Journal of Global Optimization
https://doi.org/10.1007/s10898-019-00796-9

S.I.: MOA 2018

An ADMM-based location–allocation algorithm for nonconvex constrained multi-source Weber problem under gauge

Jianlin Jiang[1,2] · Su Zhang[3] · Yibing Lv[2] · Xin Du[4] · Ziwei Yan[1]

Received: 25 December 2018 / Accepted: 5 June 2019

Abstract
Multi-source Weber problem (MSWP) is a classical nonconvex and NP-hard model in facility location. A well-known method for solving MSWP is the location–allocation algorithm which consists of a location phase to locate new facilities and an allocation phase to allocate customers at each iteration. This paper considers the more general and practical case of MSWP called the constrained multi-source Weber problem (CMSWP), i.e., locating multiple facilities with the consideration of the gauge for measuring distances and locational constraints on new facilities. According to the favorable structure of the involved location subproblems after reformulation, an alternating direction method of multipliers (ADMM) type method is contributed to solving these subproblems under different distance measures in a uniform framework. Then a new ADMM-based location–allocation algorithm is presented for CMSWP and its local convergence is theoretically proved. Some preliminary numerical results are reported to verify the effectiveness of proposed methods.

The first author was supported by National Natural Science Foundation of China No. 11571169. The second author was supported by National Natural Science Foundation of China No. 71772094.

✉ Su Zhang
zhangsu@nankai.edu.cn

Jianlin Jiang
jiangjianlin@nuaa.edu.cn

Yibing Lv
yibinglv@yangtzeu.edu.cn

Xin Du
duxin@shu.edu.cn

Ziwei Yan
yanziwei880@163.com

1 Department of Mathematics, College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

2 School of Information and Mathematics, Yangtze University, Jingzhou 434023, China

3 Business School, Nankai University, Tianjin 300071, China

4 School of Mechanical Engineering and Automation and Shanghai Key Laboratory of Power Station Automation Technology, Shanghai University, Shanghai 200444, China

Published online: 11 June 2019

Keywords Multi-source Weber problem · Nonconvex · Location–allocation · ADMM · Gauge

1 Introduction

The classical multi-source Weber problem (MSWP) is to find the locations of m new facilities in order to minimize the sum of the transportation costs between these facilities and n customers. Assume that the involved transportation costs are proportional to the corresponding distances. More specifically, the mathematical model of MSWP is as follows:

$$\begin{aligned} \text{MSWP: } \min \ & \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}\|x_i - a_j\| \\ \text{s.t.} \ & \sum_{i=1}^{m} w_{ij} = s_j, \ \ j = 1, 2, \dots, n, \\ & x_i \in R^2, \ \ i = 1, 2, \dots, m, \end{aligned} \tag{1}$$

where

(1) $a_j \in R^2$ is the location of the jth customer, $j = 1, 2, \dots, n$;
(2) $x_i \in R^2$ is the location of the ith facility to be determined, $i = 1, 2, \dots, m$;
(3) $s_j \geq 0$ is the given demand required by the jth customer and $w_{ij} \geq 0$ denotes the unknown allocation from the ith facility to the jth customer;
(4) $\|\cdot\|$ is the Euclidean distance.

MSWP has wide applications in operations research, marketing, urban planing, supply chain management etc., see e.g. [15,25,29]. This problem is nonconvex [11] and NP-hard [33], and a few numerical algorithms have been proposed for solving it, see e.g. [3,10,47]. In particular, the location–allocation algorithm, also called Cooper algorithm since it was presented originally by Cooper [10], has received much attention. Its attractive characteristic is that each of its iterations consists of a location phase and an allocation phase and the algorithm alternately allocates customers in allocation phase and locates facilities in location phase until no further improvement is possible, see e.g. [10,15,26,29].

Note that each facility is capable of providing sufficient services for the targeted customers in MSWP, thus each customer is ultimately served only by its nearest facility in order to minimize the total sum of the system, see, e.g., [3,10,29]. Therefore, MSWP is equivalently rewritten as:

$$\begin{aligned} \text{MSWP}^{'} : \min \ & \sum_{j=1}^{n} s_j \min_{i=1,\dots,m} \|x_i - a_j\| \\ \text{s.t.} \ & x_i \in R^2, \ i = 1, 2, \dots, m. \end{aligned} \tag{2}$$

In classical MSWP model, the distances between the customers and the facilities are measured by l_2-norm, which is later extended to some general norms such as l_p-norm and block norm in the literature, see e.g. [32,44]. The symmetry property of the norm assures that the distance from one point to another is always equal to the distance back. Nevertheless, [46] has highlighted the fact that in numerous real situations the symmetry of the distance is violated, e.g. transportation in rush-hour traffic, flight in the presence of wind, navigation in the presence of currents, etc. In spite of the pioneering work of [46], the asymmetric distance has started to attract the researchers' interest until the recent several decades, and some progress has been made in both theoretical and computational aspects recently, see e.g. [6,27,34,39]. In this paper, the *gauges* (symmetric or asymmetric), rather than widely-used symmetric norms, is used to measure the distances between the customers and the facilities, and then both symmetric and asymmetric distances can be involved in our generalized MSWP model.

It is also noticed that not every point on the plane is suitable to locate the new facilities due to some actual situations. For example, in a health care system we can't build a health center or a hospital where there is a mountain or a lake. Samely, the location of a facility in the health care system can't be very close to heavily polluting and noisy factories. In practice, after considering relevant practical factors the locations of new facilities are usually restricted in some specific regions, which are usually represented by convex subsets of the plane. Then, we introduce *locational constraints* on new facilities in our generalized MSWP model. By choosing appropriate locational constraints on facilities, e.g. R^2 or the subset of R^2, both the unconstrained and constrained problems can be taken into consideration.

The rest of this paper is organized as follows. The formulation of the generalized multi-source Weber problem is given in Sect. 2, which is shown to be nonconvex and NP-hard. The spirit of well-known location–allocation algorithm is also discussed in this section. In Sect. 3, the location subproblems in location phase are solved by proposed ADMM-type method to find the optimal locations of facilities. The special cases of l_1-norm, l_2-norm and one particular asymmetric distance measure are considered firstly, and then are generalized into the uniform framework. In Sect. 4, a new location–allocation algorithm is proposed for solving the generalized multi-source Weber problem and its local convergence is proved. Preliminary numerical results are reported in Sect. 5 to verify the efficiency of proposed ADMM-type method and location–allocation algorithm. Finally, some conclusions are drawn in Sect. 6.

2 Model formulation

Let B be a compact convex set in R^2 and the interior of B contains the origin, then the gauge of B is defined by

$$\gamma(x) = \inf\left\{\lambda > 0 \mid \frac{x}{\lambda} \in B\right\}, \qquad \forall\, x \in R^2. \tag{3}$$

B is called as the unit ball of $\gamma(\cdot)$ due to

$$B = \{x \in R^2 \mid \gamma(x) \leq 1\}. \tag{4}$$

This definition of gauge from a compact convex set was firstly introduced by Minkowski [35]. The gauge $\gamma(\cdot)$ satisfies the following properties:

(1) $\gamma(x) \geq 0$ and $\gamma(x) = 0 \;\Leftrightarrow\; x = 0$;
(2) $\gamma(tx) = t\gamma(x)$ for any $t \geq 0$;
(3) $\gamma(x + y) \leq \gamma(x) + \gamma(y)$ for any x and $y \in R^2$.

It follows from (2) and (3) that any gauge $\gamma(x)$ is a convex function of x. The distance measuring function can be derived from a gauge by

$$D(x, y) = \gamma(y - x). \tag{5}$$

When B is symmetric around the origin, according to the definition of (3) we have

$$\gamma(-x) = \gamma(x), \quad \forall\, x \in R^2. \tag{6}$$

It follows that

$$D(x, y) = D(y, x), \tag{7}$$

which means the distance measured by $\gamma(\cdot)$ is symmetric. On the contrary, when B is not symmetric around the origin, (6) does not hold any more and the distance measured by $\gamma(\cdot)$ is asymmetric. Thus, with different compact convex sets used as unit balls, various gauges (symmetric or asymmetric) can be employed to measure distances according to the requirements of practical applications.

With the consideration of the gauge for measuring distances and locational constraints on facilities, the constrained multi-source Weber problem (abbreviated as CMSWP) is formulated as follows:

$$\begin{aligned} \text{CMSWP: } \min \quad & \sum_{j=1}^{n} s_j \min_{i=1,\ldots,m} \gamma(x_i - a_j) \\ \text{s.t.} \quad & x_i \in X_i \subseteq R^2,\ i = 1, 2, \ldots, m, \end{aligned} \tag{8}$$

where

(1) a_j, x_i, s_j are defined as the same as those in MSWP (2);
(2) X_i denotes the locational constraints for the ith facility and we assume X_i, $i = 1, \ldots, m$, are closed convex subsets of R^2;
(3) $\gamma(\cdot)$ is the gauge generated by a given compact convex set $B \subset R^2$.

It is clear that the CMSWP model is the more general and practical case of MSWP (1) and MSWP′ (2). Notice that the locational constraint X_i can also be chosen as R^2. If all X_i, $i = 1, \ldots, m$, are R^2, the CMSWP (8) is a unconstrained location problem; otherwise, (8) is a constrained one. Thus, both constrained and unconstrained problems can be considered in the CMSWP (8) by using various locational constraints in practice. If $m = 1$ and $\gamma(\cdot)$ is the Euclidean distance, the CMSWP (8) also includes two well-known facility location problems as its special cases: it becomes the classical single-source Weber problem when $X = R^2$ and it becomes a constrained single-source Weber problem when $X \subset R^2$.

Proposition 1 *The CMSWP (8) is nonconvex and NP-hard.*

Proof Since MSWP is a special problem of CMSWP and note that MSWP is a nonconvex and NP-hard problem, it follows that CMSWP is nonconvex and NP-hard. □

To solve the classical multi-source Weber problem, the location–allocation type algorithms are particularly popular for overcoming the difficulty caused by the nonconvexity and the NP-hardness. Hence, in this paper, we are interested in applying a location–allocation algorithm in the spirit of Cooper's work to solve the CMSWP (8). Consequently, some location subproblems and allocation subproblems occur in location phase and allocation phase, respectively.

To illustrate it, let $\mathcal{N} = \{1, 2, \ldots, n\}$ and $\widetilde{A} = \{a_j : j \in \mathcal{N}\}$ denote the set of locations of all the customers and $P^k = \{A_1^k, A_2^k, \ldots, A_m^k\}$ with $\cup_{i=1}^m A_i^k = \widetilde{A}$ and $A_i^k \cap A_{i'}^k = \emptyset$ (for $i \neq i'$) denote the disjoint partition of $\widetilde{A}$ at the kth iteration (where each A_i^k, $i = 1, \ldots, m$, is called as a cluster). At the $(k+1)$th iteration, the location phase finds the candidates of locations of facilities for the given partition P^k. Let $n_i^k = Card(A_i^k)$, i.e., the number of elements in A_i^k, and rearrange the customers in A_i^k as a_j^i, $j = 1, 2, \ldots, n_i^k$, i.e., $A_i^k = \{a_1^i, a_2^i, \ldots, a_{n_i^k}^i\}$. In order to get the locations of facilities for the partition P^k, we solve the following m constrained single-source Weber problems (abbreviated as CSSWP)

$$\begin{aligned} \text{CSSWP:} \quad & \min \;\; C_i(x_i) = \sum_{j=1}^{n_i^k} s_j \; \gamma(x_i - a_j^i) \\ & \text{s.t.} \quad x_i \in X_i \subseteq R^2, \end{aligned} \tag{9}$$

for each i, $i = 1, 2, \ldots, m$. Recall that the classical single-source Weber problem (SSWP) in facility location has the following formulation

$$\text{SSWP:} \min_{x \in R^2} \;\; C(x) = \sum_{j=1}^{d} s_j \; \|x - a_j\|, \tag{10}$$

where $\|\cdot\|$ is the Euclidean distance. It is clear that SSWP (10) can be regarded as a special problem of (9) in the sense that the gauge is chosen as Euclidean norm and all locational constraints X_i are set as R^2.

After the location phase, the allocation phase involves an allocation or a partition, which depends on the x_i^{k+1} $(i = 1, 2, \ldots, m)$ generated by solving (9). More specifically, the set of customers is partitioned newly according to the nearest center reclassification (NCR) method (see e.g. [17]), i.e., the new partition $\{A_1^{k+1}, A_2^{k+1}, \ldots, A_m^{k+1}\}$ is generated so that x_i^{k+1} is the nearest facility for each customer in A_i^{k+1}.

Then a central task of the location–allocation algorithm is to solve the involved CSSWP (9) in the location phase efficiently. Contrary to CMSWP (8), CSSWP (9) has the convexity property which is revealed by the following proposition.

Proposition 2 *The CSSWP (9) in location phase is convex.*

Proof Note that all constraints X_j, $j = 1, \ldots, m$, are closed convex sets and $\gamma(\cdot)$ is a convex function, thus it follows directly that the CSSWP is a convex problem. □

While CSSWP (9) is convex, the nondifferentiability unfortunately occurs, e.g., when a new facility coincides with one customer, which is usually called singularity. This unexpected characteristic of singularity is also shared by SSWP since SSWP is a special case of CSSWP. As an important facility location problem, SSWP has been well studied and many approaches have been suggested for solving it. The first attractive contribution to this aspect, denoted by Weiszfeld method, was due to [45]. Weiszfeld method was later adopted by Cooper algorithm [10] in the location phase to solve the location subproblems. The so-called Newton-Bracketing (NB) method was utilized to solve the SSWP in [29] and then with NB method used in location phase, the Cooper-NB algorithm was proposed for solving MSWP. Both the Weiszfeld method and the NB method, however, need to use the gradients of $\|x_i - a_j\|$ in the iteration, thus their implementations may terminate unexpectedly when the singularity happens (which is unavoidable and uncontrollable). How to improve the Weiszfeld method and the NB method in the the case of singularity and make them computationally preferable become the main challenges in this study and still deserve more extensive investigations, see, e.g., [4,7,28,37,43].

In this paper, an alternating direction method of multipliers (ADMM) type method is proposed for solving CSSWP. The ADMM proposed originally in [18,19] is a popular method for solving convex separable problems and it is closely related to Douglas-Rachford splitting method [14] which was well known previously in the research area of partial differential equation. The reader is referred to [2,12,20,21,41] and references cited therein for the comprehensive review of ADMM and its novel applications arising in various areas. For the

CSSWP, the proposed ADMM-type method can deal with the undesirable singularity and successfully get the optimal solution even in the case of singularity, which will be guaranteed by theoretical analysis and verified by numerical experiments later.

A new location–allocation algorithm, with the proposed ADMM-type method used in the location phase and the NCR rule used in the allocation phase, is then developed for CMSWP. Its local convergence is also proved. Preliminary numerical results are reported, which demonstrate the effectiveness of the ADMM-type method and new location–allocation algorithm.

3 ADMM-type method for the location subproblem

This section aims at solving the involved CSSWP (9) in the location phase by an ADMM-type method. At first, we reformulate the CSSWP (9) to make it separable. When applying the ADMM to solve the reformulated location subproblem, x, y and λ are updated alternatively. The main computational challenge comes from the y-subproblem. The examples of y-subproblems with l_1-norm, l_2-norm and one asymmetric distance measure are analyzed and then a uniform framework is derived to provide a unified closed-form solution. At last, we prove the global convergence of proposed ADMM-type method for the reformulated CSSWP under the simple framework of variational inequality.

3.1 ADMM-type method for the problems with different gauges

For convenience and succinctness, with the assumption that A_i^k contains d customers, i.e., $n_i^k = d$, throughout this section we ignore some superscripts and subscripts in (9) and consider the simplified model of CSSWP(9) without confusion:

$$\begin{aligned} \min \quad & \sum_{j=1}^{d} s_j\, \gamma(x - a_j) \\ \text{s.t.} \quad & x \in X. \end{aligned} \tag{11}$$

Introducing an auxiliary variable y to the problem (11), it can be rewritten as

$$\begin{aligned} \text{CSSWP}': \ \min \quad & \sum_{j=1}^{d} s_j\, \gamma(y_j) \\ \text{s.t.} \quad & x - a_j = y_j, \quad j = 1, \ldots, d, \\ & x \in X. \end{aligned} \tag{12}$$

According to the property of gauge and the convexity of X, it follows that the problem (12) is a convex optimization problem with linear constraint. Let $y = (y_1, \ldots, y_d)^T$ and introduce a Lagrange multiplier $\lambda = (\lambda_1, \ldots, \lambda_d)^T$ to the linear constraint, then the augmented Lagrangian function of (12) is

$$\mathcal{L}_\beta(x, y, \lambda) = \sum_{j=1}^{d} s_j\, \gamma(y_j) - \sum_{j=1}^{d} \lambda_j^T\, (x - a_j - y_j) + \frac{\beta}{2} \sum_{j=1}^{d} \|x - a_j - y_j\|^2, \tag{13}$$

where $\beta > 0$ is a penalty parameter for the linear constraints, and $\|\cdot\|$ indicates l_2-norm throughout the paper.

The classical augmented Lagrangian method (ALM) [24,40] can be applied for (12) directly. For a given β, the ALM generates the new iterate $(x^{k+1}, y^{k+1}, \lambda^{k+1})$ from (x^k, y^k, λ^k) via the following iterative scheme.

$$\begin{cases} (x^{k+1}, y^{k+1}) = \text{Argmin}_{x\in X, y} \ \mathcal{L}_\beta(x, y, \lambda^k) & (14a) \\ \lambda_j^{k+1} = \lambda_j^k - \beta(x^{k+1} - a_j - y_j^{k+1}), \ j = 1, \ldots, d. & (14b) \end{cases}$$

The ALM has been well studied in the literature, which has become a benchmark method for constrained optimization. It has been shown that ALM is essentially the application of proximal point algorithm (PPA) to the dual problem, and thus the convergence of ALM is well known in terms of the dual variable. The detailed convergence results of ALM can be referred to, e.g., [1,31,36].

The direct application of ALM leads to the expensive tasks of minimizing x and y simultaneously in (14a) of each iteration. It should be pointed out that the CSSWP′ (12) has favorable structure in the sense that its objective function is separable into d individual functions, which is usually beneficial to developing efficient methods. Unfortunately, when ALM is applied to (12) directly, the favorable structure is ignored.

To exploit the favorable structure of separable problem, the alternating direction method of multipliers can be applied to solve the CSSWP′ (12). The spirit of ADMM is to decompose the subproblem of ALM (e.g. the (x, y)-subproblem in (14a)) into several subproblems in the Gauss-Seidel fashion and then solve these subproblems individually at each iteration. The decomposition strategy of ADMM makes it possible to exploit these favorable structure. In addition, Gauss-Seidel-type methods generally have faster convergence rate than the corresponding Jacobian-type methods in that they absorb the latest information as soon as possible.

Now we concentrate on presenting an efficient ADMM-type method to solve the CSSWP′ (12), and thus solve the CSSWP (9). When the ADMM is applied to (12), it results in the following iterative scheme.

$$\begin{cases} x^{k+1} = \text{Argmin}_{x\in X} \ \mathcal{L}_\beta(x, y^k, \lambda^k), & (15a) \\ y^{k+1} = \text{Argmin}_{y} \ \mathcal{L}_\beta(x^{k+1}, y, \lambda^k), & (15b) \\ \lambda_j^{k+1} = \lambda_j^k - \beta(x^{k+1} - a_j - y_j^{k+1}), \ j = 1, \ldots, d. & (15c) \end{cases}$$

In order to accelerate the convergence of ADMM, the penalty parameter β can vary according to certain self-adaptive strategies, see e.g. [12,23]. For the simplification of our discussion, the parameter β is set to be a constant positive number here. Next we will discuss how to solve the subproblems of the ADMM (15) for the CSSWP′ (12).

(I) The variable x^{k+1}. According to (13) and (15a), x^{k+1} can be obtained by solving

$$\begin{aligned} x^{k+1} &= \text{Argmin}_{x\in X} \ \frac{\beta}{2}\sum_{j=1}^{d} \|x - a_j - y_j^k\|^2 - \sum_{j=1}^{d} \lambda_j^{k\,T} (x - a_j - y_j^k) \\ &= \text{Argmin}_{x\in X} \ \sum_{j=1}^{d} \|x - a_j - y_j^k\|^2 - \frac{2}{\beta}\left(\sum_{j=1}^{d} \lambda_j^k\right)^T x \\ &= \text{Argmin}_{x\in X} \ d\|x\|^2 - 2\left(\sum_{j=1}^{d}(a_j + y_j^k)\right)^T x - \frac{2}{\beta}\left(\sum_{j=1}^{d} \lambda_j^k\right)^T x \end{aligned}$$

$$= \operatorname{Argmin}_{x\in X} \left\| x - \frac{1}{d}\sum_{j=1}^{d}\left(a_j + y_j^k + \frac{1}{\beta}\lambda_j^k\right)\right\|^2. \tag{16}$$

It is obvious that the x-subproblem (15a) has the closed-form solution

$$x^{k+1} = P_X\left(\frac{1}{d}\sum_{j=1}^{d}\left(a_j + y_j^k + \frac{1}{\beta}\lambda_j^k\right)\right), \tag{17}$$

where $P_X(v)$ is the projection of v onto the convex set X, i.e.,

$$P_X(v) = \operatorname{Argmin}\,\{\|u - v\|_2 \mid u \in X\}.$$

(II) The variable y^{k+1}. y^{k+1} is the solution to the following problem

$$y^{k+1} = \operatorname{Argmin}_y \sum_{j=1}^{d} s_j\,\gamma(y_j) + \sum_{j=1}^{d}\lambda_j^{k\,T}\,y_j + \frac{\beta}{2}\sum_{j=1}^{d}\|y_j - x^{k+1} + a_j\|^2. \tag{18}$$

Due to the favorable structure of the CSSWP′ (12), it can be separated into d subproblems with each subproblem for one variable, i.e.,

$$\begin{aligned} y_j^{k+1} &= \operatorname{Argmin}_{y_j}\ s_j\,\gamma(y_j) + \lambda_j^{k\,T}\,y_j + \frac{\beta}{2}\|y_j - x^{k+1} + a_j\|^2 \\ &= \operatorname{Argmin}_{y_j}\ s_j\,\gamma(y_j) + \frac{\beta}{2}\|y_j - x^{k+1} + a_j + \frac{1}{\beta}\lambda_j^k\|^2 \\ &= \operatorname{Argmin}_{y_j}\ \omega\,\gamma(y_j) + \frac{1}{2}\|y_j - p\|^2, \end{aligned} \tag{19}$$

where $p = x^{k+1} - a_j - \frac{1}{\beta}\lambda_j^k$ and $\omega = \frac{s_j}{\beta}$. Then it turns to solve d subproblems separately.

The norms l_1, l_2 and l_∞ have been well used in the literature of facility location to measure symmetric distances. In this paper, we consider both symmetric distance and asymmetric distance by the adoption of gauge. In the following we will take the examples of l_1-norm, l_2-norm and one asymmetric distance measure to illustrate how to solve the y_j-subproblem at first and then unify the solutions in a uniform framework.

Case 1 $\gamma(\cdot) = l_1(\cdot)$. The y_j-subproblem (19) turns to be

$$y_j^{k+1} = \operatorname{Argmin}_{y_j}\ \omega\|y_j\|_1 + \frac{1}{2}\|y_j - p\|^2. \tag{20}$$

We state the following proposition to give the solution to (20).

Proposition 3 *For an arbitrary $\omega > 0$ and $p \in R^2$, the solution to the following optimization problem*

$$\min_{y\in R^2}\left\{\omega\|y\|_1 + \frac{1}{2}\|y - p\|^2\right\}$$

is given by $S_\omega(p) \in R^2$ with the element

$$(S_\omega(p))_i := \max\{abs(p(i)) - \omega, 0\}\cdot sign(p(i)),\quad i = 1, 2,$$

where $abs(\cdot)$ and $sign(\cdot)$ are the absolute-value function and the sign function respectively.

Proof See, e.g., [5,48] and thus we omit it. □

Based on Proposition 3, the y_j-subproblem (20) with l_1 distance measure has the closed-form solution $S_\omega(p)$.

Case 2 $\gamma(\cdot) = l_2(\cdot)$. Then the y_j-subproblem has the following formulation

$$y_j^{k+1} = \mathrm{Argmin}_{y_j} \quad \omega\|y_j\| + \frac{1}{2}\|y_j - p\|^2. \tag{21}$$

Its solution can be characterized by the following proposition.

Proposition 4 *For an arbitrary $\omega > 0$ and $p \in R^2$, the solution to the problem (21) is given by*

$$y_j^{k+1} = \begin{cases} \frac{\|p\|-\omega}{\|p\|}p, & \|p\| \geq \omega, \\ 0, & otherwise. \end{cases} \tag{22}$$

Proof Denote $y_j = (f_1, f_2)^T$ and $p = (p_1, p_2)^T$, then (21) can be rewritten as

$$y_j^{k+1} = \mathrm{Argmin} \quad \left\{\omega\sqrt{f_1^2 + f_2^2} + \frac{1}{2}\left((f_1 - p_1)^2 + (f_2 - p_2)^2\right) \mid (f_1, f_2) \in R^2\right\}. \tag{23}$$

Let $f_1 = t\sin\alpha$ and $f_2 = t\cos\alpha$ with $t \geq 0$ and $\alpha \in [0, 2\pi]$. Substituting f_1 and f_2 into (23), we get

$$\begin{aligned} y_j^{k+1} &= \mathrm{Argmin} \ \left\{\omega\sqrt{f_1^2 + f_2^2} + \frac{1}{2}(f_1^2 + f_2^2 - 2f_1p_1 - 2f_2p_2) \mid (f_1, f_2) \in R^2\right\} \\ &= \mathrm{Argmin} \ \left\{\omega t + \frac{1}{2}(t^2 - 2tp_1\sin\alpha - 2tp_2\cos\alpha) \mid t \geq 0, \alpha \in [0, 2\pi]\right\} \\ &= \mathrm{Argmin} \ \left\{\frac{1}{2}t^2 - [\|p\|\sin(\alpha + \theta) - \omega]t \mid t \geq 0, \alpha \in [0, 2\pi]\right\}, \end{aligned} \tag{24}$$

where $\cos\theta = p_1/\|p\|$ and $\sin\theta = p_2/\|p\|$. Let $q = \|p\|\sin(\alpha + \theta) - \omega$, then due to $\alpha \in [0, 2\pi]$ we have $q \in [-\|p\| - \omega, \|p\| - \omega]$. Thus, (24) can be written as

$$\begin{aligned} y_j^{k+1} &= \mathrm{Argmin} \ \left\{\frac{1}{2}t^2 - qt \mid t \geq 0, q \in [-\|p\| - \omega, \|p\| - \omega]\right\} \\ &= \mathrm{Argmin} \ \left\{\frac{1}{2}(t - q)^2 - \frac{1}{2}q^2 \mid t \geq 0, q \in [-\|p\| - \omega, \|p\| - \omega]\right\}. \end{aligned} \tag{25}$$

There are two possibilities as follows.

(a) $||p|| < \omega$. It follows that $q < 0$. No matter which q is chosen in $[-\|p\| - \omega, \|p\| - \omega]$, (25) always achieves its optimal value 0 at the point of $t = 0$;

(b) $||p|| \geq \omega$. Then we have $0 \in [-\|p\| - \omega, \|p\| - \omega]$. For $q \in [-\|p\| - \omega, 0]$, (25) always achieves the optimal value 0 at $t = 0$. On the other hand, for a given $q \in [0, \|p\| - \omega]$, (25) gets the optimal value $-\frac{q^2}{2}$ (note that $-\frac{q^2}{2} \leq 0$) at $t = q$. Furthermore, when $q \in [0, \|p\| - \omega]$, $-\frac{q^2}{2}$ is a monotonically decreasing function of q. Thus, for the case of $||p|| \geq \omega$, the optimal solution to (25) is $t = \|p\| - \omega$, which also means $\sin(\alpha + \theta) = 1$. Since θ is a given vector determined by p, α can be chosen as $\alpha = \frac{\pi}{2} - \theta$. Thus, the optimal solution to (24) is $y_j^{k+1} = (t\sin\alpha, t\cos\alpha)^T = (t\cos\theta, t\sin\theta)^T = (\|p\| - \omega)(p_1/\|p\|, p_2/\|p\|)^T = \frac{\|p\|-\omega}{\|p\|}p$.

Hence, (22) can be obtained from (a) and (b) and the proof is complete. □

Case 3 $\gamma(\cdot)$ with the unit ball asymmetric around the origin. Such gauge will result in the asymmetric distance measure. To show how to solve the y_j-subproblem with asymmetric distance, we take an example of $\gamma(\cdot)$ with the unit ball defined by

$$B = \left\{ (x_1, x_2)^T \in R^2 \mid \frac{(x_1 - 1)^2}{2} + x_2^2 \leq 1 \right\}. \tag{26}$$

Proposition 5 *Let $\gamma(\cdot)$ be defined by (3) with the unit ball (26). For an arbitrary $\omega > 0$ and $p \in R^2$, the solution to the optimization problem (19) is given by*

$$y_j^{k+1} = \begin{cases} \frac{\|\hat{p}\| - \sqrt{2}\omega}{\|\hat{p}\|} \hat{p}, & \|\hat{p}\| \geq \sqrt{2}\omega, \\ 0, & otherwise, \end{cases} \tag{27}$$

where $\hat{p} = p + (\omega, 0)^T$.

Proof Denote $y_j = (f_1, f_2)^T$, thus for $\nu > 0$ we have $y_j/\nu = (f_1/\nu, f_2/\nu)^T$. According to the definition of gauge,

$$\gamma(y_j) = \inf \left\{ \nu > 0 \mid \frac{y_j}{\nu} \in B \right\}. \tag{28}$$

The infimum of (28) is achieved when $\frac{y_j}{\nu} \in \mathbf{bd}(B)$, where $\mathbf{bd}(B)$ means the boundary of B. Thus, substituting $y_j/\nu = (f_1/\nu, f_2/\nu)^T$ into the boundary of B, it follows

$$(f_1/\nu - 1)^2 + 2f_2^2/\nu^2 = 2,$$

which is equivalent to

$$\nu^2 + 2f_1\nu - (f_1^2 + 2f_2^2) = 0. \tag{29}$$

Note that ν is nonnegative, thus the solution to (29) is given by

$$\nu = \sqrt{2(f_1^2 + f_2^2)} - f_1. \tag{30}$$

Combining (28) with (30), we get

$$\gamma(y_j) = \sqrt{2(f_1^2 + f_2^2)} - f_1. \tag{31}$$

Set $f_1 = t \sin\alpha$ and $f_2 = t\cos\alpha$ with $t \geq 0$ and $\alpha \in [0, 2\pi]$. Substituting (31) into the y_j-subproblem (19), it turns out to be

$$\begin{aligned} y_j^{k+1} &= \text{Argmin } \left\{ \omega \left(\sqrt{2(f_1^2 + f_2^2)} - f_1 \right) + \frac{1}{2}\left((f_1 - p_1)^2 + (f_2 - p_2)^2 \right) \mid (f_1, f_2) \in R^2 \right\} \\ &= \text{Argmin } \left\{ \omega(\sqrt{2}t - t\sin\alpha) + \frac{1}{2}(t^2 - 2tp_1 \sin\alpha - 2tp_2\cos\alpha) | t \geq 0, \alpha \in [0, 2\pi] \right\} \\ &= \text{Argmin } \left\{ \frac{1}{2}t^2 - \left((p_1 + \omega)\sin\alpha + p_2\cos\alpha - \sqrt{2}\omega \right) t \mid t \geq 0, \alpha \in [0, 2\pi] \right\} \\ &= \text{Argmin } \left\{ \frac{1}{2}t^2 - \left(\|\hat{p}\| \sin(\alpha + \theta) - \sqrt{2}\omega \right) t \mid t \geq 0, \alpha \in [0, 2\pi] \right\} \\ &= \text{Argmin } \left\{ \frac{1}{2}t^2 - \hat{q}t \mid t \geq 0, \hat{q} \in [-\|\hat{p}\| - \sqrt{2}\omega, \|\hat{p}\| - \sqrt{2}\omega] \right\} \\ &= \text{Argmin } \left\{ \frac{1}{2}(t - \hat{q})^2 - \frac{1}{2}\hat{q}^2 \mid t \geq 0, \hat{q} \in [-\|\hat{p}\| - \sqrt{2}\omega, \|\hat{p}\| - \sqrt{2}\omega] \right\}, \end{aligned} \tag{32}$$

where $\cos\theta = \hat{p}_1/\|\hat{p}\|$, $\sin\theta = \hat{p}_2/\|\hat{p}\|$, and $\hat{q} = \|\hat{p}\| \sin(\alpha+\theta) - \sqrt{2}\omega$.

With the argument similar to (a) and (b) in **Case 2**, Proposition 5 can be proved and thus the proof is complete. □

Case 1–3 illustrate how to solve the y_j-subproblem (19) for some particular gauges including the symmetric distance measure (l_1-norm and l_2-norm) and the asymmetric distance measure (the gauge defined by (26)). In the following, we provide a unified form to characterize the solution to (19) for the general gauge based on the Moreau's decomposition. For more details about Moreau's decomposition, please refer to, e.g., [9].

Proposition 6 *For an arbitrary $\omega > 0$, $p \in R^2$ and any gauge $\gamma(\cdot)$, the solution to the y_j-subproblem (19) is given as*

$$y_j^{k+1} = p - \omega P_{\gamma^d(u)\le 1}(p/\omega), \tag{33}$$

where $P_{\gamma^d(u)\le 1}(\cdot)$ is the projection of a vector onto the unit ball $\gamma^d(u) \le 1$.

Proof Let $\varphi : R^2 \to R$ be a function and Prox_φ be the proximity operator of φ, i.e.,

$$\text{Prox}_\varphi(p) = \text{Argmin}\left\{\varphi(y) + \frac{1}{2}\|y-p\|^2 \mid y \in R^2\right\}.$$

According to (19), the solution to the y_j-subproblem is

$$y_j^{k+1} = \text{Prox}_{\omega\gamma}(p). \tag{34}$$

Applying the Moreau's decomposition, we have

$$p = \text{Prox}_{\omega\gamma}(p) + \omega\text{Prox}_{\gamma^*/\omega}(p/\omega), \tag{35}$$

where $\gamma^*(\cdot)$ is the conjugate of $\gamma(\cdot)$. Then the following equation holds,

$$y_j^{k+1} = p - \omega\text{Prox}_{\gamma^*/\omega}(p/\omega). \tag{36}$$

According to the definition of proximity operator, the second term $\text{Prox}_{\gamma^*/\omega}(p/\omega)$ of (36) can be rewritten as

$$\text{Prox}_{\gamma^*/\omega}(p/\omega) = \text{Argmin}\left\{\gamma^*(u) + \frac{\omega}{2}\|u - p/\omega\|^2 \mid u \in R^2\right\}. \tag{37}$$

For the gauge $\gamma(\cdot)$, it is easy to verify that its conjugate $\gamma^*(\cdot)$ is

$$\gamma^*(u) = \begin{cases} 0, & \gamma^d(u) \le 1, \\ +\infty, & otherwise. \end{cases} \tag{38}$$

Therefore, (37) and (38) reveals that

$$\text{Prox}_{\gamma^*/\omega}(p/\omega) = P_{\gamma^d(u)\le 1}(p/\omega). \tag{39}$$

Hence, the equation (33) can be obtained from (36) and (39) directly and the proof is complete. □

For general gauge, Proposition 6 provides a unified closed form to calculate the solution to the y_j-subproblem (19).

Remark 1 For example, we can apply Proposition 6 to solve the y_j-subproblem of the case $\gamma(\cdot) = l_\infty(\cdot)$. That is, consider

$$y_j^{k+1} = \text{Argmin}_{y_j} \ \ \omega\|y_j\|_\infty + \frac{1}{2}\|y_j - p\|^2. \tag{40}$$

According to the result (33), the solution has the following formulation

$$y_j^{k+1} = p - \omega P_{\|\cdot\|_1 \le 1}\,(p/\omega). \tag{41}$$

For $u, v \in R^2$, define $\max(u, v) := (\max(u_1, v_1), \max(u_2, v_2))^T$ and $\min(u, v) := (\min(u_1, v_1), \min(u_2, v_2))^T$, then $P_{\|\cdot\|_1 \le 1}\,(p/\omega)$ can be calculated by

$$P_{\|\cdot\|_1 \le 1}\,(p/\omega) = Q^{-1} \max\left(-\frac{\sqrt{2}}{2}e,\ \min\left(Q\ p/\omega,\ \frac{\sqrt{2}}{2}e\right)\right), \tag{42}$$

where $e = (1, 1)^T$ and

$$Q = \begin{pmatrix} \cos(\pi/4) & -\sin(\pi/4) \\ \sin(\pi/4) & \cos(\pi/4) \end{pmatrix}.$$

Combining (41) with (42), the y_j-subproblem of the case $\gamma(\cdot) = l_\infty(\cdot)$ has the closed-form solution in this unified framework.

(III) The Lagrange multiplier (dual variable) λ^{k+1}. The multiplier λ_j^{k+1}, $j = 1, \ldots, d$, are updated trivially according to (15c).

Based on previous discussion on the iterative scheme and the corresponding subproblems, we are ready to present our proposed ADMM-type method for solving the CSSWP′ (12) as follows.

Algorithm 1 The proposed ADMM-type method for (12)

Given the parameter $\beta > 0$ and initial iterate $w^0 = (x^0, y^0, \lambda^0)$.
For $k = 0, 1, 2 \ldots$, **do**:

$$\left\{\begin{aligned} x^{k+1} &= P_X\left(\frac{1}{d}\sum_{j=1}^{d}\left(a_j + y_j^k + \frac{1}{\beta}\lambda_j^k\right)\right), && \text{(43a)} \\ y_j^{k+1} &= p_j - \omega_j P_{\gamma^d(u)\le 1}\,(p_j/\omega_j), \quad j = 1, \ldots, d, && \text{(43b)} \\ \lambda_j^{k+1} &= \lambda_j^k - \beta(x^{k+1} - a_j - y_j^{k+1}), \quad j = 1, \ldots, d, && \text{(43c)} \end{aligned}\right.$$

where $p_j = x^{k+1} - a_j - \frac{1}{\beta}\lambda_j^k$ and $\omega_j = \frac{s_j}{\beta}$.

3.2 Convergence of proposed ADMM-type method

One essential concern with regard to the proposed ADMM-type method is its convergence. The proposed method is the application of classical ADMM for the convex problem CSSWP′ with $d+1$ separable blocks $(x, y_1, \ldots, y_d)$. A recent paper [8] reveals that the convergence of classical ADMM for the generic m-block separable convex problem ($m \ge 3$) is not guaranteed, even though its numerical efficiency has been verified by many satisfactory applications (e.g. [38,42]). Fortunately, thanks to the favorable structure of the CSSWP′, solving the y_j-subproblem ($j = 1, \ldots, d$) separately is indeed equivalent to solving the

y-subproblem (18). This observation is quite important because it implies that the proposed ADMM-type method can be regarded as the ADMM for two-block separable problem (x is one block and $y = (y_1, \ldots, y_d)$ is the other block). The convergence of ADMM for two-block separable convex optimization problem has been well studied, see, e.g., [13,16,30]. In this paper, under the framework of variational inequality (VI) we give one simple convergence proof of the ADMM for two-block problem by following the results in [22].

Consider general problem of the CSSWP′ (12) given as follows

$$\begin{aligned} \min_{x,y} \quad & F(x) + G(y) \\ \text{s.t.} \quad & Ax + By = b, \\ & x \in X, \; y \in Y, \end{aligned} \tag{44}$$

where $F : R^e \to R$ and $G : R^l \to R$ are convex functions, $X \in R^e$ and $Y \in R^l$ are closed and convex sets and $b \in R^\sigma$. When $F(x) = 0$, $G(y) = \sum_{j=1}^d s_j \gamma(y_j)$, $A = (I_2, \ldots, I_2)^T$, $B = -I_{2d}$, and $b = (a_1, \ldots, a_d)^T$, (44) is reduced to the CSSWP′ (12).

Let $\mathcal{L}_\beta(x, y, \lambda)$ be the augmented Lagrangian function of (44), i.e.,

$$\mathcal{L}_\beta(x, y, \lambda) = F(x) + G(y) - \lambda^T(Ax + By - b) + \frac{\beta}{2}\|Ax + By - b\|^2, \tag{45}$$

where $\beta > 0$ is a penalty parameter. Applying the classical ADMM for (44), we can get the following iterative scheme

$$\left\{ \begin{aligned} x^{k+1} &= \text{Argmin}_{x \in X} \;\; F(x) - \lambda^{k\,T} Ax + \frac{\beta}{2}\|Ax + By^k - b\|^2, && \text{(46a)} \\ y^{k+1} &= \text{Argmin}_{y \in Y} \;\; G(y) - \lambda^{k\,T} By + \frac{\beta}{2}\|Ax^{k+1} + By - b\|^2, && \text{(46b)} \\ \lambda^{k+1} &= \lambda^k - \tau\beta(Ax^{k+1} + By^{k+1} - b). && \text{(46c)} \end{aligned} \right.$$

For the iterate $w^k = (x^k, y^k, \lambda^k)$ generated by the ADMM (46), we have the following theoretical results which is useful for the convergence analysis.

Theorem 1 *Assume the solution set of the general problem* (44) *is nonempty. Then for any given* $\beta > 0$, $\tau \in (0, (1 + \sqrt{5})/2)$ *and* $w^0 = (x^0, y^0, \lambda^0)$, *the sequence* $\{w^k\}$ *generated by the ADMM (46) satisfies*

$$\lim_{k\to\infty} A(x^k - x^*) = 0 \quad \textit{and} \quad \lim_{k\to\infty} B(y^k - y^*) = 0, \tag{47}$$

where (x^*, y^*) *is the optimal solution to* (44).

Proof The optimal condition of the general problem (44) can be characterized by the following VI problem: finding $w^* = (x^*, y^*, \lambda^*) \in X \times Y \times R^\sigma$ such that

$$\left\{ \begin{array}{lll} (x - x^*)^T (f(x^*) - A^T\lambda^*) & \geq 0, & \forall\, x \in X, \\ (y - y^*)^T (g(y^*) - B^T\lambda^*) & \geq 0, & \forall\, y \in Y, \\ (\lambda - \lambda^*)^T (Ax^* + By^* - b) & \geq 0, & \forall\, \lambda \in R^\sigma, \end{array} \right. \tag{48}$$

where $f(x)$ and $g(y)$ are the gradients (or subgradients) of $F(x)$ and $G(y)$, respectively. Since the solution set of (44) is nonempty, the solution set of (48) is also nonempty.

For any given $\beta > 0$, $\tau \in (0, (1 + \sqrt{5})/2)$ and $w^0 = (x^0, y^0, \lambda^0)$, the following iterative scheme in VI form is presented in [22] to solve (48)

$$\left\{ \begin{array}{l} (x - x^{k+1})^T \left(f(x^{k+1}) - A^T\left(\lambda^k - \beta(Ax^{k+1} + By^k - b)\right)\right) \geq 0, \;\; \forall\, x \in X, \\ (y - y^{k+1})^T \left(g(y^{k+1}) - B^T\left(\lambda^k - \beta(Ax^{k+1} + By^{k+1} - b)\right)\right) \geq 0, \;\; \forall\, y \in Y, \\ \lambda^{k+1} = \lambda^k - \tau\beta(Ax^{k+1} + By^{k+1} - b). \end{array} \right. \tag{49}$$

As is well known, the VI scheme (49) is equivalent to the ADMM scheme (46). Thus, the iterate generated by (49) is the same as that generated by (46) and both are denoted by $w^k = (x^k, y^k, \lambda^k)$ here. For the generated sequence, according to Theorem 3 in [22], we have the result

$$\lim_{k\to\infty} \left\| \begin{matrix} A(x^k - x^*) \\ B(y^k - y^*) \\ \lambda^k - \lambda^* \end{matrix} \right\|_G = 0, \tag{50}$$

where G is a positive definite matrix and $w^* = (x^*, y^*, \lambda^*)$ is a solution to the VI problem (48).

It follows that

$$\lim_{k\to\infty} \left\| \begin{matrix} A(x^k - x^*) \\ B(y^k - y^*) \\ \lambda^k - \lambda^* \end{matrix} \right\| = 0, \tag{51}$$

and thus

$$\lim_{k\to\infty} \left\| \begin{matrix} A(x^k - x^*) \\ B(y^k - y^*) \end{matrix} \right\| = 0. \tag{52}$$

On the other hand, due to the equivalence between the general problem (44) and the VI problem (48), (x^*, y^*) is also the optimal solution to (44). Combining it with (52), this theorem is thus derived and the proof is complete. □

Now we are ready to prove the global convergence of proposed ADMM-type method for solving the CSSWP′ (12).

Theorem 2 *For any given $\beta > 0$ and $w^0 = (x^0, y^0, \lambda^0)$, the sequence $\{w^k\}$ generated by the ADMM-type method converges to $w^* = (x^*, y^*, \lambda^*)$ where (x^*, y^*) is the optimal solution to the CSSWP′ (12).*

Proof It is obvious that the optimal value of CSSWP (11) is attained, and so is CSSWP′ (12). It follows that the solution set of CSSWP′ (12) is nonempty and thus the assumption of Theorem 1 is satisfied.

According to Theorem 1 (here τ is chosen as 1), the sequence $\{w^k\}$ generated by the ADMM-type method satisfies (47). It follows directly from (47) that

$$\lim_{k\to\infty} A^T A(x^k - x^*) = 0 \quad \text{and} \quad \lim_{k\to\infty} B^T B(y^k - y^*) = 0. \tag{53}$$

Note that for the CSSWP′ (12) the matrix A and B are given by $A = (I_2, \ldots, I_2)^T$ and $B = -I_{2d}$. Then both A and B are full column rank and thus $A^T A$ and $B^T B$ are nonsingular. Correspondingly, we can get

$$\lim_{k\to\infty} x^k = x^* \quad \text{and} \quad \lim_{k\to\infty} y^k = y^*. \tag{54}$$

In addition, according to (51) the sequence $\{\lambda^k\}$ satisfies

$$\lim_{k\to\infty} \lambda^k = \lambda^*. \tag{55}$$

Combining (54) with (55), it follows that $\{w^k\}$ is convergent to $w^* = (x^*, y^*, \lambda^*)$ where (x^*, y^*) is the optimal solution to the CSSWP′ (12). Thus, we complete the proof. □

Remark 2 According to Theorem 1, when a parameter $\tau \in (0, (1+\sqrt{5})/2)$ is included in the update scheme of Lagrange multiplier (43c), i.e.,

$$\lambda_j^{k+1} = \lambda_j^k - \tau\beta(x^{k+1} - a_j - y_j^{k+1}), \quad j = 1, \ldots, d, \tag{56}$$

the global convergence of ADMM method is still guaranteed. For simplification, τ is always chosen as 1 in our numerical experiments.

4 Location–Allocation-ADMM algorithm for CMSWP

In the spirit of Cooper's work, a new location–allocation algorithm, denoted as the location–allocation-ADMM algorithm (LAAA), is presented in this section for solving the CMSWP (8). Each iteration of LAAA mainly consists of the location phase and the allocation phase. The CSSWP (9) in the location phase is solved by the proposed ADMM-type method; while the update of the partition of customers is according to the nearest center reclassification (NCR) as in Cooper algorithm. Note that LAAA differs from Cooper algorithm and Cooper-NB algorithm in another sense that the locations of facilities are restricted into some nonempty closed convex sets and the distances are measured by the gauge, thus it is more applicable and practical. We will prove that the iterative sequence generated by the LAAA is convergent to a local optimal solution to the CMSWP.

4.1 LAAA

Recall that $\{A_1^k, A_2^k, \ldots, A_m^k\}$ denotes the disjoint partition of customers at the kth iteration. To simplify the discussion, we first introduce three terminologies as follows:

(1) *visited*. A partition of customers, $P = \{A_1, A_2, \ldots, A_m\}$, is said to be *visited* when for this partition we have solved the m related CSSWPs and determined the locations of m new facilities.
(2) *tied clusters*. Let $\{x_1, \ldots, x_m\}$ be the locations of facilities obtained in the location phase, which will lead to a partition of customers $P = \{A_1, A_2, \ldots, A_m\}$. For one customer a_j if there exist several clusters $A_{j_1}, \ldots, A_{j_q}$ $(q \geq 2)$ such that $\gamma(x_{j_1} - a_j) = \cdots = \gamma(x_{j_q} - a_j) = \min_{k=1,\ldots,m} \gamma(x_k - a_j)$ (this means that all of $x_{j_1} \ldots, x_{j_q}$ provide the minimal distance to a_j), $A_{j_1}, \ldots, A_{j_q}$ are called as *tied clusters* for customer j.
(3) *neighbor partition*. Let P be a partition of customers. For each customer a_j, if it has tied clusters with regard to $\{x_1, \ldots, x_m\}$, we reallocate it from current cluster to another tied cluster; otherwise, we keep it in current cluster. After we deal with all customers a_j $(j = 1, \ldots, n)$ in such a way, another partition P' is generated. If $P' \neq P$, P' is called as a *neighbor partition* of P.

Now we propose the location–allocation-ADMM algorithm for solving the CMSWP (8) as follows.

Algorithm 2 Location–Allocation-ADMM algorithm for CMSWP

Given an initial partition $P^0 = \{A_1^0, \ldots, A_m^0\}$ and initial location of facilities $x^0 = (x_1^0, \ldots, x_m^0)$.

Step 0. Set $k = 0$ and $t = 0$ (t records the number of reassignments in one iteration).
Step 1. Location Phase:
Solve the involved m CSSWPs (9) by using the ADMM-type method with the initial

iterate (x_i^k, y^k, λ^k) where $y_j^k = x - a_j$ for $a_j \in A_i^k$, and thus find the location of facility x_i^{k+1} for each cluster A_i^k, $i = 1, \ldots, m$.
Step 2. If $C_i(x_i^{k+1}) = C_i(x_i^k)$, set $x_i^{k+1} = x_i^k$. If $x_i^{k+1} = x_i^k$ for all $i = 1, \ldots, m$, go to Step 5.
Step 3. Allocation Phase:
Update the partition of customers in the spirit of NCR:
For $j = 1, 2, \ldots, n$, do:

$d_{ij} := \gamma(x_i^{k+1} - a_j)$ for $i = 1, \ldots, m$;
if $a_j \in A_h^k$ and $d_{lj} = \min_{i=1,\ldots,m;\ i \neq h}\{d_{ij}\} < d_{hj}$,
set $A_h^k = A_h^k \backslash \{a_j\},\ \ A_l^k = A_l^k \cup \{a_j\}$;
$t = t + 1$.

Step 4. If $t > 0$, set $P^{k+1} = P^k$, $k = k + 1$ and go to Step 1.
Step 5. If there exists one unvisited neighbor partition P' of P^k, set $P^{k+1} = P'$; otherwise, set $P^{k+1} = P^k$. Let $k = k + 1$, $t = 0$ and go to Step 1.

Remark 3 The allocation phase of LAAA, Step 3, is to allocate the customers to the facilities according to NCR and a partition of customers is generated. Besides Step 3, Step 5 also generates a partition of customers. Different from Step 3, however, an unvisited neighbor partition (P') of the current one is chosen in Step 5 for the next iteration.

Remark 4 A sequence $\{x^k, P^k\}$ will be generated in Algorithm 2 and we will prove in the next subsection that the infinite sequence $\{x^k\}$ is convergent to a local solution of CMSWP. However, notice that when there is no unvisited neighbor partition of P^k in Step 5 (as we will explain in the next subsection, this case will occur in the implementation of Algorithm 2), the iterates succeeding x^k will keep unchanged. According to the convergence property, it implies that x^k is actually a local solution to CMSWP, and thus Algorithm 2 can be terminated with finding the local solution x^k as this case occurs.

4.2 Convergence of LAAA

This section aims at analyzing the convergence of proposed LAAA (Algorithm 2). Some notations are first introduced as follows. We define an ordered pair (x, P), where $x = (x_1, \ldots, x_m)^T$ is the location of facilities and $P = \{A_1, A_2, \ldots, A_m\}$ is a partition of customers. We further define a function $\zeta(x, P)$ for (x, P) as follows:

$$\zeta(x, P) = \sum_{i=1}^{m} \sum_{j=1}^{n_i} s_j\ \gamma(x_i - a_j^i), \tag{57}$$

where $n_i = Card(A_i)$ and the customers in A_i are rearranged as a_j^i, $j = 1, 2, \ldots, n_i$. It is obvious that $\zeta(x, P)$ represents the objective functional value of the CMSWP (8) at (x, P).

Proposition 7 *$\zeta(x^{k+1}, P^{k+1}) \leq \zeta(x^k, P^k)$ and $\zeta(x^{k+1}, P^{k+1}) < \zeta(x^k, P^k)$ if $x^{k+1} \neq x^k$.*

Proof Since x_i^{k+1} is the optimal solution to the CSSWP (9) for the cluster A_i^k in the partition P^k, i.e., x_i^{k+1} is the solution to the following convex optimization problem

$$\begin{aligned} \min\ & \sum_{j=1}^{n_i^k} s_j\ \gamma(x_i - a_j^i) \\ \text{s.t.}\ & x_i \in X_i, \end{aligned} \tag{58}$$

then it follows that

$$\zeta(x^{k+1}, P^k) \le \zeta(x^k, P^k). \tag{59}$$

If the next partition P^{k+1} is generated by Step 4 (i.e., $t > 0$), based on the principle of NCR we have

$$\zeta(x^{k+1}, P^{k+1}) \le \zeta(x^{k+1}, P^k). \tag{60}$$

Otherwise, the next partition P^{k+1} will be generated by Step 5. In this case, P^{k+1} is a neighbor partition of P^k. Since the reassignment of a customer between two tied clusters does not change the distance, we can get

$$\zeta(x^{k+1}, P^{k+1}) = \zeta(x^{k+1}, P^k). \tag{61}$$

By Combining (59), (60) and (61), it follows that

$$\zeta(x^{k+1}, P^{k+1}) \le \zeta(x^k, P^k). \tag{62}$$

If $x^{k+1} \neq x^k$, there exist at least one i such that $x_i^{k+1} \neq x_i^k$. It means that x_i^k is not the solution to (58) but x_i^{k+1} is. Thus, we have

$$\zeta(x^{k+1}, P^k) < \zeta(x^k, P^k). \tag{63}$$

Together with (60) and (61) we can get

$$\zeta(x^{k+1}, P^{k+1}) < \zeta(x^k, P^k). \tag{64}$$

Thus, the proof is complete. □

Let SP be the set of all partitions of customers and $\{(x^k, P^k)\}$ denote the iterative sequence generated by LAAA with the initial iterate (x^0, P^0). According to Proposition 7, the following theorem about the convergence of $\{(x^k, P^k)\}$ can be proved.

Theorem 3 *The sequence $\{(x^k, P^k)\}$ generated by LAAA satisfies*

$$\lim_{k\to\infty} (x^k, P^k) = (x^*, P^*) \in X \times SP \quad and \quad \lim_{k\to\infty} \zeta(x^k, P^k) = \zeta(x^*, P^*). \tag{65}$$

Proof We first show that $x^{k+1} \neq x^k$ for all $k \in N$ is not true. It is proved by contradiction. If it is true, according to the monotonicity by Proposition 7 the objective functional values of the iterates in $\{(x^k, P^k)\}$ are strictly decreased. It follows that for any k, $P^k \neq P^e$, $e = 1, \ldots, k-1$ (otherwise, x^{k+1} will be equal to x^{e+1} and thus the objective functional values will not be strictly decreased). Accordingly, all partitions in the sequences are not identical. However, note a fact that the number of all partitions of customers is no more than m^n. This results in a contradiction with that all partitions are not identical. Therefore, $x^{k+1} \neq x^k$ will not be always true.

Hence, there exists some $K \in N$ such that $x^{K+1} = x^K$, i.e., the condition of Step 2 is satisfied. Thus the algorithm goes to Step 5. In Step 5 of LAAA, two cases are required to be considered for the partition P^K as follows.

Case a P^K has a neighbor partition, say P', that has not been visited. In this case, P' will be chosen as the next partition to continue the iterations of LAAA. Note that when such case occurs an unvisited partition will be visited. Recall again the total number of partitions is no more than m^n, thus this case will not occur any more after a certain number of iterations.

Case b All neighbor partitions of P^K have already been visited, or there is no neighbor partition for P^K. In this case, we have $P^{k+1} = P^k$, and thus the iterates succeeding (x^K, P^K) remain unchanged. It follows that the sequence $\{(x^k, P^k)\}$ is convergent to $(x^*, P^*) := (x^K, P^K) \in X \times SP$, i.e.,

$$\lim_{k\to\infty} (x^k, P^k) = (x^*, P^*) \in X \times SP. \tag{66}$$

Therefore, the first assertion is proved.

The second assertion can be derived directly from the first one since the function $\zeta(x, P)$ is continuous. □

Based on Theorem 3, it is easy to prove the local convergence of LAAA for the nonconvex CMSWP as follows.

Theorem 4 *The sequence $\{x^k\}$ generated by LAAA is convergent to a local solution to the CMSWP.*

Proof According to the proof of Theorem 3, after a certain number of iterations, say K, the iterates will be unchanged and convergent to $(x^*, P^*) = (x^K, P^K)$. Furthermore, all neighbor partition of P^K (if there exist) have already been visited.

We will investigate the ordered pair (x^*, P^*) where $x^* = (x_1^*, \dots, x_m^*)^T$ and $P^* = \{A_1^*, \dots, A_m^*\}$. Let a_j be one customer which satisfies $a_j \in A_t^*$, then there are two possibilities for a_j.

(a) a_j has no tied cluster, i.e., x_t^* is the only facility which provides the minimal distance to a_j. Then we can find a neighborhood of x^* with the radius δ_j, $U(x^*, \delta_j)$, such that when x moves in $U(x^*, \delta_j)$, x_t is the facility which still provides the minimal distance to a_j among all facilities. Thus a_j remains in the tth cluster A_t^*.
(b) a_j has tied clusters $A_{j_1}^*, \dots, A_{j_q}^*$. Let $U(x^*, \delta_j)$ be one neighborhood of x^* with the radius δ_j. When x moves in $U(x^*, \delta_j)$, the customer a_j is possible to switch from A_t^* to another cluster because another facility may provide the smaller distance than x_t^*. However, provided that δ_j is small enough, it is sure that a_j can only be assigned from A_t^* to $A_{t'}^*$ with $t' \in \{j_1, \dots, j_q\}$.

Setting δ as

$$\delta := min(\delta_1, \dots, \delta_n),$$

it follows from (a) and (b) that when $x' \in U(x^*, \delta)$ the corresponding partition with regard to x', say P', will be P^* (for (a)) or one neighbor partition (if there exist) of P^* (for (b)). Recall that all neighbor partitions of P^* (if there exist) as well as P^* itself have already been visited, thus we can conclude that x^* is the best location of facilities for all of these partitions, therefore

$$\zeta(x^*, P^*) \le \zeta(x', P'), \ \forall x' \in U(x^*, \delta). \tag{67}$$

As a result, x^* is a local optimal solution to the CMSWP (8) and the proof is complete. □

5 Numerical results

This section reports some numerical results to verify the computational efficiency of proposed algorithms. The first subsection is to implement the ADMM-type method to solve a number of CSSWPs (SSWPs). To demonstrate the efficiency of Algorithm 1, we first compare it with the well-known Weiszfeld method [45] and Newton-Bracketing (NB) method [29] for SSWP, which is a special case of CSSWP. For the more general CSSWP, we demonstrate that Algorithm 1 is effective for solving the CSSWPs with different distance measures including l_1, l_2, l_∞-norms and one asymmetric gauge. In the second subsection, we apply the proposed LAAA to solve a large number of randomly generated nonconvex CMSWPs with different sizes and show that Algorithm 2 is attractive in practice. For simplicity, we choose the parameter β of Algorithms 1 and 2 as 0.05. All the programming codes were written by MATLAB 9.0 and run on a ASUS laptop (Intel Core i7-6700HQ 2.60GHz).

5.1 The efficiency of ADMM-type method

We first compare the proposed ADMM-type method with the Weiszfeld method [45] and the NB method [29]. The Weiszfeld method and the NB method are perhaps the most standard methods for solving SSWP in the literature. This paper proposes the ADMM-type method (Algorithm 1) to solve CSSWP. Note that SSWP (10), a classical and important model in facility location, is a special case of CSSWP. Hence, the ADMM-type method is applicable to solving SSWP. In the following, we first solve a number of SSWPs by ADMM-type method, Weiszfeld method and NB method and show that even for such a special problem the proposed ADMM-type method is more efficient than the other two methods.

To make the comparison clearer, we first state the iterative schemes of Weiszfeld method and NB method. The scheme of Weiszfeld method is given as

$$x^{k+1} = \sum_{j=1}^{d} \frac{s_j \left\| x^k - a_j \right\|^{-1} a_j}{\sum_{l=1}^{d} s_l \left\| x^k - a_l \right\|^{-1}}, \tag{68}$$

and the key calculation in the iterative scheme of NB method is

$$x^{k+1} = x^k - \frac{C(x^k) - M^k}{\left\| \nabla C(x^k) \right\|^2} \nabla C(x^k), \tag{69}$$

where $\nabla C(x)$ is the gradient of $C(x)$, i.e.,

$$\nabla C(x) = \sum_{j=1}^{d} s_j \frac{x - a_j}{\left\| x - a_j \right\|}, \tag{70}$$

and M^k is a value between the upper bound and lower bound of the optimal functional value of SSWP. It is clear from (68)–(70) that once the singularity occurs, i.e., the current iterate x^k coincides with one customer, the next iterate x^{k+1} is undetermined.

Assume there are 9 customers whose locations denoted by a_i $(i = 1, \dots, 9)$ are given by the columns of the following matrix:

$$\begin{pmatrix} -1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 & 1 \\ -1 & -1 & -1 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

and the weights between customers and facilities are all 1. We consider the following SSWP

$$\begin{aligned} \min \quad & \sum_{j=1}^{9} \| x - a_j \| \\ \text{s.t.} \quad & x \in R^2. \end{aligned}$$

Table 1 ADMM, Weiszfeld and NB for small-size SSWP with different initial iterates

Initial iterate	ADMM-type method		Weiszfled method		NB method	
	Iteration	CPU	Iteration	CPU	Iteration	CPU
a_1	5.615	0.5954	/	/	/	/
a_2	6.099	0.6825	/	/	/	/
a_3	4.980	0.5087	/	/	/	/
a_4	5.590	0.5775	/	/	/	/
a_5	5.455	0.5562	/	/	/	/
a_6	5.304	0.5450	/	/	/	/
a_7	6.107	0.6787	/	/	/	/
a_8	5.648	0.6125	/	/	/	/
a_9	6.161	0.6921	/	/	/	/
Randomly	5.623	0.6212	7.611	2.1883	28.847	10.9380

It is obvious that the optimal solution to this simple problem is (0, 0).

Table 1 reports the computing time (units of 10^{-4} second) and the number of iterations of ADMM-type method, Weiszfeld method and NB method with different initial x^0. For the ADMM-type method, each element of the initial y^0 and λ^0 is randomly generated in $(-1, 1)$ and the average results of 100 tests are reported. The data in the "randomly" row represent the average results for solving the SSWP 100 times with x^0 randomly generated in $(-1, 1)^2$. The stopping criterion is set as

$$\|x^{k+1} - x^k\| < 10^{-6}.$$

When we choose the initial iterate x^0 as the locations of customers (Rows 1-9 of Table 1), the singularity occurs. Thus both the Weiszfeld method and the NB method terminate unexpectedly, which have been shown in Table 1. There are some efforts to deal with the singularity of $x^k = a_j$, e.g., the modified Weiszfeld method in [43] and simply replacing $\|x^k - a_j\|$ with 0 in the NB method [29]. While the modified Weiszfeld method in [43] ensures the convergence, the computation workload increases greatly. By replacing $\|x^k - a_j\|$ with 0 in [29], the iteration of NB method can continue but the convergence is still a problem. However, the convergence of ADMM-type method is theoretically guaranteed and it can get the optimal solution quickly in the case of singularity. On the other hand, even with the randomly generated initial iterate for which the singularity does not occur (if it occurs, we ignore such initial iterate and generate another new one randomly), it is obvious that the proposed ADMM-type method outperforms the Weiszfeld method and the NB method significantly, as revealed by the row of "randomly".

Then we compare the computational efficiency of Algorithm 1 to the Weiszfeld method and the NB method for solving the SSWP with various sizes. For each d, the locations of customers are randomly generated in $(-250, 250)^2$ and all s_j $(j = 1, 2, \ldots, d)$ are randomly generated in (1, 10). The initial iterates x^0 (as well as each component of y^0 and λ^0 for ADMM-type method) are randomly generated in $(-250, 250)^2$. The stopping criterion is set as $\|x^{k+1} - x^k\| < 10^{-6}$. Table 2 reports the average computing time (units of second) and the average number of iterations of ADMM-type method, Weiszfeld method and NB method for solving the SSWPs 100 times for each size.

It is shown in Table 2 that the computation time of ADMM-type method is much less than the Weiszfeld method and the NB method for SSWPs even for large-scale problems.

Table 2 ADMM, Weiszfeld and NB for SSWPs with different sizes

d	ADMM-type method		Weiszfled method		NB method	
	Iteration	CPU	Iteration	CPU	Iteration	CPU
100	107.92	0.0010	35.30	0.0269	65.97	0.0280
200	116.24	0.0016	32.53	0.0320	65.25	0.0492
400	123.97	0.0027	30.45	0.0556	62.79	0.1013
600	132.23	0.0036	30.36	0.0876	67.22	0.1544
800	154.38	0.0049	29.81	0.1134	64.50	0.1925
1000	169.60	0.0058	29.34	0.1328	63.67	0.2722
5000	206.94	0.1525	28.90	0.6457	64.42	1.1354
10,000	238.48	0.2956	28.78	1.3090	65.81	2.2728
50,000	247.79	1.6831	28.72	6.7666	63.03	11.2227
100,000	270.81	3.6062	28.63	12.7655	62.75	22.6563

Combining this with the results from Table 1, it is easy to conclude that the proposed ADMM-type method is much preferable to the Weiszfeld method and the NB method.

Further we apply ADMM-type method to solve the CSSWP in which the gauge is used to measure the distances and locational constraints are imposed to new facilities. That is,

$$\begin{aligned} \min \quad & \sum_{j=1}^{d} s_j \gamma(x - a_j) \\ \text{s.t.} \quad & x \in X = \{x \in R^2 |\ \|x - c\| \le r\}. \end{aligned}$$

For each d, the locations of customers are randomly generated in $(-250, 250)^2$ and all s_j are randomly generated in $(1, 10)$. Each element of the initial iterate is randomly generated in $(-250, 250)$ and the stopping criterion is $\|x^{k+1} - x^k\| < 10^{-6}$. In the locational constraint the center c is randomly generated in $(-250,\ 250)^2$ and the radius r is randomly chosen in $[10, 15]$.

To demonstrate the robustness of Algorithm 1, the CSSWPs with different distance measures, including l_1, l_2, l_∞-norms and one asymmetric gauge, are solved. Specifically, the asymmetric gauge is generated with the unit ball given by (26), i.e.,

$$\frac{(x_1 - 1)^2}{2} + x_2^2 \le 1.$$

Table 3 reports the average computing time (units of second) and the average number of iterations of ADMM-type method for solving the CSSWPs with different distance measures 100 times for each size.

According to Table 3, it is easy to observe that the ADMM-type method is very efficient for solving the CSSWP with different distance measures. Combining these results with those in Tables 1 and 2 together, the efficiency of the proposed ADMM-type method is evidently verified.

Comparing the results of ADMM-type method in Tables 2 and 3, it is easy to see that the constrained problem CSSWP need much less workload than the unconstrained problem SSWP. It is reasonable since the feasible set is reduced for the constrained problem. Another interesting fact from Table 3 is that the workload for the CSSWP with l_1-norm and l_∞-norm

Table 3 ADMM-type method for CSSWPs with different distance measures

d	l_1-norm		l_2-norm		l_∞-norm		Gauge	
	Iteration	CPU	Iteration	CPU	Iteration	CPU	Iteration	CPU
100	103.34	0.0016	56.42	0.0010	98.27	0.0014	103.88	0.0021
200	165.01	0.0023	63.39	0.0011	119.02	0.0016	109.82	0.0034
400	170.52	0.0027	70.44	0.0017	148.63	0.0023	114.99	0.0047
600	183.15	0.0054	77.35	0.0022	160.46	0.0031	103.94	0.0066
800	243.97	0.0058	82.72	0.0029	193.85	0.0050	113.21	0.0069
1000	296.83	0.0083	89.21	0.0036	220.27	0.0062	129.80	0.0097
5000	303.66	0.1459	89.86	0.0697	232.06	0.2605	161.67	0.1995
10,000	281.28	0.1965	91.30	0.0994	254.40	0.4437	154.24	0.3063
50,000	257.35	1.0560	94.63	0.4439	289.28	2.7636	170.46	1.5594
100,000	255.82	4.3861	95.85	1.8756	305.24	10.5222	183.21	4.6862

is greater than that with l_2-norm and the particular gauge . The reason is possibly that the l_1-norm and l_∞-norm are block norms while the l_2-norm and the particular gauge are round.

We further compare the workload for the CSSWP with l_2-norm and the particular gauge, it is found that the workload with the particular gauge is greater than that with l_2-norm, which is due to the asymmetric property of gauge. For the CSSWP with l_1-norm and l_∞-norm, we also find that the workload for the CSSWP with l_∞-norm is greater than that with l_1-norm even when they have similar numbers of iterations. The reason is that the calculation of y_j^{k+1} for the CSSWP with l_∞-norm is based on (41) and (42), which is time-consuming.

5.2 The efficiency of LAAA

Numerical results in Tables 1, 2 and 3 have shown the computational efficiency of ADMM-type method for solving SSWP and CSSWP. Therefore, it is reasonable to expect the effective computational performance of the location–allocation algorithm embedded by ADMM-type method in its location phase for solving MSWP and CMSWP. Recall that when the Weiszfeld method or the NB method is used in the location phase to solve the location subproblems, it results in two other well-known methods, Cooper algorithm and Cooper-NB algorithm, which are always regarded as the standard and popular methods for MSWP.

We first compare Algorithm 2 with the Cooper algorithm and the Cooper-NB algorithm to solve the MSWP (1), in which there are no constraints on the locations of new facilities and the Euclidean distances (l_2-norm) are used. We test the MSWPs with $n = 100, 500, 1000$ and $m = 2, 4, 6, 8, 10$. All locations of the customers are randomly generated in $(-250, 250)^2$ and all s_j $(j = 1, 2, \ldots, n)$ are randomly generated in $(1, 10)$. For the location subproblems, the initial iterates are randomly chosen in $(-250, 250)^2$ and the stopping criterion is $\|x^{k+1} - x^k\| < 10^{-4}$. Table 4 reports the average computing time in seconds, the average number of outer iterations and the average number of inner iterations of the LAAA, the Cooper algorithm and the Cooper-NB algorithm for testing the MSWP of each size 50 times. It is clear that the numbers of outer iterations of three algorithms are the same, as reported in Table 4.

It is shown in Table 4 that the LAAA, the Cooper algorithm and the Cooper-NB algorithm are all efficient for solving MSWP even for large-scale problems. Nevertheless, the preference

Table 4 LAAA, Cooper–Weiszfeld and Cooper-NB for MSWP with different sizes

m	n	LAAA (Algorithm 2)		Cooper–Weiszfled		Cooper-NB		Out iter.
		Inn. iter.	CPU	Inn. iter.	CPU	Inn. iter.	CPU	
2	100	815.3	0.0087	358.2	0.0851	617.8	0.1132	7.0
	500	1605.8	0.0405	466.8	0.5618	935.7	0.8435	10.8
	1000	2376.0	0.0947	678.3	1.5340	1386.5	2.3801	16.1
4	100	1193.9	0.0115	610.3	0.0725	961.5	0.0875	5.7
	500	2342.1	0.0469	876.5	0.5043	1831.1	0.7988	10.9
	1000	3229.8	0.1254	1246.0	1.5204	2851.4	2.7242	17.0
6	100	1401.2	0.0126	892.8	0.0704	1406.4	0.0873	6.0
	500	3856.3	0.0722	1428.5	0.5559	3229.7	0.9540	13.3
	1000	6237.0	0.1857	2120.8	1.6917	5283.5	3.2101	21.9
8	100	1592.8	0.0160	1468.2	0.0893	2430.3	0.1204	8.1
	500	5461.3	0.1075	2305.5	0.7199	5322.1	1.2530	16.9
	1000	8258.1	0.2666	3017.3	1.8260	7626.2	3.4581	24.1
10	100	1268.2	0.0157	1799.4	0.0906	2823.9	0.1176	7.8
	500	3817.9	0.1028	2549.1	0.6559	5564.5	1.0384	15.5
	1000	8984.4	0.2844	4006.9	2.0422	11, 387.4	4.2093	29.6

of LAAA is obvious, which justifies our previous expectations. When m is fixed, the number of outer iterations increases with n increasing. For fixed n, the number of outer iterations nearly increases with m increasing. The reason is that for large-scale problem it is more possible to reassign customers from current cluster to another one.

Note that in this paper we extends the MSWP (1) to CMSWP (8) with considering the locational constraints and the distance measure of gauge. Correspondingly, the constrained and unconstrained problems with the symmetric or asymmetric distances are all involved in the CMSWP. Thus, it is necessary to apply LAAA to solve the CMSWP with symmetric and asymmetric gauges and report the corresponding numerical results.

Similarly, we test the CMSWPs with $n = 100, 500, 1000$ and $m = 2, 4, 6, 8, 10$. For each (m, n), the locations of customers are randomly generated in $(-250, 250)^2$ and all s_j are randomly generated in $(1, 10)$. For the location subproblems, the initial iterates are randomly generated in $(-250, 250)^2$ and the stopping criterion is chosen as $\|x^{k+1} - x^k\| < 10^{-4}$. The locational constraint are set as $X_i = \{x_i \in R^2 | \; \|x_i - c_i\| \leq r_i\}$, where c_i is randomly generated in $(-250, \;250)^2$ and r_i is randomly chosen in $[10, 15]$. The asymmetric gauge $\gamma(\cdot)$ is also generated with the unit ball given by (26). Table 5 reports the average computing time in seconds, the average number of outer iterations and the average number of inner iterations of LAAA for solving the CMSWP of each size 100 times.

Table 5 shows that the LAAA is very effective for the CMSWP with different distance measures, and the involved computational workload is pretty small. Moreover, the computational workload is not too sensitive to the size of CMSWP. These desirable properties contribute to making LAAA promising in practice.

In addition, comparing the results of LAAA for CMSWP with l_2-norm in Table 5 to those for MSWP with l_2-norm in Table 4, the workload of LAAA for constrained problem is less than that for unconstrained problem, which is similar to the ADMM-type method for SSWP and CSSWP. The same conclusion for the problems with gauge can also be drawn

Table 5 LAAA for CMSWP (MSWP) with different distance measures

m	n	l_2-norm and constrained			Gauge and constrained			Gauge and unconstrained		
		Out iter.	Inner iter.	CPU	Out iter.	Inner iter.	CPU	Out iter.	Inner iter.	CPU
2	100	2.7	133.4	0.0059	2.4	187.5	0.0076	6.0	920.1	0.0172
	500	3.5	216.0	0.0114	2.7	211.3	0.0152	9.5	2073.2	0.0674
	1000	4.8	259.9	0.0251	3.3	328.2	0.0293	9.6	2221.9	0.1157
4	100	3.5	380.8	0.0073	3.4	477.3	0.0114	9.5	2545.0	0.0386
	500	5.2	631.7	0.0230	4.9	887.2	0.0322	14.6	3543.4	0.1010
	1000	7.8	1184.4	0.0634	5.6	1168.8	0.0636	23.4	8953.4	0.3620
6	100	4.4	821.3	0.0161	4.0	1053.9	0.0236	9.1	3943.2	0.0578
	500	5.9	1383.9	0.0400	5.8	1886.3	0.0607	16.7	8647.1	0.1961
	1000	8.0	2001.8	0.0882	8.3	2797.6	0.1025	20.8	16, 562.8	0.4506
8	100	5.0	1106.6	0.0222	4.7	1616.6	0.0360	9.8	4755.2	0.0692
	500	6.1	1822.2	0.0537	6.2	3610.5	0.0983	17.9	13, 492.7	0.2691
	1000	8.6	3569.5	0.1102	9.0	4913.1	0.1698	24.0	21, 540.9	0.4954
10	100	4.0	1215.5	0.0275	5.5	1956.0	0.0505	7.2	3475.5	0.0665
	500	7.5	2086.0	0.0859	5.6	2683.5	0.0720	10.5	13, 172.0	0.2250
	1000	9.0	4783.1	0.1040	14.1	6339.2	0.2524	20.8	18, 374.3	0.4503

by the comparison of Columns 6–8 (gauge and constrained) to Columns 9–11 (gauge and unconstrained).

6 Conclusion

This paper investigates a nonconvex multi-source Weber problem where the gauge are employed to measure the distances and the locational constraints are imposed on new facilities. A new location–allocation algorithm is presented for this nonconvex problem, and its attractive characteristic is that the involved single-source Weber problems with different distance measures in location phase are solved by efficient alternating direction method of multipliers in a uniform framework. The local convergence of proposed location–allocation algorithm is proved. Preliminary numerical results justify the preferable merits of proposed algorithms.

References

1. Bertsekas, D.P.: Constrained Optimization and Lagrange Multiplier Methods. Academic Press, New York (1982)
2. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**, 1–122 (2011)
3. Brimberg, J., Hansen, P., Mladenovic, N., Taillard, E.D.: Improvements and comparison of heuristics for solving the uncapacitated multisource Weber problem. Oper. Res. **48**(3), 444–460 (2000)
4. Brimberg, J., Love, R.F.: Global convergence of a generalized iterate procedure for the minisum location problem with l_p distances. Oper. Res. **41**(6), 1153–1163 (1993)
5. Bruckstein, A.M., Donoho, D.L., Elad, M.: From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Rev. **51**(1), 34–81 (2009)
6. Carrizosa, E., Conde, E., Oz, M.M., Puerto, J.: Simpson points in planar problems with locational constraints—the polyhedral gauge case. Math. Oper. Res. **22**, 297–300 (1997)
7. Chandrasekaran, R., Tamir, A.: Open questions concerning Weiszfeld's algorithm for the Fermat–Weber location problem. Math. Program. **44**, 293–295 (1989)
8. Chen, C.H., He, B.S., Ye, Y.Y., Yuan, X.M.: The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. Math. Program. **155**, 57–79 (2016)
9. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward–backward splitting. Multiscale Model. Simul. **4**(4), 1168–1200 (2005)
10. Cooper, L.: Heuristic methods for location–allocation problems. SIAM Rev. **6**, 37–53 (1964)
11. Cooper, L.: Solutions of generalized location equilibrium models. J. Reg. Sci. **7**, 1–18 (1967)
12. Dai, Y.H., Han, D.R., Yuan, X.M., Zhang, W.X.: A sequential updating scheme of the Lagrange multiplier for separable convex programming. Math. Comput. **86**, 315–343 (2017)
13. Deng, W., Yin, W.: On the global and linear convergence of the generalized alternating direction method of multipliers. J. Sci. Comput. **66**(3), 889–916 (2016)
14. Douglas, J., Rachford, H.H.: On the numerical solution of the heat conduction problem in 2 and 3 space variables. Trans. Am. Math. Soc. **82**, 421–439 (1956)
15. Drezner, Z.: Facility Location: A Survey of Applications and Methods. Springer, New York (1995)
16. Eckstein, J., Bertsekas, D.P.: On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program. **55**, 293–318 (1992)
17. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, Boston (1990)
18. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite-element approximations. Comput. Math. Appl. **2**, 17–40 (1976)
19. Glowinski, R., Marrocco, A.: Approximation par *éléments* finis d'ordre un et *ré*solution par *pé*nalisation-dualit*é* d'une classe de probl*è*mes non lin*é*aires. RAIRO **R2**, 41–76 (1975)
20. Guo, K., Han, D.R., Yuan, X.M.: Convergence analysis of Douglas–Rachford splitting method for "strongly + weakly" convex programming. SIAM J. Numer. Anal. **55**, 1549–1577 (2017)
21. He, B.S., Tao, M., Yuan, X.M.: Alternating direction method with Gaussian back substitution for separable convex programming. SIAM J. Optim. **22**, 313–340 (2012)

22. He, B.S., Yang, H.: Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities. Oper. Res. Lett. **23**(3–5), 151–161 (1998)
23. He, B.S., Yang, H., Wang, S.: Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. J. Optim. Theory Appl. **106**(2), 337–356 (2000)
24. Hestenes, M.R.: Multiplier and gradient methods. J. Optim. Theory Appl. **4**, 303–320 (1969)
25. Iyigun, C., Israel, A.: A generalized Weiszfeld method for the multi-facility location problem. Oper. Res. Lett. **38**, 207–214 (2010)
26. Jiang, J.L., Yuan, X.M.: A heuristic algorithm for constrained multi-source Weber problem—the variational inequality approach. Eur. J. Oper. Res. **187**, 357–370 (2008)
27. Katz, I.N., Vogl, S.R.: A Weiszfeld algorithm for the solution of an asymmetric extension of the generalized Fermat location problem. Comput. Math. Appl. **59**, 399–410 (2010)
28. Kuhn, H.W.: A note on Fermat's problem. Math. Program. **4**, 98–107 (1973)
29. Levin, Y., Ben-Israel, A.: A heuristic method for large-scale multi-facility location problems. Comput. Oper. Res. **31**, 257–272 (2004)
30. Lions, P.L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. SIAM J. Numer. Anal. **16**, 964–979 (1979)
31. Liu, Y.F., Liu, X., Ma, S.Q.: On the nonergodic convergence rate of an inexact augmented Lagrangian framework for composite convex programming. Math. Oper. Res. **44**(2), 632–650 (2019)
32. Love, R.F., Morris, J.G.: Mathematical models of road travel distances. Manag. Sci. **25**, 130–139 (1979)
33. Megiddo, N., Supowit, K.J.: On the complexity of some common geometric location problems. SIAM J. Comput. **13**, 182–196 (1984)
34. Michelot, C., Lefebvre, O.: A primal–dual algorithm for the Fermat–Weber problem involving mixed gausess. Math. Program. **39**, 319–335 (1987)
35. Minkowski, H.: Theorie der konvexen Körper, Gesammelte Abhandlungen. Teubner, Berlin (1911)
36. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Spriger, Berlin (2006)
37. Ostresh, L.M.: On the convergence of a class of iterative methods for solving the Weber location problem. Oper. Res. **26**, 597–609 (1978)
38. Peng, Y.G., Ganesh, A., Wright, J., Xu, W.L., Ma, Y.: Robust alignment by sparse and low-rank decomposition for linearly correlated images. IEEE Trans. Pattern Anal. Mach. Intell. **34**, 2233–2246 (2012)
39. Plastria, F.: Asymmetric distances, semidirected networks and majority in Fermat–Weber problems. Ann. Oper. Res. **167**, 121–155 (2009)
40. Powell, M.J.D.: A method for nonlinear constrained total-variation problems. In: Fletcher, R. (ed.) Optimization, pp. 283–298. Academic Press, New York (1969)
41. Sun, D.F., Toh, K.C., Yang, L.Q.: A convergent 3-block semiproximal alternating direction method of multipliers for conic programming with 4-type constraints. SIAM J. Optim. **25**, 882–915 (2015)
42. Tao, M., Yuan, X.M.: Recovering low-rank and sparse components of matrices from incomplete and noisy observations. SIAM J. Optim. **21**, 57–81 (2011)
43. Vardi, Y., Zhang, C.H.: A modified Weiszfeld algorithm for the Fermat–Weber location problem. Math. Program. **90**, 559–566 (2001)
44. Ward, J.E., Wendell, R.E.: Using block norms for location modelling. Oper. Res. **33**, 1074–1090 (1985)
45. Weiszfeld, E.: Sur le point pour lequel la somme des distances de *n* points donnés est minimum. Tohoku Math. J. **43**, 355–386 (1937)
46. Witzgall, C.: Optimal location of a central facility, mathematical models and concepts. Report 8388, National Bureau of Standards, Washington DC, USA (1964)
47. Yao, Z.S., Lee, L.H., Jaruphongsa, W., Tan, V., Hui, C.F.: Multi-source facility location–allocation and inventory problem. Eur. J. Oper. Res. **207**, 750–762 (2010)
48. Yang, J.F., Sun, D.F., Toh, K.C.: A proximal point algorithm for log-determinant optimization with group lasso regularization. SIAM J. Optim. **23**, 857–893 (2013)