ADMM-TYPE METHODS FOR GENERALIZED MULTI-FACILITY WEBER PROBLEM

SHUN ZHANG
Department of Mathematics, College of Science
Nanjing University of Aeronautics and Astronautics
29 Yudao Street, Qinhuai District, Nanjing, 210016, China

JIANLIN JIANG
1 Department of Mathematics, College of Science
Nanjing University of Aeronautics and Astronautics
29 Yudao Street, Qinhuai District, Nanjing, 210016, China
2 School of Information and Mathematics, Yangtze University
1 Nanhuan Road, Jingzhou, 434023, China

SU ZHANG*
Business School, Nankai University
94 Weijin Road, Nankai District, Tianjin, 300071, China

YIBING LV
School of Information and Mathematics, Yangtze University
1 Nanhuan Road, Jingzhou, 434023, China

YUZHEN GUO
Department of Mathematics, College of Science
Nanjing University of Aeronautics and Astronautics
29 Yudao Street, Qinhuai District, Nanjing, 210016, China

(Communicated by Jie Sun)

Abstract. The well-known multi-facility Weber problem (MFWP) is one of fundamental models in facility location. With the aim of enhancing the practical applicability of MFWP, this paper considers a generalized multi-facility Weber problem (GMFWP), where the gauge is used to measure distances and the locational constraints are imposed to new facilities. This paper focuses on developing efficient numerical methods based on alternating direction method of multipliers (ADMM) to solve GMFWP. Specifically, GMFWP is equivalently reformulated into a minmax problem with special structure and then some ADMM-type methods are proposed for its primal problem. Global convergence of proposed methods for GMFWP is established under mild assumptions. Preliminary numerical results are reported to verify the effectiveness of proposed methods.

2010 Mathematics Subject Classification. Primary: 90C30; Secondary: 90B85.
Key words and phrases. Multi-facility Weber problem, singularity, gauge, locational constraints, alternating direction method of multipliers.

This work was supported by National Natural Science Foundation of China (Grant No. 11971230, 12071234) and National Social Science Fund of China (Grant No. 18BGL063).

* Corresponding author: Su Zhang.
1. Introduction. Let \( a_1, \cdots, a_n \) be the locations of \( n \) customers in the plane. The well-known multi-facility Weber problem (MFWP) is to find the locations of \( m \) new facilities \( x_1, \cdots, x_m \) such that the sum of transportation costs between \( n \) customers and \( m \) facilities and the costs between \( m \) facilities is minimized. More specifically, the mathematical model of MFWP is as follows:

\[
\text{MFWP: } \min_{x_i \in \mathbb{R}^2} \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \|x_i - a_j\| + \sum_{i=1}^{m-1} \sum_{l=i+1}^{m} v_{il} \|x_i - x_l\|, \tag{1}
\]

where \( w_{ij} > 0 \) is the weight associated with the unit transportation cost between \( x_i \) and \( a_j \), \( v_{il} > 0 \) is the weight associated with the unit cost between \( x_i \) and \( x_l \), and \( \| \cdot \| \) is the Euclidean distance.

As is known, MFWP is one of the fundamental models in the field of facility location. Due to the wide applications in operations research, marketing, urban planning etc., MFWP has attracted wide attentions and many authors are devoted to both promoting its theoretical development and presenting effective numerical algorithms, see, e.g. [7, 8, 24, 29]. The first attractive contribution to this aspect, denoted by Miehle algorithm (MA), is due to [24], where an extension of Weiszfeld procedure [36] was proposed for MFWP. Ostresh [26] proved later that the Miehle algorithm is a descent method. Rosen and Xue [31], however, constructed counterexamples showing that Miehle algorithm may stop or converge to one nonoptimal point.

How to improve the performance of MA and make it computationally preferable thus become the main challenges in the study of MFWP. The famous hyperboloid approximation procedure (HAP) is proposed in this effort, which is perhaps the most standard method for MFWP at present. The idea of HAP is to introduce a small positive scalar \( \varepsilon \) to the actual objective function to get a smooth approximating problem which is then solved by Miehle algorithm, see e.g. [9, 30]. The global convergence of HAP can be found in [30]. Recall that for HAP the larger the value of \( \varepsilon \), the faster the convergence to optimal value of approximating function but the worse the accuracy of approximation to the actual nondifferentiable function. So HAP can be improved by successively reducing the value of \( \varepsilon \), which is called SHAP. For more details about HAP please refer to [6, 29].

Note that the distances in MFWP (1) are Euclidean distances, i.e., measured by \( l_2 \)-norm. As in MFWP (1), the distances of facility location are usually measured by norms such as \( l_p \)-norms and block norms in the literature. [23] discussed the approximations about the weighted \( l_p \)-norms based on statistical evidence, and [35] investigated the use of block norms to obtain good approximation of actual distances. The symmetry property of the norm assures that the distance from one point to another is always equal to the distance back. Nevertheless, [38] has highlighted the fact that the symmetry of the distance is violated in numerous real situations. The examples include transportation in rush-hour traffic, flight in the presence of wind, navigation in the presence of currents, transportation on an inclined terrain, etc. Ever since the pioneering work of [38], the asymmetric distance has attracted the researchers’ interest, and some progress has been made in its theoretical and computational aspects recently, see e.g. [2, 3, 20, 28].

It is also clear that the locations of new facilities in MFWP can be chosen anywhere in the plane. However, it is not always this case in practice; that is to say, not every point in the plane is suitable to locate the new facilities due to some actual situations. For example, a fire station can not be built within an area where
nobody lives; a hospital or medical center cannot be very close to heavily polluting
and noisy factories. Therefore, after we consider some relevant practical factors
the locations of new facilities are usually restricted in some specific regions such as
convex subsets of the plane.

In this paper we consider more general case of MFWP to enhance its practical
applicability. Firstly, the gauge rather than the widely-used symmetric norm is
used to measure distance between points. With the usage of gauge, both symmetric
and asymmetric distances are allowed to be considered in our model. Secondly,
the locational constraints are imposed for new facilities. By choosing appropriate
locational constraints for facilities, both constrained and unconstrained location
problems can be investigated by our model.

The rest of this paper is organized as follows. The general MFWP is formulated
in Section 2. In Section 3, it is reformulated into a minmax problem and then
the corresponding primal problem and dual problem are given. Then in Section 4,
some ADMM-type methods are proposed for solving the primal problem. Prelimi-
nary numerical results are reported in Section 5 to verify the efficiency of proposed
ADMM-type methods. Finally, some conclusions are drawn in the last section.

2. Model formulation. Let \( B \) be a compact convex set in \( R^2 \) whose interior
contains the origin, then the gauge based on \( B \) is defined by

\[
\gamma(x) = \inf \{ \lambda > 0 \mid \frac{x}{\lambda} \in B \}, \quad \forall x \in R^2.
\]

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

\[
B = \{ x \in R^2 \mid \gamma(x) \leq 1 \}.
\]

This definition of a gauge from a compact convex set was first introduced by
Minkowski [25]. The gauge \( \gamma(\cdot) \) satisfies the following properties:

1) \( \gamma(x) \geq 0 \) and \( \gamma(x) = 0 \iff x = 0; \)
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

\[
D(x, y) = \gamma(y - x).
\]

When \( B \) is symmetric around the origin, according to the definition of (2) we have

\[
\gamma(-x) = \gamma(x), \quad \forall x \in R^2.
\]

It follows that

\[
D(x, y) = D(y, x),
\]

which means the distance measured by \( \gamma(\cdot) \) is symmetric. On the contrary, when
\( B \) is not symmetric around the origin, (5) does not hold any more and then 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 in location problems, depending on the requirements of practical
applications.

With considering the gauge to measure distances and the locational constraints
on facilities, the generalized multi-facility Weber problem (abbreviated as GMFWP)
is formulated as:

\[
\text{GMFWP:} \quad \min \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} \gamma(x_i - a_j) + \sum_{i=1}^{m} \sum_{l \neq i, l=1}^{m} v_{il} \gamma(x_i - x_l) \\
\text{s.t.} \quad x_i \in X_i \subseteq \mathbb{R}^2, \quad i = 1, 2, \ldots, m,
\]

where \(\gamma(\cdot)\) is the gauge generated by a given compact convex set \(B \subset \mathbb{R}^2\) and \(X_i\) denotes the locational constraint for the \(i\)-th facility. For simplification, all \(X_i, i = 1, \ldots, m\), are assumed to be closed convex subsets of \(\mathbb{R}^2\) throughout this paper. Note that \(X_i\) can also be chosen as \(\mathbb{R}^2\). If all \(X_i\) are \(\mathbb{R}^2\), the GMFWP becomes an unconstrained location problem.

Remark 1. The GMFWP (7) includes some well-known location problems as its special cases: if \(m = 1\), \(X = \mathbb{R}^2\) and \(\gamma(\cdot)\) is the Euclidean distance, (7) is the classical Weber problem; if \(m = 1\) and \(\gamma(\cdot)\) is the Euclidean distance, (7) is a constrained Weber problem; if all \(X_i = \mathbb{R}^2\) and \(\gamma(\cdot)\) is the Euclidean distance, (7) is the classical multi-facility Weber problem (1).

Though the gauge is used and locational constraints are considered, fortunately we have the following proposition for GMFWP which reveals its convexity.

Proposition 1. The GMFWP (7) is convex.

Proof. Note that all constraints \(X_j, j = 1, \ldots, m\), are closed convex sets and the function \(\gamma(\cdot)\) is also convex, then it follows immediately that the GMFWP is convex.

While GMFWP (7) is convex, it is obvious that the nondifferentiability will occur similarly as MFWP. This means that both GMFWP and MFWP share the unexpected characteristic of singularity.

3. The primal and dual reformulations of GMFWP. For the gauge \(\gamma(\cdot)\) defined by (2), there always exists a dual gauge \(\gamma^d(\cdot)\) defined by

\[
\gamma^d(z) = \max \{z^T x \mid \gamma(x) \leq 1\}.
\]

(8)

Let \(B^d\) be the unit ball of the dual gauge \(\gamma^d(\cdot)\), which is also convex and compact. \(B^d\) is exactly the polar set of \(B\). It is well known that the dual gauge of \(\gamma^d(\cdot)\) is again \(\gamma(\cdot)\), i.e.,

\[
\gamma(x) = \max \{z^T x \mid \gamma^d(z) \leq 1\},
\]

(9)

which can be rewritten as

\[
\gamma(x) = \max_{z \in B^d} z^T x.
\]

(10)

For more details about the gauge and dual gauge, as well as their properties, the readers can be referred to [28].

According to (10), GMFWP (7) is equivalent to the following min-max problem:

\[
\min_{x_i \in X_i} \max_{z_{ij}, z_{il}} \sum_{i=1}^{m} \sum_{j=1}^{n} z_{ij}^T (x_i - a_j) + \sum_{i=1}^{m} \sum_{l \neq i, l=1}^{m} z_{il}^T (x_i - x_l),
\]

(11)

where each \(z_{ij}\) is a vector in \(B_i^d = \{\xi \in \mathbb{R}^2 \mid \gamma^d(\xi) \leq w_{ij}\}\) for \(i = 1, \ldots, m\) and \(j = 1, \ldots, n\); each \(z_{il}\) is a vector in \(B_i^d = \{\xi \in \mathbb{R}^2 \mid \gamma^d(\xi) \leq v_{il}\}\) for \(i = 1, \ldots, m\) and \(l = 1, \ldots, n (l \neq i)\).
We define the following matrix

$$A_1 = \begin{pmatrix} I_2 & \cdots & I_2 \\ \vdots & \ddots & \vdots \\ I_2 & \cdots & I_2 \end{pmatrix}^T,$$  \hspace{1cm} (12)

which consists of $mn \times m$ blocks. Each block of $A_1$ is either $2 \times 2$ identity matrix ($I_2$) or $2 \times 2$ zero matrix (blank block), and each block column includes $n$ identity matrices while the others are all zero matrices. We also define $m$ matrices $N_t$, $t = 1, \cdots, m$, as

$$N_1 = \begin{pmatrix} I_2 \\ -I_2 \\ \vdots \\ -I_2 \end{pmatrix}, \quad N_2 = \begin{pmatrix} -I_2 & I_2 \\ -I_2 & I_2 \\ \vdots & \vdots \\ -I_2 & I_2 \end{pmatrix}, \quad \cdots, \quad N_m = \begin{pmatrix} -I_2 \\ -I_2 \\ \vdots \\ -I_2 \end{pmatrix}.$$

Similar to $A_1$, each block of $N_t$ is also either $2 \times 2$ identity matrix or $2 \times 2$ zero matrix. Based on these $N_t$, $t = 1, \cdots, m$, we define the following matrix

$$A_2 = (N_1, N_2, \cdots, N_m)^T.$$

By introducing the following $B^d$, $z$, $X$ and $x$

$$B^d = B^d_1 \times \cdots \times B^d_1 \times B^d_1 \times \cdots \times B^d_m \times B^d_{m-1} \times \cdots \times B^d_1 \times B^d_{m-1},$$

$$z = (z^T_{11}, \cdots, z^T_{1n}, \cdots, z^T_{m1}, \cdots, z^T_{m1}, \cdots, z^T_{m1}, \cdots, z^T_{m1}, \cdots, z^T_{m1})^T,$$

$$X = X_1 \times \cdots \times X_m, \quad x = (x_1^T, \cdots, x_m^T)^T,$$

and introducing the matrix $A$ and vector $b$

$$A = (A_1^T, A_2^T)^T, \quad b = (a_1^T, \cdots, a_n^T, \cdots, a_1^T, \cdots, a_n^T, 0, \cdots, 0)^T,$$

where each zero block is a $1 \times 2$ zero matrix and there are totally $m(m-1)$ zero blocks in $b$, (11) can be written in a compact form as follows:

$$\min_{x \in X} \max_{z \in B^d} z^T (Ax - b).$$  \hspace{1cm} (15)

We further rewrite (15) as

$$\min_{x} \max_{z} z^T Ax + \delta_X(x) - (z^T b + \delta_{B^d}(z)), $$

where $\delta_X$ and $\delta_{B^d}$ are the indicator functions of $X$ and $B^d$, respectively.

Let $g(x) = \max_{z} z^T Ax + \delta_X(x) - (z^T b + \delta_{B^d}(z))$, then it follows that

$$g(x) = \delta_X(x) + \max_{z} z^T (Ax - b) - \delta_{B^d}(z)$$

$$= \delta_X(x) + \delta_{B^d}^*(Ax - b),$$

where $\delta_{B^d}^*$ is the conjugate of $\delta_{B^d}$. Similarly, let $h(z) = \min_{x} z^T Ax + \delta_X(x) - (z^T b + \delta_{B^d}(z))$, then we have

$$h(z) = -(z^T b + \delta_{B^d}(z)) + \min_{x} z^T Ax + \delta_X(x)$$

$$= -(z^T b + \delta_{B^d}(z)) - \max_{x} -z^T Ax - \delta_X(x)$$

$$= -(z^T b + \delta_{B^d}(z)) - \delta_X^*(-A^T z),$$  \hspace{1cm} (18)
where $\delta_X$ is the conjugate of $\delta_X$. According to (17) and (18), it is clear that the min-max problem (16) is a primal-dual reformulation of the following primal problem:

$$(P) \quad \min_x \delta_X(x) + \delta_B^*(Ax - b)$$

or of the following dual problem:

$$(D) \quad \max_z -(z^Tb + \delta_{B^*}(z)) - \delta_X^*(-A^Tz).$$

It thus follows that we can solve the primal problem (P) or the dual problem (D) instead of solving (16) directly. When we compare the scales of (P) and (D), it is found that the number of variables of (P) is much smaller than that of (D) (the number of variables of (P) is $m$ and the number of variables of (D) is $m(m+n-1)$). This observation encourages us to solve the GMFWP (7) by solving the primal problem (P) in an efficient way.

**Remark 2.** The $l_p$-norm (especially $l_1, l_2, l_\infty$-norm) and block norm are frequently used to measure distances in the literature, see, e.g., [23, 35]. It is noted that the gauge used in this paper is an extension of norm. When the gauge $\gamma(\cdot)$ is chosen as $l_1, l_2, l_\infty$-norm, the dual $\gamma^d(\cdot)$ will be $l_\infty, l_2, l_1$-norm, respectively; and when $\gamma(\cdot)$ is a block norm, its dual is also a block norm.

**Remark 3.** The paper [19] considered similar generalized MFWP. However, in [19] a variational inequality (VI) approach was developed and new projection-type methods under VI were proposed for solving GMFWP. This paper reformulates GMFWP as a primal problem of the equivalent min-max problem and will propose ADMM-type methods to solve it. The numerical experiments to be reported later verify the computational preference of ADMM-type methods to projection-type methods.

**Remark 4.** A recent paper [18] also applied ADMM technique to solve facility location problem. In fact, these two papers solve two different facility location problems. The paper [18] considered the nonconvex constrained multi-source Weber problem, where ADMM was employed to solve the subproblem of the proposed location-allocation algorithm. Instead, this paper considers a generalized MFWP, which is convex itself. Thus we need to modify ADMM in a different way here.

4. **ADMM-type methods for the primal problem (19).** This section aims to develop efficient numerical methods for the primal problem (P) by exploiting its favourable structure. With introducing an auxiliary variable $y$, the primal problem (P) can be rewritten as

$$\min_{x,y} \quad \delta_X(x) + \delta_{B^*}^*(y)$$

s.t. \quad Ax - b = y. \quad (21)

Let $L_\beta(x, y, \lambda)$ be the augmented Lagrangian function of (21), which has the form of

$$L_\beta(x, y, \lambda) = \delta_X(x) + \delta_{B^*}^*(y) - \lambda^T(Ax - y - b) + \frac{\beta}{2}\|Ax - y - b\|^2,$$ \quad (22)

where $\lambda$ is the Lagrange multiplier (dual variable) and $\beta > 0$ is a penalty parameter for the linear constraint.

Revisiting the formulation of (21), we can easily see that its objective function is separated into two individual functions without coupled variables. In other words,
(P) is formulated as a linearly constrained two-block separable convex minimization problem. A large number of separable convex problems arises from various practical areas such as image processing and statistical learning, and thus this type of problems has been well studied in the past several decades. It is evident that the favorable separable structure is very beneficial for developing efficient numerical methods. To exploit the separable structure, operator splitting methods have been well discussed in the literature. The alternating direction method of multipliers (ADMM) proposed originally in [11, 13] is one of the most popular operator splitting methods, which is closely related to the well-known Douglas-Rachford splitting method in partial differential equation. For the comprehensive review of ADMM and its novel applications in various areas, the reader can be referred to [1, 5, 14, 15, 21, 32, 37] and references cited therein.

This paper concentrates on presenting some efficient ADMM-type numerical methods to solve the two-block separable problem (21) and thus solve the GM-FWP (7). Applying the ADMM to (21), it will result in the following ADMM-type iterative scheme.

\[
\begin{align*}
\begin{cases}
x^{k+1} &= \text{Argmin}_x \mathcal{L}_\beta(x, y^k, \lambda^k), \quad (23a) \\
y^{k+1} &= \text{Argmin}_y \mathcal{L}_\beta(x^{k+1}, y, \lambda^k), \quad (23b) \\
\lambda^{k+1} &= \lambda^k - \beta(Ax^{k+1} - y^{k+1} - b). \quad (23c)
\end{cases}
\]

In order to accelerate the convergence of ADMM, the penalty parameter \( \beta \) can also vary self-adaptively according to certain strategies, see, e.g., [5]. For the simplification of our discussion, the parameter \( \beta \) is set to be a constant positive number throughout this paper. Next we will discuss how to solve the subproblems of ADMM-type method (23).

I) The variable \( x^{k+1} \). According to (22) and (23a), \( x^{k+1} \) can be obtained by solving

\[
\begin{align*}
x^{k+1} &= \text{Argmin}_x \delta_X(x) + \delta_{\beta}(y^k) - \lambda^k (Ax - y^k - b) + \frac{\beta}{2} \|Ax - y^k - b\|^2 \\
&= \text{Argmin}_x \delta_X(x) + \frac{\beta}{2} \|Ax - y^k - b - \frac{1}{\beta} \lambda^k\|^2. \quad (24)
\end{align*}
\]

The \( x \)-subproblem cannot be easily solved because of general structure of the matrix \( A \). We know that the efficiency of ADMM-type methods heavily depends on how fast the subproblems are solved. To overcome this difficulty, we apply the linearization technique for \( x \)-subproblem. We linearize the quadratic term \( \frac{\beta}{2} \|Ax - y^k - b - \frac{1}{\beta} \lambda^k\|^2 \) of \( x \)-subproblem (24) and add a proximal term \( \frac{r}{2} \|x - x^k\|^2 \) to approximate the objective function as follows

\[
\begin{align*}
x^{k+1} &= \text{Argmin}_x \delta_X(x) + \frac{\beta}{2} \cdot 2x^TA^T(Ax^k - y^k - b - \frac{1}{\beta} \lambda^k) + \frac{r}{2} \|x - x^k\|^2 \\
&= \text{Argmin}_x \delta_X(x) + \frac{r}{2} \|x - x^k + \frac{\beta}{r} A^T(Ax^k - y^k - b - \frac{1}{\beta} \lambda^k)\|^2. \quad (25)
\end{align*}
\]

The linearized \( x \)-subproblem can be also seen as a special application of PPA because of the additional proximal term \( \frac{r}{2} \|x - x^k\|^2 \). It is obvious that (25) is quite simple enough to have the following closed-form solution

\[
x^{k+1} = P_X(x^k - \frac{\beta}{r} A^T(Ax^k - y^k - b - \frac{1}{\beta} \lambda^k)), \quad (26)
\]
where $P_X(v)$ is the projection of $v$ onto the set $X$, i.e.,

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

Thus, no inner iteration is required for the $x$-subproblem by using the linearization strategy.

II) The variable $y^{k+1}$. Let $p = Ax^{k+1} - b - \frac{1}{\beta}\lambda^k$, then it is easy to see that $y^{k+1}$ is the solution to the following problem

$$y^{k+1} = \text{Argmin}_y \{\delta_X(x^{k+1}) + \delta_\delta^*(y) - \lambda^k T (Ax^{k+1} - y - b) + \beta^2 \|Ax^{k+1} - y - b\|^2 \}$$

$$= \text{Argmin}_y \{\delta_\delta^*(y) + \frac{\beta}{2} \|y - Ax^{k+1} + b + \frac{1}{\beta}\lambda^k\|^2 \}$$

$$= \text{Argmin}_y \{\delta_\delta^*(y) + \frac{\beta}{2} \|y - p\|^2 \}. \quad (27)$$

Using Moreau’s decomposition, we can characterize the closed-form solution to $y$-subproblem (27) by the following proposition.

**Proposition 2.** For $\beta > 0$ and the vector $p$ defined in (27), the solution to the $y$-subproblem is given as

$$y^{k+1} = p - \frac{1}{\beta} P_{B\delta}(\beta p). \quad (28)$$

**Proof.** Let $\varphi : \mathbb{R}^{2m(m+n-1)} \to \mathbb{R}$ be a function and $\text{Prox}_{\delta \varphi}$ be the proximity operator of function $\varphi$, i.e.,

$$\text{Prox}_{\delta \varphi}(p) = \text{Argmin}_y \{\varphi(y) + \frac{1}{2\beta} \|y - p\|^2 \}.$$ 

Thus, according to (27) the solution to $y$-subproblem is

$$y^{k+1} = \text{Prox}_{\delta^*/\beta}(p). \quad (29)$$

It follows from Moreau’s decomposition that

$$p = \text{Prox}_{\delta^*/\beta}(p) + \frac{1}{\beta} \text{Prox}_{\beta \delta^*}(\beta p) = \text{Prox}_{\delta^*/\beta}(p) + \frac{1}{\beta} \text{Prox}_{\beta \delta}(\beta p), \quad (30)$$

where the last equation is from the proposition of conjugate that $\delta^{**} = \delta$.

For the second term $\text{Prox}_{\beta \delta}(\beta p)$ of (30), according to the definition of proximity operator it satisfies

$$\text{Prox}_{\beta \delta}(\beta p) = \text{Argmin}_y \{\delta_{B\delta}(y) + \frac{1}{2\beta} \|y - \beta p\|^2 \} = P_{B\delta}(\beta p). \quad (31)$$

Substituting (31) into (30), we obtain

$$\text{Prox}_{\delta^*/\beta}(p) = p - \frac{1}{\beta} P_{B\delta}(\beta p). \quad (32)$$

Combining it with (29), this proposition is derived directly. \qed

III) The Lagrange multiplier (dual variable) $\lambda^{k+1}$. The multiplier $\lambda^{k+1}$ is updated trivially according to (23c).

Based on the previous discussion on the iterative scheme and the corresponding subproblems, now we are ready to present our first ADMM-type method for (19), denoted as ADMM-a, as follows.
Given the parameter $\beta > 0$, $r > 0$ and initial iterate $w^0 = (x^0, y^0, \lambda^0)$. For $k = 0, 1, 2, \ldots$, do:

$$
\begin{align*}
    x^{k+1} &= P_X(x^k - \frac{\beta}{r} A^T (Ax^k - y^k - b - \frac{1}{\beta} \lambda^k)), \\
    y^{k+1} &= p - \frac{1}{\beta} P_{B^+}(\beta p), \\
    \lambda^{k+1} &= \lambda^k - \beta (Ax^{k+1} - y^{k+1} - b),
\end{align*}
$$

where $p = Ax^{k+1} - b - \frac{1}{\beta} \lambda^k$.

As we have explained, the proposed ADMM-a is in fact a linearized ADMM method for (19), in which the $x$-subproblem is linearized and thus its closed-form solution can be derived easily. The linearized ADMM has been well studied in the past several years, see, e.g., [15, 22, 39, 34]. Our proposed ADMM-a is a special scenario of that of the generic linearized ADMM and thus its convergence can be found in the literature. Let $\{w^k\}$ be the sequence generated by ADMM-a with $w^k = (x^k, y^k, \lambda^k)$, and we state the convergence result for (19) as follows with omitting the proof.

**Theorem 4.1.** For any given $\beta > 0$ and an arbitrary initial iterate $w^0 = (x^0, y^0, \lambda^0)$, the sequence $\{w^k\}$ converges to $w^* = (x^*, y^*, \lambda^*)$, where $x^*$ is the optimal solution to the primal problem (19) provided $rI - \beta A^T A \succ 0$.

**Remark 5.** A positive parameter $\tau$ can be added to the update scheme of dual variable (33c), i.e.,

$$
\lambda^{k+1} = \lambda^k - \tau \beta (Ax^{k+1} - y^{k+1} - b),
$$

and (34) is reduced to (33c) for $\tau = 1$. The convergence of ADMM for $\tau \in (0, (1 + \sqrt{5})/2)$ has been established, see, e.g., [10, 12]. It has been shown that large $\tau \in (0, (1 + \sqrt{5})/2)$ can accelerate the convergence of ADMM, and therefore $\tau$ is always chosen as 1.618 in numerical implementation.

To ensure the convergence of ADMM-a, it follows from Theorem 4.1 that the parameter $\beta$ and $r$ should satisfy $rI - \beta A^T A \succ 0$. Note that

$$
rI - \beta A^T A \succ 0 \iff r > \beta \lambda_{\text{max}}(A^T A),
$$

where $\lambda_{\text{max}}(A^T A)$ is the largest eigenvalue of $A^T A$. It means that $r$ should not be too small. On the other hand, however, $r$ can not be too large; otherwise, the linearized $x$-subproblem (25) will be far from the original $x$-subproblem (23a), which may result in a slow convergence of ADMM-a. Then, it leads to a natural concern about how to choose the value of the parameter $r$.

In the following, a self-adaptive adjustment of the value of $r$ is presented which can alleviate the restriction of (35) for $r$ to some extent. Based on this self-adaptive manner, we propose the second ADMM-type method (denoted as ADMM-b) for (19). It consists of two steps: the prediction step and the correction step, where the prediction step produces a tentative point $\tilde{w}^k$ and the correction step generates the next iterate $w^{k+1}$. The iterative framework of ADMM-b is elaborated in the following.
Algorithm 2. ADMM-b for (19)

Given the parameter $\beta > 0$ and initial iterate $w^0 = (x^0, y^0, \lambda^0)$. For given $w^k = (x^k, y^k, \lambda^k)$ and a small constant $r > 0$, $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1})$ is generated via the following scheme.

**Step 1. Prediction step.**

**Step 1.1.** Find $\tilde{w}^k = (\tilde{x}^k, \tilde{y}^k, \tilde{\lambda}^k)$ such that

\[
\begin{align*}
\tilde{x}^k &= P_X(x^k - \frac{\beta}{r} A^T(Ax^k - y^k - b - \frac{1}{\beta} \lambda^k)), \\
\tilde{y}^k &= p - \frac{1}{\beta} P_B(\beta p), \\
\tilde{\lambda}^k &= \lambda^k - \beta(Ax^{k+1} - y^{k+1} - b),
\end{align*}
\]  

(36a) \hspace{1cm} (36b) \hspace{1cm} (36c)

where $p = Ax^{k+1} - b - \frac{1}{\beta} \lambda^k$.

**Step 1.2.** If the following inequality is not satisfied,

\[
\beta \| A^T A (x^k - \tilde{x}^k) \| \leq \nu r \| x^k - \tilde{x}^k \|, \quad \nu \in (0, 1),
\]  

(37)

then set $r = 2r$ and go to Step 1.1.

**Step 2. Correction step.** Let $v = (y, \lambda)$, then update $w^{k+1}$ via

\[
\begin{align*}
x^{k+1} &= \tilde{x}^k + \frac{\beta}{r} A^T A (x^k - \tilde{x}^k), \\
v^{k+1} &= \tilde{v}^k.
\end{align*}
\]  

(38a) \hspace{1cm} (38b)

**Remark 6.** It is obvious that the prediction step of ADMM-b is as the same as that of ADMM-a. In other words, ADMM-a provides a tentative point for ADMM-b in the prediction step, and then based on such a point the new iterate $w^{k+1}$ is generated in the correction step. The workload of correction step can be almost negligible since the term $A^T A (x^k - \tilde{x}^k)$ has been already computed when we check the inequality (37) in Step 1.2.

**Remark 7.** The parameters $\beta$ and $r$ are required to satisfy the condition (37) in ADMM-b. Thus, for given $\beta$, instead of using a fixed value the parameter $r$ can be adjusted in a self-adaptive manner based on (37). When we compare the two conditions (35) and (37), it is easy to see that (35) implies (37). Thus, the restriction on $\beta$ and $r$ of ADMM-b is more relaxed than ADMM-a with the adoption of self-adaptive strategy.

ADMM-b is an application of linearization technique to our problem (21), and the convergence of ADMM-b can be obtained from [15]. Analogous to ADMM-a, we omit the convergence proof of ADMM-b in detail and just give its convergence result.

**Theorem 4.2.** For any given $\beta > 0$ and an arbitrary initial iterate $w^0 = (x^0, y^0, \lambda^0)$, the sequence $\{w^k\}$ converges to $w^* = (x^*, y^*, \lambda^*)$ where $x^*$ is the optimal solution to the primal problem (19) provided that the condition (37) is satisfied.

Recall that the two ADMM-type methods proposed above are linearized ADMM. The linearization technique is quite important for ADMM when one or more subproblems are hard to solve. With the adoption of linearization technique, the difficult subproblems may be simple enough to have closed-form solutions or can be
ADMM-TYPE METHODS FOR GMFWP

Easily solved up to a high precision, as ADMM-a and ADMM-b have shown. On the other hand, however, the following two points should be concerned for the linearized ADMM. The first one is that the linearized subproblem are not the original ADMM subproblem, which will affect the performance of linearized ADMM. The second point is how to choose an appropriate value for \( r \) in \( x \)-subproblem (25) so as to enhance the convergence. While the inequalities (35) and (37) ensure the convergence of ADMM-a and ADMM-b, respectively, we really don’t know which value of \( r \) can result in fast convergence of two methods.

The mentioned two points thus inspire us to consider whether an ADMM-type method without the adoption of linearization technique can be developed for solving (21). Thanks to the favorable structure of (21), we can present our third ADMM-type method which does not adopt linearization technique for subproblems in the following.

First, similar to the variable \( y \) in (21) we introduce another auxiliary variable \( z \) to (21), then its formulation can be rewritten as

\[
\begin{align*}
\min_{x,y,z} & \quad \delta X(x) + \delta^*_B(y) \\
\text{s.t.} & \quad Az - b = y, \\
& \quad x - z = 0.
\end{align*}
\]  

(39)

It is clear that the formulation (39) is a three-block separable convex minimization problem whose variables are composed of \( x, y \) and \( z \).

The augmented Lagrangian function of (39) is written as

\[
L_\beta(x,y,z,\lambda) = \delta X(x) + \delta^*_B(y) - \lambda^T_1(Az - y - b) - \lambda^T_2(x - z) + \frac{\beta}{2} ||Az - y - b||^2 + \frac{\beta}{2} ||x - z||^2.
\]  

(40)

According to the augmented Lagrangian function (40), we can apply the ADMM for three-block separable problem to (39) which results in the following iteration scheme.

\[
\begin{align*}
x^{k+1} &= \text{Argmin}_x L_\beta(x, y^k, z^k, \lambda^k), \\
y^{k+1} &= \text{Argmin}_y L_\beta(x^{k+1}, y, z^k, \lambda^k), \\
z^{k+1} &= \text{Argmin}_z L_\beta(x^{k+1}, y^{k+1}, z, \lambda^k), \\
\lambda_1^{k+1} &= \lambda_1^k - \beta(Az^{k+1} - y^{k+1} - b), \\
\lambda_2^{k+1} &= \lambda_2^k - \beta(x^{k+1} - z^{k+1}).
\end{align*}
\]  

(41)

In the following we will discuss how to solve the subproblems of (41) in detail.

**I)** The variable \( x^{k+1} \). It follows from (40) and (41a) that \( x^{k+1} \) is the solution of the following optimization problem

\[
x^{k+1} = \text{Argmin}_x \delta X(x) - \lambda_2^k T(x - z^k) + \frac{\beta}{2} ||x - z^k||^2
\]  

(42)

Now this \( x \)-subproblem is quite simple since the coefficient of \( x \) in the quadratic term is the identity matrix, and thus it has the closed-form solution as follows

\[
x^{k+1} = P_X(z^k + \frac{1}{\beta} \lambda_2^k).
\]
The variable $y^{k+1}$. It is easy to see that $y^{k+1}$ satisfies

$$y^{k+1} = \arg\min_y \delta_{B^*}(y) - \lambda_1^k (A z^k - y - b) + \frac{\beta}{2} \| A z^k - y - b \|^2$$

$$= \arg\min_y \delta_{B^*}(y) + \frac{\beta}{2} \| y - A z^k + b \| + \frac{1}{\beta} \lambda_1^k \| 2$$

$$= \arg\min_y \delta_{B^*}(y) + \frac{\beta}{2} \| y - q \|^2,$$

where $q = A z^k - b - \frac{1}{\beta} \lambda_1^k$. Analogy to Proposition 2, using Moreau’s decomposition we can obtain the closed-form solution of $y$-subproblem as follows

$$y^{k+1} = q - \frac{1}{\beta} p_{B^*}(\beta q).$$

The variable $z^{k+1}$. We can get $z^{k+1}$ via solving

$$z^{k+1} = \arg\min_z - \lambda_1^k (A z - y^{k+1} - b) - \lambda_2^k (x^{k+1} - z) \quad (44)$$

$$+ \frac{\beta}{2} \| A z - y^{k+1} - b \|^2 + \frac{\beta}{2} \| x^{k+1} - z \|^2.$$ This is an unconstrained optimization problem with regard to $z$ and its optimal solution can be obtained by solving $g(z^{k+1}) = 0$, where $g(\cdot)$ is the gradient function of (44), i.e.,

$$- A^T \lambda_1^k + \lambda_2^k + \beta A^T (A z - y^{k+1} - b) + \beta (z - x^{k+1}) = 0.$$ Therefore, the solution to $z$-subproblem is given as

$$z^{k+1} = \frac{1}{\beta} (A^T A + I)^{-1} u,$$

where the vector $u = A^T \lambda_1^k - \lambda_2^k + \beta A^T (y^{k+1} + b) + \beta x^{k+1}$.

The Lagrange multipliers $\lambda_1^{k+1}$ and $\lambda_2^{k+1}$. The two multipliers are updated according to (41d) and (41e), respectively, which are trivial computation.

Now we are at the stage to present the following ADMM-type method, denoted as ADMM-c, to solve (19).

**Algorithm 3. ADMM-c for (19)**

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

$$x^{k+1} = P_X (z^k + \frac{1}{\beta} \lambda_2^k), \quad (45a)$$

$$y^{k+1} = q - \frac{1}{\beta} p_{B^*}(\beta q), \quad (45b)$$

$$z^{k+1} = \frac{1}{\beta} (A^T A + I)^{-1} u, \quad (45c)$$

$$\lambda_1^{k+1} = \lambda_1^k - \beta (A z^{k+1} - y^{k+1} - b), \quad (45d)$$

$$\lambda_2^{k+1} = \lambda_2^k - \beta (x^{k+1} - z^{k+1}), \quad (45e)$$

where $q = A z^k - b - \frac{1}{\beta} \lambda_1^k$ and $u = A^T \lambda_1^k - \lambda_2^k + \beta A^T (y^{k+1} + b) + \beta x^{k+1}$. 
With introducing a new auxiliary variable $z$ to the model (21), all subproblems of ADMM-c have closed-form solutions, as indicated by (45a)-(45e). Recall that the $x$-subproblem (24) is hard to solve due to the existence of the general matrix $A$. With the variable $z$ introduced, however, the matrix $A$ has moved from the $x$-subproblem (42) to the $z$-subproblem (44). While ADMM-c does not adopt the linearization technique as in ADMM-a and ADMM-b, both two subproblems (42) and (44) are now easy to solve. This idea contributes to the improvement and effectiveness of ADMM-c for GMFWP, which will be verified by the numerical results later.

One essential concern with regard to ADMM-c is its convergence. The proposed ADMM-c is the application of classical ADMM for three-block separable convex problem (39). A recent paper [4], however, has revealed that the convergence of classical ADMM for the generic three-block separable convex problem is not guaranteed, even though its numerical efficiency has been verified empirically by many satisfactory applications (e.g. [27, 33]). Fortunately, due to the favorable structure of (39), the proposed ADMM-c has global convergence, i.e., for arbitrary initial iterate $w_0$ the iterate sequence $\{w_k\}$ generated by ADMM-c is convergent to the optimal solution to (39). To see this, let us have a closer look at the iterative scheme of $x$ and $y$. First, we couple $x$ with $y$ to form the vector $(x, y)$. Based on the augmented Lagrangian function (40), we apply the ADMM for two-block convex problem, then the iterative scheme turns to be as follows.

\[
\begin{align*}
(x^{k+1}, y^{k+1}) &= \text{Argmin}_{x,y} L_{\beta}(x, y; z^k, \lambda^k), \\
z^{k+1} &= \text{Argmin}_z L_{\beta}(x^{k+1}, y^{k+1}, z), \\
\lambda_1^{k+1} &= \lambda_1^k - \beta(Az^{k+1} - y^{k+1} - b), \\
\lambda_2^{k+1} &= \lambda_2^k - \beta(x^{k+1} - z^{k+1}).
\end{align*}
\]

By dropping the terms irrelevant to $x$ and $y$, the subproblem (46a) can be written as

\[
(x^{k+1}, y^{k+1}) = \text{Argmin}_{x,y} \delta_X(x) + \delta_{B_d}(y) + \lambda_1^k y - \lambda_2^k x + \frac{\beta}{2} \|Az - y - b\|^2 + \frac{\beta}{2} \|x - z\|^2
\]

where $q$ is defined as in (43). It means that for $(x^{k+1}, y^{k+1})$ in (46a) the variables $x^{k+1}$ and $y^{k+1}$ are obtained by solving the $x$-subproblem and $y$-subproblem of ADMM-c independently. It follows that ADMM-c can be regarded as the classical ADMM for two-block separable convex problem, and thus its global convergence can be established as follows.
Theorem 4.3. For any given $\beta > 0$ and an arbitrary initial iterate $w^0 = (x^0, y^0, z^0, \lambda^0)$, the sequence $\{w^k\}$ converges to $w^* = (x^*, y^*, z^*, \lambda^*)$, where $x^*$ is the optimal solution to the primal problem (19).

Proof. Let $u = (x^T, y^T)^T$ and $u^* = (x^*T, y^*T)^T$. It follows from Theorem 1 in [18] that we have
\[
\lim_{k \to \infty} C(u^k - u^*) = 0, \quad \lim_{k \to \infty} D(z^k - z^*) = 0 \quad \text{and} \quad \lim_{k \to \infty} \lambda^k - \lambda^* = 0,
\]
(47)
where
\[
C = \begin{pmatrix} 0 & -I \\ I & 0 \end{pmatrix}, \quad D = \begin{pmatrix} A \\ -I \end{pmatrix},
\]
and $(x^*, y^*, z^*)$ is the optimal solution to (39).

According to (47), we can get
\[
\lim_{k \to \infty} C^T C(u^k - u^*) = 0 \quad \text{and} \quad \lim_{k \to \infty} D^T D(z^k - z^*) = 0.
\]
(48)
Note that
\[
C^T C = I \quad \text{and} \quad D^T D = I + A^T A,
\]
(49)
and thus
\[
\lim_{k \to \infty} u^k - u^* = 0 \quad \text{and} \quad \lim_{k \to \infty} z^k - z^* = 0.
\]
(50)
Therefore, the sequence $\{w^k\}$ converges to $w^* = (x^*, y^*, z^*, \lambda^*)$. Since $(x^*, y^*, z^*)$ is the optimal solution to (39), it is clear that $x^*$ is the optimal solution to (19).
The proof is complete.

Recall that as the linearization technique is adopted, the parameters $\beta$ and $r$ of ADMM-a and ADMM-b should satisfy some particular conditions to ensure the global convergence, e.g., $rI - \beta A^T A \succ 0$ for ADMM-a. Compared with ADMM-a and ADMM-b, the global convergence condition of ADMM-c is more loose than two previous ADMM-type methods.

5. Numerical results. This section reports some numerical results to verify the computational efficiency of ADMM-type methods. We will implement the proposed ADMM-a, ADMM-b and ADMM-c to solve a number of GMFWPs including unconstrained problems and constrained problems and also including problems with symmetric distances and asymmetric distances. To demonstrate the efficiency of ADMM-type methods, in the first subsection we compare ADMM-a, ADMM-b and ADMM-c with the well-known methods in facility location for MFWP and in the second subsection we compare them with some methods in the variational inequality area. The parameters of ADMM-type methods are given as follows: $\beta = 0.01$ and $r = \beta \cdot \lambda_{\text{max}}(A^T A) + 1$ for ADMM-a; $\beta = 0.01$ and $r = \beta \cdot \lambda(A^T A) + 10$ for ADMM-b, where $\lambda(\cdot)$ denotes the average of eigenvalues; $\beta_1 = 0.05$ (for the first constraint) and $\beta_2 = 50$ (for the second constraint) for ADMM-c. All the programming codes were written by MATLAB 9.0 and were run on an ASUS laptop (Intel Core i7-6700HQ 2.60GHz).

5.1. ADMM-type methods for MFWP. Miehle algorithm (MA) and hyperboloid approximation procedure (HAP) are the most important methods for solving MFWP. This paper proposes ADMM-type methods to solve GMFWP. Note that MFWP, as a classical model in facility location, is a special case of GMFWP.
Hence, ADMM-type methods are applicable to solving MFWP. In the following, we will compare ADMM-a, ADMM-b and ADMM-c with MA and HAP for solving MFWP and show that even for such a special problem the proposed ADMM-type methods are efficient.

First we compare ADMM-a, ADMM-b and ADMM-c with MA. The following two examples were constructed in [31].

**E1 and E2:** Example 1 (E1) and Example 2 (E2) in [31] (for more details please refer to [31]).

It has been shown in [31] that the iterate sequences generated by MA stop at a nonoptimal point of E1 and converge to a nonoptimal point of E2. This means that the convergence of MA to the optimal solution is not guaranteed.

We further construct the following two examples.

**E1’ and E2’:** E1 and E2 with the initial iterate \(x_1^0\) and \(x_2^0\) chosen as the locations of two customers.

As the location of customers are chosen as initial iterates in \(E1’\) and \(E2’\), the undesirable singularity will occur. The two examples are generated to show the difference of the performance of ADMM-type methods with MA in case of singularity.

Note that the location of customers, the weights and the initial iterates are particularly chosen in \(E1, E2, E1’, \) and \(E2’\). For well justification of the efficiency of ADMM-type methods, we construct the following randomly generated problems.

**E3:** \(E1\) (E2) but all customers \(a_j\) and the initial iterate \(x_1^0, x_2^0\) are randomly generated in \([-250, 250]^2\), \(w_{ij}\) and \(v_{il}\) are randomly generated in \([1, 100]\).

Note that MA is a Jacobi-like method because the locations of new facilities change at the same time. An alternative method potentially accelerates the convergence of MA by applying new information of facility locations once it becomes available. This leads to the so-called Gauss-Seidel-like Miehle algorithm (GMA).

We solve the five examples by MA, GMA, ADMM-a, ADMM-b and ADMM-c and the numerical results are reported in Table 1, where “CPU” means the computing time, “Iter.” gives the number of iterations and “Obj.” reports the objective functional values. The stopping criterion is chosen as \(\|x^{k+1} - x^k\| < 10^{-6}\).

Each example is solved by five methods with the same initial iterate \(x_1^0\) and \(x_2^0\). The initial \(y^0, z^0\) and \(\lambda^0\) in ADMM-a, ADMM-b and ADMM-c are zero. For \(E1’, E2’\) and \(E3\), we test each example for one hundred scenes with each scene solved by five methods and report the average results.

For \(E1\), MA achieves a point, \((1.50591 \text{ 0.00000})\) and \((1.50591 \text{ 0.00000})\), in the first iteration and then stops at it from the second iteration. However, it is not the optimal solution of \(E1\), which is \((0,0)\) and \((0,0)\). When GMA is applied to solve \(E1\), the point obtained by GMA, \((1.65749,0)\) and \((1.65749,0)\), is also not the optimal one. In fact, this point is even worse than that from MA. While ADMM-a, ADMM-b and ADMM-c need more iterations for \(E1\), they can get their optimal solution and provide the optimal objective functional value. For \(E2\), with the adoption of the Gauss-Seidel idea GMA needs much less number of iterations than MA. When we compare the points obtained by five methods, ADMM-a, ADMM-b and ADMM-c provide the optimal solution but the other two methods do not.

For \(E1’\) and \(E2’\), the undesirable singularity occurs. Thus, MA and GMA cannot proceed the iterations and then their numerical results are labeled as “/” in
Table 1. MA, GMA, ADMM-a, ADMM-b and ADMM-c for MFWP

| Problem | CPU | MA  | GMA  | ADMM-a | ADMM-b | ADMM-c |
|---------|-----|-----|------|--------|--------|--------|
| E1      |     | 0.00003 | 0.00005 | 0.0117 | 0.0096 | 0.0070 |
| Iter.   | 2   | 9   | 240  | 135    | 62     |
| Obj.    | 182.23 | 182.43 | 181.42 | 181.42 | 181.42 |
| E2      |     | 0.0069 | 0.0004 | 0.0127 | 0.0097 | 0.0095 |
| Iter.   | 206 | 5   | 240  | 135    | 82     |
| Obj.    | 182.19 | 182.68 | 181.42 | 181.42 | 181.42 |
| $E1'$   |     |     |      |        |        |        |
| CPU     |     | 0.0125 | 0.0084 | 0.0081 |
| Iter.   | /   | /   | 244.99 | 118.45 | 70.28 |
| Obj.    |     |     | 181.42 | 181.42 | 181.42 |
| $E2'$   |     |     |      |        |        |        |
| CPU     |     | 0.0117 | 0.0076 | 0.0073 |
| Iter.   | /   | /   | 244.31 | 117.05 | 69.32 |
| Obj.    |     |     | 181.42 | 181.42 | 181.42 |
| E3      |     | 0.1774 | 0.1670 | 0.0095 | 0.0109 | 0.0092 |
| Iter.   | 4502.30 | 56.70 | 203.15 | 168.30 | 89.10 |
| Obj.    | 65724.12 | 68142.84 | 65317.20 | 65317.20 | 65317.19 |

Table 1. ADMM-a, ADMM-b and ADMM-c, however, are able to deal with the singularity and in such case ADMM-a, ADMM-b and ADMM-c can still get the optimal solution, which is consistent with the analytic results.

For the randomly generated $E3$, it is found that the solution obtained by GMA is the worst among five methods. MA and GMA need more than 15 times computing time than proposed ADMM-type methods, but the points are still not the optimal one of $E3$. Compared to MA and GMA, ADMM-a, ADMM-b and ADMM-c can get the optimal solution of $E3$.

Next we will compare ADMM-a, ADMM-b and ADMM-c with HAP. An alternative of HAP using the new available information was proposed in [30]. This alternative is called Gauss-Seidel-like HAP (GHAP). The parameter $\varepsilon$ is chosen as 5, 0.5, 0.05, respectively, and we report the numerical results of HAP and GHAP under different values of $\varepsilon$. SHAP successively reduces the value of $\varepsilon$ during its implementation to accelerate the convergence of HAP under small value of $\varepsilon$. This technique is also applicable to GHAP and GHAP with such technique is called SGHAP in this paper. For SHAP and SGHAP, the $\varepsilon$ is reduced from 5 to 0.05 with its value multiplied by 0.1 each time. For convenience, we call HAP, GHAP, SHAP and SGHAP as HAP-type methods.

We test MFWP with $n = 500$ and $m = 2, 4, 6, 8, 10$. All customers are randomly generated in $[-250, 250]^2$; the weights between customers and facilities are randomly chosen in $[1, 5]$ and the weights between facilities are chosen as 100. The stopping criterion is set as

$$\|x^{k+1} - x^k\| < 10^{-4}.$$

The initial iterate is randomly generated in $[-250, 250]^2$. For each pair $(n, m)$, we solve 100 randomly generated MFWPs by HAP, GHAP, SHAP, SGHAP, ADMM-a, ADMM-b and ADMM-c using the same randomly generated customers and initial iterates. The numerical results are reported in Table 2, where “CPU” and “Iter.” are as the same as those in Table 1 and “Dobj.” reports the difference of objective functional values between ADMM-a and other methods. Actually the value of “Dobj.” is the functional value of other methods minus that of ADMM-a. Therefore, if this value is positive, the point obtained by other method is worse than that by
ADMM-TYPE METHODS FOR GMFWP

| m | CPU | T | Iter. | HAP | GHAP | SHAP | SGHAP | ADMM-a | ADMM-b | ADMM-c |
|---|-----|---|------|-----|------|------|-------|--------|--------|--------|
| 2 | CPU | 1.3522 | 1.5621 | 0.7419 | 1.3930 | 4.9024 | 2.8907 | 1.6763 | 0.0784 | 0.9984 | 0.9964 |
| 2 | ADMM-a | 25.92 | 2.14 | 2.14 | 2.14 | 0 | -0.66 | -0.45 |
| 4 | CPU | 3.4443 | 8.6824 | 23.9782 | 1.7648 | 4.6689 | 12.6427 | 4.2423 | 2.9516 | 0.1061 | 0.1041 |
| 4 | ADMM-a | 1.1054 | 15.81 | 17.81 | 15.81 | 0 | -1.35 |
| 6 | CPU | 4.1408 | 14.4112 | 44.5874 | 2.6341 | 4.1586 | 23.5066 | 4.9028 | 3.8909 | 0.1092 | 0.1079 |
| 6 | ADMM-a | 61.00 | 1666.40 | 4485.80 | 2337.60 | 490.80 | 384.80 | 197.44 | 260.48 | 118.38 |
| 8 | CPU | 73.83 | 73.85 | 22.29 | 2.3383 | 73.99 | 22.29 | 22.28 | 0 | -0.30 | -1.47 |
| 8 | ADMM-a | 25.91 | 24.09 | 24.09 | 24.09 | 24.09 | 24.09 | 24.09 | 24.09 | 24.09 |
| 10 | CPU | 9.8884 | 21.2481 | 7.6862 | 2.2726 | 14.7744 | 46.6910 | 12.9670 | 7.7526 | 0.1780 | 0.2480 | 0.1068 |
| 10 | ADMM-a | 27.74 | 24.09 | 24.09 | 24.09 | 24.09 | 24.09 | 24.09 | 24.09 | 24.09 |

ADMM-a; otherwise, it is better. It is clear that the smaller the value of “Dobj.”, the better the point obtained by the method.

According to the last three columns in Table 2, it can be seen that the performances of ADMM-a, ADMM-b and ADMM-c are quite similar, and ADMM-c outperforms ADMM-a and ADMM-b a little.

When we compare ADMM-a, ADMM-b and ADMM-c with four HAP-type methods, it is easy to observe that ADMM-a, ADMM-b and ADMM-c are preferable to HAP-type methods especially for large-scale problems. Firstly, we compare them from the quality of obtained points. We have known that the points obtained by ADMM-a, ADMM-b and ADMM-c are almost the optimal solution of MFWP. However, the points obtained by HAP-type methods are clearly not the optimal solution, which is indicated by the positive values of “Dobj.”. Secondly, we compare five methods from the computing workload. It is clear that the workloads of ADMM-a, ADMM-b and ADMM-c are less than HAP-type methods especially for large-scale MFWP. With the value of m increasing, the increment speed of workload of HAP-type methods is much faster than ADMM-type methods. In addition, we also expect to decrease ε of HAP-type methods to improve the quality of obtained points. However, it is found during our experiments that though the quality of obtained points is improved, the computing time and number of iterations increase drastically with smaller ε.

Thus, with the comparison of ADMM-a, ADMM-b and ADMM-c to some well-known methods in facility location area, their efficiency is verified.

5.2. ADMM-type methods for GMFWP. In this subsection we will apply ADMM-type methods to solve GMFWP (7). Recently in the paper [19], GMFWP is reformulated into a linear variational inequality (LVI). Note that there exist some methods for LVI, which are also applicable to solving GMFWP. To verify the efficiency of ADMM-a, ADMM-b and ADMM-c, we compare them with some methods for LVI to solve GMFWP. The first method to be compared is the projection and contraction (PC) method presented in [17]:

$$
\begin{align*}
    z^{k+1} &= P_{B^*}(z^k + \gamma \alpha_k (AP_X (x^k - A^T z^k) - b)) , \\
    x^{k+1} &= P_X (x^k - \gamma \alpha_k A^T P_{B^*}(Ax^k - b + z^k)) ,
\end{align*}
$$

where $B = I - \mu A^T A$ and $\gamma \in (0, 1)$.
where $\gamma$ is a positive constant. The second one is the customized proximal point algorithm (PPA) proposed in [16]:

$$
\begin{align*}
\tilde{z}^k &= P_{B_d} \left( z^k + \frac{Ax^k - b}{s} \right), \\
\tilde{x}^k &= P_X \left( x^k - \frac{2AT\tilde{z}^k - ATz^k}{r} \right), \\
u^{k+1} &= u^k - \gamma(u^k - \tilde{u}^k),
\end{align*}
$$

where $r$, $s$ and $\gamma$ are positive constants. The convergence of PPA under the condition of $rs > ||ATA||$ and $\gamma \in (0, 2)$ can be found in [16].

The paper [19] proposed a new projection-type method under the LVI framework for GMFWP, whose iterative scheme is as follows.

$$
\begin{align*}
\tilde{z}^k &= P_{B_d} \left( z^k + \frac{Ax^k - b}{s} \right), \\
\tilde{x}^k &= P_X \left( x^k - \frac{AT\tilde{z}^k}{r} \right), \\
u^{k+1} &= u^k - \gamma W(u^k - \tilde{u}^k),
\end{align*}
$$

where $W$ is an invertible matrix and $r$, $s$ and $\gamma$ are positive constants. Let the matrix $Q$ defined as

$$
Q = \begin{pmatrix} rI & 0 \\ A & sI \end{pmatrix}.
$$

If $H := QW^{-1} > 0$ and $G := Q^T + Q - \gamma W^T HW > 0$, then the projection-type method converges [19].

The parameters of the above three methods are given as follows.

- PC method: $\gamma = 1.9$;
- PPA: $r = p \cdot \sqrt{t||ATA||}$, $s = \frac{2}{p} \cdot \sqrt{ts||ATA||}$, $t = 1.01$, $p = 2$, $\gamma = 1.2$;
- Projection-type method: $W = \begin{pmatrix} I & 0 \\ A/s & I \end{pmatrix}$, $\gamma = 1$, $r = p \cdot \sqrt{t||ATA||}$, $s = \frac{1}{p} \cdot \sqrt{t||ATA||}$, $t = 1.01$, $p = 2$.

We first test GMFWP under Euclidean distances with $n = 50, 100, 500, 1000, 2000$ and $m = 2, 4, 6, 8, 10$. All customers are randomly generated in $[-250, 250]^2$; the weights between customers and facilities are randomly chosen in $[1, 3]$ and the weights between facilities are chosen in $[1, 5]$. The locational constraints 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 $[10, 15]$, $i = 1, \cdots, m$. The initial iterate is randomly generated in $[-250, 250]^2$ and the stopping criterion is given as

$$
||x^{k+1} - x^k|| < 10^{-4}.
$$

For each given pair $(n, m)$, we test the GMFWPs for 50 scenes with each scene solved by six methods using the same randomly generated initial iterates. Table 3 reports the average computing time in seconds (CPU) and the average number of iterations (Iter.) of six methods to solve the GMFWPs under Euclidean distances (symmetric distances).
Table 3. Numerical results for GMFWP under Euclidean distances

| n   | m   | PC method | PPA | Projection | ADMM-a | ADMM-b | ADMM-c |
|-----|-----|-----------|-----|-----------|--------|--------|--------|
|     |     | CPU | Iter. | CPU | Iter. | CPU | Iter. | CPU | Iter. | CPU | Iter. |
| 2   | 0.0206 | 34.08 | 0.0077 | 29.10 | 0.0077 | 28.72 | 0.0017 | 24.58 | 0.0031 | 24.52 | 0.0085 | 78.36 |
| 4   | 0.0414 | 39.46 | 0.0010 | 34.02 | 0.0010 | 32.20 | 0.0004 | 30.10 | 0.0009 | 25.38 | 0.0194 | 83.76 |
| 6   | 0.0808 | 54.34 | 0.0287 | 44.24 | 0.0286 | 42.82 | 0.0066 | 41.94 | 0.0109 | 44.18 | 0.0314 | 111.10 |
| 8   | 0.1847 | 59.34 | 0.0407 | 44.12 | 0.0411 | 42.22 | 0.0114 | 42.32 | 0.0142 | 47.78 | 0.0672 | 123.04 |
| 10  | 0.3244 | 74.12 | 0.0601 | 58.28 | 0.0622 | 61.02 | 0.0189 | 78.34 | 0.0437 | 88.12 | 0.0996 | 157.16 |
| 100 | 0.0803 | 19.00 | 0.0139 | 30.84 | 0.0137 | 30.42 | 0.0021 | 30.72 | 0.0065 | 38.80 | 0.0083 | 64.70 |
| 500 | 0.1244 | 25.90 | 0.0378 | 33.28 | 0.0377 | 32.02 | 0.0083 | 45.38 | 0.0144 | 56.46 | 0.0210 | 66.42 |
| 1000| 0.2345 | 39.10 | 0.0602 | 34.78 | 0.0522 | 32.36 | 0.0113 | 46.92 | 0.0217 | 51.34 | 0.0391 | 79.92 |
| 2000| 0.2592 | 44.92 | 0.0743 | 37.16 | 0.0706 | 34.54 | 0.0102 | 45.78 | 0.0283 | 55.96 | 0.0532 | 73.00 |

According to Table 3, six methods can solve the randomly generated GMFWPs under Euclidean distance and with locational constraints efficiently. It is shown that the proposed ADMM-type methods are preferable to the PC method, PPA and projection-type methods for solving the GMFWPs of all scales. For large-scale GMFWP, the workload of ADMM-type methods is only about 10% that of other methods. In addition, it is also found that ADMM-c is preferable to ADMM-a and ADMM-b for large-scale problems.

Since ADMM-type methods can solve the GMFWP whose distances are measured by asymmetric gauge, we test some randomly generated GMFWPs under gauge with different scales. The asymmetric gauge \( \gamma(\cdot) \) is generated with the unit ball set as \((x + 2)^2 + 2y^2 \leq 8\).

To verify the efficiency of ADMM-type methods for GMFWP under gauge, we also compare ADMM-a, ADMM-b and ADMM-c with the other three methods simultaneously. The performance of six methods is reported in Table 4.

It is easy to observe that the proposed ADMM-a, ADMM-b and ADMM-c are efficient for GMFWP under gauge and they outperform the PC method, PPA and projection-type method for the randomly generated asymmetric problem.

When we compare the performance of three ADMM-type methods for large-scale problems in Table 3 and Table 4, we found that ADMM-c is a little better than the other two methods in Table 3 but much better than them in Table 4. This implies that ADMM-c is robust to the measuring distance function (norm or gauge). Moreover, it follows from Table 3-4 that the computing time and number of iterations of three ADMM-type methods is not so sensitive to the size of GMFWP. All of these desirable properties contribute to making the proposed ADMM-type methods especially ADMM-c promising in practice. According to Table 3-4, we also
| n | m | PC method | PPA | Projection | ADMM-a | ADMM-b | ADMM-c |
|---|---|-----------|-----|-----------|--------|--------|--------|
|   |   | CPU | Iter. | CPU | Iter. | CPU | Iter. | CPU | Iter. |
| 50 | 2 | 0.0226 | 26.76 | 0.0125 | 32.02 | 0.0094 | 23.56 | 0.0862 | 69.28 | 0.0072 | 69.74 | 0.0102 | 131.80 |
| 4 | 0.1120 | 54.80 | 0.0394 | 58.72 | 0.0352 | 51.36 | 0.0112 | 77.88 | 0.0172 | 89.92 | 0.0306 | 146.72 |
| 8 | 0.0433 | 60.60 | 0.0600 | 69.42 | 0.0522 | 34.18 | 0.0241 | 105.90 | 0.0370 | 115.60 | 0.0615 | 145.30 |
| 10 | 0.4527 | 79.68 | 0.1171 | 67.64 | 0.1048 | 85.74 | 0.0664 | 121.54 | 0.0874 | 122.90 | 0.1414 | 159.46 |
| 100 | 2 | 0.0345 | 91.94 | 0.0241 | 34.18 | 0.0182 | 28.13 | 0.0079 | 85.28 | 0.0099 | 89.46 | 0.0109 | 106.30 |
| 4 | 0.0511 | 77.74 | 0.0426 | 38.96 | 0.0357 | 28.28 | 0.0164 | 90.60 | 0.0206 | 94.42 | 0.0201 | 67.06 |
| 6 | 0.0269 | 30.30 | 0.0852 | 48.28 | 0.0767 | 49.06 | 0.0382 | 111.66 | 0.0422 | 114.86 | 0.0425 | 59.42 |
| 8 | 0.3399 | 90.12 | 0.1242 | 51.00 | 0.0974 | 40.60 | 0.0491 | 132.14 | 0.0706 | 134.12 | 0.0680 | 50.44 |
| 10 | 0.4320 | 42.00 | 0.1546 | 52.70 | 0.1249 | 32.26 | 0.0695 | 138.94 | 0.1074 | 139.06 | 0.0905 | 106.70 |
| 500 | 2 | 0.1886 | 8.30 | 0.0808 | 31.66 | 0.0695 | 26.06 | 0.0206 | 95.26 | 0.0218 | 98.42 | 0.0122 | 49.72 |
| 4 | 0.3542 | 8.46 | 0.1718 | 32.98 | 0.1539 | 27.34 | 0.0431 | 100.30 | 0.0509 | 102.48 | 0.0399 | 40.48 |
| 6 | 0.1961 | 8.86 | 0.3502 | 40.86 | 0.2628 | 34.32 | 0.0580 | 114.80 | 0.0615 | 117.78 | 0.0654 | 45.94 |
| 8 | 2.0873 | 14.38 | 0.5367 | 49.78 | 0.3985 | 45.96 | 0.1230 | 124.48 | 0.1375 | 127.12 | 0.0854 | 54.44 |
| 10 | 5.7430 | 19.54 | 0.7466 | 55.32 | 0.5923 | 32.42 | 0.1628 | 134.62 | 0.1987 | 135.96 | 0.1459 | 69.00 |
| 1000 | 2 | 0.3517 | 8.28 | 0.1729 | 32.26 | 0.1312 | 23.92 | 0.0363 | 75.32 | 0.0344 | 80.06 | 0.0202 | 33.08 |
| 4 | 2.1254 | 11.90 | 0.3249 | 31.55 | 0.3915 | 26.67 | 0.0743 | 120.12 | 0.0899 | 120.38 | 0.0497 | 40.06 |
| 6 | 0.3411 | 22.54 | 0.6087 | 50.74 | 0.6334 | 39.40 | 0.1279 | 111.23 | 0.1378 | 113.46 | 0.0950 | 49.84 |
| 8 | 8.2144 | 24.80 | 1.2100 | 55.32 | 0.9067 | 44.00 | 0.1770 | 120.34 | 0.1971 | 120.72 | 0.1256 | 59.38 |
| 10 | 30.6992 | 25.24 | 1.6418 | 59.94 | 1.6451 | 42.04 | 0.2456 | 130.70 | 0.2699 | 130.74 | 0.1729 | 54.84 |
| 2000 | 2 | 1.9512 | 9.20 | 0.3199 | 25.10 | 0.2638 | 19.80 | 0.0647 | 61.30 | 0.0595 | 61.70 | 0.0458 | 32.20 |
| 4 | 8.0222 | 10.30 | 1.1069 | 35.60 | 0.8122 | 34.20 | 0.1408 | 87.10 | 0.1436 | 91.40 | 0.0816 | 35.00 |
| 6 | 34.8110 | 17.70 | 7.9293 | 75.50 | 3.9660 | 52.90 | 0.2665 | 121.90 | 0.2821 | 122.70 | 0.1948 | 57.90 |
| 8 | 73.2160 | 19.19 | 10.1599 | 61.50 | 2.0928 | 51.00 | 0.3592 | 126.90 | 0.4011 | 127.70 | 0.2537 | 59.00 |
| 10 | 144.2117 | 19.40 | 14.8099 | 71.50 | 3.2529 | 67.70 | 0.5460 | 163.30 | 0.5600 | 170.80 | 0.3418 | 67.50 |

found that due to the asymmetric property of gauge the computational load of six methods for GMFWP under gauge are much greater than those for GMFWP under Euclidean distances.

6. Conclusion. A generalized multi-facility Weber problem is investigated in this paper. This problem considers the locational constraints on facilities and employs the gauge to measure the distances. With the consideration of the primal problem of reformulated minmax problem, some ADMM-type methods are proposed for solving the generalized problem. The convergence of proposed methods is established under mild assumption. Preliminary numerical results justify the preferable merits of proposed ADMM-type methods.

How to solve other problems arising in facility location computationally by the ADMM approach deserves further and more extensive investigation in our future research.

Acknowledgments. The authors thank Professor Defeng Sun for valuable suggestions.

REFERENCES

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers, Now Foundations and Trends, 2011.
[2] E. Carrizosa, E. Conde, M. Muñoz-Márquez and J. Puerto, Simpson points in planar problems with locational constraints. The polyhedral-gauge case, Math. Oper. Res., 22 (1997), 291–300.
[3] M. Cera, J. A. Mesa, F. A. Ortega and F. Plastria, Locating a central hunter on the plane, J. Optim. Theory Appl., 136 (2008), 155–166.
[4] C. Chen, B. He, Y. Ye and X. Yuan, The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent, Math. Program., 155 (2016), 57–79.
[5] Y.-H. Dai, D. Han, X. Yuan and W. Zhang, A sequential updating scheme of the Lagrange multiplier for separable convex programming, *Math. Comp.*, 86 (2017), 315–343.

[6] F. Daneshzand and R. Shoelhe, Multifacility location problem, in *Facility Location*, Contributions to Management Science, Physica, Heidelberg, 2009, 69–92.

[7] Z. Drezner, *Facility Location. A Survey of Applications and Methods*, Springer Series in Operations Research, Springer-Verlag, New York, 1995.

[8] Z. Drezner and H. W. Hamacher, *Facility Location. Applications and Theory*, Springer-Verlag, Berlin, 2002.

[9] J. W. Eyster, J. A. White and W. W. Wierwille, On solving multifacility location problems using a hyperboloid approximation procedure, *AIIE Trans.*, 5 (1973), 1–6.

[10] M. Fortin and R. Glowinski, On decomposition-coordination methods using an augmented Lagrangian, in *Studies in Mathematics and Its Applications*, 15, Elsevier, Amsterdam, 1983, 97–146.

[11] D. Gabay and B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, *Comput. Math. Appl.*, 2 (1976), 17–40.

[12] R. Glowinski, *Lectures on Numerical Methods for Non-linear Variational Problems*, Research Lectures on Mathematics and Physics, 65, Tata Institute of Fundamental Research, Bombay; sh Springer-Verlag, Berlin-New York, 1980.

[13] R. Glowinski and A. Marrocco, Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité, d’une classe de problèmes de Dirichlet non linéaires, *Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Rouge Anal. Numér.*, 9 (1975), 41–76.

[14] K. Guo, D. Han and X. Yuan, Convergence analysis of Douglas–Rachford splitting method for “strongly + weakly” convex programming, *SIAM J. Numer. Anal.*, 55 (2017), 1549–1577.

[15] B. He, F. Ma and X. Yuan, Optimally linearizing the alternating direction method of multipliers for convex programming, *Comput. Optim. Appl.*, 75 (2020), 361–388.

[16] B. He, X. Yuan and W. Zhang, A customized proximal point algorithm for convex minimization with linear constraints, *Comput. Optim. Appl.*, 56 (2013), 559–572.

[17] B. S. He, A modified projection and contraction method for a class of linear complementarity problems, *J. Comput. Math.*, 14 (1996), 54–63.

[18] J. Jiang, S. Zhang, Y. Lv, X. Du and Z. Yan, An ADMM-based location-allocation algorithm for nonconvex constrained multi-source Weber problem under gauge, *J. Global Optim.*, 76 (2020), 793–818.

[19] J. Jiang, S. Zhang, S. Zhang and J. Wen, A variational inequality approach for constrained multifacility Weber problem under gauge, *J. Ind. Manag. Optim.*, 14 (2018), 1085–1104.

[20] I. N. Katz and S. R. Vogl, A Weiszfeld algorithm for the solution of an asymmetric extension of the generalized Fermat location problem, *Comput. Math. Appl.*, 59 (2010), 399–410.

[21] X. Li, D. Sun and K.-C. Teh, A Schur complement based semi-proximal ADMM for convex quadratic conic programming and extensions, *Math. Program.*, 155 (2016), 333–373.

[22] Z. Lin, R. Liu and H. Li, Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning, *Mach. Learn.*, 99 (2015), 287–325.

[23] R. F. Love and J. G. Morris, Mathematical models of road travel distances, *Manag. Sci.*, 25 (1979), 117–210.

[24] W. Miehle, *Link-length minimization in networks*, *Operations Res.*, 6 (1958), 232–243.

[25] H. Minkowski, *Theorie der Konvexen Körper, Gesammelte Abhandlungen*, Teubner, Berlin, 1911.

[26] L. M. Ostresh, *The multifacility location problem: applications and descent theorems*, *J. Regional Sci.*, 17 (2006), 409–419.

[27] Y. Peng, A. Ganesh, J. Wright, W. Xu and Y. Ma, RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 2010.

[28] F. Plastria, *Asymmetric distances, semidirected networks and majority in Fermat-Weber problems*, *Ann. Oper. Res.*, 167 (2009), 121–155.

[29] F. Plastria, *The Weiszfeld algorithm: Proof, amendments, and extensions*, in *Foundations of Location Analysis*, International Series in Operations Research & Management Science, 155, Springer, New York, 2011, 357–389.

[30] J. B. Rosen and G. L. Xue, On the convergence of a hyperboloid approximation procedure for the perturbed Euclidean multifacility location problem, *Oper. Res.*, 41 (1993), 1164–1171.
[31] J. B. Rosen and G. L. Xue, On the convergence of Michle’s algorithm for the Euclidean multifacility location problem, *Oper. Res.*, 40 (1992), 188–191.
[32] D. Sun, K.-C. Toh and L. Yang, A convergent 3-block semiproximal alternating direction method of multipliers for conic programming with 4-type constraints, *SIAM J. Optim.*, 25 (2015), 882–915.
[33] M. Tao and X. Yuan, Recovering low-rank and sparse components of matrices from incomplete and noisy observations, *SIAM J. Optim.*, 21 (2011), 57–81.
[34] X. Wang and X. Yuan, The linearized alternating direction method of multipliers for Dantzig selector, *SIAM J. Sci. Comput.*, 34 (2012), A2792–A2811.
[35] J. E. Ward and R. E. Wendell, Using block norms for location modeling, *Oper. Res.*, 33 (1985), 1074–1090.
[36] E. Weiszfeld, Sur le point pour lequel la somme des distances de n points donnés est minimum, *Tohoku Math. J.*, 43 (1937), 355–386.
[37] Z. Wen, D. Goldfarb and W. Yin, Alternating direction augmented Lagrangian methods for semidefinite programming, *Math. Program. Comput.*, 2 (2010), 203–230.
[38] C. Witzgall, *Optimal Location of a Central Facility*, Mathematical Models and Concepts, Report 8388, National Bureau of Standards, Washington D.C., 1964.
[39] X. Zhang, M. Burger and S. Osher, A unified primal-dual algorithm framework based on Bregman iteration, *J. Sci. Comput.*, 46 (2011), 20–46.

Received July 2020, revised September 2020.

E-mail address: nuazhangs@126.com
E-mail address: jiangjianlin@nuaa.edu.cn
E-mail address: zhangsu@nankai.edu.cn
E-mail address: yibinglv@yangtzeu.edu.cn
E-mail address: guoyuzhen@nuaa.edu.cn