A new solution approach for solving the 2-facility location problem in the plane with block norms *

Andrea Maier

July 24, 2015

Department of Mathematics, University of Kaiserslautern,
P.O. Box 3049, 67653 Kaiserslautern, Germany
maier@mathematik.uni-kl.de

Abstract

Motivated by the time-dependent location problem over T time-periods introduced in Maier and Hamacher (2015) we consider the special case of two time-steps, which was shown to be equivalent to the static 2-facility location problem in the plane. Geometric optimality conditions are stated for the median objective. When using block norms, these conditions are used to derive a polygon grid inducing a subdivision of the plane based on normal cones, yielding a new approach to solve the 2-facility location problem in polynomial time. Combinatorial algorithms for the 2-facility location problem based on geometric properties are deduced and their complexities are analyzed. These methods differ from others as they are completely working on geometric objects to derive the optimal solution set.

Keywords: optimal dynamic locations, optimal trajectory, block norm, public event

1 Introduction

A location problem is considered which arises from the special case of the dynamic location problem introduced in Maier and Hamacher (2015). The latter is motivated by the modeling of sales- and security personnel, that is part of a larger project in which security and commercial issues in public events are tackled.

At a public event visitors are changing their position depending on their interest. By experience organizers of the event can often estimate how people are moving and at which time. To improve profit, security, etc. there is often sales- or security personnel walking around. This staff will be modeled by a trajectory in the plane which can be identified with a time-dependent location model over discrete time-steps.

In this report, we will in particular deal with the single facility dynamic location problem with two time-steps which is equivalent to the static 2-facility location problem. We consider the distance function induced by block norms and give a new algorithm based on the geometric properties of the optimality conditions that are given later. For a literature summary on planar time-dependent location problems we refer to Maier and Hamacher (2015). The problem definition will be given in Section 2 and optimality conditions will be stated. These optimality conditions are based on a 2-commodity flow in an associated network. That flow defines geometric objects such as convex polygons (cells), facets (edges) and extreme points (vertices) of these polygons.

In Section 3 the interrelation between the geometric objects and the divergence of the flow is analyzed. Related work was done by Wagner (2015), who uses methods of functional analysis to

*This work was in part supported by a grant of the German Ministry of Research and Technology (BMBF) under grant FKZ 13N12826

derive a dual grid which is similar to our results. The two approaches differ from each other as ours emphasizes geometric properties.
Moreover, in Section 4 this interrelation is used to derive algorithms for the two facility location problem and to analyze their complexity. In contrast to other approaches the methods in this report will not only consider single points in the plane to derive the optimal solution set, they consider whole geometric objects. In the last section, a summary of our results is given and further improvements are discussed.

2 Problem definition

In the dynamic single facility location problem, the trajectory of a sales or security person is given by points $x_t \in \mathbb{R}^2$ for a set of equidistant, discrete time-steps $t \in \mathcal{T} := \{1, \dots, T\}$. As $X = (x_t)_{t \in \mathcal{T}} \in \mathbb{R}^{2T}$ can be interpreted as a *trajectory* in the plane, the dynamic single facility problem could also be coined *optimal trajectory problem*.
Each change in the position of the staff induces some cost $v_t \in \mathbb{R}_{>0}$ ($t \in \mathcal{T} \setminus T$) per unit distance between x_t and x_{t+1}. For each time-step, there are M demand points $d_{tm} \in \mathbb{R}^2$ and weights $w_{tm} \geq 0$. We denote the common index set with $\mathcal{M} := \{1, \dots, M\}$. With $\mathcal{D}_t := \{d_{tm} \in \mathbb{R}^2, m \in \mathcal{M}\}$ we refer to the set of demand points for a specific time-step $t \in \mathcal{T}$. For the unweighted version all weights are equal to 1.
Various possible objective functions and constraints were considered in Maier and Hamacher (2015). In this paper we focus on the *median objective* given by

$$\min_{X \in \mathbb{R}^{2T}} F(X) := \sum_{t=1}^{T-1} v_t \|x_{t+1} - x_t\| + \sum_{t=1}^{T} \sum_{m=1}^{M} w_{tm} \|d_{tm} - x_t\|. \quad (1)$$

Using the classification scheme of Hamacher and Nickel (1998) the problem can be classified by $1/\mathbb{R}^2/dyn/\bullet/\Sigma$, where the first entry specifies the number of facilities, the second entry denotes that we are considering the problem in the plane, the third that it is a dynamic problem, the fourth specifies the distance measure or $\bullet$ is used if we do not want to specify it and the last entry will show which objective function is used. The optimal solution set will be identified with $\mathcal{X}^*(1/\mathbb{R}^2/dyn/\bullet/\Sigma)$ or $\mathcal{X}^*$ if it is clear which problem is considered.
In Maier and Hamacher (2015) it has been shown that $1/\mathbb{R}^2/dyn/\bullet/\Sigma$ is a special case of the static multi-facility problem $N/\mathbb{R}^2/stat/\bullet/\Sigma$, therefore known algorithms can be applied. Considering block norms which have been introduced by Ward and Wendell (1980), there are already several algorithms in existing literature, cf. Fliege (1998), Idrissi et al. (1989) and Ward and Wendell (1985). The latter gives an LP-Formulation of the problem, such that it can be solved by linear programming algorithms in polynomial time.

Subsequently we will use the following notations: For a set $S \subset \mathbb{R}^2$, $bd(S), int(S)$ stands for the boundary, resp. interior of S. The dot product will be denoted by $\langle \cdot, \cdot \rangle$.

Definition 2.1
Let $B \subset \mathbb{R}^2$ be a compact, convex, symmetric polytope with 0 as its center. Then the *block norm* with unit ball B is given by

$$\gamma(v) := \inf\{\mu \geq 0 \mid v \in \mu B\}. \quad (2)$$

With $\mathsf{Ext}(B) := \{b_1, \dots, b_R\}$ we refer to the extreme points of B.

Definition 2.2
The *polar set* of B is defined by

$$B^\circ = \left\{v \in \mathbb{R}^2 \;\middle|\; \langle b_r, v \rangle \leq 1, \forall r = 1, \dots, R\right\} \quad (3)$$

with extreme points of $\mathsf{Ext}(B^\circ) = \{b_1^\circ, \dots, b_R^\circ\}$. Its corresponding norm will be denoted by $\gamma^\circ(\cdot)$.

We denote the common index set of extreme points of B and B° with $\mathcal{R}$. It can be shown that the block norm can be calculated as (see Ward and Wendell (1985))

$$\begin{aligned}\gamma(v) &= \min\left\{\sum_{r=1}^{R}|\beta_r| \;\middle|\; v=\sum_{r=1}^{R}\beta_r b_r\right\} \\ &= \max\left\{|\langle v, b_r^\circ\rangle| \mid r=1,\dots,R\right\}.\end{aligned} \tag{4}$$

Definition 2.3
The *normal cone* $K(p)$ to B° at $p \in B^\circ$ is defined by

$$K(p) := \{x \in \mathbb{R}^n \mid \langle x, q-p\rangle \le 0 \;\forall q \in B^\circ\}.$$

As $1/\mathbb{R}^2/dyn/\bullet/\Sigma$ is a special case of the multi-facility location problem, the following optimality conditions can be derived using the results of Lefebvre et al. (1990):

Theorem 2.4 (Lefebvre et al. (1990))**.**

(i) If X is an optimal solution to $1/\mathbb{R}^2/dyn/\bullet/\Sigma$, then there exist vectors $p_{tm} \in \mathbb{R}^2$ for $t \in \mathcal{T}$, $m \in \mathcal{M}$ and $\tilde{p}_t$ for $t \in \mathcal{T} \cup \{0\}$ with $\tilde{p}_0 = \tilde{p}_T = 0$, satisfying the conservation constraints:

$$\sum_{m\in\mathcal{M}} w_{tm}p_{tm} + v_t\tilde{p}_t - v_{t-1}\tilde{p}_{t-1} = 0 \qquad \forall t \in \mathcal{T}, \tag{5}$$

the cone conditions

$$x_t \in d_{tm} + K(p_{tm}) \qquad \forall t \in \mathcal{T}, m \in \mathcal{M}, \tag{6}$$

$$x_t \in x_{t+1} + K(\tilde{p}_t) \qquad \forall t \in \mathcal{T}\setminus T, \tag{7}$$

and ball conditions

$$p_{tm} \in B^\circ \qquad \forall t \in \mathcal{T}, m \in \mathcal{M}, \tag{8}$$

$$\tilde{p}_t \in B^\circ \qquad \forall t \in \mathcal{T}\setminus T. \tag{9}$$

(ii) Let $p = (p_{tm})_{m\in\mathcal{M},t\in\mathcal{T}}$, $\tilde{p} = (\tilde{p}_t)_{t\in\mathcal{T}}$ and $P = (p,\tilde{p})$ be a vector satisfying the conservation constraints (5) and the ball conditions (8) and (9) on p and $\tilde{p}$. If there exists an $\tilde{X}$ such that the pair $(\tilde{X}, P)$ also satisfies the cone conditions, then $\tilde{X}$ is optimal. Moreover, X' is also an optimal solution if and only if (X', P) also satisfies these cone conditions.

In Lefebvre et al. (1990) a graph is associated with the location problem and P can be identified as a flow on this graph. Constraints (5) are the *flow conservation constraints* of this graph. The left hand side of equation (5) is equal to the *divergence* value introduced in Lefebvre et al. (1990) and will be denoted by $\mathsf{div}_t(P)$. Moreover, if (X, P) fulfills equations (6)-(9), then $\mathsf{div}_t(P) \in \partial F(X)$, where $\partial F(X)$ denotes the subdifferential of F at X. For the interested reader we refer to Plastria (1992), Rockafellar (1997) and Wagner (2015).

Definition 2.5
Let $p = (p_{tm})_{t\in\mathcal{T},m\in\mathcal{M}} \in \mathbb{R}^{2TM}$ and $\tilde{p} = (\tilde{p}_t)_{t\in\mathcal{T}\cup\{0\}}$ with $\tilde{p}_0 = \tilde{p}_T = 0$. Then the weighted *divergence* for $1/\mathbb{R}^2/dyn/\bullet/\Sigma$ at time-step $t \in \mathcal{T}$ for $P = (p,\tilde{p})$ is given by

$$\mathsf{div}_t(P) := \sum_{m\in\mathcal{M}} w_{tm}p_{tm} + v_t\tilde{p}_t - v_{t-1}\tilde{p}_{t-1}.$$

Definition 2.6
Note the ball conditions (8) can be rewritten for each time-step $t \in \mathcal{T}$ as

$$p_t = (p_{tm})_{m\in\mathcal{M}} \in (B^\circ)^M := \{(p_m)_{m\in\mathcal{M}} \mid p_m \in B^\circ, m \in \mathcal{M}\}. \tag{8a}$$

Then for each time-step $t \in \mathcal{T}$ and $p \in (B^\circ)^M$ the cone conditions (6) define *geometric objects*:

$$G_t^p := \left\{ x \in \mathbb{R}^2 \;\middle|\; x \in \bigcap_{m \in \mathcal{M}} d_{tm} + K(p_m) \right\}.$$

More precisely, if G_t^p is non-empty, it is either a *proper cell* of dimension 2, an *edge* (*facet*) of a cell or a *vertex* (*intersection point*), i.e., an extreme points of a cell. The set of extreme points will be denoted by $\mathcal{I}(\mathcal{D}_t)$. The *set of geometric objects for a time-step* $t \in \mathcal{T}$ is given by:

$$\mathcal{G}_t := \left\{ G_t^p \;\middle|\; G_t^p \neq \emptyset, p \in (B^\circ)^M \right\}.$$

Definition 2.7
Let $G \in \mathcal{G}_t$ be a geometric object, we define the set of *possible flows associated with* G as

$$\mathcal{P}_t^G := \left\{ p \in (B^\circ)^M \;\middle|\; G = \bigcap_{m \in \mathcal{M}} d_{tm} + K(p_m) \right\}.$$

Remark 2.8
The sets G_t^p and $\mathcal{P}_t^G$ are inverse to each other since $p \in \mathcal{P}_t^G$ implies $G = G_t^p \in \mathcal{G}_t$ and $G = G_t^p \in \mathcal{G}_t$ implies $p \in \mathcal{P}_t^G$.

3 Geometric properties of the divergence values

In Maier and Hamacher (2015) a polynomial algorithm for the 2-facility location problem with block norms $2/P/stat/\gamma/\Sigma$ is given. This algorithm constructs a finite dominating set and finds the optimal solution by complete enumeration. How to replace the complete enumeration by a search along an improving direction is an open question. In this report the basics for a search direction to find the optimal solution set $\mathcal{X}^*(2/P/stat/\gamma/\Sigma)$ more efficiently are introduced.
The conservation constraint for the two facility location problem

$$\mathsf{div}_1(P) = \sum_{m \in \mathcal{M}} w_{1m} p_{1m} + v_1 \tilde{p}_1 = 0$$
$$\mathsf{div}_2(P) = \sum_{m \in \mathcal{M}} w_{2m} p_{2m} - v_1 \tilde{p}_1 = 0$$

imply

$$v_1 \tilde{p_1} = \underbrace{\sum_{m \in \mathcal{M}} w_{2m} p_{2m}}_{\text{A}} = - \underbrace{\sum_{m \in \mathcal{M}} w_{1m} p_{1m}}_{\text{B}}.$$

The sums (A) and (B) are nothing else than the divergence values for the two static single facility location problems $1/\mathbb{R}^2/stat, w = w_t/\gamma/\Sigma$ with weights $w_t = (w_{tm})_{m \in \mathcal{M}}$ for each $t = 1, 2$. This gives the motivation for the following definition.

Definition 3.1
The *set of divergence values associated with a geometric object* $G \in \mathcal{G}_t$ is defined as

$$\mathsf{DiVal}_t(G) = \left\{ \sum_{m \in \mathcal{M}} w_{tm} p_{tm} \;\middle|\; (p_{tm})_{m \in \mathcal{M}} \in \mathcal{P}_t^G \right\}.$$

Hence, for two geometric objects $G_1 \in \mathcal{G}_1$, $G_2 \in \mathcal{G}_2$ all possible values for $v_1 \tilde{p}_1$ have to lie in $-\mathsf{DiVal}_1(G_1) \cap \mathsf{DiVal}_2(G_2)$ to fulfill the conservation constraints. If $\tilde{p}_1$ also satisfies the cone conditions (7), then we have an optimal flow P and we can derive $\mathcal{X}^*(2/\mathbb{R}^2/stat/\gamma/\Sigma)$. Therefore, we will first take a closer look at the divergence for the single facility location problem.

Static Case $1/\mathbb{R}^2/stat/\gamma/\Sigma$

The objective function of $1/\mathbb{R}^2/stat/\gamma/\Sigma$ is given by

$$\min_{x\in\mathbb{R}^2} F_{stat}(x) := \sum_{m\in\mathcal{M}} w_m\gamma(x-d_m).$$

The cone and ball conditions can be restated as

$$\begin{aligned} &x \in d_m + K(p_m), && m\in\mathcal{M},\\ &p_m \in B^\circ, && m\in\mathcal{M}, \end{aligned}$$

and the conservation constraints as:

$$\mathsf{div}(p) = \sum_{m\in\mathcal{M}} w_m p_m = 0.$$

As there is only one time-step we omit the index t and denote the sets of geometric objects with $\mathcal{G}$ and for each $G \in \mathcal{G}$ the set of possible flows with $\mathcal{P}^G$.

Theorem 3.2.
Any pair of different proper cells of dimension 2 have different unique divergence sets, i.e., for $C_1, C_2 \in \mathcal{G}$ with $dim(C_1) = dim(C_2) = 2$ and $C_1 \neq C_2$:

$$\begin{aligned} \mathsf{DiVal}(C_1) &\neq \mathsf{DiVal}(C_2), \ \textit{and}\\ |\mathsf{DiVal}(C_1)| &= |\mathsf{DiVal}(C_2)| = 1. \end{aligned}$$

Proof. Let C_1 and C_2 be two different proper cells of dimension 2 with flows $p^1 = (p^1_m)_{m\in\mathcal{M}} \in \mathcal{P}^{C_1}$ and $p^2 = (p^2_m)_{m\in\mathcal{M}} \in \mathcal{P}^{C_2}$. As the cells are proper, each flow value p^i_m is one of the extreme points of B° and thus uniquely determined, therefore $|\mathsf{DiVal}(C_1)| = |\mathsf{DiVal}(C_2)| = 1$.
Suppose that $\mathsf{div}(p^1) = \mathsf{div}(p^2)$, i.e.,

$$\sum_{m\in\mathcal{M}} w_m p^1_m = \sum_{m\in\mathcal{M}} w_m p^2_m. \tag{10}$$

By (4) and the definition of the normal cones, we also have for all $x \in int(C_1)$ and $m \in \mathcal{M}$:

$$\begin{aligned} \gamma(x-d_m) &= \langle x-d_m, p^1_m\rangle\\ &\geq \langle x-d_m, p^2_m\rangle. \end{aligned}$$

Since $C_1 \neq C_2$, there is at least one $m \in \mathcal{M}$ for which $p^1_m \neq p^2_m$ holds and as a consequence, we have a strict inequality for at least one m. Summing up, we get

$$\begin{aligned} \sum_{m\in\mathcal{M}} \langle x, w_m p^1_m\rangle - \langle d_m, w_m p^1_m\rangle &= \sum_{m\in\mathcal{M}} w_m\langle x-d_m, p^1_m\rangle\\ &> \sum_{m\in\mathcal{M}} w_m\langle x-d_m, p^2_m\rangle\\ &= \sum_{m\in\mathcal{M}} \langle x, w_m p^2_m\rangle - \langle d_m, w_m p^2_m\rangle\\ &= \langle x, \sum_{m\in\mathcal{M}} w_m p^2_m\rangle - \sum_{m\in\mathcal{M}} \langle d_m, w_m p^2_m\rangle\\ &\overset{(10)}{=} \langle x, \sum_{m\in\mathcal{M}} w_m p^1_m\rangle - \sum_{m\in\mathcal{M}} \langle d_m, w_m p^2_m\rangle\\ &= \sum_{m\in\mathcal{M}} \langle x, w_m p^1_m\rangle - \langle d_m, w_m p^2_m\rangle \end{aligned}$$

$$\implies \sum_{m\in\mathcal{M}} \langle d_m, w_m p^1_m\rangle < \sum_{m\in\mathcal{M}} \langle d_m, w_m p^2_m\rangle,$$

which is independent of x. By the same argument, we must have for $x \in int(C_2)$:

$$\sum_{m\in\mathcal{M}} \langle d_m, w_m p_m^2\rangle \; < \; \sum_{m\in\mathcal{M}} \langle d_m, w_m p_m^1\rangle,$$

giving a contradiction. □

For the next proof the following definition is needed.

Definition 3.3
Each extreme point of B defines a fundamental direction $f_r = \{\lambda b_r \mid \lambda \geq 0\}$ for $r \in \mathcal{R}$. A *construction line* defined by f_r at point d is given by:

$$L^r(d) := \left\{x \in \mathbb{R}^2 \;\middle|\; x = d + f_r\right\}.$$

Theorem 3.4. *Let E be the common edge of two proper adjacent cells C_1 and C_2, i.e., $E = C_1 \cap C_2$, each cell identified with a unique flow $p^1 = (p_m^1)_{m\in\mathcal{M}}$ and $p^2 = (p_m^2)_{m\in\mathcal{M}}$, respectively. Then any flow p^E defining the edge E must lie strictly between $\mathsf{div}(p^1)$ and $\mathsf{div}(p^2)$, i.e.,*

$$\mathsf{div}(p^E) \in \mathsf{DiVal}(E) = \left\{\mathsf{div}(p^1) + \lambda\left[\mathsf{div}(p^2) - \mathsf{div}(p^1)\right] \;\middle|\; \lambda \in (0,1)\right\}.$$

Proof. Since the two cells have a common edge E, there must be a construction line separating them. W.l.o.g. assume that the construction line passes through $d_j, \ldots, d_M$ for a $j \in \mathcal{M}$, i.e., $E \subset L^r(d_m)(m = j, \ldots, M)$, $r \in \mathcal{R}$.
Note that for all $m = 1, \ldots j-1$ we have $p_m^1 = p_m^2 = b_{i_m}^\circ$ for some extreme points $b_{i_m}^\circ \in \mathsf{Ext}(B^\circ)$. In case of $m \in \{j, \ldots, M\}$, the flows of the two cells are different, therefore, we must separate between those with demand points on one side of the edge and those with demand points on the other side. W.l.o.g let

$$\begin{aligned} &p_m^1 = b_{i_m}^\circ = b_i^\circ \text{ and } p_m^2 = b_{i_m+1}^\circ = b_{i+1}^\circ && m \in \{j, \ldots, k\}, \\ &p_m^1 = -b_{i_m+1}^\circ = -b_{i+1}^\circ \text{ and } p_m^2 = -b_{i_m}^\circ = -b_i^\circ && m \in \{k+1, \ldots, M\} \end{aligned}$$

for two consecutive extreme points b_i° and b_{i+1}° of B°.
Note, that the only possibility that the flow P^E defines this edge is that

$$p_m^E = p_m^1 = p_m^2 = b_{i_m}^\circ \qquad \forall m = 1, \ldots, j-1$$

and either

1. for $m = j, \ldots M$

$$p_m^E = p_m^1 + \lambda_m(p_m^2 - p_m^1) = \begin{cases} = b_i^\circ + \lambda_m(b_{i+1}^\circ - b_i^\circ) & \text{if } m \in \{j, \ldots, k\} \\ = -b_{i+1}^\circ + \lambda_m(-b_i^\circ + b_{i+1}^\circ) & \text{if } m \in \{k+1, \ldots, M\} \end{cases}$$

   with $\lambda_m \in [0,1]$ and there exists at least one $m \in \{j, \ldots, M\}$ such that $\lambda_m \in (0,1)$.

2. there exist flows with $p_{m_1}^E = p_{m_1}^1, p_{m_2}^E = p_{m_2}^2$ for $m_1 \neq m_2$ and $m_1, m_2 \in \{j, \ldots M\}$.

In both cases we have

$$0 \leq \lambda_m \leq 1 \text{ and } 0 < \sum_{m=j}^{M} \lambda_m < M - j + 1. \tag{11}$$

We get the following description of all possible divergence values:

$$
\begin{aligned}
\mathsf{div}(p^E) &= \sum_{m=1}^{j-1} w_m p_m^1 + \sum_{m=j}^{M} w_m \left[p_m^1 + \lambda_m (p_m^2 - p_m^1)\right] \\
&= \sum_{m=1}^{j-1} w_m b_{i_m}^{\circ} + \sum_{m=j}^{k} w_m \left[b_i^{\circ} + \lambda_m (b_{i+1}^{\circ} - b_i^{\circ})\right] + \sum_{m=k+1}^{M} w_m \left[-b_{i+1}^{\circ} + \lambda_m (-b_i^{\circ} + b_{i+1}^{\circ})\right] \\
&= \underbrace{\sum_{m=1}^{j-1} w_m b_{i_m}^{\circ} + \sum_{m=j}^{k} w_m b_i^{\circ} - \sum_{m=k+1}^{M} w_m b_{i+1}^{\circ}}_{\mathsf{div}(p^1)} + \sum_{m=j}^{M} \lambda_m w_m (b_{i+1}^{\circ} - b_i^{\circ}) \\
&= \mathsf{div}(p^1) + \left[\sum_{m=j}^{M} w_m \lambda_m\right] (b_{i+1}^{\circ} - b_i^{\circ}) \\
&= \mathsf{div}(p^1) + \bar{\lambda} \left[\sum_{m=j}^{M} w_m\right] (b_{i+1}^{\circ} - b_i^{\circ}) \\
&= \mathsf{div}(p^1) + \bar{\lambda} \sum_{m=j}^{M} w_m (b_{i+1}^{\circ} - b_i^{\circ}) \\
&\qquad \text{for suitable } \bar{\lambda} \in (0,1) \text{ since } \textstyle\sum \lambda_m w_m \in (0, \sum w_m) \text{ by (11)} \\
&= \mathsf{div}(p^1) + \bar{\lambda} \sum_{m=1}^{M} w_m (p_m^2 - p_m^1) \qquad \text{by definition of } p_m^i \\
&= \mathsf{div}(p^1) + \bar{\lambda} \left[\mathsf{div}(p^2) - \mathsf{div}(p^1)\right].
\end{aligned}
$$

□

Definition 3.5
For any intersection point $x \in \mathcal{I}(\mathcal{D}_t)$ define the set of *surrounding cells* of x as

$$\mathsf{SurCells}_t(x) := \{C \in \mathcal{G}_t \mid x \in C, \dim(C) = 2\}.$$

Since we consider the static case, we will omit t.

Theorem 3.6. *The divergence values of the cells around an intersection point $x \in \mathcal{I}(\mathcal{D})$ form a convex, symmetric polygon*

$$conv(\{\mathsf{div}(p) \mid p \in \mathcal{P}^C, C \in \mathsf{SurCells}(x)\})$$

with $\{\mathsf{div}(p) \mid p \in \mathcal{P}^C, C \in \mathsf{SurCells}(x)\}$ as extreme points.

Proof. As seen in the previous proof, going from one cell C_1 to an adjacent cell C_2 changes the divergence by $\left[\sum_{m=j}^{M} w_m\right] (b_{i+1}^{\circ} - b_i^{\circ})$ adapting the notation from the preceding proof. Moreover, if $b_r \in B$ is the ray defining the construction line separating C_1 and C_2, then b_r is orthogonal to $(b_{i+1}^{\circ} - b_i^{\circ})$ by definition of B° and the normal cones. Since the construction lines going through an intersection point define a symmetric "star", the statement immediately follows (cf. Figure 3.1). □

Theorem 3.7. *The divergence set of an intersection point $x \in \mathcal{I}(\mathcal{D})$ is the interior of the polygon formed by the unique divergences of the cells around it:*

$$\mathsf{DiVal}(x) = int(conv(\{\mathsf{div}(p) \mid p \in \mathcal{P}^C, C \in \mathsf{SurCells}(x)\})).$$

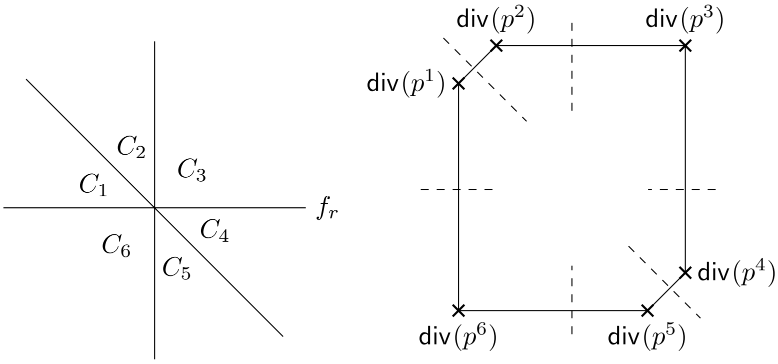


Figure 3.1: Divergence sets of edges orthogonal to construction lines.

Proof.
By the previous Theorem 3.6, we already know that the divergence of the surrounding cells of an intersection point x form a symmetric polygon. Note that no divergence of a flow $p^x \in \mathcal{P}^x$ can be outside $conv(\{\mathsf{div}(p) \mid p \in \mathcal{P}^C, C \in \mathsf{SurCells}(x)\})$, otherwise the cone conditions $x \in \bigcap_{m \in \mathcal{M}} d_m + K(p_m^x)$ cannot be satisfied.
For convenience, let $\mathsf{SurCells}(x) = \{C_1, \ldots, C_z\}$ for a $z \leq R$ and the corresponding flows $p^i = (p_m^i)_{m \in \mathcal{M}} \in \mathcal{P}^{C_i}$ for $i = 1, \ldots, z$.

Define

$$\begin{aligned} S_m &:= \arg\max_{b \in B^\circ} \langle x - d_m, b \rangle \\ &= \begin{cases} \{b_{i_m}^\circ\} & \text{if } |\arg\max\{\langle x - d_m, b\rangle \mid b \in B^\circ\}| = 1 \\ conv(b_{i_m}^\circ, b_{i_m+1}^\circ) & \text{otherwise} \end{cases} \\ &\quad \text{for a suitable } i_m \in \mathcal{R}. \end{aligned}$$

By (4) and the definition of the normal cone we know that

$$p_m \in S_m \forall m \in \mathcal{M} \implies x \in \bigcap_{m \in \mathcal{M}} d_m + K(p_m). \tag{12}$$

In addition, by defining $w_m S_m := \{w_m p_m \mid p_m \in S_m\}$, we get for the the Minkowski sum the following equations:

$$\begin{aligned} \sum_m conv(w_m S_m) &= conv(\sum_m w_m S_m) \\ &= conv(\{\mathsf{div}(p^1), \ldots, \mathsf{div}(p^z)\}) \end{aligned} \tag{13}$$

where the last equation holds because $\mathsf{div}(p^i)$ are the extreme points of the convex hull. By Theorem 3.2 and Theorem 3.4 we know that each p with $\mathsf{div}(p) \in bd(conv(\{\mathsf{div}(p^1), \ldots, \mathsf{div}(p^z)\}))$, the cone conditions yield either cells or edges, therefore, we have to subtract them. By equations (12) and (13) we get the desired result.

□

Remark 3.8
The sets $\mathsf{DiVal}(G)$ for $G \in \mathcal{G}$ with $dim(G) \geq 1$ form a polygon grid that corresponds to the dual grid derived in the dissertation of Wagner (2015), which was developed independently and published while this report was written. The former uses the dual of $1/\mathbb{R}^2/stat/\gamma/\Sigma$ and methods of subdifferential calculus to derive this grid. In contrast, this report considers whole geometric

shapes for which all possible divergences are derived. The relation between the subdifferential of $F_{stat}(x)$ and the theory in this report is given by

$$\partial F_{stat}(x) = \begin{cases} cl(\mathsf{DiVal}(x)), \text{if } x \in \mathcal{I}(\mathcal{D}), \\ cl(\mathsf{DiVal}(E)), \text{if } x \in ri(E) \text{ for an edge } E, \\ cl(\mathsf{DiVal}(C)), \text{if } x \in int(C) \text{ for a cell } C, \end{cases}$$

where $cl(\cdot)$ is the closure and ri denotes the relative interior of a set.

As consequence of the previous theorems the divergences of the geometric objects form a polygonal grid.

Theorem 3.9. *The set of all points in* $\{x \in \mathbb{R}^2 \mid \exists G \in \mathcal{G} : x \in \mathsf{DiVal}(G)\}$ *is* $\left(\sum_{m\in\mathcal{M}} w_m\right) B^\circ$.

Proof. From the previous results we know that the sets $\mathsf{DiVal}(G)$ for $G \in \mathcal{G}$ with $dim(G) \geq 1$ form a polygon grid. All the points within the grid cells are in $\mathsf{DiVal}(x)$ for some $x \in \mathcal{I}(\mathcal{D})$. Moreover, the boundary of this grid is obtained by the unbounded cells and edges in $\mathcal{G}$, thus, by showing that those objects form the boundary of $\left(\sum_{m\in\mathcal{M}} w_m\right) B^\circ$, we are done.
By the structure of the construction lines, there are $2R$ unbounded cells in $\mathcal{G}$ with exactly two extreme rays $b_r, b_{r+1} \in \mathsf{Ext}(B)$. W.l.o.g. these cells are given by

$$C_r = \bigcap_{m\in\mathcal{M}} d_m + K(b_r^\circ) \qquad\qquad r \in \mathcal{R}.$$

Thus, their divergence is given by $\mathsf{DiVal}(C_r) = \{\sum_{m\in\mathcal{M}} w_m b_r^\circ\}$, which are the extreme points of $\left(\sum_{m\in\mathcal{M}} w_m\right) B^\circ$. Again, by the same argument as in the proof of Theorem 3.6, the divergence of the other unbounded cells lie exactly between those of the extreme points, which finishes the proof. □

4 A geometric method for the 2-facility location problem

In this section a new algorithm based on the set of geometric objects is developed and its complexity is analyzed. In the end it is shown how to modify the algorithm to improve its running time.

4.1 A new subdivision based on the cone conditions

Going back to the 2-facility case, or equivalently the dynamic location problem with only two time-steps, we know from the optimality conditions that it is only necessary to consider the geometric objects with $G_t \in \mathcal{G}_t$ with $\mathsf{DiVal}_t(G_t) \cap v_1 B^\circ \neq \emptyset$, otherwise the conservation constraints cannot be satisfied. This motivates the next definition.

Definition 4.1
For any $v \in \mathbb{R}_{\geq 0}, t \in \mathcal{T}$, the *feasible objects with respect to weight* v are

$$\mathcal{FG}_t(v) := \{G \in \mathcal{G}_t \mid \mathsf{DiVal}_t(G) \cap vB^\circ \neq \emptyset\}.$$

Moreover, for $v = v_1$, where v_1 is the weight between the new locations in the two facility case, the optimal solutions lie in these polygons. Note, that $\mathcal{FG}_t(v)$ forms a star-like polygon, not necessarily convex.

Definition 4.2
The *boundary* objects of $\mathcal{FG}_t(v)$ are defined as

$$\mathsf{bdObj}(\mathcal{FG}_t(v)) := \{G \in \mathcal{FG}_t(v) \mid \mathsf{DiVal}_t(G) \cap bd(vB^\circ) \neq \emptyset\},$$

and the *interior* objects as

$$\mathsf{intObj}(\mathcal{FG}_t(v)) = \mathcal{FG}_t(v) \setminus \mathsf{bdObj}(\mathcal{FG}_t(v)).$$

Definition 4.3
Two geometric objects $G_1, G_2 \in \mathsf{bdObj}(\mathcal{FG}_t(v))$ are called $\mathsf{bdObj}(\mathcal{FG}_t(v))$-*adjacent* if

$$G_1 \cap G_2 \neq \emptyset, \quad \text{and}$$
$$\nexists G \in \mathsf{bdObj}(\mathcal{FG}_t(v)) \setminus \{G_1 \cup G_2\} : G \subset G_1 \cap G_2.$$

If v is very large, the boundary $\mathsf{bdObj}(\mathcal{FG}_t(v))$ may be empty.

In the following, we will only consider geometric objects inside $\mathcal{FG}_t$.

Theorem 4.4. *Let $v \in \mathbb{R}_{\geq 0}$ be given. Then the normal cones originating at the objects $G \in \mathcal{FG}_t(v)$ and defined by their divergence, i.e. the sets that are given by*

$$\begin{aligned}\mathsf{Cone}_t(G) &:= G + K(\mathsf{DiVal}_t(G)) \\ &:= \{x \in \mathbb{R}^2 \mid \exists g \in G, r \in K(\mathsf{DiVal}_t(G)) : x = g + r\},\end{aligned}$$

with

$$K(\mathsf{DiVal}_t(G)) := \bigcup_{q \in \mathsf{DiVal}_t(G)} K(1/v \cdot q),$$

give a subdivision of the plane for each $t \in \mathcal{T}$.

Proof. Note that for all objects $G \in \mathsf{intObj}(\mathcal{FG}_t(v))$ we have $\mathsf{DiVal}_t(G) \subset int(vB^\circ)$ and as consequence $\mathsf{Cone}_t(G) = G$. Therefore, we only have to consider the cases where $G \in \mathsf{bdObj}(\mathcal{FG}_t(v))$. As the divergence values move orthogonal to the position of the geometric objects, none of the cones on the boundary point to the inside of $\mathcal{FG}_t(v)$ and, therefore, can only intersect with $\mathsf{intObj}(\mathcal{FG}_t)$ on the actual boundary of the objects themselves. By similar arguments, we know that $\bigcup_{G \in \mathcal{FG}_t(v)} \mathsf{Cone}_t(G) = \mathbb{R}^2$.
It remains to show that $int(\mathsf{Cone}_t(G_1)) \cap int(\mathsf{Cone}_t(G_2)) = \emptyset$ for $G_1, G_2 \in \mathsf{bdObj}(\mathcal{FG}_t(v))$ with $G_1 \neq G_2$. It is enough to consider the $\mathsf{bdObj}(\mathcal{FG}_t(v))$-adjacent geometric objects.

$G_1 = C$ Cell / $G_2 = E$ Edge: There are two cases:

Case 1: The edge is contained in the cell:
By definition E and C can only be $\mathsf{bdObj}(\mathcal{FG}_t(v))$-adjacent if $\mathsf{DiVal}_t(C) = \{\mathsf{div}_t(p^C)\}$ and $\mathsf{DiVal}_t(E)$ are on a common facet of B°. Therefore, it is not possible for them to have common points in the interior of $\mathsf{Cone}_t(C)$ and $\mathsf{Cone}_t(E)$ as E must be orthogonal to that facet.

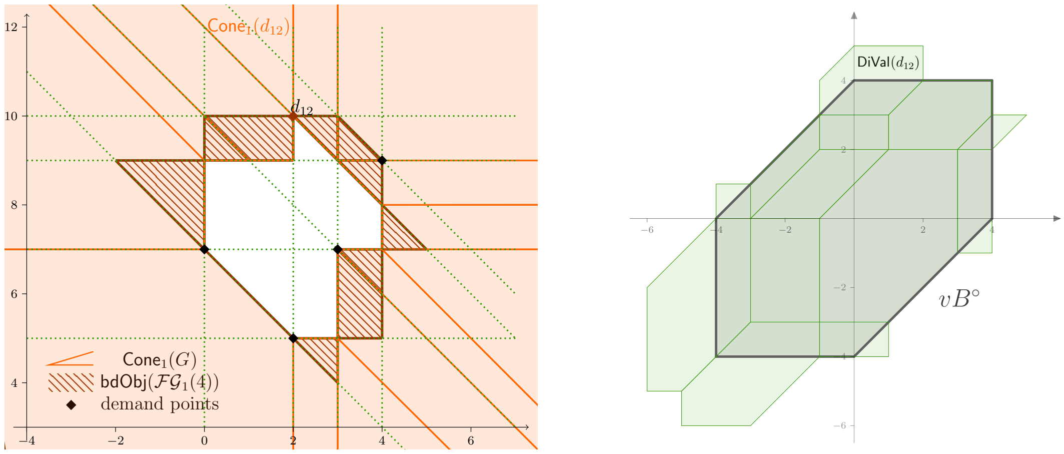
Case 2: In the second case, the edge is not part of the cell. However, by definition of our adjacency this case it is not possible at all, since otherwise there would be an intersection point $x \in \mathsf{bdObj}(\mathcal{FG}_t(v))$ with $x \in C$ and $x \in E$.

$G_1 = E$ Edge / $G_2 = x$ intersection point: Since both entities are in $\mathsf{bdObj}(\mathcal{FG}_t(v))$, x must be contained in the edge E by the same argumentation as above. Moreover, considering the closure of $\mathsf{DiVal}_t(x)$, then $\mathsf{DiVal}_t(E)$ must be one of its edges, precisely the edge orthogonal to E on the same side as E lies to x. As consequence, their cones $\mathsf{Cone}_t(E)$ and $\mathsf{Cone}_t(x)$ must at least intersect at their boundary. The fact that $\mathsf{DiVal}_t(E)$ defines a half-line $\{x \in \mathbb{R}^2 \mid \langle a, x\rangle + b = 0\}$, separating vB° into two parts, with $\mathsf{DiVal}_t(x)$ contained in one side, w.l.o.g. $\mathsf{DiVal}_t(x) \subset \{x \in \mathbb{R}^2 \mid \langle a, x\rangle + b > 0\}$ and $\mathsf{DiVal}(E) \subset \{x \in \mathbb{R}^2 \mid \langle a, x\rangle + b = 0\}\}$ finishes this case.

$G_1 = C$ Cell / $G_2 = x$ intersection point: As argued above the intersection point must be contained in the cell. There are two unique edges of C containing the intersection point x. The divergence of these edges are orthogonal to the edges themselves, therefore, $int(\mathsf{Cone}_t(C))$ cannot contain x, otherwise $\mathsf{DiVal}_t(x) \cap vB^\circ = \emptyset$. The same argumentation holds for showing $\mathsf{Cone}_t(x) \cap C = \emptyset$. Moreover, since $\mathsf{DiVal}_t(p^C) \notin \mathsf{DiVal}_t(x)$, but $\mathsf{DiVal}_t(p^C) \in cl(\mathsf{DiVal}_t(x))$, $\mathsf{Cone}_t(C)$ and $\mathsf{Cone}_t(x)$ have no common point in the interior.

G_1/G_2: Cell / Cell, Edge / Edge , intersection point / intersection point: By definition of our adjacency the two objects cannot be one of these combinations. □

Example 4.5
Consider the following set of demand points ($t = 1$) $d_{11} = (2,5), d_{12} = (2,10), d_{13} = (4,9), d_{14} = (0,7), d_{15} = (3,7)$, with weights $w_{1m} = 1, \quad \forall m \in \mathcal{M} \setminus \{5\}$ and $w_{15} = 2$. Figure 4.1 shows the subdivision obtained from the block norm defined by the polytope with extreme points $(1,0),(1,-1),(0,-1),(-1,0),(-1,1),(0,1)$. Its dual ball is given by the extreme points $(1,0)$, $(1,1)$, $(0,1)$, $(-1,0)$, $(-1,-1)$, $(0,-1)$. Figure 4.1 illustrates the cones and the divergence sets of the geometric objects in $\mathcal{FG}_1(4)$ with $v = 4$.



(a) Illustration of the subdivision defined by $\mathsf{Cone}_1(G)$ for $G \in \mathsf{bdObj}(\mathcal{FG}_1(4))$

(b) Illustration of $\mathsf{DiVal}_1(G)$ for $G \in \mathcal{FG}_1(4)$

Figure 4.1: Illustration of the cones generated by the boundary of $\mathcal{FG}_1(4)$

4.2 The Algorithm

When solving the static location problem $1/P/stat, w = (w_{tm})_{m\in\mathcal{M}}/\gamma/\Sigma$ for each time-step $t = 1,2$ separately, one gets solution objects G_t with flow p_t satisfying

$$G_t = \bigcap_{m\in\mathcal{M}} d_{tm} + K(p_{tm}),$$
$$p_{tm} \in B^\circ,$$
$$\sum_{m=1}^{M} p_{tm} = 0.$$

Therefore, the ball and cone conditions for the demand points (6) and (8) are satisfied and by setting $\tilde{p}_1 = 0$ the ball conditions (9) and the flow conservation constraints (5) as well. However, the only condition that may violate the optimal conditions for the dynamic problem case is the cone condition (7) which can be rewritten as

$$G_1 \cap (G_2 + K(\tilde{p}_1)) \neq \emptyset. \tag{7a}$$

In the following a process to improve the violated conditions (7a) while the others remain satisfied will be described. The following notation will be used:

Notation 4.6. *For the two facility case, the sets $\mathcal{FG}_t(v_1)$ describe the sets of polygons that have the possibility to be optimal. We will omit the v_1 and denote the sets just as $\mathcal{FG}_t$.*

First we introduce a distance function between two geometric objects to measure this improvement:

Definition 4.7
Between two geometric objects G_1 and G_2 (not necessarily bounded), the *distance* is given by

$$\mathsf{Dist}(G_1, G_2) := \min\{\|x_1 - x_2\|_2 \mid x_1 \in G_1, x_2 \in G_2\}.$$

Later, it is shown that each iteration step strictly decreases the distance $\mathsf{Dist}(G_1, G_2 + K(\tilde{p}_1))$ until it is 0 and consequently, the cone condition (7a) is satisfied.

Definition 4.8
For any $p \in \mathbb{R}^2$ the geometric object $G = \mathsf{GO}_t(p)$ containing p in its divergence set $\mathsf{DiVal}_t(G)$ is defined as

$$\mathsf{GO}_t(p) = \begin{cases} G_t, & \text{if } p \in \mathsf{DiVal}_t(G_t), \\ \emptyset, & \text{if no such } G_t \text{ exists.} \end{cases}$$

Note that GO_t is well defined, since $\mathsf{GO}_t(p)$ is unique.

Next G_1 and G_2 is updated. It is proved that the algorithm terminates after finitely many of these updates with the optimal solution set.

Update Step

Consider iteration k and two geometric objects $G_t^k \in \mathcal{G}_t$ $(t = 1, 2)$ with an associated flow $q^k := v_1\tilde{p}^k$ satisfying $q^k \in -\mathsf{DiVal}_1(G_1^k) \cap \mathsf{DiVal}_2(G_2^k)$ and $\mathsf{Dist}(G_1^k, G_2^k + K(\tilde{p}_1)) > 0$, like the initial solutions of $\mathcal{X}^*(1/P/stat, w = (w_{tm})_{m\in\mathcal{M}}/\gamma/\Sigma)$ do if they are not optimal.
Consider moving q^k in the direction of $\Delta p = \bar{x}_1 - \bar{x}_2$ with $(\bar{x}_1, \bar{x}_2) \in \{(x_1, x_2) \in G_1^k \times G_2^k + K(\tilde{p}_1^k) \mid \|x_1 - x_2\|_2 \ minimal\}$. Note that it is possible that $\bar{x}_2 \notin G_2^k$. Define

$$\begin{aligned} \bar{G}_1 &= \left\{ x \in G_1^k \;\middle|\; \langle x, -\Delta p\rangle = \max_{\tilde{x}\in G_1^k} \langle \tilde{x}, -\Delta p\rangle \right\} \\ \bar{G}_2 &= \left\{ x \in G_2^k \;\middle|\; \langle x, \ \ \Delta p\rangle = \max_{\tilde{x}\in G_2^k} \langle \tilde{x}, \ \ \Delta p\rangle \right\}. \end{aligned} \tag{14}$$

Note that $\bar{G}_t$ is either an edge or an intersection point with $\bar{x}_1 \in \bar{G}_1$ and $\bar{x}_2 \in \bar{G}_2 + K(\tilde{p}_1^k)$. By construction of the divergence grid it holds that $(-1)^t(q^k + \lambda\Delta p) \in \mathsf{DiVal}_t(\bar{G}_t)$ for λ sufficiently small. Choose

$$\lambda = \min\{\lambda^1, \lambda^2\}, \tag{15a}$$

with

$$\begin{aligned} \lambda^1 &:= \max\{\lambda \in \mathbb{R}_{>0} \mid -(q^k + \lambda\Delta p) \in cl(\mathsf{DiVal}_1(\bar{G}_1) \cap v_1 B^\circ)\}, \\ \lambda^2 &:= \max\{\lambda \in \mathbb{R}_{>0} \mid \quad q^k + \lambda\Delta p \ \ \in cl(\mathsf{DiVal}_2(\bar{G}_2) \cap v_1 B^\circ)\}, \end{aligned} \tag{15b}$$

update

$$q^{k+1} := q^k + \lambda\Delta p,$$

and

$$\begin{aligned} G_1^{k+1} &:= \mathsf{GO}_1(-q^{k+1}) \\ G_2^{k+1} &:= \mathsf{GO}_2(\ \ q^{k+1}). \end{aligned}$$

Note, that either G_1^{k+1} or G_2^{k+1} is an edge or a cell if $q^{k+1} \in int(v_1 B^\circ)$.
The resulting algorithm iterates the update step until G_1^k and $G_2^k + K(\tilde{p}^k)$ have a nonempty intersection, i.e $\mathsf{Dist}(G_1^k, G_2^k + K(\tilde{p}^k)) = 0$ (see Algorithm 4.1).

Algorithm 4.1: Solving $1/P/dyn, T = 2/\gamma/\Sigma$

input : $1/P/dyn, T = 2/\gamma/\Sigma$
output: All optimal solutions with corresponding flow P

$G_1^0 \leftarrow \mathcal{X}^*(1/P/stat/w = w_1/\Sigma)$;
$G_2^0 \leftarrow \mathcal{X}^*(1/P/stat/w = w_2/\Sigma)$;
$q^0 \leftarrow (0, 0)$;
$k \leftarrow 0$;

do
  $\tilde{p}_1^k \leftarrow {}^{q^k}/_{v_1}$;
  $(\bar{x}_1, \bar{x}_2) \in \{(x_1, x_2) \in G_1^k \times (G_2^k + K(\tilde{p}_1^k)) \mid \|x_1 - x_2\|_2 \; minimal\}$;
  $\Delta p \leftarrow \bar{x}_1 - \bar{x}_2$;
  if $\|\Delta p\| = 0$ **then**
    Calculate $\mathcal{X}^*$ by using the cone conditions;
    return $\mathcal{X}^*$
  end
  Calculate $\bar{G}_1, \bar{G}_2$ according to (14);
  Calculate λ_1, λ_2 according to (15b);
  $\lambda \leftarrow \min\{\lambda^1, \lambda^2\}$;
  $q^{k+1} \leftarrow q^k + \lambda \Delta p$;
  $G_1^{k+1} \leftarrow \mathsf{GO}_1(-q^{k+1})$;
  $G_2^{k+1} \leftarrow \mathsf{GO}_2(q^{k+1})$;
  $k \leftarrow k + 1$;
while $\|\Delta p\| > 0$;

Example 4.9
Consider the $1/P/dyn, T = 2/\gamma/\Sigma$ problem with the following data:

- $d_{11} = (2, 5), d_{12} = (2, 10), d_{13} = (4, 9), d_{14} = (0, 7), d_{15} = (3, 7)$,
- $d_{21} = (17, 5), d_{22} = (16, 13), d_{23} = (17, 11), d_{24} = (18, 12), d_{25} = (19, 11)$
- $w_{tm} = 1, \quad \forall (t, m) \in \mathcal{T} \times \mathcal{M} \setminus \{(1, 5)\}$ and $w_{15} = 2$,
- $v_1 = 4$.
- $B = conv((1, 0), (1, -1), (0, -1), (-1, 0), (-1, 1), (0, 1))$

The optimal solutions for the static problems are given by

$$G_1^0 = \tilde{x}_1^0 := (3, 7)$$
$$G_2^0 = C_2^0 := conv((17, 11), (17, 12), (18, 11)).$$

The following update steps of Algorithm 4.1 are visualized in Figures 4.2 and 4.3:

1. $\Delta p = \begin{pmatrix} 3 \\ 7 \end{pmatrix} - \begin{pmatrix} 17 \\ 11 \end{pmatrix} = \begin{pmatrix} -14 \\ -4 \end{pmatrix} \implies \mathsf{Dist}(G_1^0, G_2^0 + K(0)) > 0$
   - $\bar{G}_1 = \begin{pmatrix} 3 \\ 7 \end{pmatrix}, \bar{G}_2 = \begin{pmatrix} 17 \\ 11 \end{pmatrix}$
   - $\lambda = 0.2$
   - $q^1 = -\begin{pmatrix} 2.8 \\ 0.8 \end{pmatrix}$
   - $G_1^1 = \tilde{x}_1^0 = \mathsf{GO}_1(-q^1)$
   - $G_2^1 := E_2^1 := conv((17, 11), (16, 12)) = \mathsf{GO}_2(q^1)$
2. $\Delta p = \begin{pmatrix} 3 \\ 7 \end{pmatrix} - \begin{pmatrix} 16 \\ 12 \end{pmatrix} = \begin{pmatrix} -13 \\ -5 \end{pmatrix} \implies \mathsf{Dist}(G_1^1, G_2^1 + K(\tilde{p}^1)) > 0$

- $\bar{G}_1 = \begin{pmatrix} 3 \\ 7 \end{pmatrix}, \bar{G}_2 = \begin{pmatrix} 16 \\ 12 \end{pmatrix}$
- $\lambda = {}^1/_{70}$
- $q^2 = -\begin{pmatrix} 3 \\ {}^6/_7 \end{pmatrix}$
- $G_1^2 := E_1^2 := conv((3,7),(4,7)) = \mathsf{GO}_1(-q^2)$
- $G_2^2 := \tilde{x}_2^2 := \begin{pmatrix} 16 \\ 12 \end{pmatrix} = \mathsf{GO}_2(q^2)$

3\. - 5. For the iteration steps 3-5 compare with Figures 4.2 and 4.3

6. $\Delta p = \begin{pmatrix} 0 \\ -3 \end{pmatrix} \implies \mathsf{Dist}(G_1^5, G_2^5 + K(\tilde{p}^5)) > 0$
   - $\bar{G}_1 = \begin{pmatrix} 4 \\ 8 \end{pmatrix}, \bar{G}_2 := conv((11,11),(16,11))$
   - $\lambda = {}^1/_3$
   - $q^6 = -\begin{pmatrix} 4 \\ 3 \end{pmatrix}$
   - $G_1^6 = \mathsf{GO}_1(-q^6) = C_1^6 := conv((3,9),(4,9),(4,8))$
   - $G_2^6 = \mathsf{GO}_2(q^6) = C_2^6 := conv((12,11),(16,11),(16,6))$

7. $\Delta p = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \implies \mathsf{Dist}(G_1^6, G_2^6 + K(\tilde{p}^6)) = 0.$

   The optimal solution is given by the objects in $\mathcal{X}^* := \{(x_1, y_1, x_2, y_2) \in \mathcal{X}_1^* \times \mathcal{X}_2^* \mid y_1 = y_2\}$ with $\mathcal{X}_1^* := C_1^6$ and $\mathcal{X}_2^* := conv((13,9)(16,9),(16,8),(14,8))$. The corresponding flow is given by $\tilde{p}_1 = {}^{q^6}/_{v_1} = \begin{pmatrix} -1 \\ -0.75 \end{pmatrix}$ and

$$p_{11} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad p_{12} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad p_{13} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \quad p_{14} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad p_{15} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$
$$p_{21} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad p_{22} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \quad p_{23} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \quad p_{24} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \quad p_{24} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}.$$

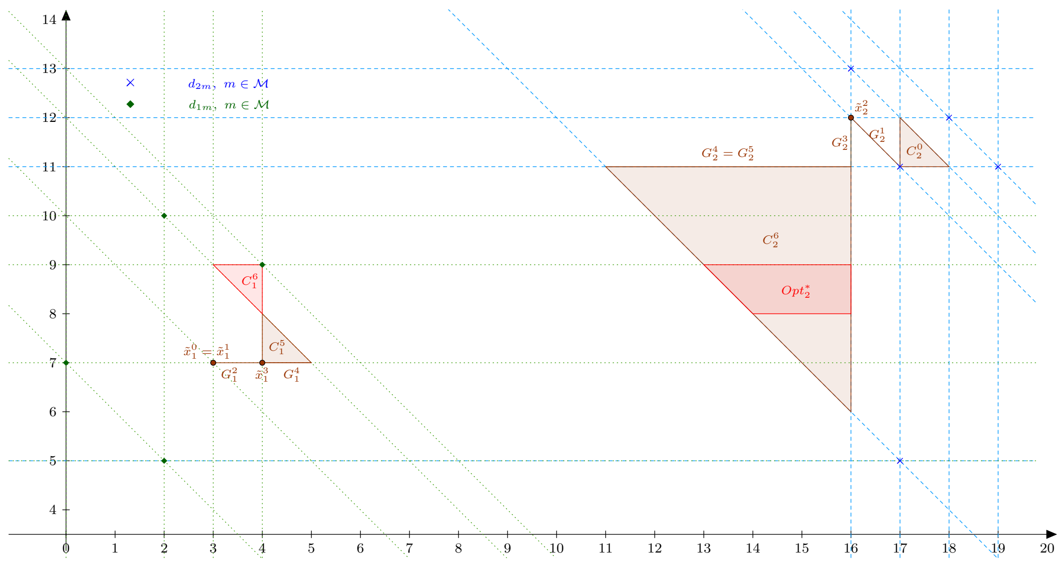
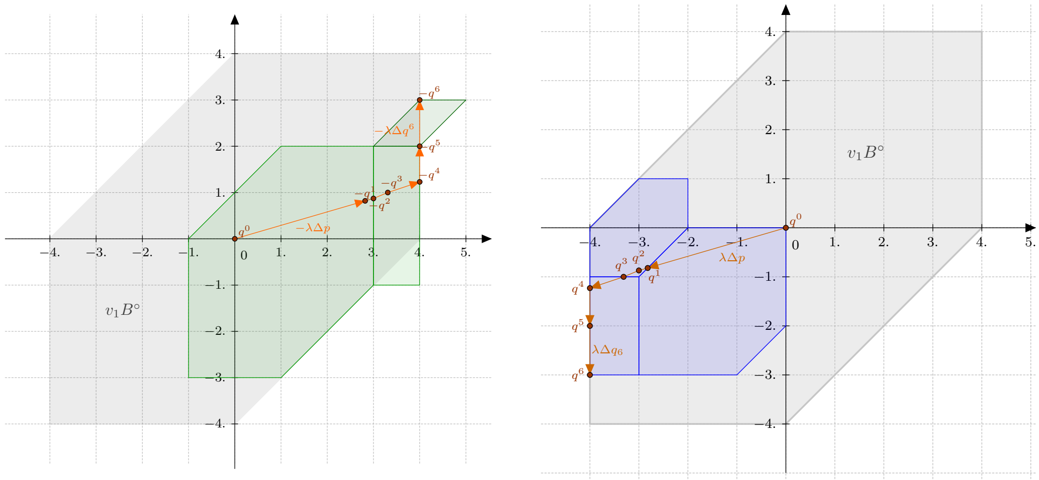


Figure 4.2: Visualization of Example 4.9



(a) $\mathsf{DiVal}_1(G_1^k)$ (b) $\mathsf{DiVal}_2(G_2^k)$

Figure 4.3: Divergence sets of the geometric objects considered in Example 4.9

Correctness

By definition of q^{k+1} and GO_t, we have $q^{k+1} \in v_1 B^\circ$ and the condition

$$q^{k+1} \in -\mathsf{DiVal}_1(G_1^{k+1}) \cap \mathsf{DiVal}_2(G_2^{k+1}) \tag{16}$$

still holds after the update step.

By (16) and the definition of $\mathsf{DiVal}_t(G_t^{k+1})$ there exists flows $p_t^{k+1} = (p_{tm}^{k+1})_{m\in\mathcal{M}} \in \mathcal{P}_t^{G_t^{k+1}}$ and $\tilde{p}^{k+1} := {}^{q^{k+1}}/_{v_1}$ fulfilling the conservation constraints:

$$\sum_{m\in\mathcal{M}} w_{1m} p_{1m}^{k+1} - v_1 \tilde{p}^{k+1} = 0$$
$$\sum_{m\in\mathcal{M}} w_{1m} p_{2m}^{k+1} + v_1 \tilde{p}^{k+1} = 0.$$

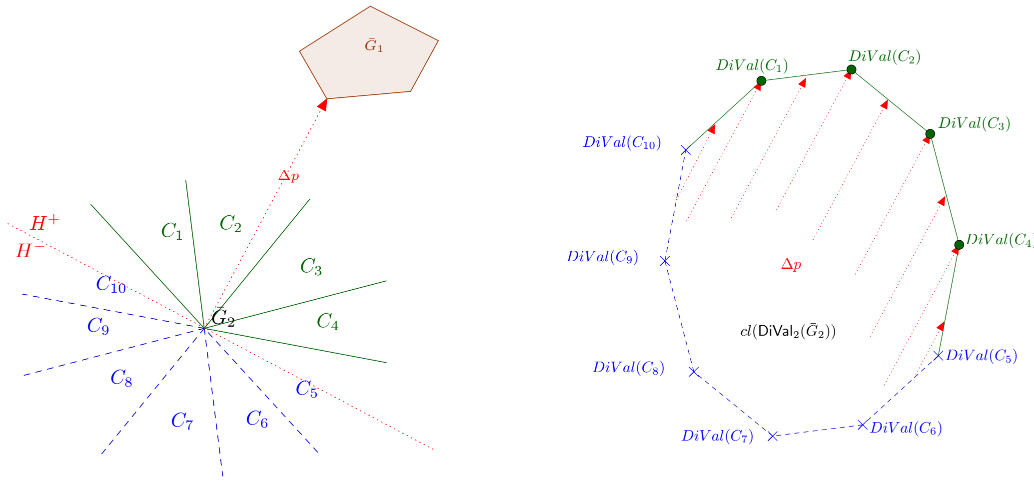
Moreover, as $p_t^{k+1} \in \mathcal{P}_t^{G_t^{k+1}}$ the cone and ball conditions for the demand points (6) and (8) are still satisfied

$$G_t^{k+1} = \bigcap_{m\in\mathcal{M}} d_{tm} + K(p_{tm}^{k+1}), \qquad t = 1,2$$
$$p_{tm}^{k+1} \in B^\circ, \qquad t = 1,2; m \in \mathcal{M}.$$

It only remains to show, that the algorithm will find an optimal solution. First of all, we need the following Lemma.

Lemma 4.10. *In $2/P/stat/\gamma/\Sigma$, there always exists a flow $P = (p, \tilde{p})$ defining an optimal solution for which $v_1\tilde{p}_1 \in bd(-\mathsf{DiVal}_1(G_1) \cap v_1 B^\circ)$ or $v_1\tilde{p}_1 \in bd(\mathsf{DiVal}_2(G_2) \cap v_1 B^\circ)$ for $G_t = \bigcap_{m\in\mathcal{M}} d_{tm} + K(p_{tm})$.*

Proof. We only have to prove the case when $v_1\tilde{p}_1 \in int(v_1 B^\circ)$: Assume we have an optimal solution set $\mathcal{X}^*$ for which neither $v_1\tilde{p}_1 \in bd(-\mathsf{DiVal}_1(G_1) \cap v_1 B^\circ)$ nor $v_1\tilde{p}_1 \in bd(\mathsf{DiVal}_2(G_2) \cap v_1 B^\circ)$ holds, which means G_1, G_2 are intersection points. Since $v_1\tilde{p}_1$ lies in the interior of $v_1 B^\circ$, there is only one solution $X^* \in \mathcal{X}^*$ for which $x_1 = x_2 = x^* \in \mathbb{R}^2$ holds. Considering any $q \in bd(-\mathsf{DiVal}_1(x^*) \cap \mathsf{DiVal}_2(x^*))$, then $G_t := \mathsf{GO}_t(q)$ defines a cell or edge for $t = 1$ or $t = 2$ containing x^*. By definition there exist $p_t \in \mathcal{P}_t^{G_t}$ such that the cone, ball and conservation constraints for $\tilde{p} = {}^1/_{v_1} \cdot q$ are still satisfied. Note that by Theorem 2.4, it is not possible that q defines more optimal solutions than X^*. $\square$



(a) Original space (b) $\mathsf{DiVal}_2(\bar{G}_2)$

Figure 4.4: Possible G_2^{k+1} for a flow change Δp, where $\bar{G}_2 \in \mathcal{I}(\mathcal{D}_2)$.

Lemma 4.11. *The cone conditions strictly improve in each iteration step, i.e.:*

$$\mathsf{Dist}(G_1^{k+1}, G_2^{k+1} + K(\tilde{p}_1^{k+1})) < \mathsf{Dist}(G_1^k, G_2^k + K(\tilde{p}_1^k)).$$

Proof. Consider the two endpoints $(\bar{x}_1, \bar{x}_2) \in \{(x_1, x_2) \in G_1^k \times (G_2^k + K(\tilde{p}_1^k)) \mid \|x_1 - x_2\|_2 \text{ minimal}\}$ and $\Delta p := \bar{x}_1 - \bar{x}_2$. Without loss of generality we consider the case where $\lambda = \lambda^2$ in equation (15a). Let H be the hyperplane orthogonal to Δp going through $\bar{x}_2$, i.e $H = \{x \in \mathbb{R}^2 \mid \langle x, \Delta p\rangle = \langle \bar{x}_2, \Delta p\rangle\}$. Let $H^+ = \{x \in \mathbb{R}^2 \mid \langle x, \Delta p\rangle > \langle \bar{x}_2, \Delta p\rangle\}$ the side containing G_1^k and H^- the other one like shown in Figure 4.4.
It follows that $G_2^k \subset H^- \cup H$ and by definition, $\bar{G}_2 \subset G_2^k$ is either an intersection point or an edge orthogonal to Δp, therefore $\bar{G}_2 \subset H$.
However, as all edges of $cl(\mathsf{DiVal}_2(\bar{G}_2))$ are orthogonal to the edges $E_2 \in \mathcal{G}_2$ with $\bar{G}_2 \subset E_2$, Δp can only intersect one of the divergence values corresponding to the objects in H^+, therefore, $G_2^{k+1} + K(\tilde{p}_1^{k+1}) \subset H^+$ (cf. Figure 4.4). In the special case where $\bar{G}_2$ is an edge E, Δp is orthogonal to that edge and points directly to the cell in H^+ containing E.
In both cases we have $\mathsf{Dist}(G_1^{k+1}, G_2^{k+1} + K(\tilde{p}_1^{k+1})) < \mathsf{Dist}(G_1^k, G_2^k + K(\tilde{p}_1^k))$.

□

Complexity

As we are considering geometric objects, we need to make use of methods in Computational Geometry (CG) to calculate the worst case complexity of Algorithm 4.1. Not only preprocessing can affect the running time drastically, but also the way one stores the objects. There might be the case where preprocessing has a higher theoretically complexity than the algorithm itself, however, as it is only done once, the practical running time might become a lot faster. About storing the objects: one can decrease the theoretical running time of an algorithm by using a lot of storage space, however in praxis, it might not be advisable to overload the memory of a computer, consequently, one needs to find a trade-off between preprocessing, space and actual running time. The following paragraphs will give an idea how to calculate the necessary objects in Algorithm 4.1. In the following we will assume that all vertices of a polygon are sorted in clockwise order. Moreover, we assume that the extreme points of B and B° are given, such that

$$K(b_i^\circ) = \{x \in \mathbb{R}^2 \mid x = \lambda_1 b_i + \lambda_2 b_{i+1}, \lambda_1, \lambda_2 \geq 0\}, \tag{17}$$

starting from the one with minimal angle to the x-axis. This is necessary to find an $r \in \mathcal{R}$ such that for a given $x \in \mathbb{R}^2$ we can find the cone with $x \in K(b_r^\circ) = \{x \in \mathbb{R}^2 \mid x = \lambda_1 b_r + \lambda_2 b_{r+1}, \lambda_1, \lambda_2 \geq 0\}$ in $\mathcal{O}(\log R)$.

Calculating $\bar{\mathbf{G}}_1, \bar{\mathbf{G}}_2$ and $\boldsymbol{\Delta}\mathbf{p}$: We distinguish the following two cases

G_1 and $G_2 + K(q^k/v_1)$ are bounded: If the vertices of two convex polygons with m and n vertices are sorted in clockwise order, then the minimum distance can be found in $\mathcal{O}(m+n)$, using the methods of Toussaint (1984) and McKenna and Toussaint (1985). Toussaint and McKenna give an $\mathcal{O}(m+n)$ algorithm to find the closest pair of vertices between those two polygons. Once we know which vertices are closest, checking whether the minimal distance might be between an edge and a vertex can be done in constant time. As the number of vertices of G_t is less than $2M$ for any $G_t \in \mathcal{G}_t$, we have a linear running time $\mathcal{O}(M)$.

G_1 and/or $G_2 + K(q^k/v_1)$ is unbounded: Note, that even if $q^k \in int(v_1 B^\circ)$ the two polygons can be unbounded whenever the cell or edge is unbounded itself. Therefore, G_1^k and $G_2^k + K(q^k/v_1)$ can have at most two rays $r_{1,2}^t$. One gets the same running time of $\mathcal{O}(M)$ by consecutively looking at the rays and iterating through the vertices of both objects successively. Hence, the unbounded case does not worsen the running time.

In both cases, Δp and $\bar{G}_t$ can automatically be derived, which gives a worst case of $\mathcal{O}(M)$.

Calculating $cl(\mathsf{DiVal}_1(\bar{G}_1)), cl(\mathsf{DiVal}_2(\bar{G}_2))$: As there is more than one method to calculate $\mathsf{DiVal}_t(\bar{G}_t)$, there are different worst case complexities. Note that the worst case is achieved whenever $\bar{G}_t \in \mathcal{I}(\mathcal{D}_t)$. The pseudo-code in Procedure 4.2 will describe how to calculate $\mathsf{DiVal}_t(\bar{G}_t)$ in this case.

Procedure 4.2: Calculate $\mathsf{DiVal}_t(x)$ for $x \in \mathcal{I}(\mathcal{D}_t)$

```
extreme_points ← ∅;
previous ← None;
for r ∈ R do
    DiVal(C_r) ← 0;
    for m ∈ M do
        Choose ε = ε(m, x) sufficiently small;
        x̃ ← x + ε (b_{r+1} + b_r)/‖b_{r+1} + b_r‖;
        b° ← arg max_{b∈B°} ⟨x̃ − d_m, b⟩;
        DiVal_t(C_r) ← DiVal_t(C_r) + w_m b°;
    end
    if DiVal_t(C_r) ≠ previous then
        extreme_points ← extreme_points ∪ {DiVal(C_r)};
        previous ← DiVal_t(C_r);
    end
end
return extreme_points
```

In this procedure ϵ is chosen sufficiently small, guaranteeing that $\tilde{x} \in int(C_r)$ for $C_r \in \mathsf{SurCells}_t(x)$. By the previous arguments one can find the two fundamental directions closest to x, however, not intersecting x in $\mathcal{O}(\log R)$. Calculating the distance between x and the fundamental directions needs $\mathcal{O}(1)$ and choosing any ϵ smaller than this distance will guarantee that $\tilde{x}$ lies in $int(C_r)$. Altogether, we have $\mathcal{O}(MR \log R)$ in the worst case. Note that calculating $\mathsf{DiVal}_t(\bar{G}_t)$ for $\bar{G}_t$ being an edge or a cell can be done similarly to Procedure 4.2 with the difference that we do not have to iterate over every $r \in \mathcal{R}$. Moreover, the extreme points are already sorted in clockwise order.

Calculating $\boldsymbol{\lambda^1, \lambda^2}$: We need to intersect a ray with a polygon in (15b). The running time achieves its worst case whenever $\bar{G}_t \in \mathcal{I}(\mathcal{D}_t)$. Assume we have a polygon of R vertices $v_1, \dots, v_R$ given in clockwise order and a ray $\lambda \Delta p$ with starting point q^k. We already know that the ray has its starting point in the polygon and, therefore, intersects it. It is possible to find the intersection in $\mathcal{O}(\log R)$:
Starting with $v_m = v_1$ and v_n ($n = \lfloor R/2 \rfloor$) we check on which side of the ray the two vertices are. This can be done in constant time. Only in the first step, we have to check on which side q^k is of the line from v_1 through v_n, otherwise we might go in the wrong direction.
W.l.o.g. assume $q^k + \lambda \Delta p$ is intersecting the polygon in one of the edges from $a = v_m$ to $b = v_n$. Set $l := \lfloor \frac{m+n}{2} \rfloor$. If $q^k + \lambda \Delta p$ is intersecting the line between v_m and v_l, set $m = l$, else set $n = l$ and repeat this process until $n = m + 1$ or $q^k + \lambda \Delta p$ is intersecting v_l.
As calculating $\mathsf{DiVal}_t(\bar{G}_t) \cap v_1 B^\circ$ is more time consuming than intersecting the two polygons separately with $q^k + \lambda \Delta p$ and taking the one with minimum length, we have to do this procedure for all three polygons $\mathsf{DiVal}_1(\bar{G}_1), \mathsf{DiVal}_2(\bar{G}_2)$ and $v_1 B^\circ$. Also note that in an implementation one would be using the indices of $\mathsf{SurCells}_t(x)$ to get the vertices $v_1, \dots, v_R$ and also return a tuple

- (r, r): $q^k + \lambda \Delta p$ intersects the divergence of the cell contained in $\bar{G}_t + K(b_r^\circ)$.
- $(r, r+1)$: $q^k + \lambda \Delta p$ intersects divergence of the edge contained in $\bar{G}_t + K(\frac{b_r^\circ + b_{r+1}^\circ}{2})$.

Such indices can be calculated in $\mathcal{O}(\log R)$ once we know where $q^k + \lambda \Delta p$ is intersecting $\mathsf{DiVal}_t(\bar{G}_t)$.

Calculating $\mathbf{G_t^{k+1}}$: Note that the running time of calculating G_t^{k+1} is dominated by the case where G_t^{k+1} changes, i.e. $q^{k+1} \in bd(\mathsf{DiVal}_t(\bar{G}_t))$, otherwise $G_t^{k+1} = \bar{G}_t$. Let (r_1, r_2) be the indices returned from calculating λ^1 and λ^2. Assume $\bar{G}_t \in \mathcal{I}(\mathcal{D}_t)$ (the case $\bar{G}_t$ being an edge goes similar). By (17) we know that G_t^{k+1} lies in the cone starting at $\bar{G}_t$ and spanned by b_{r_1} and b_{r_2}. It can be calculated like shown in Procedure 4.3. The correctness follows by the same argumentation as used in calculating $\mathsf{DiVal}_t(\bar{G}_t)$.
As argued in the previous paragraphs calculating ϵ as well as calculating the set A can be done in $\mathcal{O}(\log R)$. As each normal cone can be rewritten as an intersection of two half-planes, each cell or edge is an intersection of at most $4M$ half-planes. By Preparata and Shamos (1985) this intersection can be calculated in $\mathcal{O}(M \log M)$. Therefore, Procedure 4.3 takes $\mathcal{O}(M(\log M + \log R))$ time.

Procedure 4.3: Calculate G_t^{k+1} if $\bar{G}_t \in \mathcal{I}(\mathcal{D}_t)$

```
r_1, r_2 ← indices_from_calculating(λ^i);
for m ∈ M do
    Choose ϵ sufficiently small;
    x̃ ← Ḡ_t + ϵ (b_{r_1} + b_{r_2}) / ‖b_{r_1} + b_{r_2}‖;
    A ← arg max_{b∈B°} ⟨x̃ − d_m, b⟩ ;                         // |A| ≤ 2
    H_m ← calculate_halfplanes(d_m, b_i°) for b_i° ∈ A ;        // |H_m| ≤ 4
end
C_r ← intersect(H_1, ..., H_M);
C_r ← sort_vertices;
return C_r;
```

Calculate optimal solution set: Each G_t^{k+1} has at most $2M$ extreme points and $K(\tilde{p}^{k+1})$ has at most two rays. The optimal solution set $\mathcal{X}^*$ can be calculated by intersecting G_2^{k+1} with each ray starting from the extreme points of G_1^{k+1}. Then we get a partition of G_2^{k+1}, for which the corresponding solutions in G_1^{k+1} can be described. Obtaining this partition and the description of $\mathcal{X}^*$ can be calculated in $\mathcal{O}(M^3)$.

Number of Iterations: Note that there are $\mathcal{O}(M^2R^2)$ geometric objects for each time-step t. Although, this algorithm is only considering objects with common divergence value, it might happen that the algorithm is "zigzagging". Therefore, in the worst case, the algorithm has $\mathcal{O}(M^4R^4)$ iterations. The previous paragraph can be summarized in the following theorem:

Theorem 4.12. *Algorithm 4.1 with Procedures 4.2 and 4.3 runs in* $\mathcal{O}(M^5R^4(\log M + R\log R))$ *time.*

Remark 4.13
It is possible to improve this bound by doing preprocessing and storing all the necessary objects, such that they do not need to be calculated on the spot. However, more storage space is needed. Assuming that one stores all the cells with extreme points, rays and their divergence values, the intersection points $x \in \mathcal{I}(\mathcal{D}_t)$ with reference to the surrounding cells $C \in \mathsf{SurCells}_t(x)$, Algorithm 4.1 can be implemented in $\mathcal{O}(M^4R^4(M + \log R))$ time using $\mathcal{O}(M^3R^2(\log M + R\log R)$ preprocessing and $\mathcal{O}(M^2R^2(M + R))$ space.

4.3 A Hybrid Algorithm for $2/\mathbb{R}^2/\text{stat}/\gamma/\Sigma$

In this section, we will deal with the zigzagging problem from the previous algorithm. Getting rid of it would mean needing $\mathcal{O}(M^2R^2)$ iterations. The next algorithm is called hybrid, because it is a mixture of Algorithm 4.1 and a method to solve the single facility location problem. First it is checked whether the two facilities coincide checking the optimality conditions for the geometric objects in the boundary $\mathsf{bdObj}(\mathcal{FG}_t)$ for $t = 1, 2$ and then it solves the singles facility case.
If $\sum_{m\in\mathcal{M}} w_{tm} < v_1$ for at least one $t \in \{1, 2\}$, then for any $X \in \mathcal{X}^*$ the two facilities coincide by Theorem 3.9, as the divergences of the objects lie strictly in the interior of v_1B°. This result also confirms the ones by Lefebvre et al. (1990) and Plastria (1992). Suppose $\sum w_m \geq v_1$ for both $t = 1, 2$. Then there is the possibility that $x_1^* \neq x_2^*$.
Let us first assume the optimal solution is in $\mathsf{bdObj}(\mathcal{FG}_t)$. By the proof of Lemma 4.10 we know that one of the optimal geometric objects G_t is either a cell or an edge, or, if both of them are intersection points, $\tilde{p}_1$ has to be an extreme point of B°.
Using the same update steps like in Algorithm 4.1, we can start with two elements G_1^0 and G_2^0 that satisfy $q^k = v_1b_1^\circ \in (-1)^t\mathsf{DiVal}_t(G_t^0)$. By the methods described in the previous sections such an element can be calculated in $\mathcal{O}(M^3R^2(\log M + R\log R))$ time. We set $r = 1$ and in each iteration $k = 1, 2, \dots$ we set $\Delta p = b_{r+1} - b_r$ until $q^k = v_1b_{r+1}$. Then we update r to $r + 1$ and continue until we reach $r = R$, like shown in Algorithm 4.4.
However, in this case the procedure can terminate without finding any solution, since we only considered the boundary $\mathsf{bdObj}(\mathcal{FG}_t)$.
If no solution is found, we know that $x_1 = x_2$ as $\tilde{p}_1 \in int(B^\circ)$. Hence, we can solve a static problem with $2M$ facilities $1/\mathbb{R}^2/stat, w = (w_{tm})_{m\in\mathcal{M},t\in\mathcal{T}}/\gamma/\Sigma$.
Considering the number of iterations, the set $\mathsf{DiVal}_t(G_t)$ of a geometric object $G_t \in \mathcal{G}_t$ can intersect $bd(v_1B^\circ)$ on more than one position and, therefore, can be considered multiple times. The next theorem shows that the number of times one element is considered is bounded.

Theorem 4.14. *Algorithm 4.4 terminates after $\mathcal{O}(M^2R^2)$ iterations.*

Proof. Cells and edges can only be considered once, or respectively twice. Therefore, we assume that $G_t^k \in \mathcal{I}(\mathcal{D}_t)$. All edges of $\mathsf{DiVal}_t(G_t^k)$ must be parallel to some facet of B°. Therefore, using the update method in this section a geometric object can only be considered at most $R/2$ times. To see this enumerate the edges of v_1B° in clockwise order $e_1, \dots, e_R$, where $e_r := b_{r+1} - b_r$ $(r \in \mathcal{R})$. Let $\mathsf{DiVal}_t(G_t^k) \cap e_r \neq \emptyset$ for some $r \in \mathcal{R}$ and $G_t^k \neq G_t^{k+1}$. Thus, $\mathsf{DiVal}_t(G_t^{k+1}) \cap ri(e_r) \neq \emptyset$. Assume, G_t^k is considered in another iteration, i.e. $G_t^{k'} = G_t^k$ for some $k' > k$. Then G_t^k intersects another edge $e_{r'}$ with $r' > r$, however, as all edges of $\mathsf{DiVal}_t(G_t^k)$ are parallel to one in $\{e_1, \dots, e_r\}$, it follows that $r' \geq r + 2$.
Suppose an object $\mathsf{DiVal}_t(G_t)$ intersects v_1B° at j occasions (for $3 \leq j \leq R/2$). Then by the previous argumentation, there can only be at most $\mathcal{O}(R/j)$ objects with that property. Therefore, summing up over the number of times one object can be considered, one gets a total of $\mathcal{O}(1 \cdot M^2R^2 + 2 \cdot M^2R^2 + \sum_{j=3}^{R/2} j \cdot R/j) = \mathcal{O}(M^2R^2)$ iterations, taking all reconsiderations into account. □

The following theorem can be derived using the same complexity results as before.

Theorem 4.15. *Algorithm 4.4 runs in $\mathcal{O}(M^3R^2(\log M + R\log R))$.*

Note that it is also possible to adjust the search direction along the boundary by improving the distance between the geometric objects like done in the previous subsection. However, this does not change the worst case running time.

Remark 4.16
By the same argumentation as before, Algorithm 4.4 can be implemented in $\mathcal{O}(M^2R^2(M + \log R))$ using $\mathcal{O}(M^3R^3(\log M + \log R))$ preprocessing and $\mathcal{O}(M^2R^2(M + R))$ space.

Algorithm 4.4: Solving $1/P/dyn, T = 2/\gamma/\Sigma$

input : $1/P/dyn, T = 2/\gamma/\Sigma$
output: All optimal solutions with corresponding flow P

if $\sum_{m\in\mathcal{M}} w_{tm} < v_1$ *for some* $t \in \{1,2\}$ **then**
  return solve$(1/\mathbb{R}^2/static, w = (w_{tm})_{m\in\mathcal{M},t\in\mathcal{T}}/\gamma/\Sigma)$;
end

/* *Initialization* */;
$q^0 \leftarrow v_1 b_1^\circ$;
$G_1^0 \leftarrow \mathsf{GO}_1(-q^0)$;
$G_2^0 \leftarrow \mathsf{GO}_2(q^0)$;
$r \leftarrow 1$;
$k \leftarrow 0$;

while $\mathsf{Dist}(G_1^k, G_2^k + K(q^k/v_1)) > 0$ **do**
  if $q^k = v_1 b_{r+1}^\circ$ **then**
    $r \leftarrow r + 1$;
    if $r = R + 1$ **then**
      return $\mathcal{X}^*(1/\mathbb{R}^2/stat, w = (w_{tm})_{m\in\mathcal{M},t\in\mathcal{T}}/\gamma/\Sigma)$;
    else
      $\Delta p \leftarrow b_{r+1}^\circ - b_r^\circ$; // $b_{R+1}^\circ = b_1^\circ$
    end
  end
  /* *Calculate step-size* */;
  Calculate $\bar{G}_1, \bar{G}_2$ according to (14);
  Calculate λ_1, λ_2 according to (15b);
  $\lambda \leftarrow \min\{\lambda^1, \lambda^2\}$;
  /* *Update step* */;
  $q^{k+1} \leftarrow q^k + \lambda \Delta p$;
  $G_1^{k+1} \leftarrow \mathsf{GO}_1(-q^{k+1})$;
  $G_2^{k+1} \leftarrow \mathsf{GO}_2(q^{k+1})$;
  $k \leftarrow k + 1$;
end
Calculate optimal solution set $\mathcal{X}^*$ with G_1^k, G_2^k and q^k;
return $\mathcal{X}^*$;

5 Conclusion and Future Research

This report reconsiders the $1/\mathbb{R}^2/dyn/\gamma/\Sigma$ Problem, focusing on the case with only two time-steps. The latter is equivalent to the static 2-facility median problem in the plane $2/\mathbb{R}^2/stat/\gamma/\Sigma$. A geometric description of the divergence values for each geometric object induced by the cone conditions (6) is derived. This representation is used to state two new solution approaches for the 2-facility location problem that are also analyzed in their complexity. As the two algorithms are using methods of computational geometry the trade-off between preprocessing, space and running time is pointed out. It is possible to decrease the actual running time at the cost of increasing preprocessing and storage space. However, it might not be advisable to store all the geometric objects. How this will effect the running time in praxis has not been tested yet.
Moreover, one can also use *polyhedral gauges* instead of block norms. Gauges can be defined like block norms, however, they are lacking the symmetry property. Most of the results in this report can be adapted to polyhedral gauges. In addition, there are further objectives and extensions stated in Maier and Hamacher (2015) which have not been considered yet.

References

J. Fliege. Solving convex location problems with gauges in polynomial time. Technical report, Fakultät für Mathematik, TU Dortmund, 1998. Ergebnisberichte des Instituts für Angewandte Mathematik, Nummer 158.

H. W. Hamacher and S. Nickel. Classification of location models. *Location Science*, 6(1–4):229 – 242, 1998. ISSN 0966-8349. doi: http://dx.doi.org/10.1016/S0966-8349(98)00053-9. URL http://www.sciencedirect.com/science/article/pii/S0966834998000539.

H. Idrissi, O. Lefebvre, and C. Michelot. Duality for constrained multifacility location problems with mixed norms and applications. *Annals of Operations Research*, 18(1):71–92, 1989. ISSN 0254-5330. doi: 10.1007/BF02097796. URL http://dx.doi.org/10.1007/BF02097796.

O. Lefebvre, C. Michelot, and F. Plastria. Geometric interpretation of the optimality conditions in multifacility location and applications. *Journal of Optimization Theory and Applications*, 65 (1):85–101, 1990.

A. Maier and H. W. Hamacher. A finite dominating set algorithm for a dynamic location problem in the plane. Technical Report 155, Fachbereich Mathematik, 2015.

M. McKenna and G. T. Toussaint. Finding the minimum vertex distance between two disjoint convex polygons in linear time. *Computers & Mathematics with Applications*, 11(12):1227 – 1242, 1985. ISSN 0898-1221. doi: http://dx.doi.org/10.1016/0898-1221(85)90109-9. URL http://www.sciencedirect.com/science/article/pii/0898122185901099.

F. Plastria. When facilities coincide: Exact optimality conditions in multifacility location. *Journal of Mathematical Analysis and Applications*, 169(2):476 – 498, 1992. ISSN 0022-247X. doi: http://dx.doi.org/10.1016/0022-247X(92)90091-Q. URL http://www.sciencedirect.com/science/article/pii/0022247X9290091Q.

F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag New York, Inc., New York, NY, USA, 1985. ISBN 0-387-96131-3.

R.T. Rockafellar. *Convex Analysis*. Convex Analysis. Princeton University Press, 1997. ISBN 9780691015866.

G.T. Toussaint. An optimal algorithm for computing the minimum vertex distance between two crossing convex polygons. *Computing*, 32(4):357–364, 1984. ISSN 0010-485X. doi: 10.1007/BF02243778. URL http://dx.doi.org/10.1007/BF02243778.

A. Wagner. A new duality based approach for the problem of locating a semi-obnoxious facility, 2015.

J. E. Ward and R. E. Wendell. A new norm for measuring distance which yields linear location problems. *Operations Research*, 28:836–844, 1980.

J. E. Ward and R. E. Wendell. Using block norms for location modeling. *Operations Research*, 33 (5):1074–1090, 1985.