J. Appl. Math. & Informatics Vol. 33(2015), No. 1 - 2, pp. 157 - 172
http://dx.doi.org/10.14317/jami.2015.157

# ALGORITHM FOR WEBER PROBLEM WITH A METRIC BASED ON THE INITIAL FARE†

LEV A. KAZAKOVTSEV* AND PREDRAG S. STANIMIROVIC

Abstract. We introduce a non-Euclidean metric for transportation systems with a defined minimum transportation cost (initial fare) and investigate the continuous single-facility Weber location problem based on this metric. The proposed algorithm uses the results for solving the Weber problem with Euclidean metric by Weiszfeld procedure as the initial point for a special local search procedure. The results of local search are then checked for optimality by calculating directional derivative of modified objective functions in finite number of directions. If the local search result is not optimal then algorithm solves constrained Weber problems with Euclidean metric to obtain the final result. An illustrative example is presented.

AMS Mathematics Subject Classification : 90B85.
*Key words and phrases* : Location problem, Weber problem, Radar metric.

## 1. Introduction

Weber problem [29] is a continuous optimization problem for finding a point $X^* \in \mathbb{R}^n$ satisfying

$$X^* = \arg\min_{X\in\mathbb{R}^n} \sum_{i=1}^{N} w_i \|A_i - X\|. \tag{1}$$

Here, $A_i \in \mathbb{R}^n, i = 1, \dots, N$ are some known demand points, $w_i \in \mathbb{R}, w_i \geq 0$ are some weighting coefficients, $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ is a vector norm [20].

Main appearances of the Weber problem include the warehouse location [10, 7], positioning computer and communication networks [14], locating base stations of wireless networks. Solving a Weber problem (searching for a centroid) is a step of many clustering algorithms [25, 19, 9].

Received February 14, 2014. Revised July 8, 2014. Accepted July 24, 2014. *Corresponding author. †The 2nd author gratefully acknowledges support from the Research Project 174013 of the Serbian Ministry of Science

The problem (1) was originally formulated by Weber [29] with Euclidean norm ($\|\cdot\| = l_2(\cdot)$) and it is generalized to $l_p$ norms and other metrics [29, 6].

Detailed explanation of various norms and metrics is presented in [21, 18, 8]. The $l_p$ norms play an important role in the theory and practice of location problems. The most common distance metrics in continuous space are Euclidean ($l_2$), rectangular ($l_1$) and Chebyshev ($l_\infty$) metrics but other metrics are also important for specific cases [1, 8, 23]. Various distance metrics can be used for solving clustering problems [26, 30]. In [16], authors consider norm approximation and approximated solution for Weber problems with an arbitrary metric using random search [15]. Problems with barriers are described in [18]. In special cases, such problems can be transformed into discrete problems [22].

In the case of public transportation systems, the price usually depends on distance. However, some minimum price is usually defined. For example, the initial fare of the taxi cab may include some distance, usually 1-5 km. Having rescaled the distances so that this distance included in the initial price is equal to 1, we can define the price function $d_P$ as

$$d_P(X,Y) = \max\{\|X-Y\|, 1\} \quad \forall X, Y \in \mathbb{R}^n, \tag{2}$$

where $\|\cdot\|$ is a vector norm. We use the term "taxi metric" to denote the metric defined by (2). In this paper, we consider $\|\cdot\|$ as Euclidean norm in $\mathbb{R}^2$ only ($\|\cdot\|_2$).

In clustering problems, such metric can be used to describe the distance between the samples and the core of the cluster [27] with fixed core diameter. A metric which neglects the distances smaller than some pre-defined observational error $\mathcal{E}$ is equivalent with our "taxi" metric:

$$d_E(X,Y) = \max\{\|X-Y\| - \mathcal{E}, 0\} = \mathcal{E}\left(d_P\left(\frac{X}{\mathcal{E}}, \frac{Y}{\mathcal{E}}\right) - 1\right).$$

The Radar Screen [3] metric is a very similar norm metric with the distance function defined by

$$d_{rs}(X,Y) = ||X-Y||_{rs} = \min\{1, ||X-Y||_2\} \forall X, Y \in \mathbb{R}^n. \tag{3}$$

The Weber problem with the Radar Screen metric is a special case of the problem considered in [11]. Unlike (3), our distance function (2) is convex and our approach significantly differs from that proposed in [11].

The paper is organized as follows. In Chapter 2, we restate some basic definitions and describe existing algorithms and investigate some features of the objective function. In Chapter 3, we restate the algorithm for the Weber problem with new metric. In chapter 4, we give a simple example and results of the algorithm.

## 2. Preliminaries

The single-facility Weber problem (1) in $\mathbb{R}^2$ (planar problem) with "taxi metric" (2) can be formulated as

$$X^* = \arg\min_{X\in\mathbb{R}^2} f(X) = \arg\min_{X\in\mathbb{R}^2} \sum_{i=1}^{N} w_i \max\{1, ||A_i - X||_2\}$$
$$= \arg\min_{X\in\mathbb{R}^2} \sum_{i=1}^{N} w_i \max\left\{1, \sqrt{(x_1-a_1^i)^2+(x_2-a_2^i)^2}\right\}. \tag{4}$$

Here, $A_i = (a_1^i, a_2^i), i = 1, \ldots, N$, $X = (x_1, x_2)$.

The problem proposed by Weber is based on the Euclidean metric

$$X^* = \arg\min_{X\in\mathbb{R}^2} f_E(X) = \arg\min_{X\in\mathbb{R}^2} \sum_{i=1}^{N} w_i ||A_i - X||_2$$
$$= \arg\min_{X\in\mathbb{R}^2} \sum_{i=1}^{N} w_i \sqrt{(x_1-a_1^i)^2+(x_2-a_2^i)^2}. \tag{5}$$

The most common algorithm for Weber problem with the metrics induced by the $l_p$ norms is Weiszfeld procedure [28, 10].

For the simplicity, we assume that

$$w_i > 0,\ i = 1, \ldots, N. \tag{6}$$

**Lemma 2.1.** *If*

$$\exists \mathcal{S}_E \subset \mathbb{R}^2 : \|X - A_i\|_2 \le 1\ \forall X \in \mathcal{S}_E, i = 1, \ldots, N$$

*then any point $X \in \mathcal{S}_E$ is a solution of problem* (4)*. Moreover, any $X' \notin \mathcal{S}_E$ is not a minimizer of* (4)*.*

*Proof.* Let us assume that $X^* \in \mathcal{S}_E$. Then for arbitrary $\forall X \in \mathbb{R}^2$ we have

$$f(X^*) = \sum_{i=1}^{N} w_i \le \sum_{i=1}^{N} w_i \max\{1, \|X - A_i\|_2\} = f(X),$$

which implies $f(X^*) = \min\{f(X),\ X \in \mathbb{R}^2\}$. □

Lemma 2.1 describes the case when the non-iterative solution is possible. More several cases when the non-iterative approach is applicable are described in [2].

Let us denote the set

$$\mathcal{R}_0 = \{X \in \mathbb{R}^2 |\ \|X - A_i\|_2 \ge 1,\ i = 1, \ldots, N\}. \tag{7}$$

**Lemma 2.2.** *If $X^*$ is the solution of the problem* (4) *and $X^* \in \mathcal{R}_0$ then $X^*$ is the solution of Weber problem* (5) *and vice versa.*

*Proof.* Under the assumption $X^* \in \mathcal{R}_0$ we have $f(X^*) = f_E(X^*)$. □

**Lemma 2.3.** *The objective function of problem* (4) *is convex.*

*Proof.* The sum of convex functions $f_i(X) = \max\{1, ||X - A_i||_2\},\ i = 1, \ldots, N$ is convex. □

For any arbitrary point $X \in \mathbb{R}^2$, let us denote the sets of demand point indices

$$S_{\leq}(X) = \{i \in \{1, \ldots, N\} |\ \|X - A_i\|_2 \leq 1\}, \tag{8}$$

$$S_{\geq}(X) = \{i \in \{1, \ldots, N\} |\ \|X - A_i\|_2 \geq 1\}, \tag{9}$$

$$S_{>}(X) = \{i \in \{1, \ldots, N\} |\ \|X - A_i\|_2 > 1\} \tag{10}$$

and a set of points (a region)

$$\mathcal{R}(X) = \{Y \in \mathbb{R}^2 |\ S_{\leq}(X) = S_{\leq}(Y) \text{ or } S_{\geq}(X) = S_{\geq}(Y)\ \forall i = 1, \ldots, N\} \tag{11}$$

The regions $\mathcal{R}(X)$ of any point $X \in \mathbb{R}^2$ are bounded by arcs [11, 12] of radius 1 with centres in points $A_i, i = 1, \ldots, N$ (see Fig. 1). In [4], authors prove that the quantity of regions is quadratically bounded by the number of demand points.

An algorithm for solving constrained Weber problems with regions bounded by arcs is proposed in [17].

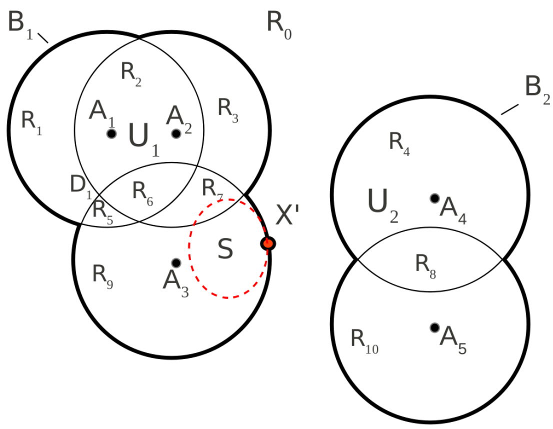


FIGURE 1. Illustration of the problem (4), its regions $\mathcal{R}_k, k = \overline{1, M}$ and unions $U_1$, $U_2$.

Let our problem have $M$ different regions $\mathcal{R}_k$, $k = 1, \ldots, M$:

$$(X \in \mathcal{R}_k) \ \Leftrightarrow\ (\mathcal{R}_k = \mathcal{R}(X)).$$

Note that $\mathcal{R}_0$ was introduced above. The border point of each region belong to at least one other region.

The algorithm for enumerating all the disc intersection points is given in [5]. Let our problem has $I$ disc intersections $D_1, \ldots, D_I$.

**Lemma 2.4.** *If $X^*$ is a solution of the problem* (4) *and $X^* \in \mathcal{R}_k$, $k = 0, \dots, M$ and $S_>(X^*) \neq \emptyset$ then $X^*$ is the solution of the following constrained Weber problem with the Euclidean metric:*

$$\arg\min_{X \in \mathbb{R}^2} f_{\mathcal{R}_k}(X) = \arg\min_{X \in \mathbb{R}^2} \sum_{i \in S_>(X^*)} w_i ||A_i - X, ||_2 \tag{12}$$

$$X \in \mathcal{R}_k. \tag{13}$$

*Proof.* The value of the objective function for $X \in \mathcal{R}_k$ is

$$\begin{aligned} f(X) = \sum_{i=1}^{N} w_i d_P(X, A_i) &= \sum_{i \in S_\leq(X)} w_i + \sum_{i \in S_>(X)} w_i ||X - A_i||_2 \\ &= \sum_{i \in S_\leq(X^*)} w_i + \sum_{i \in S_>(X^*)} w_i ||X - A_i||_2. \end{aligned} \tag{14}$$

Since the first summand in (14) is constant, we have an equivalent problem (12) with the constraint (13). □

The solution of constrained optimization problems with convex objective functions coincides with the solution of the corresponding unconstrained problem or lays on the border of the forbidden region [12] (moreover, the solution of the constrained problem is said to be visible from the solution of the unconstrained problem).

**Corollary 2.5.** *If $X^*$ is a solution of problem* (4) *then it is the solution of the unconstrained problem* (12) *or $\exists i \in \{1, \dots, N\}$ which satisfies $\|A_i - X^*\|_2 = 1$.*

Let us denote by $U_q$, $q = 1, \dots, N_U$ the unions of regions $\mathcal{R}_k, k = 1, \dots, M$ surrounded by the region $\mathcal{R}_0$:

$$U_q = \bigcup_{k: \mathcal{R}_k \in U_q} \mathcal{R}_k, \; q = 1, \dots, N_U,$$

$$\bigcup_{q=1}^{N_U} U_q = \bigcup_{k=1}^{M} \mathcal{R}_k.$$

Denote also the borders of those unions by

$$B_q = U_q \cap \mathcal{R}_0$$

and set of points of all the borders as

$$\mathcal{B} = \bigcup_{q=1,\dots,N_U} B_q.$$

**Lemma 2.6.** *If $X^{**}$ is the unique solution of the problem* (5) *and $X^*$ is a solution of the problem* (4) *then*

$$\exists U_{q'}, \, q' \in \{1, \dots, N_U\} : X^* \in U_{q'}, X^{**} \in U_{q'}.$$

*Proof.* Let us consider a constrained problem with the Euclidean metric

$$\arg\min_{X\in\mathbb{R}^2} f_E(X), \tag{15}$$

$$X \in \mathcal{R}_0. \tag{16}$$

Let $X'$ be a solution of this problem.

As the objective function of this problem is convex, two cases are possible.

Case 1. $X' = X^{**}$.

Case 2. Solution $X'$ of this constrained problem lies on the borderline of the feasible set, i.e. $X' \in \mathcal{B}$. Moreover, $X'$ is visible from $X^{**}$.

In Case 1, in accordance with Lemma 2.2, $X' \neq X^{**}$ unless $X' \in \mathcal{B}$. Thus, if $X^{**} \in U_{q'}$ then $X' \in U_{q'}$. From $X' \in \mathcal{R}_0$, we have $X' \in B_{q'}$. Let us denote the set (see Fig. 1)

$$\mathcal{S} = \{X \in \mathbb{R}^2 |\ f(X) \leq f(X')\}.$$

From $f(X) = f_E(X)\ \forall X \in \mathcal{R}_0$, $X'$ is the solution of the constrained problem

$$\arg\min_{X\in\mathbb{R}^2} f(X),$$

$$X \in \mathcal{R}_0.$$

From the convexity of the objective function $f(\cdot)$ immediately follows that $\mathcal{S}$ is convex. Let us denote a set $\mathcal{X}'_S$ of optimizers of the constrained problem (15) – (16). From

$$\nexists X'' \in \mathcal{R}_0 :\ f(X'') \leq f(X'),$$

we have

$$\nexists X'' \in \mathcal{B} :\ f(X'') \leq f(X').$$

Thus,

$$\nexists X'' \in \mathcal{B} \setminus \mathcal{X}' :\ X'' \in \mathcal{S}.$$

Therefore, the set $S$ does not contain any barriers $B_q$ of the unions $U_q$ ($q = 1, \ldots, N_U$) except the points from $\mathcal{X}'_S$ and $\exists X' \in \mathcal{X}'_S :\ X' \in U_{q'}$. Since $X' \in U_{q'}$ and $(\mathcal{S} \cap \mathcal{R}_0) \subset \mathcal{B}$, we have

$$\mathcal{S} \subset U_{q'}.$$

Since $X^*$ is the optimizer of (4), $f(X^*) \leq f(X')$. Thus, $X^* \in \mathcal{S}$ and $X^* \in U_{l'}$. □

**Lemma 2.7.** *Let $\mathcal{X}'_S$ be the set of solutions of the constrained problem* (12)-(13). *Let $\mathcal{G}_k$ be the set of border points of region $\mathcal{R}_k$. Then the set $\mathcal{G} \cap \mathcal{X}'_S$ is finite unless $S_>(X^*) = \emptyset \forall X^* \in \mathcal{X}'_S$.*

*Proof.* The case $||A_i - X||_2 \leq 1 \forall i = 1, ..., N, X \in X'$ Let $X^* \in \mathcal{X}'_S$ be an arbitrary point. If $S_>(X^*) \neq \emptyset$ then, being the Weber problem with the Euclidean metric, problem (12) has a strictly convex objective function unless all its demand points $A_i,\ i \in S_>(X^*)$ are collinear. In this case, the problem has exactly one solution.

If the demand points are collinear, the solution coincides with one of demand point $A_{i'}, i' \in S_>(X^*)$ or all points of some line segment $A_{i'}A_{i''}, i' \in$

$S_>(X^*), i'' \in S_>(X^*)$ are solutions. The border $\mathcal{G}$ is formed by arcs. Thus, it has finite number of intersections with the line segment. □

The algorithms proposed in the next section are based on the lemmas above.

Algorithms for both constrained and unconstrained Weber problem with Euclidean metric are well investigated, see [12, 13, 29]. We use these algorithms as subroutines in our algorithm.

## 3. Algorithm description

Our algorithm starts the local search procedure from the initial point which is calculated by the Weiszfeld procedure as the solution of the unconstrained Weber problem with the Euclidean metric (5). If the solution $X^*$ satisfies $X^* \in \mathcal{R}_0$ (i.e. $\|X^* - A_i\|_2 \geq 1$, $i = 1, \dots, N$) then, in accordance with Lemma 2.2, $X^*$ is the solution of problem (4). Otherwise, algorithm continues further search from point $X^*$.

Having solved problem (12) with constraint $X \in \mathcal{R}(X^*)$, we obtain a new solution $X^*$ or a set of solutions. If the unique solution all points from the solution set belong to the border of the union of regions $U_{q'}$ then, in accordance with Lemma 2.6, we have the optimal solution.

If the unique solution $X^*$ or every point of the solution set does not contain any border points of region $\mathcal{R}(X^*)$, due to convexity of the objective function, we have the solution final and algorithm stops.

If the solution $X^*$ lays on the borderline of region $\mathcal{R}(X^*)$ or the solution set contains any border points then we must solve the constrained Weber problem for the regions containing $X^*$. If there are some better solutions, continue with the best solution. Otherwise, stop.

Since the objective function is convex, we can use any local search procedure. The following heuristics provides the significant speed-up. First, the value of the objective function is calculated for the circle intersection points of the region $\mathcal{R}(X^*)$ (i.e. its angular points) where $X^*$ if the solution of the unconstrained Weber problem (5). This intersection $X^{**}$ with the best result is chosen as an initial point for the further search. The local search procedure continues then with the neighbor intersection points (i.e. the intersection points which are the ends of the arcs starting from $X^{**}$). When the local search stops at some intersection $X^{**}$, our algorithm checks if this point is the local minimum in each of its neighbor regions. If it is not the local (and global) minimum, the search continues with solving constrained Weber problem as described above.

If a temporal solution $X^{**}$ is an intersection point, the algorithm checks if this point is the local minimum in each of its regions. The angular point of the convex region is the point of minimum of the function in this region if all possible directional derivatives are non-negative. But our regions can be non-convex.

Let us denote $\mathcal{P}(\mathcal{R}_k)$ such a convex polygon that all its vertices coincide with the angular points of the region $\mathcal{R}_k$. Then region

$$\varrho(\mathcal{R}_k) = \mathcal{R}_k \cap \mathcal{P}(\mathcal{R}_k)$$

is convex.

Let us denote two rays $l_1$ and $l_2$ with initial point $X^{**}$ and an angle $\phi \in (0, \pi)$ between them such that all points of the region $\varrho(\mathcal{R}_k)$ are situated between $l_1$ and $l_2$ and both $l_1$ and $l_2$ are tangent to the borderline of the region $\varrho(\mathcal{R}_k)$. All possible directions from $X^{**}$ in the region $\varrho(\mathcal{R}_k)$ lay between $l_1$ and $l_2$.

From the convexity of region $\varrho(\mathcal{R}_k)$ and the objective function (12), if

$$\frac{\partial f_{\mathcal{R}_k}}{\partial \overline{l_1}}(X^{**}) > 0, \frac{\partial f_{\mathcal{R}_k}}{\partial \overline{l_2}}(X^{**}) > 0 \tag{17}$$

then $X^{**}$ is the minimum point of (12) in $\varrho(\mathcal{R}_k)$. Here, $\frac{\partial f_{\mathcal{R}_k}}{\partial \overline{l_1}}$ and $\frac{\partial f_{\mathcal{R}_k}}{\partial \overline{l_2}}$ are directional derivatives with directions $l_1$ and $l_2$ correspondingly. From $\varrho(\mathcal{R}_k) \subset \mathcal{R}_k$, this point $X^{**}$ is the minimum point in $\mathcal{R}_k$.

If $X^{**}$ is the point of local minimum for all regions which it joins then $X^{**}$ is the solution of problem (4) and solving the constrained Weber problem (12), (13) is not needed. The experiments on the randomly generated problems and the rescaled problems from [24] show that solving the constrained Weber problem is not needed in most cases.

In our algorithm, regions $\mathcal{R}_k$ are enumerated as follows. The number $k$ is an array of $N$ digits, one digit for each of the demand points. The $i$th digit is set to 1 if $||X - A_i|| < 1$ for all internal points $X$ of region $\mathcal{R}_k$. If $||X - A_i|| > 1$ then the $i$th digit is set to 0. For example, region $\mathcal{R}_6$ (see Fig. 1) in new notation is $\mathcal{R}_{11100}$. Using this method of enumeration, it is not necessary to enumerate all regions at the first steps of the algorithm.

Analogous method of enumeration is used for intersection points $D_j$. The index $j$ contains $N$ digits. If $||D_j - A_i|| > 1$ then the $i$th digit is set to 0. If $||D_j - A_i|| < 1$ then the $i$th digit is set to 1. If $||D_j - A_i|| = 1$ then the $i$th digit is set to 2. For example, the angulous point $D_1$ (see Fig. 1) of regions $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_5$ and $\mathcal{R}_6$ in the proposed algorithm is denoted as $D_{12200}$ (it is an internal point of the circle with center in $A_1$, border point of circles with centers in $A_2$ and $A_3$ and it is situated outside circles with centers in $A_4$ and $A_5$).

With this notation, it is easy to determine the region or regions for any arbitrary point $X^*$. We use the following algorithm (here, $k$ is an array of digits).

Note that the region $\mathcal{R}_0$, see (7), in this notation is $\mathcal{R}_{000\ldots0}$.

**Algorithm 3.1.** Determine the region index

**Require:** Coordinates $X = (x_1, x_2)$ ot the point, coordinates of the demand points $A_i = (a_1^i, a_2^i), i = \overline{1, N}$.

**Step 1:** for $i = 1, \ldots, N$ do:

**Step 1.1:** If $\|A_i - X\| = 1$ then $k[i] = 2$;

**Step 1.2:** else if $\|A_i - X\| < 1$ then $k[i] = 1$;

**Step 1.3:** else $k[i] = 0$;

**Step 1.4:** Continue Step 1;

**Step 2:** $R_{array}[1] = \{k\}$; $N_r = 1$;

**Step 3:** For $i$ in $\{\overline{1,N}\}$ do:
**Step 3.1:** if $k[i] = 2$ then
**Step 3.2:** $\mathcal{P}_{array} = R_{array}$;
**Step 3.3:** for $j$ in $\{\overline{1,N_r}\}$ do:
**Step 3.3.4:** $R_{array}[j][k] = 0$; $\mathcal{P}_{array}[j][k] = 1$; Continue Step 2.3;
**Step 3.4:** Add all elements of $\mathcal{P}_{array}$ to the end of the array $R_{array}$; $N_r = N_r * 2$;
**Step 3.5:** Continue Step 2;
**Step 4:** STOP, return $R_{array}$ and number of its elements $N_r$.

The algorithm above returns a set (an array) $R_{array}$ of region indexes $k$ such that $X \in \mathcal{R}_k$. Steps 1 to 1.4 form an array of digits describing the distance from the given point to each of the demand points: digits 0,1 and 2 mean distance more than 1, less than 1 and equal to 1, correspondingly. In Steps 3 to 3.5, array $R_{array}$ of indexes is formed. Initially, it contains one element coinciding with the array $k$ formed in Steps 1 to 1.4. For each demand point having distance equal to 1 from the given point (digit 2 in array $k$), array $R_{array}$ is duplicated: instead of digit 2, digit 0 is substituted in the first copy of the initial array $R_{array}$ and digit 1 in its second copy. Thus, array $R_{array}$ contains $2^{e_1}$ indexes where $e_1$ is quantity of the demand points having distance equal to 1 from point $X$.

For any intersection point $D_j$, the index $j$ is known and we can start this algorithm for such point from Step 2 assuming $k = j$.

For determining the set of the neighbor intersection points for a given intersection point $D_j$, we use the following algorithm.

**Algorithm 3.2.** Form a list of neighbor angular points

**Require:** An index $j^*$ of the intersection point $D_{j^*}$ (here, $j^*$ is an array of digits 0,1,2), a set of all intersection points $\mathcal{D}_{all}$.
**Step 1:** $\mathcal{D}_{neighbour} = \emptyset$;
**Step 2:** For each $D_i$ in $calD_all \setminus \{D_{j^*}\}$ do:
Comment: here, the indexes $j^*$ and $i$ are considered as arrays of digits.
**Step 2.1:** $n_{common} = 0$; $b_{ok} = 1$.
**Step 2.2:** For $k = 1, \ldots, N$ do:
**Step 2.2.1:** If $i[k] = 2$ and $j^*[k] = 2$ then $n_{common} = n_{common} + 1$; ;
**Step 2.2.2:** else if $j^*[k] \neq 2$ and $j^*[k] \neq i[k]$ then $b_{ok} = 0$; break Step 2.2 and go to Step 2.3.
**Step 2.2.5:** Continue Step 2.2.
**Step 2.3:** if $n_{common} > 0$ and $b_{ok} = 1$ then $\mathcal{D}_{neighbour} = \mathcal{D}_{neighbour} \cup \{D_i\}$.
**Step 2.4:** Continue Step 2.
**Step 3:** STOP, return $\mathcal{D}_{neighbour}$.

In Steps 2 to 2.4, all known intersetion points are scanned. For the indexes of the intersection points, the notation from 3.1 is used: digit 2 in the $k$th position of the index means that distance from the intersection point to the $k$th demand point is equal to 1. In Steps 2.2 to 2.2.5, searching for digits 2 in indexes is organized.

Our algorithm for solving problem (4) is organized as follows.

**Algorithm 3.3.** Solving the location problem (4)

**Require:** Set of $N$ demand points $\mathcal{A}$ with coordinates $A_i = (a_1^i, a_2^i)$ of the demand points and their weights $w_i$, $i = 1, \ldots, N$.
**Step 1:** Solve the Weber problem with Euclidean metric (5) implementing Weiszfeld procedure, store the result to $X^*$.
**Step 2:** if $\nexists i \in \{1, \ldots, N\} : \|X^* - A_i\| \leq 1$ then STOP and return $X^*$.
**Step 3:** Determine the region $\mathcal{R}_{k^*} = \mathcal{R}(X^*)$ with Algorithm 3.1. The result is index $k$ which is an array of $N$ digits. If a set of regions is returned then we use the first one.
**Step 4:** Form the set $\mathcal{D}_{all}$ of all intersection points of the circles with centres in $A_i, i = 1, \ldots, N$ and radius 1.
**Step 5:** Form the set $\mathcal{D}$ of all angular points (intersections) of the region $\mathcal{R}_k$; Set $\mathcal{D}_{checked} = \mathcal{D}$.
**Step 6:** $\mathcal{F}^{**} = +\infty$.
**Step 7:** For each element $D_j$ of the set $\mathcal{D}$ do:
**Step 7.1:** If $f(D_j) < \mathcal{F}^{**}$ then $\mathcal{F}^{**} = f(D_j)$; $X^{**} = D_j$.
**Step 7.2:** Continue Step 7.
**Step 8:** $b_{found} = 1$.
**Step 9:** while $b_{found} = 1$ do:
**Step 9.1:** $b_{found} = 0$; Call Algorithm 3.2 to form the set $\mathcal{D}_{neighbour}$ of the neighbor intersections of $X^{**}$.
**Step 9.2:** For $X'$ in $\mathcal{D}_{neighbour} \setminus \mathcal{D}_{checked}$ do:
**Step 9.2.1:** If $f(X') < \mathcal{F}^{**}$ then $\mathcal{F}^{**} = f(X')$; $b_{found} = 1$; $X^{**} = X'$.
**Step 9.3:** $\mathcal{D}_{checked} = \mathcal{D}_{checked} \cup \mathcal{D}_{neighbour}$.
**Step 9.4:** Continue Step 9.
**Step 10:** Form the set $\mathcal{L}$ of regions joint by the set $X^{**}$ with Algorithm 3.1;
**Step 11:** $\mathcal{L}_{tosearch} = \emptyset$.
**Step 12:** For each region $\mathcal{R}_k$ in $\mathcal{L}$ do:
**Step 12.1:** For the convex region $\varrho(\mathcal{R}_k) = \mathcal{R}_k \cap \mathcal{P}(\mathcal{R}_k)$, calculate two directions (rays with initial point $X$) $l_1$ and $l_2$ tangent to the borderline of the region $\mathcal{R}_k$.
**Step 12.2:** If $\frac{\partial f_{\mathcal{R}_k}}{\partial \overline{l_1}}(X^{**}) \leq 0$ or $\frac{\partial f_{\mathcal{R}_k}}{\partial \overline{l_2}}(X^{**}) \leq 0$ then
**Step 12.2.1:** $\mathcal{L}_{tosearch} = \mathcal{L}_{tosearch} \cup \{\mathcal{R}_k\}$.
**Step 12.3:** Continue Step 12.
**Step 13:** While $\mathcal{L}_{tosearch} \neq \emptyset$ do:
**Step 13.1:** For each element $\mathcal{R}_k$ in $\mathcal{L}_{tosearch}$ do:
**Step 13.1.1:** Solve the constrained Weber problem (12)–(13) using the modified Weiszfeld procedure [11, 4], store the result to $X'$.
**Step 13.1.2:** If $f(X') < \mathcal{F}^{**}$ then $\mathcal{F}^{**} = f(X')$; $X^{**} = X'$; Determine the set $R_{array}$ of regions of $X'$ using Algorithm 3.1.
$\mathcal{L}_{tosearch} = R_{array} \setminus \mathcal{L}_{tosearch}$;
break Step 13.1 and go to Step 13.2.

**Step 13.1.3:** Continue Step 13.1.
**Step 13.2:** Continue Step 13.
**Step 14:** STOP and return $X^{**}$.

## 4. Numerical example

Let us solve the problem shown in Fig. 2

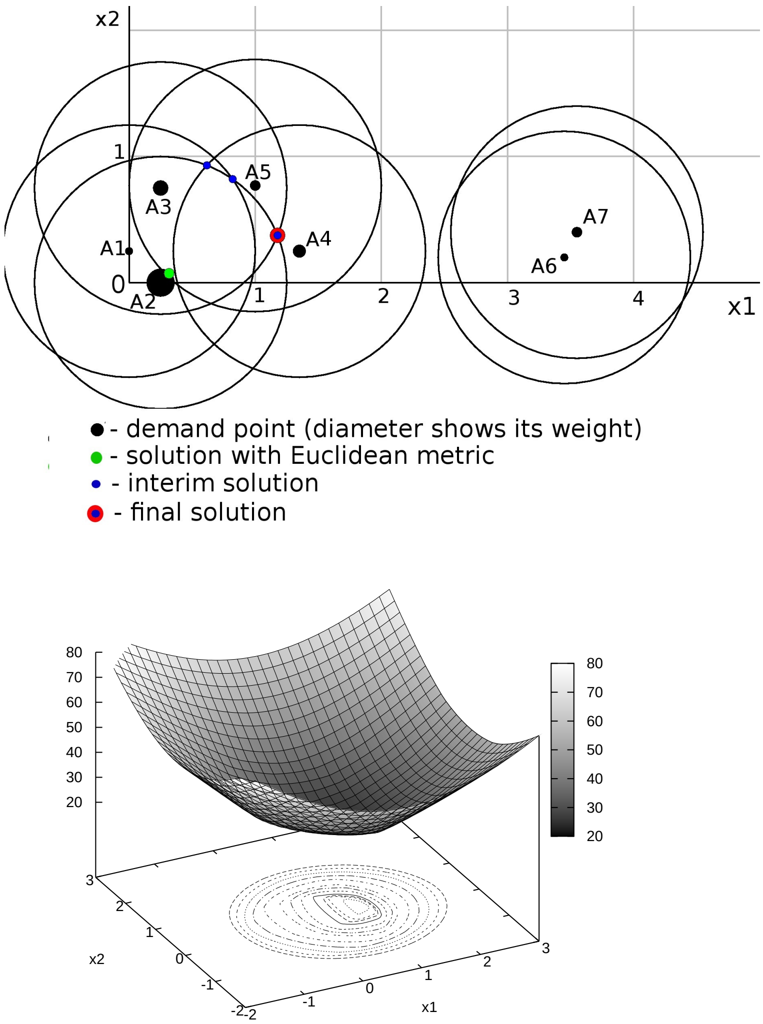


FIGURE 2. Example problem scheme and its objective function graph.

Here, $N = 7$, $A_1 = (0, 0.25)$, $A_2 = (0.25, 0)$, $A_3 = (0.25, 0.75)$, $A_4 = (1.35, 0.25)$, $A_5 = (1, 0.77)$, $A_6 = (3.45, 0.2)$, $A_7 = (3.55, 0.4)$, $w_1 = w_6 = 1$, $w_2 = 9$, $w_3 = 4$, $w_4 = 3$, $w_5 = w_7 = 2$.

The result of Weiszfeld procedure at Step 1 of Algorithm 3.3 is

$$X^* = (1.1770229, 0.375000).$$

At Step 2, this point is not in $\mathcal{R}_{0000000}$ since $\|A_1 - X^*\| < 1$. Thus, the algorithm goes on.

At Step 3, from $\|A_1 - X^*\| < 1$, $\|A_2 - X^*\| < 1$, $\|A_3 - X^*\| < 1$, $\|A_5 - X^*\| < 1$, $\mathcal{R}(X^*) = \mathcal{R}_{11101000}$.

At Step 4, our algorithm forms a set $\mathcal{D}_{all}$ of all 22 intersection points.

At Step 5, the set of angular points (intersections) of region $\mathcal{R}_{1110100}$ is

$$\begin{aligned}\mathcal{D} = \mathcal{D}_{checked} &= \{D_{1212100}, D_{1210200}, D_{1112200}\}\\ &= \{(0.616995, 0.930223), (0.020905, 0.973404), (0.387208, -0.020244)\}.\end{aligned}$$

After Step 6 and three iterations in Steps 7 to 7.2, we have

$$X^{**} = D_{1212100} = (0.616995, 0.930223), \quad \mathcal{F}^{**} = 27.886994.$$

At Step 8, a boolean variable $b_{found}$ is set to 1 and our algorithm start the iteration (Step 9).

At Step 9.1, $b_{found}$ is reset to 0. Algorithm 3.2 returns the list of the neighbor intersections for $D_{1212100}$:

$$\begin{aligned}\mathcal{D}_{neighbour} &= \{D_{2012100}, D_{1210200}, D_{1112200}, D_{2211100}\}\\ &= \{(0.675000, 0.987818), (0.020905, 0.973404),\\ &\qquad (0.387208, -0.020244), (0.820971, 0.820971)\}.\end{aligned}$$

At Step 9.2 to 9.2.1, the algorithm estimates the objective function for these intersections except $D_{1210200}$, $D_{1112200}$ and after these iterations, we have

$$X^{**} = D_{2211100} = (0.820971, 0.820971), \ \mathcal{F}^{**} = 27.223985, \ b_{found} = 1.$$

At Step 9.3, the algorithm adds $\mathcal{D}_{neighbour}$ to the set $\mathcal{D}_{checked}$ and we have $\mathcal{D}_{checked} = \{D_{1212100}, D_{1210200}, D_{1112200}, D_{2012100}, D_{2211100}\}$.

Step 9 is then repeated.

At the second iteration of Step 9.1, $b_{found}$ is reset to 0. Algorithm 3.2 returns the list of the neighbor intersections for $D_{1212100}$:

$$\begin{aligned}\mathcal{D}_{neighbour} &= \{D_{2012100}, D_{0221100}, D_{1212100}, D_{212110}\}\\ &= \{(0.675000, 0.987818), (0.177025, 0.375000),\\ &\qquad (0.616995, 0.930223), (0.983778, 0.070611)\}.\end{aligned}$$

At Step 9.2 to 9.2.1, the algorithm estimates the objective function for these intersections except $D_{2012100}$, $D_{1212100}$ and after two iterations, we have

$$X^{**} = D_{0221100} = (1.177025, 0.375000), \ \mathcal{F}^{**} = 26.209559, \ b_{found} = 1.$$

At Step 9.3, $\mathcal{D}_{checked} = \{D_{1212100}, D_{1210200}, D_{1112200}, D_{2012100}, D_{2211100}, D_{0221100}, D_{2121100}\}$, Step 9 is then repeated.

At the third iteration of Step 9.1, $b_{found}$ is reset to 0. Algorithm 3.2 returns the list of the neighbor intersections for $D_{1212100}$:

$$\begin{aligned}\mathcal{D}_{neighbour} &= \{D_{0201200}, D_{0022100}, D_{2121100}, D_{2211100}\}\\ &= \{(1.229095, -0.203404), (1.129747, 1.225443),\\ &\quad (0.983778, 0.070610), (0.820971, 0.820971)\}.\end{aligned}$$

At Step 9.2 to 9.2.1, the algorithm estimates the objective function for the intersections $D_{0201200}$ and $D_{002210}$ and after two iterations, we have no improvement of $X^{**} = D_{0221100}$ and $\mathcal{F}^{**} = 26.209559$.

Thus, $b_{found} = 0$ and the iterations of Step 9 finish.

At Step 10, the list of the regions joint by $X^{**} = D_{022110}$ is

$$\mathcal{L} = \{\mathcal{R}_{0001100}, \mathcal{R}_{0101100}, \mathcal{R}_{0011100}, \mathcal{R}_{0111100}\}.$$

Algorithm sets $\mathcal{L}_{tosearch} = \emptyset$.

In region $\mathcal{R}_{0001100}$, the direction $l_1$ is a ray on the line connecting $X^{**}$ and $D_{0201200}$ ($d_7$ in Fig. 3), $l_2$ is a ray on the line connecting $X^{**}$ and $D_{0022100}$ ($d_2$ in Fig. 3).

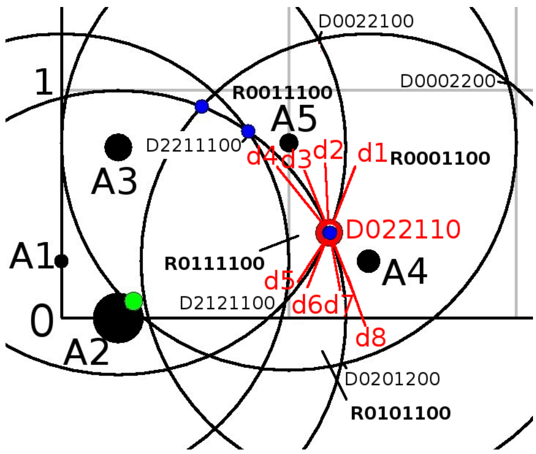


FIGURE 3. Neighbor regions and directions for directional derivatives calculations (Steps 12 to 12.3 of Algorithm 3.3).

In region $\mathcal{R}_{0011100}$, the direction $l_1$ is a ray on the line tangent to the circle with center in $A_3$ ($d_1$ in Fig. 3), $l_2$ is a ray on the line connecting $X^{**}$ and $D_{2211100}$ ($d_4$ in Fig. 3).

In region $\mathcal{R}_{0111100}$, the direction $l_1$ is a ray on the line tangent to the circle with center in $A_2$ ($d_3$ in Fig. 3), $l_2$ is a ray on the line tangent to the circle with center in $A_3$ ($d_6$ in Fig. 3).

In region $\mathcal{R}_{010110}$, the direction $l_1$ is a ray on the line tangent to the circle with center in $A_2$ ($d_8$ in Fig. 3), $l_2$ is a ray on the line connecting $X^{**}$ and $D_{2121100}$ ($d_5$ in Fig. 3).

In Steps 12 to 12.2, our algorithm calculates all directional derivatives

$$\frac{\partial(\sum\limits_{i\in\{1,2,3,6,7\}} w_i||X-A_i||_2)}{\partial\overline{d_7}}(X^{**}),\ \frac{\partial(\sum\limits_{i\in\{1,2,3,6,7\}} w_i||X-A_i||_2)}{\partial\overline{d_2}}(X^{**}),$$
$$\frac{\partial(\sum\limits_{i\in\{1,2,6,7\}} w_i||X-A_i||_2)}{\partial\overline{d_1}}(X^{**}),\ \frac{\partial(\sum\limits_{i\in\{1,2,6,7\}} w_i||X-A_i||_2)}{\partial\overline{d_4}}(X^{**}),$$
$$\frac{\partial(\sum\limits_{i\in\{1,6,7\}} w_i||X-A_i||_2)}{\partial\overline{d_3}}(X^{**}),\ \frac{\partial(\sum\limits_{i\in\{1,6,7\}} w_i||X-A_i||_2)}{\partial\overline{d_6}}(X^{**}),$$
$$\frac{\partial(\sum\limits_{i\in\{1,3,6,7\}} w_i||X-A_i||_2)}{\partial\overline{d_8}}(X^{**}) \text{ and } \frac{\partial(\sum\limits_{i\in\{1,3,6,7\}} w_i||X-A_i||_2)}{\partial\overline{d_5}}(X^{**}).$$

All values are positive (Step 12.2).

Thus, $\mathcal{L}_{tosearch} = \emptyset$ and iterations in Steps 13 to 13.2 are not performed.

The resulting point is $X^{**} = (0.177025, 0.375000)$, value of the objective function is $\mathcal{F}^{**} = 26.209559$.

## 5. Conclusion

The location problems for the systems with the minimum transportation cost can be formulated as the problems with a special metric

$$d_P(X,Y) = \min\{1,\ \|X-Y\|_2\}.$$

The proposed algorithm is able to solve such problems. The implemented local search heuristic reduces the computational complexity to the complexity of solving few constrained and one unconstrained Weber problems with Euclidean metric. However, the computational complexity of the proposed algorithm is subject to the further research.

## References


1. R.G. Brown, *Advanced Mathematics: Precalculus with discrete Mathematics and data analysis*, (A.M.Gleason, ed.), Evanston, Illinois: McDougal Littell, 1997.
2. R. Chen, *Noniterative Solution of Some Fermat-Weber Location Problems*, Advances in Operations Research (2011), Article ID 379505, Published online. 10 pages doi:10.1155/2011/379505, http://downloads.hindawi.com/journals/aor/2011/379505.pdf.
3. M.M. Deza and E. Deza, *Encyclopedya of Distances*, Springer Verlag, Berlin, Heilderberg, 2009.
4. Z. Drezner, A. Mehrez and G.O. Wesolowsky, *The facility location problem with limited distances*, Transportation Science, **25** (1991), 183-187.
5. Z. Drezner and G.O. Wesolowsky, *A maximin location problem with maximum distance constraints*, AIIE Transact., **12** (1980), 249-252.

6. Z. Drezner, K. Klamroth, A. Schobel and G.O. Wesolowsky, *The Weber problem*, in Z. Drezner and H.W. Hamacher (editors), Facility Location: Applications and Theory, Springer-Verlag, 2002, 1-36.
7. Z. Drezner, C. Scott and J.S. Song, *The central warehouse location problem revisited*, IMA Journal of Managemeng Mathematics, **14** (2003), 321-336.
8. Z. Drezner and M. Hamacher, *Facility location: applications and theory*, Springer-Verlag, Berlin, Heidelberg, 2004.
9. S. Gordon, H. Greenspan, J. Goldberger, *Applying the Information Bottleneck Principle to Unsupervised Clustering of Discrete and Continuous Image Representations*, Computer Vision. Proceedings. Ninth IEEE International Conference on, Vol.1 (2003), 370-377.
10. R.Z. Farahani and M. Hekmatfar editors, *Facility Location Concepts, Models, Algorithms and Case Studies*, Springer-Verlag Berlin Heidelberg, 2009.
11. I.F. Fernandes, D. Aloise, D.J. Aloise, P. Hansen and L. Liberti, *On the Weber facility location problem with limited distances and side constraints*, Optimization Letters, issue of 22 August 2012, 1–18, published online, doi:10.1007/s11590-012-0538-9.
12. P. Hansen, D. Peeters and J.F. Thisse, *Constrained location and the Weber-Rawls problem*, North-Holland Mathematics Studies, **59** (1981) 147-166.
13. H. Idrissi, O. Lefebvre and C. Michelot, *A primal-dual algorithm for a constrained Fermat-Weber problem involving mixed norms*, Revue francaise d'automatique, d'informatique et de recherche operationnelle. Recherche Operationnelle, **22** (1988), 313-330.
14. L.A. Kazakovtsev, *Wireless coverage optimization based on data provided by built-in measurement tools*, WASJ, **22**, Special Volume on Techniques and Technologies (2013), 8-15.
15. A.N. Antamoshkin and L.A. Kazakovtsev, *Random search algorithm for the p-median problem*, Informatica (Ljubljana), **37** (2013), 267-278.
16. L.A. Kazakovtsev, *Adaptation of the probability changing method for Weber problem with an arbitrary metric*, Facta Universitatis, (Niš) Ser. Math. Inform., **27** (2012), 289-254.
17. L.A. Kazakovtsev, *Algorithm for Constrained Weber Problem with feasible region bounded by arcs*, Facta Universitatis, (Niš) Ser. Math. Inform., **28** (2013), 271-284.
18. K. Klamroth, Single-facility location problems with barriers, *Springer Verlag*, Berlin, Heilderberg, 2002.
19. K. Liao, D. Guo, *A Clustering-Based Approach to the Capacitated Facility Location Problem*, Transactions in GIS, **12** (2008), 323-339.
20. H. Minkowski, *Gesammelte Abhandlungen*, zweiter Band, Chelsea Publishing, 2001.
21. J. Perreur and J.F. Thisse, *Central metric and optimal location*, J. Regional Science, **14** (1974), 411-421.
22. I.P. Stanimirovic, *Successive computation of some efficient locations of the Weber problem with barriers*, J. Appl. Math. Comput., **42** (2013), 193?11. DOI 10.1007/s12190-012-0637-x
23. P.S. Staminirović, M. Cirić, L.A. Kazakovtsev and I.A. Osinuga, *Single-facility Weber location problem based on the Lift metric*, Facta Universitatis, (Niš) Ser. Math. Inform., **27** (2012), 31-46.
24. E. Taillard, *Location problems*, web resource available at http://mistic.heig-vd.ch /taillard-/problemes.dir/location.html
25. E.D. Thaillard, *Heuristic Methods for Large Centroid Clustering Problems*, Journal of Heuristics, **9** (2003), 51-73.
26. A. Vimal, S.R. Valluri, K. Karlapalem, An Experiment with Distance Measures for Clustering, *International Conference on Management of Data COMAD 2008, Mumbai, India*, 241–244 (2008)
27. K. Voevodski, M.F. Balcan, H. Roglin, S.H. Teng, Y. Xia, *Min-sum Clustering of Protein Sequences with Limited Distance Information*, Proceedings of the First International Conference on Similarity-based Pattern Recognition (SIMBAD'11), Venice, Italy (2011), 192-206.

28. E. Weiszfeld, *Sur le point sur lequel la somme des distances de n points donnes est minimum*, Tohoku Mathematical Journal, **43** (1937), 335-386.
29. G. Wesolowsky, *The Weber problem: History and perspectives*, Location Science, **1** (1993), 5-23.
30. Y. Ying, P. Li, *Distance Metric Learning with Eigenvalue Optimization*, Journal of Machine Learning Research, **13** (2012), 1-26.

**Lev A. Kazakovtsev** was awarded the candidate of technical sciences degree at the Research Institute of Control Systems, Wave Processes and Technologies (Krasnoyarsk, Russian Federation) in 2002. He is an associate professor of the Krasnoyarsk State Agrarian University (Chair of Mathematical Modeling and Informatics) and Deputy Director of the Department of Information Technologies of the Siberian State Aerospace University. His research interests include the location problems, discrete optimization, random search algorithms.

Department of Information Technologies, Siberian State Aerospace University named after M. F. Reshetnev, prosp. Krasnoyarskiy Rabochiy 31, Krasnoyarsk, 660093, Russian Federation.
e-mail: levk@bk.ru

**Predrag S. Stanimirovic** received his M.Sc. in 1990 and Ph.D in 1996 at the University of Niš, Faculty of Sciences and Mathematics. He is a Full Professor at the same faculty since 2002. His research interests include operation research, linear programming, nonlinear programming, generalized inverses and symbolic computations.

Faculty of Sciences and Mathematics, University of Niš, Višegradska 33, 18000 Niš, Serbia.
e-mail: pecko@pmf.ni.ac.rs