J Glob Optim (2011) 50:713–728
DOI 10.1007/s10898-010-9632-7

A continuation approach for the capacitated multi-facility weber problem based on nonlinear SOCP reformulation

Jein-Shan Chen · Shaohua Pan · Chun-Hsu Ko

Received: 3 September 2009 / Accepted: 4 December 2010 / Published online: 24 December 2010

Abstract We propose a primal-dual continuation approach for the capacitated multi-facility Weber problem (CMFWP) based on its nonlinear second-order cone program (SOCP) reformulation. The main idea of the approach is to reformulate the CMFWP as a nonlinear SOCP with a nonconvex objective function, and then introduce a logarithmic barrier term and a quadratic proximal term into the objective to construct a sequence of convexified subproblems. By this, this class of nondifferentiable and nonconvex optimization problems is converted into the solution of a sequence of nonlinear convex SOCPs. In this paper, we employ the semismooth Newton method proposed in Kanzow et al. (SIAM Journal of Optimization 20:297–320, 2009) to solve the KKT system of the resulting convex SOCPs. Preliminary numerical results are reported for eighteen test instances, which indicate that the continuation approach is promising to find a satisfying suboptimal solution, even a global optimal solution for some test problems.

Keywords Capacitated multi-facility Weber problem · Nondifferentiable · Nonconvex · Second-order cone program · Semismooth Newton method

J.-S. Chen—Member of Mathematics Division, National Center for Theoretical Sciences, Taipei Office. The author's work is partially supported by National Science Council of Taiwan.
S. Pan—The author's work is supported by the Fundamental Research Funds for the Central Universities (SCUT), Guangdong Natural Science Foundation (No. 9251802902000001) and National Young Natural Science Foundation (No. 10901058).

J.-S. Chen (✉)
Department of Mathematics, National Taiwan Normal University, 11677 Taipei, Taiwan
e-mail: jschen@math.ntnu.edu.tw

S. Pan
School of Mathematical Sciences, South China University of Technology, 510640 Guangzhou, China
e-mail: shhpan@scut.edu.cn

C.-H. Ko
Department of Electrical Engineering, I-Shou University, 840 Kaohsiung, Taiwan
e-mail: chko@isu.edu.tw

1 Introduction

There are many different kinds of facility location problems for which various methods have been proposed; see [10,15,21,20,22,23,26,32,35,36] and references therein. The web-site maintained by EWGLA (Euro Working Group on Location Analysis) also serves as a good resource to look for related literature including survey papers, books, and journals. In this paper, we consider the newer but more difficult capacitated multi-facility Weber problem (CMFWP) which plays an important role in the operation and management science.

The CMFWP, also called the capacitated Euclidean distance location-allocation problem in other contexts, is concerned with locating a set of facilities and allocating their capacity to satisfy the demand of a set of customers with known locations so that the total transportation cost is minimized. Supply centers such as plants and warehouses may constitute the facilities, while retailers and dealers may be considered as customers. The mathematical model of CMFWP can be stated as follows:

$$\begin{aligned} \min \ & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} w_{ij} \|x_i - a_j\| \\ \text{s.t.} \ & \sum_{j=1}^{n} w_{ij} = s_i, \quad i = 1, 2, \ldots, m \\ & \sum_{i=1}^{m} w_{ij} = d_j, \quad j = 1, 2, \ldots, n \\ & w_{ij} \geq 0, \ \ i = 1, 2, \ldots, m; \ \ j = 1, 2, \ldots, n \\ & x_i \in \mathbb{R}^2, \ \ i = 1, 2, \ldots, m. \end{aligned} \tag{1}$$

In the formulation of CMFWP, m is the number of facilities to be located, n is the number of customers, s_i is the capacity of facility i, and d_j is the demand of customer j. Throughout this paper, we without loss of generality assume that the total capacity of all facilities equals the total demand of all customers, i.e.,

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j. \tag{2}$$

In addition, the allocations w_{ij} are unknown variables denoting the amount to be shipped from facility i to customer j with the unit shipment cost per unit distance being c_{ij}. If all w_{ij} are known, the CMFWP reduces to the traditional convex multi-facility location problem for which many efficient algorithms (see [10,13,22,33,34,26]) have been proposed, whereas if all x_i are fixed, it reduces to the ordinary transportation problem. For the sake of notation, in the sequel, we denote by $x_i = (x_{i1}, x_{i2})$ the unknown coordinates of facility i, and by $a_j = (a_{j1}, a_{j2})$ the given coordinates of customer j.

We see that the objective function of (1) is nondifferentiable at any points where $x_i = a_j$ for some $i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, n\}$, which precludes a direct application of effective gradient-based methods for finding its solution. Also, it is nonconvex since w_{ij} are unknown decision variables. Therefore, the CMFWP belongs to a class of nondifferentiable nonconvex optimization problems subject to $(m + n)$ linear constraints and mn nonnegativity constraints. In fact, Sherali and Nordai [30] have shown that this class of problems is NP-hard, even if all demand points a_j are located on a straight line.

For this class of problems, Cooper in his seminal work [7] first proposed an exact solution method by explicit enumeration of all extreme points of the transportation polytope, defined by the first three groups of constraints of (1). Selim [29] later presented a biconvex cutting plane procedure in his unpublished dissertation. The exact solution method, similar to Cooper's complete enumeration, can effectively deal with very small instances only. Recently, Sherali and Nordai [28] developed a branch-and-bound algorithm which is based on a partitioning of all the allocation space and finitely converges to a global optimum within a specified percentage tolerance. Apart from these exact methods, some heuristic methods have also been proposed by the reformulation linearization technique [27] or an approximating mixed integer linear programming formulation [1]. We observe that most of these methods are combinatorial primal ones which are designed by exploiting the structure of the problem itself and do not provide any information about the dual solution.

In this paper, we propose a continuous primal-dual approach by converting the CMFWP into the solution of a sequence of nonlinear convex second-order cone programs (SOCPs). Specifically, we first reformulate the CMFWP as a nonlinear SOCP with a nonconvex cost function, and then introduce a logarithmic barrier term and a quadratic proximal term into the objective to circumvent the nonconvex difficulty. Among others, the strict convexity of logarithmic function and the strong convexity of quadratic proximal term are fully used to convexify the objective function of the resulting SOCP. Such a technique is not new which is often used in the literature; see [4] for example. The SOCP reformulation has recently attracted much attention for engineering and operations research problems. However, to our best of knowledge, they are all formed into linear SOCPs for which some softwares using interior point methods [18,31] can be applied. In contrast, the nonlinear SOCP reformulation has little been used since all the aforementioned softwares are only able to solve linear SOCPs. This paper is concerned with the application of the nonlinear SOCP reformulation in the CMFWP, and its main purpose is to propose an alternative continuous approximate method to handle the class of difficult problems, instead of introducing a highly specialized method in a competition for the best solution and the fasted computation time.

This paper is organized as follows. In Sect. 2, we review the general convex SOCP and the semismooth Newton method [17] for solving it. Section 3 presents a detailed process of reformulating (1) as a nonlinear SOCP. Section 4 proposes a primal-dual continuation approach for the CMFWP by solving approximately a sequence of convexified SOCPs. In Sect. 5, we report the preliminary numerical results for some test problems from [3,28] with the continuation approach, and compare the final objective values with those yielded by the global optimization methods [28]. Finally, we conclude this paper.

Throughout this paper, $\|\cdot\|$ denotes the Euclidean norm, $\mathbb{R}^n$ denotes the space of n-dimensional real column vectors, and $\mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_m}$ is identified with $\mathbb{R}^{n_1+\cdots+n_m}$. Thus, $(x_1, \ldots, x_m) \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_m}$ is viewed as a column vector in $\mathbb{R}^{n_1+\cdots+n_m}$. The notations I and $\mathbf{0}$ denote an identity matrix and a zero matrix of suitable dimension, respectively, and $\mathrm{diag}(x_1, \ldots, x_n)$ means a diagonal matrix with $x_1, \ldots, x_n$ as the diagonal elements. Given a finite number of square matrices $Q_1, \ldots, Q_n$, we denote the block diagonal matrix with these matrices as block diagonal by $\mathrm{diag}(Q_1, \ldots, Q_n)$. For a differentiable function f, we denote by $\nabla f(x)$ and $\nabla^2_{xx} f(x)$ the gradient and the Hessian matrix of f at x, respectively. For a differentiable mapping $G : \mathbb{R}^n \to \mathbb{R}^m$, we denote by $G'(x) \in \mathbb{R}^{m\times n}$ the Jacobian of G at x. Let $\mathcal{O}$ be an open set in $\mathbb{R}^n$. If $G : \mathcal{O} \to \mathbb{R}^n$ is a locally Lipschitz continuous, then

$$\partial_B G(x) := \left\{ H \in \mathbb{R}^{m\times n} \mid \exists \{x^k\} \subseteq D_G : x^k \to x,\ G'(x^k) \to H \right\}$$

is nonempty and called the B-subdifferential of G at $x \in \mathcal{O}$, where D_G denotes the set of points at which G is differentiable. We assume that the reader is familiar with the concepts of (strongly) semismooth functions, and refer to [24,25] for details.

2 Preliminaries

The convex SOCP is to minimize a convex function over the intersection of an affine linear manifold with the Cartesian product of second-order cones, which can be described as

$$\begin{array}{ll} \text{minimize} & g(x) \\ \text{subject to} & Ax = b, \quad x \in \mathcal{K}, \end{array} \tag{3}$$

where $g : \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable convex function, A is an $m \times n$ matrix with full row rank, $b \in \mathbb{R}^m$ and $\mathcal{K}$ is the Cartesian product of second-order cones (SOCs), also called Lorentz cones. In other words,

$$\mathcal{K} = \mathcal{K}^{n_1} \times \mathcal{K}^{n_2} \times \cdots \times \mathcal{K}^{n_q}, \tag{4}$$

where $q, n_1, \ldots, n_q \geq 1, n_1 + \cdots + n_q = n$, and $\mathcal{K}^{n_i}$ denotes the SOC in $\mathbb{R}^{n_i}$ defined by

$$\mathcal{K}^{n_i} := \left\{ x_i = (x_{i1}, x_{i2}, \ldots, x_{in_i}) \in \mathbb{R}^{n_i} \mid \sqrt{x_{i2}^2 + \cdots + x_{in_i}^2} \leq x_{i1} \right\} \tag{5}$$

with $\mathcal{K}^1$ denoting the nonnegative real number set $\mathbb{R}_+$. A special case of (4) corresponds to the nonnegative orthant cone $\mathbb{R}^n_+$, i.e., $q = n$ and $n_1 = \cdots = n_q = 1$. When g is linear, clearly, (3) becomes a linear SOCP which has been investigated in many previous works and the interested reader is referred to the survey papers [2,19] and the books [5,6] for many important applications and theoretical properties.

The treatment of nonlinear convex SOCPs is much more recent and mainly focuses on the research of effective solution methods. Notice that $\mathcal{K}$ is a closed convex cone which is self-dual in the sense that $\mathcal{K}$ equals its dual cone $\mathcal{K}^* := \{y \in \mathbb{R}^n \mid \langle y, x \rangle \geq 0 \ \ \forall x \in \mathcal{K}\}$. Thus, it is easy to write the Karush-Kuhn-Tucker (KKT) optimality conditions of (3) as

$$\begin{cases} \nabla g(x) - A^T y - \lambda = 0 \\ Ax - b = 0 \\ \langle x, \lambda \rangle = 0, \quad x \in \mathcal{K}, \quad \lambda \in \mathcal{K}. \end{cases} \tag{6}$$

These conditions are also sufficient for optimality since g is convex. Based on the KKT system (6), there have been several methods proposed for solving (3), which include the smoothing Newton methods [8,12,16], the merit function method [9], and the semismooth Newton method [17]. As mentioned in the introduction, this paper is concerned with the application of the nonlinear SOCP in the CMFWP.

Since the resulting convex SOCPs in Sec. 4 will be solved with the semismooth Newton method in [17], we next review it. Let $P_{\mathcal{K}} : \mathbb{R}^n \to \mathbb{R}^n$ denote the Euclidean projection operator onto the cone $\mathcal{K}$, i.e., $P_{\mathcal{K}}(z) := \operatorname{argmin}_{y \in \mathcal{K}} \{\|z - y\|\}$ for any $z \in \mathbb{R}^n$. Then, from [12, Prop. 4.1], we have that

$$x - P_{\mathcal{K}}(x - \lambda) = 0 \iff x \in \mathcal{K}, \quad \lambda \in \mathcal{K}, \quad \langle x, \lambda \rangle = 0. \tag{7}$$

Consequently, the solution of the convex SOCP (3) is equivalent to finding the zeros of

$$\Phi(\omega) := \Phi(x, y, \lambda) := \begin{pmatrix} \nabla g(x) - A^T y - \lambda \\ Ax - b \\ x - P_{\mathcal{K}}(x - \lambda) \end{pmatrix}. \tag{8}$$

Since $P_{\mathcal{K}}$ is strongly semismooth by [16, Prop. 4.5], using [11, Theorem 19] then yields that the operator Φ is at least semismooth, and furthermore, it is strongly semismooth if $\nabla^2_{xx}g(x)$ is locally Lipschitz continuous at any $x \in \mathbb{R}^n$. The semismooth Newton method in [17] finds a zero of Φ by applying the nonsmooth Newton method [24,25] to the semismooth system $\Phi(\omega) = 0$. In other words, it generates the iterate sequence $\{\omega^k = (x^k, y^k, \lambda^k)\}$ by

$$\omega^{k+1} := \omega^k - W_k^{-1}\Phi(\omega^k), \tag{9}$$

where W_k is an arbitrary element from the B-subdifferential $\partial_B\Phi(\omega^k)$ and has the form of

$$W_k = \begin{pmatrix} \nabla^2_{xx}g(x^k) & -A^T & -I \\ A & \mathbf{0} & \mathbf{0} \\ I - V^k & \mathbf{0} & V^k \end{pmatrix}$$

for a suitable block diagonal matrix $V^k = \mathrm{diag}(V_1^k, \ldots, V_q^k)$ with $V_i^k \in \partial_B P_{\mathcal{K}^{n_i}}(x_i^k - \lambda_i^k)$.

The following two technical lemmas respectively provide the formula to compute the value of $P_{\mathcal{K}}$ at any point and the representation of each element in $V \in \partial_B P_{\mathcal{K}^n}(z)$.

Lemma 2.1 [17, Lemma 2.2] *For any given $z = (z_1, z_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, it holds that*

$$P_{\mathcal{K}^n}(z) = \max\{0, \mu_1(z)\}u_z^{(1)} + \max\{0, \mu_2(z)\}u_z^{(2)},$$

where $\mu_1(z)$, $\mu_2(z)$ and $u_z^{(1)}$, $u_z^{(2)}$ are the spectral values and the spectral vectors of z, respectively, given by

$$\mu_1(z) = z_1 - \|z_2\|, \quad \mu_2(z) = z_1 + \|z_2\|;$$
$$u_z^{(1)} = \frac{1}{2}(1, -\bar{z}_2), \qquad u_z^{(2)} = \frac{1}{2}(1, \bar{z}_2)$$

with $\bar{z}_2 = \frac{z_2}{\|z_2\|}$ if $z_2 \neq 0$ and otherwise $\bar{z}_2$ being any vector in $\mathbb{R}^{n-1}$ satisfying $\|\bar{z}_2\| = 1$.

Lemma 2.2 [17, Lemma 2.6] *Given a general point $z = (z_1, z_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, each element $V \in \partial_B P_{\mathcal{K}^n}(z)$ has the following representation:*

(a) *If $z_1 \neq \pm\|z_2\|$, then $P_{\mathcal{K}^n}(z)$ is continuously differentiable with*

$$V = P'_{\mathcal{K}^n}(z) = \begin{cases} \mathbf{0} & \text{if } z_1 < -\|z_2\| \\ I & \text{if } z_1 > \|z_2\| \\ \frac{1}{2}\begin{pmatrix} 1 & \bar{z}_2^T \\ \bar{z}_2 & H \end{pmatrix} & \text{if } -\|z_2\| < z_1 < \|z_2\|, \end{cases}$$

where

$$\bar{z}_2 := \frac{z_2}{\|z_2\|}, \quad H := \left(1 + \frac{z_1}{\|z_2\|}\right)I - \frac{z_1}{\|z_2\|}\bar{z}_2\bar{z}_2^T.$$

(b) *If $z_2 \neq 0$ and $z_1 = \|z_2\|$, then*

$$V \in \left\{I, \frac{1}{2}\begin{pmatrix} 1 & \bar{z}_2^T \\ \bar{z}_2 & H \end{pmatrix}\right\}, \quad \text{where } \bar{z}_2 := \frac{z_2}{\|z_2\|} \text{ and } H := 2I - \bar{z}_2\bar{z}_2^T.$$

(c) *If $z_2 \neq 0$ and $z_1 = -\|z_2\|$, then*

$$V \in \left\{\mathbf{0}, \frac{1}{2}\begin{pmatrix} 1 & \bar{z}_2^T \\ \bar{z}_2 & H \end{pmatrix}\right\}, \quad \text{where } \bar{z}_2 := \frac{z_2}{\|z_2\|} \text{ and } H := \bar{z}_2\bar{z}_2^T.$$

(d) *If $z = 0$, then either $V = \mathbf{0}$ or $V = I$ or V belongs to the set*

$$\left\{ \frac{1}{2} \begin{pmatrix} 1 & \bar{z}_2^T \\ \bar{z}_2 & H \end{pmatrix} \;\middle|\; H = (w_0 + 1)I - w_0 \bar{z}_2 \bar{z}_2^T \text{ for some } |w_0| \le 1 \text{ and } \|\bar{z}_2\| = 1 \right\}.$$

3 Nonlinear SOCP reformulation

In this section, we will present the detailed process of reformulating the capacitated multi-facility Weber problem (1) as a nonlinear SOCP with the form of (3). First, by introducing mn new variables t_{ij} for $i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n$, we can transform (1) into

$$\begin{aligned}
\min\ & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} w_{ij} t_{ij} \\
\text{s.t. } & \|x_i - a_j\| \le t_{ij}, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n \\
& \sum_{j=1}^{n} w_{ij} = s_i, \quad i = 1, 2, \ldots, m \\
& \sum_{i=1}^{m} w_{ij} = d_j, \quad j = 1, 2, \ldots, n \\
& w_{ij} \ge 0, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n \\
& x_i \in \mathbb{R}^2, \quad i = 1, 2, \ldots, m.
\end{aligned} \tag{10}$$

We write out all the constraints of $\|x_i - a_j\| \le t_{ij},\ i = 1, 2, \ldots, m; j = 1, 2, \ldots, n$ as below:

$$\begin{array}{lll}
\sqrt{(x_{i1} - a_{11})^2 + (x_{i2} - a_{12})^2} & \le t_{i1}, & i = 1, 2, \ldots, m, \\
\sqrt{(x_{i1} - a_{21})^2 + (x_{i2} - a_{22})^2} & \le t_{i2}, & i = 1, 2, \ldots, m, \\
\vdots & \vdots\ \vdots & \vdots \\
\sqrt{(x_{i1} - a_{n1})^2 + (x_{i2} - a_{n2})^2} & \le t_{in}, & i = 1, 2, \ldots, m.
\end{array} \tag{11}$$

Let

$$\begin{cases} u_{ij} := x_{i1} - a_{j1}, & i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n; \\ v_{ij} := x_{i2} - a_{j2}, & i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n. \end{cases} \tag{12}$$

Then, the constraints in (11) turn into

$$\begin{array}{lll}
(u_{i1})^2 + (v_{i1})^2 \le (t_{i1})^2, & t_{i1} \ge 0, & i = 1, 2, \ldots, m, \\
(u_{i2})^2 + (v_{i2})^2 \le (t_{i2})^2, & t_{i2} \ge 0, & vi = 1, 2, \ldots, m, \\
\vdots\quad \vdots\ \vdots & \vdots & \vdots \\
(u_{in})^2 + (v_{in})^2 \le (t_{in})^2, & t_{in} \ge 0, & i = 1, 2, \ldots, m.
\end{array} \tag{13}$$

Let

$$\hat{x}_{ij} := \begin{bmatrix} t_{ij} \\ u_{ij} \\ v_{ij} \end{bmatrix} \in \mathbb{R}^3, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n. \tag{14}$$

From (13) and the definition of $\mathcal{K}^3$, we have $\hat{x}_{ij} \in \mathcal{K}^3$ for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$.

It should be pointed out that we have created additional constraints through the above reformulation procedure. In particular, from (12), it follows that

$$\begin{aligned} u_{11} - u_{1k} &= a_{k1} - a_{11}, \quad k = 2, 3, \ldots, n, \\ \vdots \qquad & \qquad \vdots \\ u_{m1} - u_{mk} &= a_{k1} - a_{11}, \quad k = 2, 3, \ldots, n \end{aligned} \tag{15}$$

and

$$\begin{aligned} v_{11} - v_{1k} &= a_{k2} - a_{12}, \quad k = 2, 3, \ldots, n, \\ \vdots \qquad & \qquad \vdots \\ v_{m1} - v_{mk} &= a_{k2} - a_{12}, \quad k = 2, 3, \ldots, n. \end{aligned} \tag{16}$$

We will see that (15)–(16) can be recast as a linear system. Note that (15) and (16) are equivalent to

$$\begin{aligned} [0 \quad 1 \quad 0] \begin{bmatrix} t_{11} \\ u_{11} \\ v_{11} \end{bmatrix} + [0 \quad -1 \quad 0] \begin{bmatrix} t_{1k} \\ u_{1k} \\ v_{1k} \end{bmatrix} &= a_{k1} - a_{11}, \quad k = 2, 3, \ldots, n, \\ \vdots \qquad\qquad\qquad & \qquad \vdots \qquad \vdots \\ [0 \quad 1 \quad 0] \begin{bmatrix} t_{m1} \\ u_{m1} \\ v_{m1} \end{bmatrix} + [0 \quad -1 \quad 0] \begin{bmatrix} t_{mk} \\ u_{mk} \\ v_{mk} \end{bmatrix} &= a_{k1} - a_{11}, \quad k = 2, 3, \ldots, n \end{aligned}$$

and

$$\begin{aligned} [0 \quad 0 \quad 1] \begin{bmatrix} t_{11} \\ u_{11} \\ v_{11} \end{bmatrix} + [0 \quad 0 \quad -1] \begin{bmatrix} t_{1k} \\ u_{1k} \\ v_{1k} \end{bmatrix} &= a_{k2} - a_{12}, \quad k = 2, 3, \ldots, n \\ \vdots \qquad\qquad\qquad & \qquad \vdots \qquad \vdots \\ [0 \quad 0 \quad 1] \begin{bmatrix} t_{m1} \\ u_{m1} \\ v_{m1} \end{bmatrix} + [0 \quad 0 \quad -1] \begin{bmatrix} t_{mk} \\ u_{mk} \\ v_{mk} \end{bmatrix} &= a_{k2} - a_{12}, \quad k = 2, 3, \ldots, n. \end{aligned}$$

We can simplify (15)–(16) by introducing the following notations. More specifically, let

$$A_u := \begin{bmatrix} 0 & 1 & 0 & 0 & -1 & 0 & & & & & & \\ 0 & 1 & 0 & & & & 0 & -1 & 0 & & & \\ & \vdots & & & & & & & & \ddots & & \\ 0 & 1 & 0 & & & & & & & 0 & -1 & 0 \end{bmatrix} \in \mathbb{R}^{(n-1)\times 3n}, \tag{17}$$

$$A_l := \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & -1 & & & & & & \\ 0 & 0 & 1 & & & & 0 & 0 & -1 & & & \\ & \vdots & & & & & & & & \ddots & & \\ 0 & 0 & 1 & & & & & & & 0 & 0 & -1 \end{bmatrix} \in \mathbb{R}^{(n-1)\times 3n}, \tag{18}$$

and denote

$$b_u := \begin{bmatrix} a_{21} - a_{11} \\ a_{31} - a_{11} \\ \vdots \\ a_{n1} - a_{11} \end{bmatrix} \in \mathbb{R}^{n-1}, \quad b_l := \begin{bmatrix} a_{22} - a_{12} \\ a_{32} - a_{12} \\ \vdots \\ a_{n2} - a_{12} \end{bmatrix} \in \mathbb{R}^{n-1}.$$

Then, Eqs. (15)–(16) can be recast as the following system of linear constraints

$$\begin{bmatrix} A_u & & & \\ & A_u & & \\ & & \ddots & \\ & & & A_u \\ -- & -- & -- & -- \\ A_l & & & \\ & A_l & & \\ & & \ddots & \\ & & & A_l \end{bmatrix} \begin{bmatrix} [\hat{x}_{11}] \\ \vdots \\ [\hat{x}_{1n}] \\ -- \\ [\hat{x}_{21}] \\ \vdots \\ [\hat{x}_{2n}] \\ -- \\ \vdots \\ -- \\ [\hat{x}_{m1}] \\ \vdots \\ [\hat{x}_{mn}] \end{bmatrix} = \begin{bmatrix} b_u \\ b_u \\ \vdots \\ b_u \\ -- \\ b_l \\ b_l \\ \vdots \\ b_l \end{bmatrix}, \tag{19}$$

where the dimensions in the linear system are $2m(n-1) \times 3mn$, $3mn \times 1$, $2m(n-1) \times 1$ for the matrix, column of variables, and column of constants, respectively.

We next look into the constraints on demand and capacity, namely the two groups of equality constraints in (10). We notice that they can be recast as a linear system as below:

$$\begin{bmatrix} [1\ 1\ 1\ \cdots 1] & & & \\ & [1\ 1\ 1\ \cdots 1] & & \\ & & \ddots & \\ & & & [1\ 1\ 1\ \cdots 1] \\ ------ & ------ & ------ & ------ \\ [1\ 0\ 0 \cdots 0] & [1\ 0\ 0 \cdots 0] & \cdots & [1\ 0\ 0 \cdots 0] \\ [0\ 1\ 0 \cdots 0] & [0\ 1\ 0 \cdots 0] & \cdots & [0\ 1\ 0 \cdots 0] \\ \vdots & \vdots & \vdots & \vdots \\ [0\ 0\ 0 \cdots 1] & [0\ 0\ 0 \cdots 1] & \cdots & [0\ 0\ 0 \cdots 1] \end{bmatrix} \begin{bmatrix} w_{11} \\ \vdots \\ w_{1n} \\ -- \\ w_{21} \\ \vdots \\ w_{2n} \\ -- \\ \vdots \\ -- \\ w_{m1} \\ \vdots \\ w_{mn} \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \\ -- \\ d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}. \tag{20}$$

Again, we point it out that the dimensions of the system (20) are $(m+n) \times (mn)$, $(mn) \times 1$, $(m+n) \times 1$ for the matrix, column of variables and column of constants, respectively. Moreover, the coefficient matrix has not full row rank due to the assumption (2), but its any $m+n-1$ rows are all linear independent. For convenience, in the rest of this paper, A_w denotes the matrix composed of the first $m+n-1$ rows in the coefficient matrix of (20).

In summary, we reformulate the CMFWP (1) as the following nonlinear SOCP:

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} w_{ij} t_{ij} \\ &\text{subject to} \quad Ax = b \\ &\qquad\qquad\quad x \in (\mathcal{K}^3)^{mn} \times (\mathcal{K}^1)^{mn} \end{aligned} \tag{21}$$

where $(\mathcal{K}^3)^{mn} \times (\mathcal{K}^1)^{mn}$ denotes the Cartesian product of mn $\mathcal{K}^3$ and mn $\mathcal{K}^1$, and

$$x := \left(\hat{x}_{11}, \ldots, \hat{x}_{1n}, \ldots, \hat{x}_{m1}, \ldots \hat{x}_{mn}, w_{11}, \ldots, w_{1n}, \ldots, w_{m1}, \ldots, w_{mn}\right), \tag{22}$$

$$A := \left[\begin{array}{cccc|c} A_u & & & & \\ & A_u & & & \\ & & \ddots & & \\ & & & A_u & \mathbf{0} \\ \hline A_l & & & & \\ & A_l & & & \\ & & \ddots & & \\ & & & A_l & \\ \hline & \mathbf{0} & & & A_w \end{array}\right], \tag{23}$$

$$b := (b_u, b_u, \ldots, b_u, b_l, b_l, \ldots, b_l, s_1, \ldots, s_m, d_1, \ldots, d_{n-1}). \tag{24}$$

Notice that, by the expression (22) of x, the objective function of (21) can be rewritten as a quadratic function $x^T Qx$, where $Q = [Q_{kl}]_{4mn\times 4mn}$ is a nonsymmetric matrix with

$$Q_{kl} = \begin{cases} c_{ij} & \text{if } k = 3(i-1)n + 3(j-1) + 1,\ l = 3mn + (i-1)n + j; \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

for $k, l = 1, 2, \ldots, 4mn$ and $i = 1, 2, \ldots, m;\ \ j = 1, 2, \ldots, n$. Hence, (21) is equivalent to

$$\begin{aligned} &\text{minimize} \quad x^T Qx \\ &\text{subject to} \quad Ax = b \\ &\qquad\qquad\quad x \in (\mathcal{K}^3)^{mn} \times (\mathcal{K}^1)^{mn} \end{aligned} \tag{26}$$

where Q is a $4mn \times 4mn$ matrix given by (25), A is a $(2mn - m + n - 1) \times 4mn$ matrix with full row rank $(2mn - m + n - 1)$, and the dimension of b is $(2mn - m + n - 1) \times 1$. Since $x^T Qx = x^T Q^T x$ for any $x \in \mathbb{R}^{4mn}$, the SOCP (21) is further equivalent to

$$\begin{aligned} &\text{minimize} \quad f(x) := \frac{1}{2} x^T (Q + Q^T) x \\ &\text{subject to} \quad Ax = b \\ &\qquad\qquad\quad x \in (\mathcal{K}^3)^{mn} \times (\mathcal{K}^1)^{mn} \end{aligned} \tag{27}$$

The SOCP (27) has an advantage over (26) that its Hessian matrix is symmetric, although its objective function is still nonconvex. In the next section, we develop a primal-dual algorithm for solving the CMFWP based on the nonlinear SOCP reformulation (27).

We should point out that the Hessian matrix of the function f, i.e., $\bar{Q} = (Q + Q^T)$ and the constraint coefficient matrix A of (27) are both sparse. For example, when $m = n = 2$, $\bar{Q} = [\bar{Q}_{kl}]_{16\times 16}$ has only eight nonzero entries: $\bar{Q}_{1,13} = \bar{Q}_{13,1} = c_{11}$, $\bar{Q}_{4,14} = \bar{Q}_{14,4} = c_{12}$,

$\bar{Q}_{7,15} = \bar{Q}_{15,7} = c_{21}$, $\bar{Q}_{10,16} = \bar{Q}_{16,10} = c_{22}$, whereas $A = [A_{ij}]_{7\times 16}$ has sixteen nonzero entries: $A_{12} = 1$, $A_{15} = -1$, $A_{28} = 1$, $A_{2,11} = -1$, $A_{33} = 1$, $A_{36} = -1$, $A_{49} = 1$, $A_{4,12} = -1$, $A_{5,13} = A_{5,14} = 1$, $A_{5,15} = A_{5,16} = 1$, $A_{6,13} = A_{6,15} = 1$, $A_{7,14} = A_{7,16} = 1$. In addition, we observe that each row of $\bar{Q}$ has at most a nonzero entry c_{i_1,j_1} with $c_{i_1,j_1} \in \{c_{11}, \ldots, c_{1n}, \ldots, c_{m1}, \ldots, c_{mn}\}$, and all diagonal entries are zero.

4 Continuation approach for the CMFWP

This section designs a primal-dual approximate algorithm for the nonconvex SOCP (27). The main idea is transforming (27) into the solution of a sequence of convexified SOCP subproblems.

First, note that the variables w_{ij} are nonnegative and restricted by the linear constraints $\sum_{j=1}^{n} w_{ij} = s_i$ and $\sum_{i=1}^{m} w_{ij} = d_j$, and hence the nonconvex SOCP (27) is equivalent to

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & Ax = b \\ & 0 \le w_{ij} \le \min\{s_i, d_j\}, \quad i = 1, \ldots, m;\ j = 1, \ldots, n \\ & x \in (\mathcal{K}^3)^{mn} \times (\mathcal{K}^1)^{mn}. \end{array} \tag{28}$$

Since the logarithmic barrier function $-\sum_{i=1}^{m}\sum_{j=1}^{n}\left[\ln(w_{ij}) + \ln(\min\{s_i, d_j\} - w_{ij})\right]$ is well defined when $0 < w_{ij} < \min\{s_i, d_j\}$ for all $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, and moreover, its limit is $+\infty$ if some w_{ij} tends to 0 or $\min\{s_i, d_j\}$, we can dispense with the bound constraints on w_{ij} in (28) and obtain the following transformed problem:

$$\begin{array}{ll} \min & f(x) - \tau \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n}\left[\ln(w_{ij}) + \ln(\min\{s_i, d_j\} - w_{ij})\right] \\ \text{s.t.} & Ax = b \\ & x \in (\mathcal{K}^3)^{mn} \times (\mathcal{K}^1)^{mn} \end{array} \tag{29}$$

where $\tau > 0$ is a barrier parameter that finally tends to 0. The above operation seems to purposely make the original problem (27) more complex. However, we will see that the introduction of the logarithmic barrier term gives a help in locating the feasible solution of (27), as well as plays a certain role in convexifying $f(x)$.

Now, we introduce a quadratic proximal term $\frac{1}{2}\|x - z\|^2$ with $z \in \mathbb{R}^{4mn}$ being a given vector into the objective of (29) to further convexify $f(x)$. Define

$$g(x, z, \tau, \varepsilon) := f(x) - \tau \sum_{i=1}^{m}\sum_{j=1}^{n}\left[\ln(w_{ij}) + \ln(\min\{s_i, d_j\} - w_{ij})\right] + \frac{\varepsilon}{2}\|x - z\|^2 \tag{30}$$

where $\varepsilon > 0$ is a proximal parameter that will become larger until over some threshold.

Proposition 4.1 *Given a vector $z \in \mathbb{R}^{4mn}$, let g be defined as in (30) for any $\tau, \varepsilon > 0$. Then, there exists $\varepsilon_0 > 0$ such that g is strictly convex for any $\varepsilon > \varepsilon_0$ and $\tau > 0$.*

Proof For given $z \in \mathbb{R}^{4mn}$ and any $\tau, \varepsilon > 0$, we compute the Hessian matrix of g as

$$\begin{aligned} \nabla^2_{xx} g(x, z, \tau, \varepsilon) = Q + Q^T + \mathrm{diag}(&\underbrace{\varepsilon, \ldots, \varepsilon}_{3mn}, P_{11} + \varepsilon, \ldots, P_{1n} \\ &+ \varepsilon, \ldots, P_{m1} + \varepsilon, \ldots, P_{mn} + \varepsilon) \end{aligned}$$

with P_{ij} for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$ given by

$$P_{ij} = \frac{\tau}{(w_{ij})^2} + \frac{\tau}{(\min\{s_i, d_j\} - w_{ij})^2} > 0.$$

Since all diagonal entries of $Q + Q^T$ are zero and each row of $Q + Q^T$ has at most a nonzero entry c_{i_1, j_1} with $c_{i_1, j_1} \in \{c_{11}, \ldots, c_{1n}, c_{21}, \ldots, c_{2n}, \ldots, c_{m1}, \ldots, c_{mn}\}$, we have that

$$\left[\nabla^2_{xx} g(x, z, \tau, \varepsilon)\right]_{kk} > \sum_{l=1,\ l\neq k}^{4mn} \left[\nabla^2_{xx} g(x, z, \tau, \varepsilon)\right]_{kl}$$

for all $k = 1, 2, \ldots, 4mn$ whenever $\varepsilon > \varepsilon_0$ with

$$\varepsilon_0 := \max_{1 \le i \le m, 1 \le j \le n} \{c_{ij}\}. \tag{31}$$

In other words, the matrix $\nabla^2_{xx} g(x, z, \tau, \varepsilon)$ is strictly diagonally dominant for any $\varepsilon > \varepsilon_0$ and $\tau > 0$. From Corollary 7.2.3 of [14], it then follows that $\nabla^2_{xx} g(x, z, \tau, \varepsilon)$ is positive definite, and consequently $g(x, z, \tau, \varepsilon)$ is strictly convex, for any $\varepsilon > \varepsilon_0$ and $\tau > 0$. □

Proposition 4.1 states that for a given $z \in \mathbb{R}^{4mn}$, the function $g(x, z, \tau, \varepsilon)$ is strictly convex for any $\varepsilon > \varepsilon_0$ and $\tau > 0$. In fact, it is also strongly convex when $\varepsilon > \varepsilon_0$. In view of this, our continuation approach seeks for an approximate optimal solution of the CMFWP by solving a sequence of the following subproblems

$$\begin{array}{ll} \min & g(x, \hat{x}^k, \tau_k, \varepsilon_k) \\ \text{s.t.} & Ax = b \\ & x \in (\mathcal{K}^3)^{mn} \times (\mathcal{K}^1)^{mn} \end{array} \tag{32}$$

with a decreasing sequence $\{\tau_k\}$ and a increasing sequence $\{\varepsilon_k\}$, where $\hat{x}^k$ is a vector given by the solution of the last subproblem. For a fixed k, since the subproblem (32) is a convex SOCP whenever $\varepsilon_k > \varepsilon_0$, its solution is easy, which from Sect. 2 is equivalent to solving

$$\begin{cases} \nabla_x g(x, \hat{x}^k, \tau_k, \varepsilon_k) - A^T y - \lambda = 0 \\ Ax - b = 0 \\ x - P_{\mathcal{K}}(x - \lambda) = 0 \end{cases} \tag{33}$$

with $\mathcal{K} = (\mathcal{K}^3)^{mn} \times (\mathcal{K}^1)^{mn}$. Define the mapping $\Phi_k : \mathbb{R}^{10mn-m+n-1} \to \mathbb{R}^{10mn-m+n-1}$ by

$$\Phi_k(\omega) = \Phi_k(x, y, \lambda) := \begin{pmatrix} \nabla_x g(x, \hat{x}^k, \tau_k, \varepsilon_k) - A^T y - \lambda \\ Ax - b \\ x - P_{\mathcal{K}}(x - \lambda) \end{pmatrix}. \tag{34}$$

Then, solving the nonsmooth system (33) is equivalent to finding the zero of the operator $\Phi_k(\omega)$. We will attain this goal by using the semismooth Newton method in [17].

Next we describe the iteration steps of the continuation approach in which two fixed constants $c_1 \in (0, 1)$ and $c_2 > 1$ are used to reduce and increase the dynamic parameters τ and ε. Let $\Psi_k(w) := \frac{1}{2}\|\Phi_k(\omega)\|^2$ denote the natural merit function of system $\Phi_k(\omega) = 0$.

Algorithm 4.1 (*Continuation Approach*)

(S.0) *Given the constants* $c_1 \in (0, 1)$, $c_2 > 1$ *and* $\hat{\tau}, \hat{\varepsilon} > 0$. *Select* $\tau_0, \varepsilon_0 > 0$ *and a starting point* $\bar{\omega}^0 = (\bar{x}^0, \bar{y}^0, \bar{\lambda}^0)$ *with the last mn elements* $\bar{w}^0_{ij}$ *of* $\bar{x}^0$ *satisfying* $0 < \bar{w}^0_{ij} < min\{s_i, d_j\}$. *Let* $\hat{x}^0 := \bar{x}^0$, *and set* $k := 0$.

(S.1) *Compute (by Algorithm 4.2 below) an approximate optimal solution* $\omega^k = (x^k, y^k, \lambda^k)$ *of (33) with the starting point* $\bar{\omega}^k = (\bar{x}^k, \bar{y}^k, \bar{\lambda}^k)$.

(S.2) *If* $\tau_k < \hat{\tau}$ *and* $\varepsilon_k > \hat{\varepsilon}$, *then stop, and otherwise go to (S.3).*

(S.3) *Modify the parameters* τ_k *and* ε_k *by* $\tau_{k+1} := c_1\tau_k$ *and* $\varepsilon_{k+1} := c_2\varepsilon_k$, *and let*

$$\bar{x}^{k+1} := x^k, \quad \bar{y}^{k+1} := y^k, \quad \bar{\lambda}^{k+1} := z^k, \quad \hat{x}^{k+1} := x^k.$$

(S.4) *Set* $k := k + 1$, *and go to (S.1).*

Note that the continuation approach is a primal-dual one and the last mn components of the final iterate y^k provide an approximate dual solution of the CMFWP, which usually has a certain economic meaning associated with this class of transportation problems. In addition, the main computation work of Algorithm 4.1 is to seek an approximate solution of (33). Such a solution can be easily obtained when ε_k is large enough since, on the one hand, the mapping $\nabla_x g(x, \hat{x}^k, \tau_k, \varepsilon_k)$ is strongly monotone in this case, which together with Lemma 3.1 of [16] implies that the function $\Psi_k(\omega)$ has bounded level sets; and on the other hand, from Sect. 2, the operator Φ_k is at least semismooth which guarantees in theory a fast algorithm with a superlinear (or quadratic) convergence rate, to seek the solution of (33). Of course, we should emphasize that, when computing the approximate solution of (33), we must restrict the maximum steplength of the iterates (see Algorithm 4.2 below).

We next describe the specific iteration steps of the semismooth Newton method [17] when applying it for solving the semismooth system (33). For a given k, from Sect. 2 it follows that the main iteration step of the method is as follows:

$$\omega^{l+1} := \omega^l - W_l^{-1}\Phi_k(\omega^l), \quad W_l \in \partial_B\Phi_k(\omega^l) \ \text{ for } \ l = 0, 1, 2, \ldots,$$

where $\partial_B\Phi_k(\omega^l)$ denotes the B-subdifferential of the semismooth mapping Φ_k at the point ω^l and any element W_l of $\partial_B\Phi_k(\omega^l)$ has the expression

$$W_l := \begin{bmatrix} \nabla^2_{xx} g(x^l, \hat{x}^k, \varepsilon_k, \tau_k) & -A^T & I \\ A & \mathbf{0} & \mathbf{0} \\ I - V^l & \mathbf{0} & V^l \end{bmatrix} \tag{35}$$

for a suitable block diagonal matrix $V^l = \mathrm{diag}(V_1^l, \ldots, V_{4mn}^l)$ with $V_i^l \in \partial_B P_{\mathcal{K}^3}(x^l - \lambda^l)$ for $i = 1, 2, \ldots, 3mn$ and $V_i^l \in \partial_B P_{\mathcal{K}^1}(x^l - \lambda^l)$ for $i = 3mn + 1, \ldots, 4mn$.

Algorithm 4.2 *(Solving the subproblem (33) for Step (S.1) of Algorithm 4.1)*

(S.0) *For a given k, choose a positive integer* l_k *and let* $\omega^0 = \bar{\omega}^k = (\bar{x}^k, \bar{y}^k, \bar{\lambda}^k)$. *Set* $l := 0$.

(S.1) *If* $\Psi_k(\omega^l) \le \tau_k$ *or* $l > l_k$, *then stop, and otherwise go to (S.2).*

(S.2) *Select an element* W_l *from* $\partial_B\Phi_k(\omega^l)$ *and compute the Newton direction* $\triangle\omega^l := (\triangle x^l, \triangle y^l, \triangle\lambda^l)$ *by solving the following system*:

$$W_l \triangle\omega^l = -\Phi_k(\omega^l).$$

(S.3) *Compute the maximum steplength* α_l *of the iterate* x^l *such that the last mn elements* w_{ij}^l *of* x^l *satisfy* $0 < w_{ij}^l < min\{s_i, d_j\}$ *for* $i = 1, 2, \ldots, m$ *and* $j = 1, 2, \ldots, n$. *Let*

$$x^{l+1} := x^l + 0.99 * \alpha_l \triangle x^l, \quad y^{l+1} := y^l + \triangle y^l, \quad \lambda^{l+1} := \lambda^l + \triangle\lambda^l.$$

(S.4) *Set* $l := l + 1$, *and go to (S.1).*

Note that Algorithm 4.2 is a local optimization method for the system $\Phi_k(\omega) = 0$, but it works well (as will be shown in the next section) since Algorithm 4.1 only requires an approximate solution of $\Phi_k(\omega) = 0$. For a fixed k, $\partial_B \Phi_k(\omega^l)$ is a set and W_l may be chosen as an arbitrary element from this set. We see that the selection of $W_l \in \partial_B \Phi_k(\omega^l)$ is due to that of $V_i^l \in \partial_B P_{\mathcal{K}^3}(x^l - \lambda^l)$ for $i = 1, \ldots, 3mn$ and $V_i^l \in \partial_B P_{\mathcal{K}^1}(x^l - \lambda^l)$ for $i = 3mn + 1, \ldots, 4mn$. In the next section, when implementing Algorithm 4.1, we choose $V \in \partial_B P_{\mathcal{K}^n}(z)$ for given $z = (z_1, z_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$ in the following way:

Selection of $V \in \partial_B P_{\mathcal{K}^n}(z)$ **with** $z = (z_1, z_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$

(S.0) If $||z_1| - \|z_2\|| > 0$, then go to Step (S.1); if $\|z_2\| > 0$ and $z_1 > 0$, then go to Step (S.2); if $\|z_2\| > 0$ and $z_1 < 0$, then go to Step (S.3); otherwise we set $V := I$.

(S.1) If $z_1 < -\|z_2\|$, then set $V := \mathbf{0}$; if $z_1 > \|z_2\|$, then set $V := I$; and otherwise set

$$V := \frac{1}{2}\begin{bmatrix} 1 & \bar{z}_2 \\ \bar{z}_2 & H \end{bmatrix} \text{ with } \bar{z}_2 = \frac{z_2}{\|z_2\|} \quad \text{and} \quad H = \left(1 + \frac{z_1}{\|z_2\|}\right) I - \frac{z_1}{\|z_2\|} \bar{z}_2 \bar{z}_2^T.$$

(S.2) Compute $\bar{z}_2 = z_2/\|z_2\|$ and $H = 2I - \bar{z}_2\bar{z}_2^T$, and set $V := \frac{1}{2}\begin{bmatrix} 1 & \bar{z}_2^T \\ \bar{z}_2 & H \end{bmatrix}$.

(S.3) Compute $\bar{z}_2 = z_2/\|z_2\|$ and $H = \bar{z}_2\bar{z}_2^T$, and set $V := \frac{1}{2}\begin{bmatrix} 1 & \bar{z}_2^T \\ \bar{z}_2 & H \end{bmatrix}$.

5 Numerical results

In this section, we report the preliminary numerical results of the continuation approach for eighteen benchmark instances taken from [3,28]. Among others, twelve of them (E1–E8 and E10–13) are obtained from [3], and the others are derived from E13 by the same way as [28]. By the classification rule of [28], E1–E8 belong to small-sized test problems, E9 is a medium-sized one, while the rest 9 test problems are all large-sized.

All numerical experiments were done with a PC (Intel Pentium 4) of 2.8 GHz CPU and 512 MB memory. The computer codes were all written in Matlab 6.5. The subproblem (33) was solved by the semismooth Newton method described in Algorithm 4.2. During the tests, the parameters involved in Algorithm 4.1 were chosen as follows:

$$\tau_0 = 10^3, \quad \varepsilon_0 = 1.0, \quad \hat{\tau} = 10^{-4}, \quad \hat{\varepsilon} = 10^3, \quad c_1 = 1.05, \quad c_2 = 0.6. \tag{36}$$

The maximum iteration number l_k in Algorithm 4.2 was set to be 50. The starting point $\bar{\omega}^0 = (\bar{x}^0, \bar{y}^0, \bar{z}^0)$ were chosen as follows: the first $3mn$ elements of $\bar{x}^0$ were set to be

$$\left[\left[-1, \frac{\sum_{j=1}^{n} a_{j1}}{n}, \frac{\sum_{j=1}^{n} a_{j2}}{n}\right]^T \middle| \left[-1, \frac{\sum_{j=1}^{n} a_{j1}}{n}, \frac{\sum_{j=1}^{n} a_{j2}}{n}\right]^T \middle| \cdots \right.$$
$$\left. \middle| \left[-1, \frac{\sum_{j=1}^{n} a_{j1}}{n}, \frac{\sum_{j=1}^{n} a_{j2}}{n}\right]^T \right]$$

Table 1 Numerical comparisons of Algorithm 4.1 with global optimization methods

Problem	PLSB procedure		PLT procedure		The continuation approach			
No. (m, n)	Fval	CPU	Fval	CPU	Fval	Fer1	Fer2	CPU
E1 (2,2)	0	0.04 (s)	0	0.04 (s)	1.8566e-5	6.32e-14	8.71e-14	2.95 (s)
E2 (2,4)	247.28	0.10 (s)	247.28	0.20 (s)	247.2780	8.60e-14	2.57e-13	6.18 (s)
E3 (2,4)	214.37	0.50 (s)	214.34	0.90 (s)	222.9738	1.00e-14	1.17e-14	27.9 (s)
E4 (3,5)	24.00	3.20 (s)	24.00	2.30 (s)	31.87	1.29e-9	5.49e-9	74.8 (s)
E5 (3,5)	73.96	4.10 (s)	73.96	2.00 (s)	73.9587	9.53e-13	4.12e-12	16.9 (s)
E6 (3,9)	221.40	55.6 (s)	221.4	66.4 (s)	268.0349	1.06e-9	1.09e-8	295.2 (s)
E7 (3,9)	871.62	57.3 (s)	871.62	42.2 (s)	991.4868	1.40e-13	8.50e-13	72.7 (s)
E8 (4,8)	609.23	14 (m)	609.23	6 (m)	843.2934	1.27e-10	9.36e-13	107.5 (s)
E9 (5,10)	3260.2	13 (m)	2595.47	8 (m)	5472.3	1.08e-13	7.26e-12	16 (m)
E10 (5,15)	8215.55	23 (m)	8169.79	23 (m)	8879.3	1.03e-13	1.09e-12	37 (m)
E11 (5,20)	12935.65	13 (m)	12846.87	134 (m)	22518.04	9.10e-11	1.37e-8	100 (m)
E12 (5,20)	1582.62	29 (m)	1107.18	73 (m)	1502.4	1.49e-10	3.49e-9	90 (m)
E13 (5,30)	25292.8	31 (m)	23990.04	316 (m)	26693.03	9.30e-10	9.67e-9	302 (m)
E14 (6,10)	8045.11	21 (m)	7797.21	9 (m)	8310.6	1.34e-8	7.21e-8	30 (m)
E15 (7,10)	7142.96	21 (m)	6974.77	14 (m)	8433.1	7.16e-10	4.01e-9	40 (m)
E16 (8,10)	2179.44	31 (m)	1576.83	120 (m)	6387.3	6.70e-11	2.09e-10	58 (m)
E17 (9,10)	3403.93	21 (m)	3250.68	12 (m)	7530.4	1.12-12	3.32e-12	20 (m)
E18 (10,10)	8040.60	30 (m)	7764.05	19 (m)	8024.02	2.70e-10	5.27e-10	103 (m)

and the last mn elements $\bar{w}_{ij}^0$, $i = 1, \ldots, m$, $j = 1, \ldots, n$ were chosen as $\bar{w}_{ij}^0 = \dfrac{\min\{s_i, d_j\}}{m+n}$, $\bar{y}^0 = 0$, and $\bar{z}^0 = \bar{x}^0$. It may be worthwhile to explore other choices.

We compare the objective values given by Algorithm 4.1 with the above parameters with those yielded by the global optimization procedures in [28], i.e., the projected location space bounding procedure (PLSB) and the enhanced reformulation-linearization technique (RLT). The two global optimization procedures use the exact branch-and-bound procedure for the problems E1–E9, and the heuristic branching procedure for problems E10–E18; see [28] for the detailed description. The numerical results yielded by the continuation approach and the two global optimization procedures are summarized in Table 1.

In Table 1, the first column specifies the dimension of the CMFWP, the second column and the third column report the objective values yielded by the PLSB-based procedure and the RLT-based procedure, respectively, and the fourth column collects the results generated by Algorithm 4.1 where **Fval** represents the objective values of (1) at the final iteration, **Time** gives the CPU time used by the continuation approach for each problem, and **Fer1** and **Fer2** denote the feasibility error of the first and the second group of constraints of (1), i.e. the value of the following error functions at the final iteration:

$$\sqrt{\sum_{i=1}^{m}\left(\sum_{j=1}^{m} w_{ij} - s_i\right)^2} \quad \text{and} \quad \sqrt{\sum_{j=1}^{n}\left(\sum_{i=1}^{m} w_{ij} - d_j\right)^2}.$$

Among others, the results in the second and the third column are from [28, Table 2–3] where all computations were performed on a SUN ULTRA 1 workstation with 256 MB RAM.

Fer 1 and **Fer 2** columns of Table 1 show that Algorithm 4.1 generates at least feasible solutions for all test problems. Comparing the objective values yielded by the continuation approach with those given by the global optimization procedures, we see that Algorithm 4.1 can also find satisfying suboptimal solutions for all test problems. Particularly, for small-sized test problems E1–E8, it also yields the global optimal solutions for three test problems.

Although there exists a certain gap between the continuation approach and the global optimization procedure in terms of the solution quality, the objective values obtained may provide desirable upbounds for the application of the branch-and-bound method. Note that the design of Algorithm 4.1 does not exploit the speciality of the quadratic objective function of (28) and any specific structure of the CMFWP. In our future research work, we will consider improvement of the continuous method from these two directions.

6 Conclusions

We have proposed a primal-dual continuous approach for the capacitated multi-facility Weber problem by transforming the problem into the solution of a sequence of nonlinear convex SOCP subproblems approximately. Compared with the other methods mentioned in the introduction, this approach is an approximate one and designed by the nonlinear SOCP reformulation of the CMFWP, which has much simpler numerical implementation and provides an efficient estimation for the optimal dual solution that usually has a certain economic interpretation associated with this class of location problems. Preliminary numerical results verify the feasibility of the approach and indicate that it is promising to search for a satisfying suboptimal solution for large-scaled problems.

From Sect. 3, we see that the nonconvexity of (28) arises from its quadratic objective function. Hence, it is worthwhile to consider an approximate algorithm for the CMFWP by using effective SOCP relaxations of (28), and we leave it as a future research topic.

Acknowledgments The authors would like to thank the anonymous referee and the editor for their helpful comments which improve the presentation of this paper greatly.

References

1. Aras, N., Altinel, I.K., Orbay, M.: New heuristic methods for the capacitated multi-facility Weber problem. Nav. Res. Logist. **54**, 21–32 (2007)
2. Alizadeh, F., Goldfarb, D.: Second-order cone programming. Math. Program. **95**, 3–51 (2003)
3. Al-Loughani, L.: Algorithmic approaches for solving the Euclidean distance location-allocation problem. Ph.D. Dissertation, Industrial and System Engineering, Virginia Polytechnic Institute and State University, Blacksburgh, Virginia (1997)
4. Bertsekas, D.P.: Convexification procedures and decomposition methods for nonconvex optimization problems. J. Optim. Theor. Appl. **29**, 169–197 (1979)
5. Ben-Tal, A., Nemirovski, A.: Lectures on Modern Convex Optimization, MPS-SIAM Series on Optimization. SIAM, Philadelphia PA (2001)
6. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
7. Cooper, L.: The transportation-location problem. Operations Research **20**, 94–108 (1972)
8. Chen, X.-D., Sun, D., Sun, J.: Complementarity functions and numerical experiments for second-order cone complementarity problems. Comput. Optim. Appl. **25**, 39–56 (2003)
9. Chen, J.-S., Tseng, P.: An unconstrained smooth minimization reformulation of the second-order cone complementarity problem. Math. Program. **104**, 293–327 (2005)

10. Drezner, Z., Hamacher, H.: Facility Location: Applications and Theory. Springer, Berlin (2002)
11. Fischer, A.: Solution of the monotone complementarity problem with locally Lipschitzian functions. Math. Program. **76**, 513–532 (1997)
12. Fukushima, M., Luo, Z.-Q., Tseng, P.: Smoothing functions for second-order cone complementarity problems. SIAM J. Optim. **12**, 436–460 (2002)
13. Francis, R.L., McGinnis, L.F., White, J.A.: Facility Layout and Location: An Analytical Approach. Prentice Hall, Englewood Cliffs, NJ (1991)
14. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (1985)
15. Hajiaghayi, M., Mahdian, M., Mirrokni, V.S.: The facility location problem with general cost functions. Networks **42**, 42–47 (2003)
16. Hayashi, S., Yamashita, N., Fukushima, M.: A combined smoothing and regularization method for monotone second-order cone complementarity problems. SIAM J. Optim. **15**, 593–615 (2005)
17. Kanzow, C., Ferenczi, I., Fukushima, M.: On the local convergence of semismooth Newton methods for linear and nonlinear second-order cone programs without strict complementarity. SIAM J. Optim. **20**, 297–320 (2009)
18. Lobo, M.S., Vandenberghe, L., Boyd, S.: SOCP-Software for second-order cone programming: User's guide. Technical Report, Department of Electrical Engineering, Standford University, April (1997)
19. Lobo, M.S., Vandenberghe, L., Boyd, S., Lebret, H.: Application of second-order cone programming. Linear Algebra Appl. **284**, 193–228 (1998)
20. Mahdian, M., Ye, Y., Zhang, J.: Approximation algorithms for metric facility location problems. SIAM J. Comput. **36**, 411–432 (2006)
21. Maranas, C.D. : Global optimization in Weber's problem with attraction and repulsion Part 7. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, pp. 1423–1427. Springer, Berlin (2009)
22. Overton, M.L.: A quadratically convergent method for minimizing a sum of Euclidean norms. Math. Program. **27**, 34–63 (1983)
23. Pardalos, P.M., Du, D.: Network Design: Connectivity and Facilities Location, DIMACS Series, vol. 40. American Mathematical Society, Providence (1998)
24. Qi, L.: Convergence analysis of some algorithms for solving nonsmooth equations. Math. Oper. Res. **18**, 227–244 (1993)
25. Qi, L., Sun, J.: A nonsmooth version of Newton's method. Math. Program. **58**, 353–367 (1993)
26. Qi, L., Zhou, G.: A smoothing Newton method for minimizing a sum of Euclidean norms. SIAM J. Optim. **11**, 389–410 (2000)
27. Sherali, H.D., Adams, W.P.: A reformulation-linearization technique for solving discrete and continuous nonconvex problems. Kluwer Academic, The Netherlands (1999)
28. Sherali, H.D., Al-Loughani, I., Subramanian, S.: Global optimization procedure for the capacitated Euclidean and l_p distance multifacility location-allocation problems. Oper. Res. **50**, 433–448 (2002)
29. Selim, S.: Biconvex programming and deterministic and stochastic location allocation problems. Ph. D. disseration, School of Industrial and System Engineering, Georgia Institute of Technology, Atlanta, Georgia (1979)
30. Sherali, H.D., Nordai, F.L.: NP-hard, capacitated, balanced p-median problems on a chain graph with a continuum of link demands. Math. Oper. Res. **13**, 32–49 (1988)
31. Sturm, J.F.: Using SeDumi 1.02, a MATLAB tool box for optimization over symmetric cones. Optim. Methods Softw. **11**, 625–653 (1999)
32. Wesolowsky, G.: The Weber problem: history and perspectives. Location Sci. **1**, 5–23 (1993)
33. Xue, G.L., Ye, Y.: An efficient algorithm for minimizing a sum of Euclidean norms with applications. SIAM J. Optim. **7**, 1017–1036 (1997)
34. Xue, G.L., Rosen, J.B., Pardalos, P.M.: A polynomial time dual algorithm for the Euclidean multifacility location problemn. Oper. Res. Lett. **18**, 201–204 (1996)
35. Levin, Y., Ben-Israel, A.: A heuristic method for large-scale multi-facility local problems. Comput. Oper. Res. **31**, 257–272 (2004)
36. Zhang, J., Chen, B., Ye, Y.: A multiexchange local search algorithm for the capacitated facility location problem. Math. Oper. Res. **30**, 389–403 (2005)