A NEWTON ACCELERATION OF THE WEISZFELD ALGORITHM FOR MINIMIZING THE SUM OF EUCLIDEAN DISTANCES

YUYING LI*

Abstract. The Weiszfeld algorithm for continuous location problems can be considered as an iteratively reweighted least squares method. It exhibits linear convergence. In this paper, a Newton type algorithm with similar simplicity is proposed to solve a continuous multifacility location problem with Euclidean distance measure. Similar to the Weiszfeld algorithm, at each iteration the main computation can be solving a weighted least squares problem. A Cholesky factorization of a symmetric positive definite band matrix, typically with a relatively small band width (e.g., a band width of two for a Euclidean location problem on a plane) is required. This new algorithm can be regarded as a Newton acceleration to the Weiszfeld algorithm with fast global and local convergence. The simplicity and efficiency of the proposed algorithm makes it particularly suitable for large-scale Euclidean location problems and parallel implementation. Computational experience also suggests that the proposed algorithm performs remarkably well in the presence of degeneracy and near degeneracy. In addition, it is proven to be globally convergent. Although the local convergence analysis is still under investigation, computation results suggest that it is typically superlinearly convergent.

Key Words. location problem, Euclidean distance, Weiszfeld method, Newton method

1. Multifacility Location Problems. A location problem places facilities to optimize some explicit or implicit spatially dependent objective. In the early 17^{th} century, Fermat first proposed the simplest location problem: Given three points in the plane, find the fourth point to minimize the sum of distances to the three given points. Since then, location problems have taken many different forms. Increasing computer power and development of iterative methods for solving location problems have led to many applications.

A large set of continuous location problems involve minimizing an objective function related to the Euclidean distance measurement between different locations. The objective of this paper is to propose a Newton process which accelerates the well-known Weiszfeld algorithm for these problems.

A continuous multifacility location problem with Euclidean distance measure can be described as follows, e.g., [11]. Assume that $c_i \in \Re^l$, $i = 1 : eloc$, denotes the existing facility locations in $\Re^l$ ($l = 2$ implies that the facilities are on a plane) and the cost per unit flow between any pair of facilities depends linearly on the Euclidean distance. The objective is to locate $nloc$ new facilities in $\Re^l$, represented as $[x_1, \cdots, x_{nloc}]$, $x_i \in \Re^l$, so that the total cost is minimized, i.e.,

$$(1) \qquad \min_{[x_1,\cdots,x_{nloc}]\in\Re^{nloc\times l}} \sum_{i=1}^{nloc}\sum_{j=1}^{eloc} w_{ij}\|x_i - c_j\| + \sum_{i=1}^{nloc-1}\sum_{j=i+1}^{nloc} \omega_{ij}\|x_i - x_j\|.$$

The location problem (1) can be described in a more general and concise matrix form. It can be easily verified that an Euclidean multifacility problem belongs to a class of more

* Department of Computer Science and Advanced Computing Research Institute, Cornell University, Ithaca, NY 14853, yuying@cs.cornell.edu. Research partially supported by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under grant DE-FG02-90ER25013.A000, and in part by NSF, AFOSR, and ONR through grant DMS-8920550, and by the Cornell Theory Center which receives major funding from the National Science Foundation and IBM corporation, with additional support from New York State and members of its Corporate Research Institute.

general location problems: minimization of the sum of Euclidean norms,

$$\min_{x\in\Re^n} \psi(x) \stackrel{\text{def}}{=} \sum_{i=1}^{m} \|A_i^T x - b_i\|, \tag{2}$$

where

$$A \stackrel{\text{def}}{=} [A_1, \cdots, A_m], \quad b \stackrel{\text{def}}{=} [b_1; \cdots; b_m],$$

$A_i \in \Re^{n\times l}$, $b_i \in \Re^l$, $i = 1, \cdots, m$ and l is a positive integer. Here we follow notations in Matlab [16]: the symbol "," augments columns to a matrix of the appropriate dimension while the symbol ";" augments rows to a matrix. In subsequent presentation, we assume that A has full row rank and denote the Euclidean norm by $\|\cdot\|$. A rectangular multifacility location problem is a linear l_1 problem and can be written as a problem (2) with $l = 1$. Subsequently, we assume $l \geq 2$ unless explicitly stated otherwise.

The main difficulty with minimizing the objective function $\psi(x)$ is the presence of nondifferentiability in the Euclidean distance. The nonlinear objective function $\psi(x)$ is convex but not continuously differentiable everywhere. For example, nondifferentiability for a location problem (1) occurs whenever any new facility coincides with an existing one or any pair of new facilities coincide with each other.

The most famous method for solving (1) is the Weiszfeld algorithm [20]. It is essentially an iteratively reweighted least squares procedure exhibiting a linear convergence rate. Conn and Calamai [2, 3, 4] propose a projected direction method and a related method has also been considered by Overton [14].

Compared to the projected direction methods [2, 3, 4, 14], the Weiszfeld algorithm is simple and suitable for large-scale computation. In addition, for large-scale problems the combinatorial nature of explicitly adding and dropping activities in a projection method can potentially lead to large number of iterations. On the other hand, in spite of its elegance, the Weiszfeld algorithm can be extremely slow due to its linear convergence rate. Recently, following the development of interior points methods for linear programming problems, Anderson [1] considers a barrier function method for (2) and Conn and Overton [8] investigate a possible primal and dual method.

In this paper, we propose a globally convergent algorithm for the Euclidean location problem (2). This new algorithm can be regarded as a Newton acceleration of the Weiszfeld algorithm. It retains the computational simplicity of the Weiszfeld algorithm but drastically improves its efficiency. Similar to the Weiszfeld algorithm, the main computation each iteration is solving a weighted least squares problem with the same dimensions. However, instead of diagonal weighting, the weighting matrix for the new algorithm is a symmetric positive definite band matrix with typically a very small bandwidth l; e.g., the bandwidth l equals two for a planar location problem. Compared to the Weiszfeld algorithm, the additional cost of our accelerated version is a Cholesky factorization of the banded weighting matrix.

The main objective of this paper is to motivate, develop and analyze the new algorithm for a Euclidean location problem (2). We present its global convergence analysis. In addition, substantial computational evidence demonstrating its feasibility and performance is provided. Fast (superlinear) local convergence is observed. Subsequently, we will review the Weiszfeld algorithm in §2. Then we will consider, in §3, a Newton process which can potentially lead to fast convergence. In §5, we assemble a globally convergent algorithm by observing the connection of the Newton process to the Weiszfeld algorithm. Convergence analysis and results are presented in §6 and computational issues are discussed in §7. Computational analysis, including comparison to the Weiszfeld algorithm will be presented in §8. Concluding remarks are given in §9.

Before continuing, we introduce some notations. Let r denote the residual vector, i.e.,

$$r \stackrel{\text{def}}{=} A^T x - b. \tag{3}$$

For any $v \in \Re^m$, let diag(v) denote a diagonal matrix with v_i as the i-th diagonal element. Similarly, if $V_i \in \Re^{l \times l}$, for $i = 1, \cdots, m$, bdiag(V_i) denotes a block diagonal matrix with V_i as the ith diagonal block.

2. The Weiszfeld Algorithm. The Weiszfeld algorithm is frequently referred to in the literature and has been used to solve various location problems. It was first proposed by Weiszfeld in 1937 [20] for solving a *single* facility location problem and later reanalyzed by Kuhn [9]. It has been further generalized to a *multifacility* location problem in [12]. The most striking feature of the Weiszfeld algorithm is its simplicity and use of a least squares solve as its main computation task. Its serious drawback is slow linear convergence.

For a Euclidean location problem (2), the Weiszfeld algorithm proceeds as follows.

At iteration k, assume that $\|r_i^k\| > 0$ for all $i = 1, \cdots, m$. The new iterate x^{k+1} is the solution of the least squares problem (4):

$$\min_{x \in \Re^n} \sum_{i=1}^{m} \frac{1}{\|r_i^k\|} r_i^T r_i, \quad \text{where} \quad r = A^T x - b. \tag{4}$$

Let R^k denote the positive definite diagonal matrix in (5):

$$R^k \stackrel{\text{def}}{=} \text{bdiag}\left(\frac{I_l}{\|r_i^k\|}\right), \tag{5}$$

where I_l is the l-by-l identity matrix. If we let $d_x^k = x^{k+1} - x^k$, then d_x^k is the solution of

$$AR^k A^T d_x = -AR^k r^k. \tag{6}$$

Alternatively, d_x^k can be computed from a weighted least squares problem,

$$R^{k\frac{1}{2}} A^T d_x \stackrel{\text{LS}}{=} -R^{k\frac{1}{2}} (R^{k^{-\frac{1}{2}}} r^k). \tag{7}$$

Assuming that, at each iteration, $\|r_i^k\| > 0$ for all $1 \le i \le m$, then the sequence $\{x^k\}$ is well defined and typically converges linearly to a solution x^* [9].

3. A Local Newton Process. The linear convergence of the Weiszfeld algorithm leads to computational inefficiency. We now offer a local Newton process by inspecting the optimality conditions of a Euclidean location problem (2).

Let the columns of Z form a basis for the null space of A, i.e., $AZ^T = 0$. Then the original problem (2) is equivalent to

$$
\begin{aligned}
&\min_{r\in\Re^{m\times l}} \phi(r) \stackrel{\text{def}}{=} \sum_{i=1}^{m} \|r_i\| \\
&\text{subject to} \quad Zr = Zb,
\end{aligned}
\tag{8}
$$

where $r = A^T x - b$. For any given vector r, there exists at most one vector x satisfying $r = A^T x - b$ under our assumption that A has full row rank. We will maintain $r = A^T x - b$ in our algorithm; hence $\phi(r) \equiv \psi(x)$. This alternative formulation (8) is used to facilitate motivating our new algorithm. We note, however, that explicit knowledge of Z is not necessary in the actual implementation.

The convex function $\phi(r)$ is not differentiable if $r_i = 0$ for some i. Let g and H denote the gradient and the Hessian of $\phi(r)$:

$$
\begin{aligned}
g &\stackrel{\text{def}}{=} [\frac{r_1}{\|r_1\|};\cdots;\frac{r_m}{\|r_m\|}], \\
H &\stackrel{\text{def}}{=} \text{bdiag}(\frac{1}{\|r_i\|}(I_l - \frac{r_i r_i^T}{\|r_i\|^2})).
\end{aligned}
$$

For notational convenience, we denote $\frac{\pm 0}{0} = 1$. Hence $g_i = [1;\cdots;1]$ if $r_i = 0$.

The optimality conditions for (2) are captured in the following theorem which paraphrases the well-known optimality conditions, e.g., [14].

THEOREM 1. *Assume that at x^*, $\{A_i : r_i^* = 0\}$ are linearly independent. Then x^* is a solution of (2) if and only if $r^* = A^T x^* - b$ satisfies*

$$
\begin{aligned}
&\|r_i\|(g_i - \lambda_i) = 0, \quad 1 \le i \le m, \\
&A\lambda = 0, \quad Zr = Zb,
\end{aligned}
\tag{9}
$$

and $\|\lambda_i\| \le 1$ if $r_i = 0$.

We subsequently say that a Euclidean location problem (2) is degenerate at x if either columns $\{A_i : r_i = 0\}$ are linearly dependent or $\|\lambda_i\| = 1$ for some $r_i = 0$. For example, a multifacility location problem (1) is degenerate if there are more than $nloc$ coinciding facilities or there are more than one new facilities coinciding with the same existing facility.

Intuitively, the nonlinear system of equations (9) is a natural choice for developing a Newton process. Indeed, this is the condition under which the primal-dual algorithm of [8] is based (convergence of the proposed algorithm [8] has not been addressed in the available preprint). It can be verified that the Newton step $d_x \in \Re^n$ for (9) satisfies

$$
A(R^k G^k \text{diag}(g^k - \lambda^k) + H^k) A^T d_x = -Ag^k,
$$

where $G = \text{bdiag}(G_i)$ and G_i is the Jacobian matrix of $\|r_i\| e_l$. Since we choose to solve (2) directly, a Newton direction is useful in this context when it is tied with a global descent process of the objective function. Unfortunately the matrix $RG\text{diag}(g - \lambda)$ is unsymmetric; it is not clear how to globalize this Newton direction of (9).

Alternatively, we consider the following nonlinear systems:

$$\begin{aligned} &\text{diag}(r)(g - \lambda) = 0 \\ &Zr = Zb, \quad A\lambda = 0. \end{aligned} \tag{10}$$

Condition (9) satisfies (10) plus $g_i - \lambda_i = 0$ if $r_i \neq 0$. Although (10) captures optimality less accurately than (9), it is simpler and, as we will see later, the Newton step for (10) can be readily globalized.

Admittedly, the nonlinear system (10) is unusual for developing a Newton process: the Jacobian of the nonlinear system (10) does not exist at $r_i^* = 0$. However, similar nonlinear systems with nondifferentiability at a solution have been used in the minimization context to achieve fast local convergence [6, 5, 7, 10]. Compared to the nonlinear systems in [6, 5, 7, 10], (10) is more complicated; instead of ± 1 as in the case of a l_1 problem [6], g_i can be any point on a unit sphere $\|r_i\| = 1, r_i \in \Re^l$. Nonetheless, we believe that this Newton process leads to fast local convergence (a theoretical analysis is ongoing research). In §8, we will provide sufficient computational evidence to demonstrate that fast local convergence does occur.

A Newton process emerges by considering a Newton step for (10):

$$\begin{bmatrix} \text{diag}(g^k - \lambda^k) + \text{diag}(r^k)H^k & -r^k \\ 0 & A \\ Z & 0 \end{bmatrix} \begin{bmatrix} d \\ d_\lambda \end{bmatrix} = \begin{bmatrix} -\text{diag}(r^k)(g^k - \lambda^k) \\ 0 \end{bmatrix}. \tag{11}$$

The unsymmetric linear system (11) has dimension $2ml$-by-$2ml$. If Z is explicitly available, we can compute d_x^k and $d_\lambda^k = Z^T d_w^k$ by solving an unsymmetric ml-by-ml linear system:

$$\begin{bmatrix} \text{diag}(g^k - \lambda^k)A^T + \text{diag}(r^k)H^k A^T & -r^k Z^T \end{bmatrix} \begin{bmatrix} d_x \\ d_w \end{bmatrix} = -\text{diag}(r^k)(g^k - \lambda^k).$$

If we assume that x^k is such that $|r_i^k| > 0$, then the Newton equation (11) can be obtained by solving:

$$\begin{aligned} &A(\text{diag}(\frac{g^k - \lambda^k}{r^k}) + H^k)A^T d_x = -Ag^k, \\ &d^k = A^T d_x^k, \\ &\lambda^{k+1} = g^k + (\text{diag}(\frac{g^k - \lambda^k}{r^k}) + H^k)d^k. \end{aligned} \tag{12}$$

Assuming that (11) has a unique solution, it is then clear that $A\lambda^{k+1} = 0$ if $A\lambda^k = 0$.

Computation using (12) is appealing due to reduction of the dimension and the matrix Z does not appear in the equations. However, if there exists some components of r_i near zero, then the linear system (12) is ill conditioned and H_i^k can approach infinity. Asymptotically, it is possible to solve (12) in a relatively stable fashion with a least squares solve. We delay discussion of the computational issues until §7.

We conclude this section with an important observation: the Newton direction (12) d_x^k bears a close relation to the Weiszfeld step: setting H^k and λ^k to zero, the equation (12) becomes a simple least squares problem

$$AR^kA^Td_x = -Ag^k,$$

which is exactly the Weiszfeld step (6) (using that $g^k = R^kr^k$). We explore this observation in globalizing the local Newton process next.

4. Towards Global Convergence: Determining a Good Descent Direction. Now we make the Newton process useful by connecting it with a globally convergent process.

If we modify the Newton step (12),

$$A(\text{diag}(|\frac{g^k - \lambda^k}{r^k}|) + H^k)A^Td_x = -Ag^k, \tag{13}$$

then the solution d_x^k of the above system will provide a descent direction. If we can determine a good descent direction at every point, convergence towards a solution becomes promising. Examining the Newton step (11), a potential troubling spot is near a point satisfying

$$\text{diag}(r)(g-\lambda) = 0, \quad A\lambda = 0 \quad \text{and} \quad Zr = 0,$$

since a point satisfying above conditions may not be optimal (but the corresponding Newton step is zero).

Fortunately, we know that the Weiszfeld algorithm is a globally convergent algorithm and the Newton process described has a simple connection to the Weiszfeld algorithm. This motivates our globalization process below.

Let D_r^k denote the diagonal scaling matrix,

$$D_r^k \stackrel{\text{def}}{=} \text{diag}(|r^k|). \tag{14}$$

Let the parameter η_0 measure satisfaction of the condition $\text{diag}(r)(g-\lambda) = 0$, η_{cs} measure satisfaction of the complementarity conditions and η_{df} measure dual feasibility:

$$\eta_0^k \stackrel{\text{def}}{=} \max(\min(\frac{|r^k|}{\|r^k\|_\infty}, |g^k - \lambda^k|), \tag{15}$$

$$\eta_{cs}^k \stackrel{\text{def}}{=} \max_{1\le i\le m} (\min(\frac{{R^k}^{-1}e_m}{\|r^k\|_\infty}, |g^k - \lambda^k|)), \tag{16}$$

$$\eta_{df}^k \stackrel{\text{def}}{=} \max_{\|\lambda_i^k\|>1} (\|g_i^k - \lambda_i^k\|_\infty), \tag{17}$$

where e_m is the vector of all ones in $\Re^m$ (hence ${R^k}^{-1}e_m = [\|r_1\|e_l; \cdots; \|r_m\|e_l]$).

Let $0 < \theta^k \le 1$ be a measurement of optimality:

$$\theta^k \stackrel{\text{def}}{=} \begin{cases} 1, & \text{if } \eta_0^k < \rho_0 \max(\eta_{cs}^k, \eta_{df}^k), \\ \frac{\max(\eta_{cs}^k, \eta_{df}^k)}{1+\max(\eta_{cs}^k, \eta_{df}^k)}, & \text{otherwise,} \end{cases} \tag{18}$$

where $1 > \rho_0 > 0$ is a small constant (e.g., $\rho_0 = 10^{-3}$ in our implementation). It is clear that, if $\theta = 0$ at (x, λ) with λ satisfying $A\lambda = 0$, then x is a solution to (2).

Based on the optimality measure θ^k, we define a transition between the Weiszfeld step (6) and the Newton step (11): we replace the diagonal matrix $\mathrm{diag}(|g^k - \lambda^k|)$ in (13) by the diagonal matrix D_θ^k which is a linear combination of $\mathrm{diag}(|g^k - \lambda^k|)$ (used in defining a Newton step) and $\mathrm{diag}(|g^k|)$ (used in a Weiszfeld step):

$$D_\theta^k \stackrel{\text{def}}{=} (1 - \theta^k)\mathrm{diag}(|g^k - \lambda^k|) + \theta^k \mathrm{diag}(|g^k|). \tag{19}$$

Using the scaling matrix D_θ^k and D_r^k, at any point x^k with $|r^k| > 0$, we compute a descent direction d_x^k from

$$A(D_\theta^k D_r^{k^{-1}} + (1 - \theta^k)H^k)A^T d_x = -Ag^k. \tag{20}$$

The globalization matrix D_θ^k plays an important role in this transition from a Weiszfeld step to a Newton step. When far away from the solution, i.e., θ^k is close or equal to unity, (20) yields a step resembles the Weiszfeld step (6). When close to the solution, i.e., θ^k close to zero, (20) produces an increasingly accurate approximation to the Newton step (11). Therefore, the globalization can achieve both global convergence and local fast convergence.

5. The Proposed Newton Algorithm. In order to converge to a solution, a sufficiently large stepsize needs to be determined along a descent direction to ensure sufficient progress. A sophisticated and sequential line search procedure can reduce the appeal of a descent algorithm, particularly for large-scale problems and parallel computing environment. Indeed, the simplicity of the Weiszfeld algorithm includes the fact that no line search is needed in the computation.

To design a simple line search procedure, we gain insight, once more, by examining the Weiszfeld algorithm. In the convergence proofs for the Weiszfeld algorithm, Kuhn [9] considers the following quadratic function $q^k(r)$ at r^k:

$$q^k(r) \stackrel{\text{def}}{=} \sum_{i=1}^{m} \frac{\|r_i\|^2}{\|r_i^k\|}. \tag{21}$$

The quadratic $q^k(r)$ has the property that $q^k(r^k) = \phi(r^k)$, where $\phi(r) = \sum_{i=1}^m \|r_i\|$ is the objective function (8). Moreover, we have

$$\begin{aligned} q^k(r) - q^k(r^k) &= \sum_{i=1}^{m} \frac{(\|r_i^k\| + (\|r_i\| - \|r_i^k\|))^2}{\|r_i^k\|} - \phi(r^k), \\ &= 2(\phi(r) - \phi(r^k)) + \sum_{i=1}^{m} \frac{(\|r_i\| - \|r_i^k\|)^2}{\|r_i^k\|}. \end{aligned}$$

If $q^k(r) < q^k(r^k) = \phi(r^k)$, we have

$$\begin{aligned} \phi(r) - \phi(r^k) &\le \frac{1}{2}(q^k(r) - q^k(r^k)) \\ &< 0. \end{aligned}$$

In other words, decrease of $q^k(r)$ leads to decrease of $\phi(r)$ at r^k. This suggests that we can determine a stepsize by minimizing the quadratic $q^k(r)$.

Given a descent direction d^k for $\phi(r)$ at r^k, let α_*^k denote the minimizer of $q^k(r^k + \alpha d^k)$:

$$\alpha_*^k = -\frac{{g^k}^T d^k}{{d^k}^T R^k d^k}, \tag{22}$$

where R^k is the positive diagonal matrix defined in (5). A simple line search procedure emerges. At the kth iteration, the unit stepsize is accepted if it leads to reduction of $\phi(r)$. Otherwise, α_*^k is taken. Since $q^k(r^k + \alpha d^k) - q^k(r^k) = 2\alpha {g^k}^T d^k + \alpha^2 {d^k}^T R^k d^k$ and $q^k(r^k + \alpha_*^k d^k) - q^k(r^k) = \alpha_*^k {g^k}^T d^k$,

$$\phi(r^k + \alpha_*^k d^k) - \phi(r^k) \le \frac{\alpha_k^*}{2} {g^k}^T d^k = \frac{({g^k}^T d^k)^2}{2{d^k}^T R^k d^k}. \tag{23}$$

We now assemble the globalized Newton algorithm in FIG. 1.

The N-Weiszfeld Algorithm. Assume that x^0 such that $|r^0 = A^T x^0 - b| > 0$ is given and $\lambda^0 = 0$. Assume that ρ_0 is a small positive constant, , e.g., $\rho_0 = 10^{-3}$.

Step 1 Compute θ^k and diagonal matrices D_θ^k from (18) and $D_r^k = \text{diag}(|r^k|)$.

Step 2 Compute d_x^k as a solution of

$$A(D_\theta^k {D_r^k}^{-1} + (1-\theta^k)H^k)A^T d_x = -Ag^k,$$

and update λ^{k+1}:

$$\begin{aligned} d^k &= A^T d_x^k, \\ \lambda^{k+1} &= g^k + (D_\theta^k {D_r^k}^{-1} + (1-\theta^k)H^k)d^k. \end{aligned}$$

Step 3 Compute α_*^k:

$$\alpha_*^k = -\frac{{g^k}^T d^k}{{d^k}^T R^k d^k}.$$

If $\psi(x^k + d_x^k) < \psi(x^k)$, $\alpha^k = 1$. Otherwise $\alpha^k = \alpha_*^k$. Update:

$$x^{k+1} = x^k + \alpha^k d_x^k.$$

FIG. 1. *The N-Weiszfeld Algorithm for Minimizing Sum of Euclidean Norms*

Remark 1. The algorithm remains simple: If $\theta^k \equiv 1$ in **Step 1** of the N-Weiszfeld algorithm, then FIG. 1 describes the Weiszfeld algorithm. As is clear from FIG. 1, there is a crucial difference between the Weiszfeld algorithm and the N-Weiszfeld algorithm. The Weiszfeld step depends uniquely on the current position x^k. The descent direction (step) for the N-Weiszfeld algorithm, on the other hand, is determined by both the current position x^k and dual multiplier approximation λ^k.

Remark 2. An implicit assumption of the description of the N-Weiszfeld algorithm is that $|r^k| > 0$ at each iteration. A similar assumption, $\|r_i^k\| > 0$, is used in the description and analysis of the Weiszfeld algorithm, e.g., [9]. Next we will establish global convergence property of the new algorithm under this assumption. In §7, we will discuss how to avoid zero residual components of r computationally.

Remark 3. If we set $l = 1$, the new algorithm in FIG. 1 is essentially the algorithm of [6] for l_1 problems with a slightly different globalization matrix D_θ. For l_1 problems, $|g^k| = 1$. The globalization matrix D_θ in [6] equals diag$(|g - (1-\theta)\lambda|)$ which is better for l_1 problems in our experience.

6. Global Convergence Properties. We now prove a global convergence property of the N-Weiszfeld algorithm in FIG. 1: every limit point of the sequence $\{x^k\}$ generated by the N-Weiszfeld algorithm is a solution of (2) if the problem is nondegenerate at a limit of $\{x^k\}$.

The main convergence results are in Theorem 4. We first prove two technical lemmas.

LEMMA 2. *The matrix D_θ^k defined in (19) satisfies*

$$\theta^k \mathit{diag}(|g^k|) \le D_\theta^k \le (2+\theta^k)I, \tag{24}$$

where I is the identity matrix with the same dimension of D_θ^k.

Proof. From the definition (19) of D_θ^k,

$$D_\theta^k = (1-\theta^k)\text{diag}(|g^k - \lambda^k|) + \theta^k \text{diag}(|g^k|),$$

and so the first inequality obviously holds. Using definition (18) of θ^k and definition (17) of η_{df}^k,

$$\begin{aligned}(1-\theta^k)\text{diag}(|g^k-\lambda^k|) &\le \max(2, \frac{1}{1+\eta_{df}^k}\text{diag}(|g^k-\lambda^k|)) \\ &\le \max(2, \frac{|g_i^k-\lambda_i^k|}{1+|g_i^k-\lambda_i^k|}) \\ &\le 2.\end{aligned}$$

The proof is completed. ■

The descent direction $d^k = A^T d_x^k$ given by (10) can be considered in an alternative way:

$$\begin{bmatrix} D_\theta^k + (1-\theta^k)D_r^k H^k & -D_r^k \\ 0 & A \\ Z & 0 \end{bmatrix} \begin{bmatrix} d \\ d_\lambda \end{bmatrix} = \begin{bmatrix} -D_r^k(g^k-\lambda^k) \\ 0 \\ 0 \end{bmatrix}, \tag{25}$$

or

$$\begin{bmatrix} D_\theta^k + (1-\theta^k)D_r^k H^k & -D_r^k \\ 0 & A \\ Z & 0 \end{bmatrix} \begin{bmatrix} d^k \\ \lambda^{k+1} \end{bmatrix} = - \begin{bmatrix} D_r^k g^k \\ 0 \\ 0 \end{bmatrix}.$$

The above equations will be used in establishing our next Lemma.

LEMMA 3. *Assume that the matrix A has full row rank. If $\liminf_{k\to\infty} \theta^k > 0$, then $\lim_{k\to\infty} {g^k}^T d^k = 0$, $\lim_{k\to\infty} d^k = 0$, $\lim_{k\to\infty} D_r^k(g^k - \lambda^{k+1}) = 0$ and $\{r^k\}$ converges.*

Proof. By description of N-Weiszfeld in FIG. 1, we have

$$\phi(r^0) > \phi(r^1) > \cdots > \phi(r^k) > \phi(r^{k+1}) > \cdots. \tag{26}$$

Since $\{\phi(r^k)\}$ is bounded below by zero,

$$\lim_{k\to\infty} (\phi(r^k) - \phi(r^{k+1})) = 0. \tag{27}$$

Since $\{\phi(r^k) = \sum_{i=1}^m \|r_i^k\|\}$ is bounded, it is clear that $\{r^k\}$ is bounded. We now prove that

$$\lim_{k\to\infty} {g^k}^T d^k = 0.$$

From (20), H^k is positive definite, and $D_\theta^k \geq \theta^k \text{diag}(|g^k|)$, we have

$$\begin{aligned} -{g^k}^T d^k &\geq {d^k}^T D_\theta^k {D_r^k}^{-1} d^k, \\ &\geq \theta^k {d^k}^T \text{diag}(|g^k|) {D_r^k}^{-1} d^k. \end{aligned}$$

But ${D_r^k}^{-1}\text{diag}(|g^k|) = R^k$, hence

$$-{g^k}^T d^k \geq \theta^k {d^k}^T R^k d^k. \tag{28}$$

Using definition (22) of α_*^k, $\alpha_*^k \geq \theta^k$.

From (23) and $\alpha_*^k \geq \theta^k$, we have

$$\liminf_{k\to\infty} (\phi(r^{k+1}) - \phi(r^k)) \leq \frac{1}{2} \liminf_{k\to\infty} \theta^k {g^k}^T d^k \leq 0.$$

From $\liminf_{k\to\infty} \theta^k > 0$ and $\lim_{k\to\infty} (\phi(r^{k+1}) - \phi(r^k)) = 0$, we conclude that $\lim_{k\to\infty} {g^k}^T d^k = 0$.

Using the boundedness of $\{r^k\}$, $\liminf_{k\to\infty} \theta^k > 0$ and (28), there exists $\chi > 0$ such that

$$-{g^k}^T d^k \geq \theta^k {d^k}^T R^k d^k \geq \chi \|d^k\|^2. \tag{29}$$

Hence $\lim_{k\to\infty} d^k = 0$. Since the matrices $\{D_\theta^k + (1-\theta^k)D_r^k H^k\}$ are bounded, using (25), (24) and $\lim_{k\to\infty} d^k = 0$, we immediately obtain that $\lim_{k\to\infty} D_r^k(g^k - \lambda^{k+1}) = 0$.

From (23) and (29),

$$|\phi(r^{k+1}) - \phi(r^k)| \geq -\frac{\alpha_k^*}{2} g^{k^T} d^k \geq \frac{\chi \alpha_k^*}{2} \|d^k\|^2,$$

hence $\lim_{k\to\infty} \alpha_k^* d^k = 0$, following (27). But $\lim_{k\to\infty} d^k = 0$ and $\alpha^k \in \{1, \alpha_k^*\}$. Hence $\lim_{k\to\infty} \alpha^k d^k = 0$.

Since $\{r^k\}$ is bounded and $\{\alpha^k d^k\}$ converges to zero, the set of limit points of $\{r^k\}$ is closed and connected ([13], p.478). Using (26), $\phi(\bar{r}) = \phi(r^*)$, for any limit points $\bar{r}$ and r^* of $\{r^k\}$. But $\phi(r) = \sum_{i=1}^m \|r_i\|$ is a strictly convex function (even though $\psi(x)$ may not be strictly convex). Hence there is a unique limit point of $\{r^k\}$, i.e., $\lim_{k\to\infty} r^k = r^*$. ■

Now we are ready for the global convergence result.

THEOREM 4. *Assume that the matrix A has full row rank. Let $\{x^k\}$ be generated by the N-Weiszfeld algorithm in FIG. 1. If the columns $\{A_i :\ {r_i}^* = 0\}$ are linearly independent at a limit point x^* of $\{x^k\}$, then every limit of $\{x^k\}$ is a solution of (2).*

Proof.

If $\liminf_{k\to\infty} \max(\eta_{cs}^k, \eta_{df}^k) = 0$, then there exists a limit point which is a solution. From the monotonicity of $\phi(x^k)$, every limit point of $\{x^k\}$ is a solution.

We now prove that is a limit point of $\{\max(\eta_{cs}^k, \eta_{df}^k)\}$ by contradiction. Assume that $\liminf_{k\to\infty} \max(\eta_{cs}^k, \eta_{df}^k) > 0$. Then $\liminf_{k\to\infty} \theta^k > 0$. From the assumption that the columns $\{A_i :\ r_i^* = 0\}$ are linearly independent, there exists a unique λ^* satisfying (9). Therefore $\lim_{k\to\infty} \lambda^{k+1} = \lambda^*$.

From Lemma 3, $\{r^k\}$ converges and $\lim_{k\to\infty} \text{diag}(r^k)(g^k - \lambda^{k+1}) = 0$. Thus $\eta_0^* = 0$. Using definition (18) of θ^k, $\liminf_{k\to\infty} \max(\eta_{cs}^k, \eta_{df}^k) > 0$.

Since $\liminf_{k\to\infty} \max(\eta_{cs}^k, \eta_{df}^k) > 0$, $\eta_0^k < \rho_0 \max(\eta_{cs}^k, \eta_{df}^k)$ for sufficiently large k. Hence $\theta^k = 1$ for sufficiently large k. From (20), $g^{k^T} d^k = -d^{k^T} R^k d^k$ when $\theta^k = 1$. Therefore, using (22), $\alpha_*^k = 1$ whenever $\theta^k = 1$. Thus, for sufficiently large k,

$$r^{k+1} = r^k + d^k.$$

Using $D_r^k = \text{diag}(|r^k|)$ and $D_\theta^k = \text{diag}(|g^k|)$,

$$d^k = \text{diag}(|r^k|)\text{diag}(|g^k|^{-1})(g^k - \lambda^{k+1}).$$

Therefore

$$r^{k+1} = -\|r^k\| \text{sgn}(r^k) \lambda^{k+1}. \tag{30}$$

Hence, for any $r_{i\,j}^* = 0$ and $r_i^* \neq 0$, we immediately conclude that $\lambda_{i\,j}^* = 0$ from (30). Together with $\eta_0^* = 0$, we know that $\eta_{cs}^* = 0$.

Consider any $r_i^* = 0$. We prove that $\|\lambda_i^*\| \leq 1$ by contradiction. Assume that $r_i^* = 0$ but $\|\lambda_i^*\| > 1$.

Using (30) and $\|\lambda_i^*\| > 1$, for sufficiently large k,

$$\|r_i^{k+1}\| = \|r_i^k\| \|\lambda_i^{k+1}\| > \|r_i^k\|.$$

This implies that $\{r_i^k\}$ cannot converge to zero, which is a contradiction. Therefore $\liminf_{k\to\infty} \theta^k = 0$. The proof is complete. ■

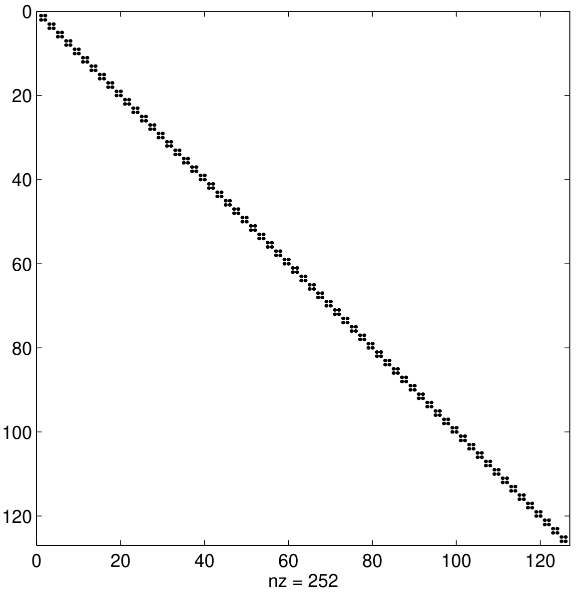

FIG. 2. *Sparsity Struction of H for a Location Problem with Three New Facilities and Twenty Exisiting Facilities*

7. A Computational Implementation. The description of the N-Weiszfeld algorithm in FIG. 1 assumes that $|r^k| > 0$, i.e., each residual component is not zero during the computation process. A similar restriction is typical in the context of an interior point method for linear programming problems [17].

First we note that, taking the unit or stepsize α_k^*, it is unlikely that a component of r^k becomes exactly zero. Moreover, the requirement $|r^k| > 0$ can be achieved using a backtracking technique in a similar fashion to an interior point algorithm paradigm for linear programming problems.

Assume that $|r^k| > 0$. If some component of $r^k + \alpha^k d^k$ is exactly zero, we can compute the maximum breakpoint which is smaller than α^k:

$$\beta^k \stackrel{\text{def}}{=} \max_{1 \le j \le ml} \{-\frac{r_j^k}{d_j^k} : 0 \le -\frac{r_j^k}{d_j^k} < \alpha^k\},$$

and backtrack $\alpha^k \leftarrow \beta^k + \max(0.975, (1-\theta^k))(\alpha^k - \beta^k)$.

Now we examine how to compute a descent direction (20), i.e., solving

$$A(D_\theta^k {D_r^k}^{-1} + (1-\theta^k)H^k)A^T d = -Ag^k.$$

We observe that the matrix $D_\theta^k {D_r^k}^{-1} + (1-\theta^k)H^k$ has the same sparsity structure of H^k; a symmetric positive definite banded matrix with a small (typically) band width l. FIG. 2 illustrates the sparsity structure of a Hessian H^k for a multifacility location problem on a plane with $m = 2$ and $n = 20$.

Exploiting the band structure of the symmetric positive definition matrix $D_\theta^k {D_r^k}^{-1} + (1 - \theta^k)H^k$, we can compute, relatively efficiently, a Cholesky factorization

$$D_\theta^k {D_r^k}^{-1} + (1-\theta^k)H^k = G^k {G^k}^T.$$

Then the solution of $A(D_\theta^k {D_r^k}^{-1} + (1-\theta^k)H^k)A^T d = -Ag^k$ can be computed by firstly solving

$$G^k y^k = g^k$$

and then computing d_x^k from the weighted least squares problem

$${G^k}^T A^T d \overset{\mathrm{LS}}{=} -y^k.$$

Similar to the weighted least squares problems in many interior point methods for linear programming problems, the symmetric positive definite weighting matrix G^k is typically ill conditioned. In [15, 19] it has been demonstrated that least squares problems with ill-conditioning diagonal weighting matrices can be solved stably; however stable algorithms are yet to be developed in the large-scale problem setting. Using a simple heristic approach suggested in [18], we attempt to compute the weighted least squares solution accurately by reordering the rows of the coefficient matrix ${G^k}^T A^T$ according to the scaling $D_\theta^k {D_r^k}^{-1}$: the largest diagonal arranged to be at the first row and descending to the smallest diagonal arranged to the last row.

8. Computational Results. We now demonstrate computationally that the proposed N-Weiszfeld algorithm converges quickly to a solution. In contrast to the Weiszfeld algorithm, fast global and local convergence of N-Weiszfeld is exhibited. We also report performance of N-Weiszfeld on large-scale problems by generating random multifacility location problems (1), i.e., the existing locations are random entries. We provide detail of the CPU time expenditures for major operations required by N-Weiszfeld. We demonstrate that, the additional per iteration cost of the proposed N-Weiszfeld algorithm over the original Weiszfeld algorithm is often negligible as problem size increases.

Both the Weiszfeld algorithm and N-Weiszfeld are implemented in Matlab [16] on a Sun Sparc-2 workstation. The Weiszfeld algorithm is obtained by simply setting $\theta^k = 1$ in N-Weiszfeld implementation.

We report the following characteristics of the computed solution:

1. opt: an optimality accuracy measure, opt $\overset{\text{def}}{=} \max(\eta_{cs}^*, \eta_{df}^*)$;
2. deg: a possible degeneracy measure:

$$\deg \overset{\text{def}}{=} \min_{1 \le i \le m} \{\|r_i\| + |\|\lambda_i\| - 1|\}.$$

The stopping criteria for both the Weiszfeld and N-Weiszfeld algorithms are the following: the computation halts when

$$\begin{array}{ll} \text{either} & \text{it} > \text{itbound}, \\ \text{or} & \max(\eta_{cs}^k, \eta_{df}^k) \le \text{tol}, \\ \text{or} & \frac{|\psi(x^{k+1}) - \psi(x^k)|}{\psi(x^{k+1})} \le 100\epsilon_{mach}, \end{array}$$

with tol $= 10^{-10}$ and the Matlab machine epsilon $\epsilon_{mach} = 1.6 \times 10^{-16}$.

Example	I ($\omega = 2$)	II ($\omega = 1$)	III($\omega = 1.414$)	IV($\omega = \sqrt{2}$)	V
N-Weiszfeld	7	6	24	39	10
Weiszfeld	73	38	>600	>600	98
Overton [14]	6	7	12	NA	29
Conn & Calamai[4]	6	9	10	NA	27
Degeneracy Measure	10^{-1}	10^{-1}	10^{-4}	10^{-8}	10^{-2}

TABLE 1
Comparisons on Small Testing Problems

The same starting point x^0 is used in both the Weiszfeld and N-Weiszfeld algorithms: a least squares solution:

$$A^T x \overset{\text{LS}}{=} b.$$

In §8.1, we compare the proposed N-Weiszfeld algorithm with the Weiszfeld algorithm using five small examples in [14]. We demonstrate that the N-Weiszfeld algorithm converges rapidly to the solution and achieves better accuracy in the presence of near degeneracy.

In §8.2, we illustrate performance of N-Weiszfeld on large-scale random multifacility location problems (1). We demonstrate that the number of iterations grows very slowly, if at all, as the problem sizes increase. Problem RMFL 4 also shows that the behavior of N-Weiszfeld is not significantly altered in the presence of degeneracy.

Finally in §8.3, we compare computation time for three major tasks: forming Hessian matrix H^k, performing Cholesky factorization and solving a weighted least squares problem.

8.1. Comparison to the Weiszfeld Algorithm. We illustrate the performance of the N-Weiszfeld and Weiszfeld algorithms with the following five examples used in [14].

Example 1 – 4 can be described in the form of (2) with the following specifications

$$\begin{array}{lll} n = 2 & l = 2 & m = 3 \\ A_1 = I & A_2 = \omega I & A_3 = I \\ b_1 = [-1; 0] & b_2 = [0; \omega] & b_3 = [1; 0]. \end{array} \tag{31}$$

Example 5. This is a multifacility location problem (1) with the following specifications:

$$l = 2, \quad m = 5, \quad n = 9$$

and

$$[w_{ij}] = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ & 0 & 1 & 10^{-2} & 10^{-1} \\ & & 0 & 10^{-2} & 10^{-1} \\ & & & 0 & 10^{-1} \\ & & & & 0 \end{bmatrix}, \quad [\omega_{ij}] = \begin{bmatrix} 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \end{bmatrix},$$

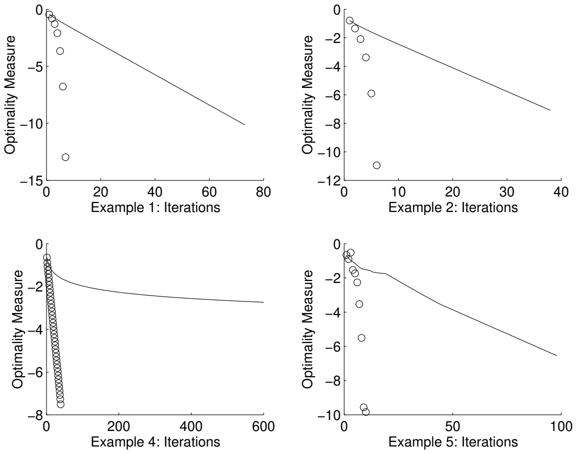

FIG. 3. *The N-Weiszfeld algorithm (circle-line) and the Weiszfeld algorithm (solid line)*

and

$$[c_1, \cdots, c_9] = \begin{bmatrix} 0 & 2 & 6 & 6 & 8 & 7 & 0 & 0 & 0 \\ 0 & 4 & 2 & 10 & 8 & 7 & 1 & 2 & 3 \end{bmatrix}.$$

For Example 3 and 4, the location problems are nearly degenerate, deg $= 4.3 \times 10^{-4}$ for Example 3 and deg $= 4.9 \times 10^{-8}$ for Example 4. The N-Weiszfeld algorithm performs satisfactorily, achieving an accuracy of 10^{-12} for Example 3 and 10^{-9} for Example 4. The Weiszfeld algorithm, on the other hand, is only able to achieve accuracy of no more than 10^{-3} in 600 iterations.

FIG. 3 depicts the changes of the logarithmic (base 10) optimality measure versus the iteration accounts. For all the five examples, the Weiszfeld algorithm exhibits linear convergence, particularly slow for Example 3 and 4 due to near degeneracy. Our new N-Weiszfeld, on the other hand, achieves fast (superlinear) local convergence for Example 1, 2 and 5. Moreover, for near degenerate problems Example 3 and 4, better accuracy is obtained.

8.2. Large-scale Random Multifacility Location Problems. We now report some computational results of the proposed N-Weiszfeld algorithm on random multifacility location (RMFL) problems (1). Each iteration number entry in Table 2-5 is an average of 10 random test problem instances. Both the best and the worst computed optimality accuracy $(\text{opt}^*_{min}, \text{opt}^*_{max})$ over these 10 instances are given. The dimension of the matrix A in (2) varies from 20×1090 to 360×21420.

Table 2-4 indicate that small numbers of iterations are required to compute an accurate solution. Moreover, the number of iterations changes slightly with the problem size. Table 5 illustrates performance of N-Weiszfeld on degenerate problems where the number of coinciding locations is greater than n. Asymptotically, superlinear convergence is observed for results in Table 2-4, while linear convergence is occasionally observed in Table 5, due to degeneracy.

RMFL 1. The existing locations are **rand**$(2, eloc)$ and the weights are set up:

$$w_{ij} = 100|\textbf{rand}| + 1,$$

$nloc = 10,\ l = 2$				
existing locations	50	100	200	300
iterations	8.8	9	10.3	12
coinciding locations	5.1	5.1	4.5	4.4
opt	$(10^{-15}, 10^{-8})$	$(10^{-14}, 10^{-9})$	$(10^{-15}, 10^{-8})$	$(10^{-15}, 10^{-8})$
deg	$(10^{-3}, 10^{-2})$	$(10^{-3}, 10^{-2})$	$(10^{-3}, 10^{-2})$	$(10^{-3}, 10^{-2})$

TABLE 2
Location Problems on a Plane with Ten New Facilities

$eloc = 120,\ l = 2$				
new locations	25	50	75	100
iterations	11.5	12.6	14.8	16.1
coinciding locations	12.8	24.5	36.8	49.8
opt	$(10^{-14}, 10^{-7})$	$(10^{-5}, 10^{-14})$	$(10^{-7}, 10^{-13})$	$(10^{-8}, 10^{-13})$
deg	$(10^{-3}, 10^{-3})$	$(10^{-4}, 10^{-2})$	$(10^{-4}, 10^{-3})$	$(10^{-4}, 10^{-3})$

TABLE 3
Random Multifacility Location Problems on a Plane with 120 *Existing Facilities*

$$w_{ij} \leftarrow 1000 w_{ij} \quad \text{for} \quad i = 1:2:nloc \text{ and } j = 1:2:nloc, \quad \text{and,}$$

$$\omega_{ij} = 100|\mathbf{rand}| + 1.$$

In this experiment, we have fixed the number of new facilities and varied the number of existing facilities. The dimension of the matrix A for Table 2 ranges from 20-by-1090 to 20-by-6090. In Table 2, the worst accuracy obtained is 10^{-8} and the best is 10^{-15}.

RMFL 2. We now fix the number of existing locations and vary the number of new facilities. The existing locations are again generated by the Matlab function **rand**$(2, eloc)$ and the weights are set up as below:

$$w_{ij} = 100|\mathbf{rand}| + 1,$$

$$w_{ij} \leftarrow 1000 w_{ij}, \quad \text{for } i = 1:2:nloc \text{ and } j = 1:2:nloc,$$

but

$$\omega_{ij} = |\mathbf{rand}| + 1.$$

The test results are summarized in Table 3. The dimensions of the matrix A range from 50-by-6600 to 200-by-33900. Performance of N-Weiszfeld on **RMFL 2** is similar to that of **RMFL 1**.

$eloc=$ 50, $nloc=$ 20				
l	6	10	14	18
iterations	7.9	7.3	7.5	8.3
coinciding locations	10	10	9.9	9.9
opt	$(10^{-15}, 10^{-7})$	$(10^{-8}, 10^{-14})$	$(10^{-15}, 10^{-7})$	$10^{-15}, 10^{-5})$
deg	$(10^{-2}, 10^{-2})$	$(10^{-2}, 10^{-1})$	$(10^{-2}, 10^{-1})$	$(10^{-2}, 10^{-1})$

TABLE 4
Random Multifacility Location Problems with Different l

$eloc=$ 120, $l=2$				
new locations	25	50	75	100
iterations	12.7	13.7	27.4	29.3
coinciding locations	63.1	326.7	734.7	1301.7
opt	$(10^{-15}, 10^{-5})$	$(10^{-15}, 10^{-4}$	$(10^{-13}, 10^{-5})$	$(10^{-11}, 10^{-5})$
deg	$(10^{-4}, 10^{-1})$	$(10^{-3}, 10^{-1})$	$(10^{-2}, 10^{-1})$	$(10^{-3}, 10^{-2})$

TABLE 5
Random Multifacility Location Problems on a Plane with 120 Existing Facilities

RMFL 3. We now fix both the number of existing locations and new locations but vary the dimension l. The weights w_{ij} and ω_{ij} are specified in the same as way as in **RMFL 2**.

The dimensions of the matrix A for Table 4 range from 40-by-2380 to 360-by-21420. It can be observed that the number of iterations grows only slightly as the dimension l increases.

RMFL 4. As in RMFL 2, we fix the number of existing locations and vary the number of new facilities. However, the weights are set up:

$$w_{ij} = 100|\textbf{rand}| + 1,$$

$$w_{ij} \leftarrow 1000 w_{ij}, \quad \text{for } i = 1:2:nloc \text{ and } j = 1:2:nloc,$$

but

$$\omega_{ij} = 100|\textbf{rand}| + 1.$$

The change of weights are set to introduce degeneracy at the solution. This experiment suggests that degeneracy can be a common occurrence for location problems. Fairly accurate solutions are still obtained and quadratic convergence is frequently observed.

8.3. Comparison of CPU time. The proposed N-Weiszfeld algorithm clearly takes significantly less iterations than the original Weiszfeld algorithm. In order to illustrate that the

N-Weiszfeld is more efficient, we now illustrate that the additional cost of the N-Weiszfeld algorithm, compared to the Weiszfeld algorithm, is negligible as problem sizes increase. There are three significant computational tasks in each step of the N-Weiszfeld algorithm:

Task 1. Forming Hessian H^k:

$$H^k = \text{bdiag}\left(\frac{1}{\|r_i^k\|}\left(I_l - \frac{r_i^k {r_i^k}^T}{\|r_i^k\|^2}\right)\right);$$

Task 2. Cholesky factorization:

$$D_\theta^k {D_r^k}^{-1} + (1-\theta^k)H^k = G^k {G^k}^T;$$

Task 3. Solving a weighted least squares problem:

$${G^k}^T A^T d \overset{\text{LS}}{=} -y^k,$$

where

$$G^k y^k = g^k.$$

In FIG. 4-6, we chart the amount of CPU time required for each operation on a random multifacility location problem. FIG. 4-6 suggest that, as l and $nloc$ increase, the least squares solves dominate the total CPU time. The cost of forming the sparse block diagonal Hessian H^k in Matlab is significant. However, in a different computing environment, such as a parallel computer, this may change.

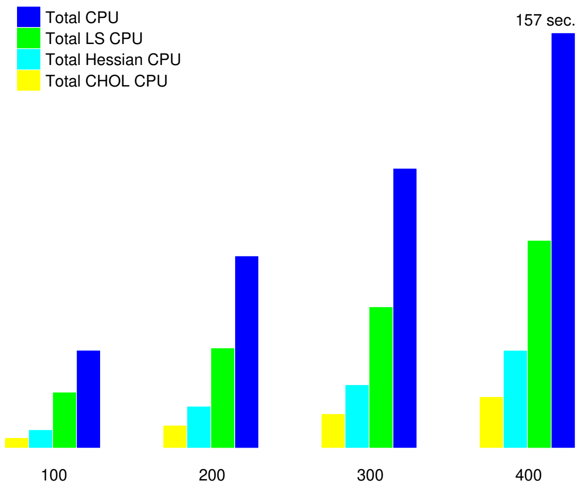

FIG. 4. *Computing Time ($l = 2$, $nloc = 30$ and $eloc = 100 : 100 : 400$)*

9. Conclusion. We have proposed a new Newton type algorithm (N-Weiszfeld) for minimizing the sum of Euclidean norms. This new algorithm has the similar attractive feature of the well-known Weiszfeld algorithm: the main computational cost per iteration is

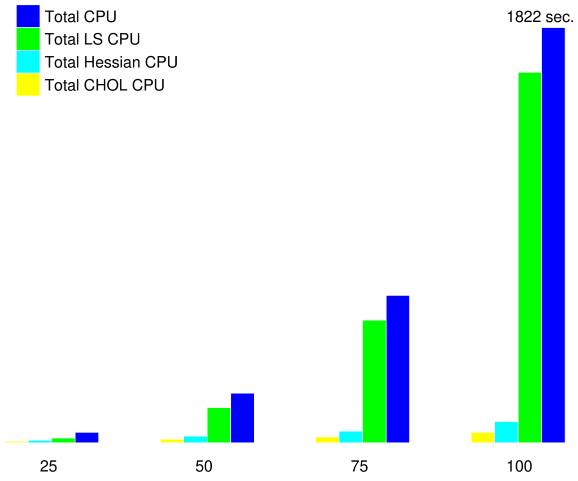

FIG. 5. *Computing Time (*$l = 2$*, eloc* $= 200$ *and nloc* $= 25 : 25 : 100$*)*

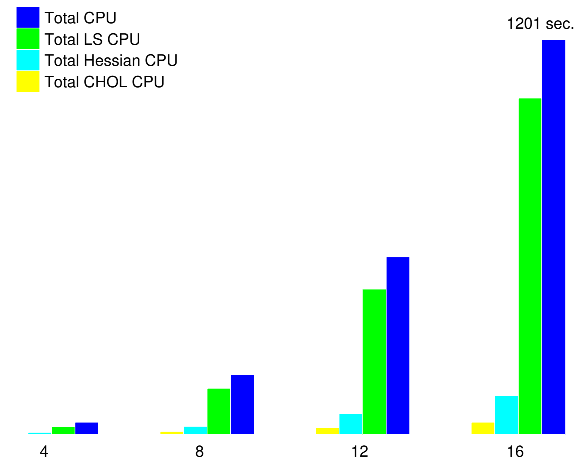

FIG. 6. *Computing Time (nloc* $= 20$*, eloc* $= 50$ *and* $l = 4 : 4 : 16$*)*

a weighted least squares solve. The fundamental difference between the Weiszfeld algorithm and the new N-Weiszfeld algorithm is that, while an improved iterate x^{k+1} is based on x^k alone for the Weiszfeld algorithm, each improvement x^{k+1} is based on both the current iterate x^k and approximate dual multipliers λ^k for N-Weiszfeld. Given a direction in the x component, the multipliers λ^{k+1} approximation can be obtained with negligible cost (a matrix-vector product).

The new N-Weiszfeld algorithm represents a transition from a Weiszfeld step to a Newton step for a system of nonlinear equations which captures the optimality conditions. We prove that every limit point of the sequence $\{x^k\}$ generated by N-Weiszfeld is a solution of the location problem (2) under the (primal) nondegeneracy assumption. The nonlinear system upon which the Newton process is derived is unusual because of its nondifferentiability. This complicates our ongoing local convergence analysis. Nonetheless, we believe that N-Weiszfeld is superlinearly convergent and our computational results certainly support this belief.

Our numerical experience clearly indicates that the proposed N-Weiszfeld algorithm is a significant improvement over the Weiszfeld algorithm in both efficiency and stability. The superiority of the N-Weiszfeld algorithm over the Weiszfeld algorithm is achieved with relative small cost (increasingly negligible for large problems): forming a Hessian matrix and a Cholesky factorization of a banded matrix with the bandwidth l.

REFERENCES

[1] K. D. ANDERSON, *An efficient newton barrier method for minimizing a sum of Euclidean norms*, SIAM Journal on Optimization, (to appear).

[2] P. H. CALAMAI AND A. R. CONN, *A stable algorithm for solving the multifacility location problem involving Euclidean distances*, SIAM Journal on Scientific and Statistical Computing, 1 (1980), pp. 512–526.

[3] P. H. CALAMAI AND A. R. CONN, *A second-order method for solving the multifacility location problems*, in Numerical Analysis: Proceedings of the Ninth Biennial Conference, Dundee, Scotland, G. Watson, ed., Springer-Verlag, Lecture Notes in Mathematics, 912, 1982.

[4] P. H. CALAMAI AND A. R. CONN, *A projected newton method for l_p norm location problems*, Mathematical Programming, 38 (1987), pp. 75–109.

[5] T. F. COLEMAN AND Y. LI, *A global and quadratically-convergent method for linear l_∞ problems*, SIAM Journal on Scientific and Statistical Computing, 29 (1992), pp. 1166–1186.

[6] ——, *A globally and quadratically convergent affine scaling method for linear l_1 problems*, Mathematical Programming, 56 (1992), pp. 189–222.

[7] T. F. COLEMAN AND Y. LI, *On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to boun ds*, Mathematical Programming, 67 (1994), pp. 189–224.

[8] A. R. CONN AND M. L. OVERTON, *A primal-dual interior point method for minimizing a sum of Euclidean distance*, preprint, 1994.

[9] H. W. KUHN, *A note on fermat's problem*, Mathematical Programming, 4 (1973), pp. 99–107.

[10] Y. LI, *A globally convergent method for l_p problems*, SIAM Journal on Optimization, 3 (1993), pp. 609–629.

[11] R. F. LOVE, J. G. MORRIS, AND G. O. WESOLOWSKY, *Facilities Location Models & Methods*, North-Holland, 1988.

[12] E. MIEHLE, *Link-length minimization in networks*, Operation Research, 6 (1958), pp. 232–243.

[13] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, 1970.

[14] M. L. OVERTON, *Minimizing a sum of Euclidean norms*, Mathematical Programming, (1983), pp. 34–63.

[15] G. W. STEWART, *On scaled projections and pseudoinverse*, Linear Algebra and its Applications, 112 (1989), pp. 189–193.

[16] THE MATHWORKS INC., *Matlab Reference guide*, The MathWorks, Natick, Mass, 1992.

[17] M. J. TODD, *Recent developments and new directions in linear programming*, in Mathematical Programming: Recent Developments and Application s, M. Iri and K. Tanabe, eds., Kluwer Academic Publishers, 1989, pp. 109–157.

[18] C. VAN LOAN, *On the method of weighting for equality-constrained least squares problems*, SIAM Journal on Numerical Analysis, 22 (1985), pp. 851–864.

[19] S. VAVASIS, *Stable numerical algorithm for equilibrium systems*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1108–1131.

[20] E. WEISZFELD, *Sur le point pour lequel la somme des distances de n points dennés est minimum*, Tôhoko Mathematics Journal, 43 (1937), pp. 355–386.