pasiūlyti arba patobulinti algoritmus, sprendžiančius vietų parinkimo uždavinius su apribojimais. Šiam tikslui pasiekti buvo išsikelti šie uždaviniai:

1. Studijuojant mokslinę literatūrą, nustatyti, kokie uždaviniai gali būti atpažinti kaip vietų parinkimo uždaviniai su apribojimais.

2. Ištirti (globalius) algoritmus, naudojamus atitinkamiems vietų parinkimo uždaviniams be apribojimų spręsti.

3. Apibendrinti šiuos algoritmus vietų parinkimo uždaviniams su apribojimais.

4. Nustatyti pasiūlytų algoritmų privalumus, trūkumus ir pritaikomumo ribas.

Ieškodami atsakymo į pirmąjį punktą, apibrėžėme 3 uždavinius, apie kuriuos jau kalbėjome: tai *vienodo ploto Voronojaus mozaikos uždavinys su ribojimu centrams*, *mažiausių kvadratų klasterizavimo uždavinys su tinklo ribojimu* ir *multi-Weber'io uždavinys su barjerais*. Likę trys punktai nulėmė šios disertacijos turinį.

Mokslinis darbo naujumas

- *Vienodo ploto Voronojaus mozaikos uždavinys su ribojimu centrams* iki šiol nebuvo nagrinėtas mokslinėje literatūroje. Disertacijoje aprašėme uždavinio sprendimo algoritmą, šis algoritmas siekia rasti globalų uždavinio sprendinį.
- *Mažiausių kvadratų klasterizavimo uždavinys su tinklo ribojimu* iki šiol nebuvo nagrinėtas mokslinėje literatūroje. Lokaliam uždavinio sprendiniui rasti pasiūlėme k-vidurkių (angl. „k-means“) tipo algoritmą. Taip pat pasiūlėme ir suprogramavome algoritmą, kuris yra tinkamas rasti globalų vidutinio dydžio uždavinių sprendinį.
- Atliktas darbas ties *multi-Weber'io uždaviniu su barjerais* papildo šios įdomios problemos tyrimus. Mūsų žiniomis, šiame darbe pasiūlytas perrankos algoritmas su šakų ir rėžių kriterijumi pirmąkart rado globalius sprendinius $60/2$, $45/3$, $40/4$, $35/5$ dydžio uždaviniams.[4]

Ginamieji teiginiai

1. Disertacijoje pasiūlytas algoritmas, kur kiekvienoje iteracijoje sprendžiama sveikaskaitinio programavimo problema, atitinkanti lokaliai tiesinę uždavinio aproksimaciją, geba rasti aukštos kokybės sprendinį *vienodo ploto Voronojaus mozaikos uždaviniui su ribojimu centrams*.
2. Darbe pasiūlytas perrankos algoritmas su iškilojo apvalkalo kirpimo kriterijumi yra tinkamas rasti globalų sprendinį vidutinio

[4] Čia žymėjimas N/K reiškia uždavinį, kur norime N „klientų“ padalinti į K klasterių, t. y. norime optimaliai parinkti K „centrų–taškų“ duotiems N „klientų–taškų“.

dydžio *mažiausių kvadratų klasterizavimo uždaviniams su tinklo ribojimu*. Didesniems atvejams pasiūlytas k-vidurkių (angl. „k-means“) tipo algoritmas efektyviai randa lokalų sprendinį.

3. Disertacijoje pasiūlytas perrankos algoritmas su apatinio rėžio kriterijumi yra tinkamas rasti globalų sprendinį tokio dydžio *multi-Weber'io uždaviniams su barjerais*, kuriems globalusis sprendinys mokslinėje literatūroje anksčiau nebuvo nustatytas.

Tyrimo aprobavimas ir publikavimas

Publikacijos. Šios disertacijos rezultatai buvo publikuoti šiuose straipsniuose:

1. Mindaugas Kepalas and Julius Žilinskas. „Solving Net-Constrained Clustering Problem“. In: *Journal of Nonlinear and Variational Analysis* (2024), pp. 987–1012.

2. Mindaugas Kepalas and Julius Žilinskas. „Global Optimization Algorithm for the Multi-Weber Problem with Polyhedral Barriers“. In: *Informatica* (2025), pp. 875–902.

Išvardyti straipsniai tiesiogiai susiję su disertacijoje nagrinėjamais uždaviniais:

- *mažiausių kvadratų klasterizavimo uždaviniu su tinklo ribojimu*;
- *multi-Weber'io uždaviniu su barjerais*.

Konferencijos. Disertacijos rezultatai buvo pristatyti 3-jose tarptautinėse konferencijose:

- Mindaugas Kepalas, Julius Žilinskas. „Optimizing Locations on a Network“. EURO 2022, liepos 3–6 d., 2022, Helsinkis, Suomija.
- Mindaugas Kepalas, Julius Žilinskas. „Facility Locations on a Network“. Europt 2022, liepos 29–30 d., 2022, Lisabona, Portugalija.
- Mindaugas Kepalas, Julius Žilinskas. „2-Dimensional Net Constrained Clustering Problem“. The 2023 World Congress on Global Optimization (WCGO 2023), liepos 10–14 d., 2023, Atėnai, Graikija.

Taip pat disertacijos tyrimai buvo pristatyti 4-iose nacionalinėse konferencijose:

- Mindaugas Kepalas, Julius Žilinskas. „Locations on Networks“. DAMSS, gruodžio 2–4 d., 2021, Druskininkai, Lietuva.
- Mindaugas Kepalas, Julius Žilinskas. „Handling Two-Dimensional Net Constraint: Its Formulation in MIP Terms and Empirical Studies of Solvers“. DAMSS, gruodžio 1–3 d., 2022, Druskininkai, Lietuva.
- Mindaugas Kepalas, Julius Žilinskas. „Minimum-Sum-Of-Squares Clustering With (Net) Constraints for Cluster-Centres“. DAMSS, lapkričio 30 – gruodžio 2 d., 2023, Druskininkai, Lietuva.
- Mindaugas Kepalas. „Keletas optimalaus taškų paskirstymo apribotoje plokštumos dalyje uždavinių“. Lietuvos jaunųjų matematikų susitikimas (LJMS), gruodžio 29 d., 2025, Vilnius, Lietuva.

Disertacijos struktūra

Disertaciją sudaro 5 pagrindiniai skyriai. Pirmajame pateikiamos matematinės darbe nagrinėjamų uždavinių formuluotės ir uždavinių pavyzdžiai su iliustracijomis. Antrajame apžvelgiama literatūra, nagrinėta doktorantūros studijų metu. Trečiasis skyrius yra skiriamas vienodo ploto Voronojaus mozaikos uždaviniui. Ketvirtajame skyriuje nagrinėjami vieno optimalaus „aptarnavimo centro“ (angl. „facility location“) uždaviniai. Paskutinis penktasis disertacijos skyrius yra skiriamas globalaus multilokacijų uždavinio sprendinio paieškoms. Galiausiai pateikiamos disertacijos išvados, literatūros sąrašas ir disertacijos santrauka lietuvių kalba.

DISERTACIJOS UŽDAVINIŲ MATEMATINĖS FORMULUOTĖS IR ILIUSTRACIJOS

Šioje disertacijoje nagrinėjami uždaviniai gali būti apibendrinti žemiau esančia formuluote:

$$\begin{aligned} \min_{P_1,\ldots,P_K} \quad & \textbf{cost}(P_1,\ldots,P_K) \\ \text{s.t.} \quad & P_k \in \mathcal{R}^{\cup}, \quad k = 1,\ldots,K. \end{aligned} \tag{S.1}$$

Čia uždavinio kintamieji yra plokštumos taškai $P_1,\ldots,P_K$, kurie privalo priklausyti plokštumos poaibiui $\mathcal{R}^{\cup} \subset \mathbb{R}^2$ (angl. „$\mathcal{R}$egion $\cup$nion“). Šis poaibis disertacijoje apibrėžiamas kaip iškilųjų simpleksų (pvz., atkarpų ir trikampių) sąjunga. Uždavinio tikslas yra rasti tokį taškų išdėstymą, kad jų „kaštai“ (t. y. funkcijos **cost** reikšmė) būtų minimalūs.

Nagrinėjami modeliai

Šioje disertacijoje nagrinėjami trys pagrindiniai uždaviniai, kurie yra formuluotės (S.1) atvejai.

Vienodo ploto Voronojaus mozaikos uždavinys su ribojimu centrams.

$$\min_{P_1,\ldots,P_K} \quad \frac{1}{K}\sum_{k=1}^{K}\left|\textbf{area}(\mathcal{V}_k \cap \mathcal{P}) - \frac{\textbf{area}(\mathcal{P})}{K}\right| \tag{S.2a}$$

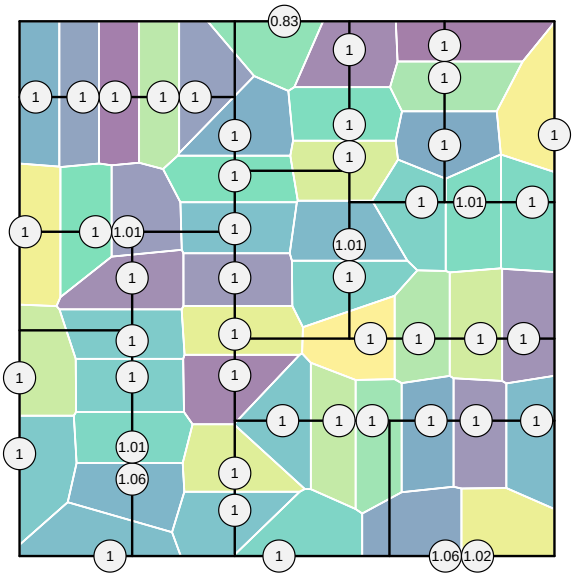
$$\text{s.t.} \quad P_k \in \mathcal{N}, \quad k = 1,\ldots,K. \tag{S.2b}$$

Šioje formuluotėje:

- $\mathcal{N}$ yra tinklas (atkarpų sąjunga),
- $\mathcal{V}_k$ yra Voronojaus ląstelė, atitinkanti tašką P_k,
- $\mathcal{P}$ yra koks nors daugiakampis.

Kitaip tariant, mūsų tikslas yra taip išdėlioti K taškų $P_1,\ldots,P_K$ laikantis ribojimo $P_k \in \mathcal{N}$, kad visos Voronojaus ląstelės turėtų kuo panašesnius plotus. Žr. iliutraciją S.1 pav.

Pasiūlytas uždavinys (S.2) yra naujas tuo požiūriu, kad centrų priklausymo tinklui sąlyga (S.2b) anksčiau nebuvo nagrinėta mokslinėje literatūroje. Be šio ribojimo vienodo ploto Voronojaus mozaikos uždavinys buvo nagrinėtas, pvz., Birgin et al. [6].



S.1 pav. Vienodo ploto Voronojaus mozaikos uždavinys su ribojimu centrams. Skaičiai apskritimuose nurodo atitinkamos Voronojaus ląstelės plotą.

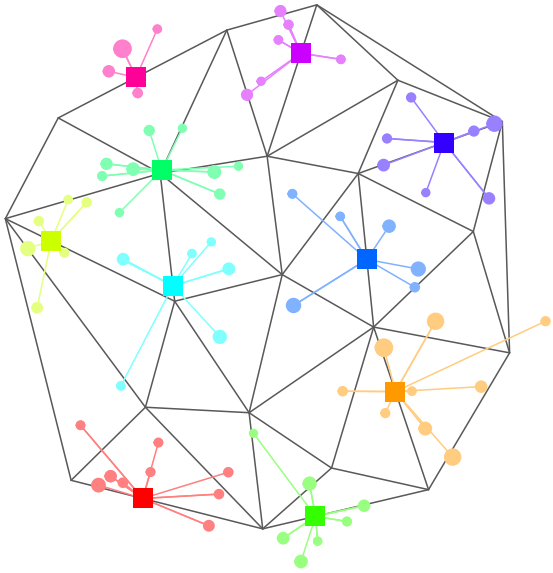
Mažiausių kvadratų klasterizavimo uždavinys su tinklo ribojimu.

$$\min_{P_1,\dots,P_K} \quad \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} w_i \left\| A_i - P_k \right\|_2^2 \tag{S.3a}$$

$$\text{s.t.} \quad P_k \in \mathcal{N}, \quad k = 1, \dots, K. \tag{S.3b}$$

Šioje formuluotėje:

- $\mathcal{N}$ yra tinklas (atkarpų sąjunga),
- (A_n, w_n), $n = 1, \dots, N$ yra duota aibė taškų su svoriais („klientai“ lokacijose A_n su „poreikiais“ w_n),
- $\mathcal{C}_k$, $k = 1, \dots, K$ yra „klientų“ padalinimas į grupes („klasterius“); t. y. apibrėžtos grupės tenkina sąlygas $\mathcal{C}_k \cap \mathcal{C}_l = \emptyset$ kai $l \neq k$ bei sąlygą $\cup_{k=1}^{K} \mathcal{C}_k = \{1, \dots, N\}$,
- $\|\bullet\|_2^2$ yra Euklido atstumo kvadratas.



S.2 pav. Mažiausių kvadratų klasterizavimo uždavinys su tinklo ribojimu. Pastebėkite, kad visi „aptarnavimo centrai“ P_k (žymimi kvadratėliais) tenkina reikalavimą $P_k \in \mathcal{N}$.

Kitaip tariant, siekiame rasti tokias „aptarnavimo centrų“ lokacijas P_k ir taip priskirti joms „klientus“ A_n, kad aptarnavimo kaštai, kurie matuojami svertine Euklido atstumo kvadratų suma, būtų minimalūs. Žr. iliustraciją S.2 pav.

Aprašytas uždavinys yra naujas tuo požiūriu, kad tinklo ribojimas centrams (S.3b) mažiausių kvadratų klasterizavimo uždaviniui pirmąkart nagrinėtas mūsų publikacijoje [A.1]. Klasikinei formuluotei be ribojimo (S.3b) yra parašyta daugybė literatūros, pavyzdžiui, k-vidurkių (angl. „k-means“) algoritmas, randantis lokalų uždavinio sprendinį, gavo savo pavadinimą pagal MacQueen [36] straipsnį. Globalaus sprendinio problema yra nagrinėjama, pvz., Aloise et al. [1], kur autoriai pasinaudoja stulpelių generavimo metodo principu (Gilmore and Gomory [24]).

Multi-Weber'io uždavinys su barjerais. Šiame uždavinyje papildomai yra apibrėžti barjerai $\{\mathcal{B}_1, \ldots, \mathcal{B}_L\}$, kuriuose keliavimas draudžia-

mas. Pateikiame matematinę uždavinio formuluotę:

$$\begin{aligned} \min_{P_1,\ldots,P_K} \quad & \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} w_i d_{\mathcal{B}}\left(A_i, P_k\right) \\ \text{s.t.} \quad & P_k \in \mathcal{R}^{\cup}, \quad k = 1, \ldots, K. \end{aligned} \tag{S.4}$$

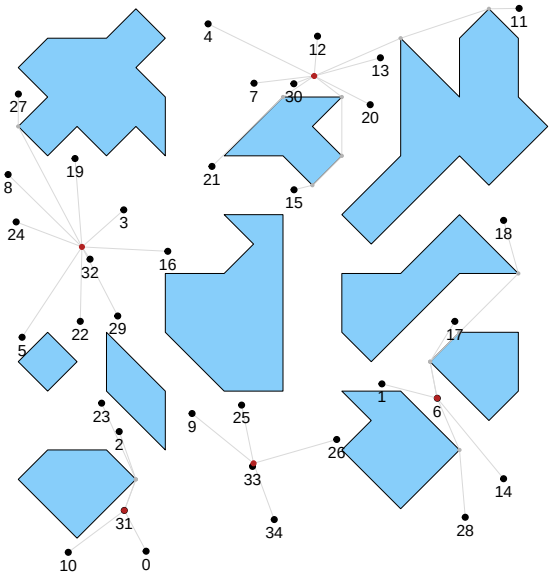
Šioje formuluotėje:

- ribojimų aibė $\mathcal{R}^{\cup}$ „aptarnavimo centrams“ yra lygi $\mathbb{R}^2 \setminus \cup_{l=1}^{L} \mathcal{B}_l$,
- (A_n, w_n), $n = 1, \ldots, N$ yra aibė taškų su svoriais („klientai“ lokacijose A_n su „poreikiais“ w_n),
- $\mathcal{C}_k$, $k = 1, \ldots, K$ yra „klientų“ padalinimas į grupes („klasterius“); t. y. apibrėžtos grupės tenkina sąlygas $\mathcal{C}_k \cap \mathcal{C}_l = \emptyset$ kai $l \neq k$ bei sąlygą $\cup_{k=1}^{K} \mathcal{C}_k = \{1, \ldots, N\}$,
- $d_{\mathcal{B}}\left(\bullet, \bullet\right)$ yra trumpiausias atstumas tarp taškų atsižvelgiant į barjerus.

Kitaip tariant, siekiame rasti tokias „aptarnavimo centrų“ lokacijas P_k ir taip priskirti joms „klientus“ A_n, kad aptarnavimo kaštai, matuojami trumpiausiais atstumais atsižvelgiant į barjerus, būtų minimalūs. Žr. iliustraciją S.3 pav.

Pasinaudodami disertacijoje ir straipsnyje [A.2] aprašytu perrankos algoritmu su šakų ir rėžių kirpimo kriterijumi, sugebėjome išspręsti $60/2$, $45/3$, $40/4$, $35/5$ dydžio[5] uždavinius. Šis algoritmas aprašytas skyriuje *Chapter 5*; apie kitus mokslinėje literatūroje taikytus metodus uždaviniui (S.4) spręsti galima pasiskaityti skyrelyje *Section 2.5.2*.

[5] Primename, kad žymėjimas N/K reiškia uždavinį, kur duotiems N „klientų-taškų“ reikia rasti K optimalių „centrų-taškų“.



S.3 pav. Multi-Weber'io uždavinys su barjerais.

DARBO IŠVADOS

Šioje disertacijoje nagrinėjami vietų optimizavimo uždaviniai su apribojimu. Pateikiame išvadas kiekvienam iš trijų pagrindinių disertacijoje nagrinėtų modelių.

1. *Vienodo ploto Voronojaus mozaikos uždavinys su ribojimu centrams:* disertacijoje pasiūlytas algoritmas (kur kiekvienoje iteracijoje uždavinys yra tiesiškai aproksimuojamas ir suformuluojama šią aproksimaciją atitinkanti tiesinio arba sveikaskaitinio programavimo problema) randa aukštos kokybės sprendinį šiam uždaviniui. Atlikus skaitinį eksperimentą, sėkmingai rasti aukštos kokybės sprendiniai uždaviniams su 50 Voronojaus taškų.

2. *Mažiausių kvadratų klasterizavimo uždavinys su tinklo ribojimu:* pasiūlytas perrankos algoritmas su iškilojo apvalkalo šakų kirpimo kriterijumi leidžia spręsti tokius uždavinių atvejus, kurių sveikaskaitinės formuluotės komerciniai matematinio optimizavimo įrankiai nepajėgia išspręsti. Pasiūlytu algoritmu $50/7$ dydžio[6] uždaviniai išsprendžiami apytiksliai per valandą, kai palyginimui sveikaskaitinę uždavinio formuluotę komercinis įrankis spręstų kelerius metus (pagal prognozę). Jeigu reikia spręsti uždavinius, didesnius negu $50/7$, lokalų sprendinį galima efektyviai nustatyti pasinaudojant disertacijoje aprašytu k-vidurkių (angl. „k-means“) tipo algoritmu.

3. *Multi-Weber'io uždavinys su barjerais:* pasiūlytas šakų ir rėžių algoritmas geba rasti globalų sprendinį uždaviniams iki dydžio[6] $60/2$, $45/3$, $40/4$, $35/5$ (norimu tikslumu). Ši disertacija yra pirmas darbas, kuriame buvo rasti globalūs sprendiniai uždavinio atvejams su daugiau negu 30 „klientų“.

[6] Čia žymėjimas N/K reiškia uždavinį, kur norime N „klientų“ padalinti į K klasterių, t. y. norime optimaliai parinkti K „centrų–taškų“ duotiems N „klientų–taškų“.

PADĖKA (ACKNOWLEDGEMENTS)

Noriu padėkoti visiems, kas lydėjo mane šioje mokslinėje kelionėje.

- Savo doktorantūros vadovui prof. dr. Juliui Žilinskui – už diskusijas, susitikimus, už pagalbą kritiniais momentais, už paliktą erdvę ieškoti savo doktorantūros tyrimų kelio ir pasitikėjimą, netgi kai „nuklysdavau į šalį“.

- Visiems DMSTI kolegoms – už patarimus įvairiais doktorantūros klausimais, pastabas ir pasiūlymus po pranešimų, už įdomius doktorantūros kursus: doc. dr. Viktorui Medvedevui, prof. dr. Olgai Kurasovai ir kitiems.

- „Kambariokui“ doc. dr. Martynui Sabaliauskui – už įdomius pokalbius matematinėmis ir kitomis temomis.

- Kitiems Instituto doktorantams – už kartu praleistą laiką konferencijose, kursuose ar kitomis progomis, už pagalbą vienas kitam įvairiais doktorantūros aspektais.

- Abiem Instituto direktoriams doc. dr. Gintautui Tamulevičiui ir prof. habil. dr. Gintautui Dzemydai – už suteiktą galimybę dirbti Institute.

- Snieguolei Meškauskienei ir Danutei Rimeisienei – už pagalbą administraciniais doktorantūros klausimais.

- Prof. dr. Virginijui Marcinkevičiui – už disertacijos LaTeX šabloną.

- Recenzentams prof. dr. Remigijui Paulavičiui ir doc. dr. Algirdui Lančinskui, perskaičiusiems pirminę („juodraštinę“) disertacijos versiją.

Ir, žinoma, dėkoju savo artimiausiems: Tėvams Danguolei ir Jonui už visokeriopą pagalbą ir palaikymą bei Justinai už neišsenkantį tikėjimą manimi: manau, kad šios kelionės be Tavęs aš nebūčiau įveikęs.

FOR NOTES / UŽRAŠAMS

Mindaugas Kepalas
Searching for the Global Solution of Constrained Location Problems
Doctoral Dissertation
Natural Sciences
Informatics (N 009)
Thesis Editor: Zuzana Šiušaitė

Mindaugas Kepalas
Globaliojo sprendinio paieška vietos parinkimo uždaviniams su apribojimais
Daktaro disertacija
Gamtos mokslai
Informatika (N 009)
Santraukos redaktorė: Dalia Gedzevičienė

Vilnius University Press
9 Saulėtekio Ave., Building III, LT-10222 Vilnius
Email: info@leidykla.vu.lt, www.leidykla.vu.lt
bookshop.vu.lt, journals.vu.lt
Print run of 20 copies