SIAM J. OPTIM.
Vol. 7, No. 4, pp. 1017–1036, November 1997

008

AN EFFICIENT ALGORITHM FOR MINIMIZING A SUM OF EUCLIDEAN NORMS WITH APPLICATIONS*

GUOLIANG XUE† AND YINYU YE‡

Abstract. In recent years rich theories on polynomial-time interior-point algorithms have been developed. These theories and algorithms can be applied to many nonlinear optimization problems to yield better complexity results for various applications. In this paper, the problem of minimizing a sum of Euclidean norms is studied. This problem is convex but not everywhere differentiable. By transforming the problem into a standard convex programming problem in conic form, we show that an ϵ-optimal solution can be computed efficiently using interior-point algorithms. As applications to this problem, polynomial-time algorithms are derived for the Euclidean single facility location problem, the Euclidean multifacility location problem, and the shortest network under a given tree topology. In particular, by solving the Newton equation in linear time using *Gaussian elimination on leaves of a tree*, we present an algorithm which computes an ϵ-optimal solution to the shortest network under a given full Steiner topology interconnecting N regular points, in $O(N\sqrt{N}(\log(\bar{c}/\epsilon) + \log N))$ arithmetic operations where $\bar{c}$ is the largest pairwise distance among the given points. The previous best-known result on this problem is a graphical algorithm which requires $O(N^2)$ arithmetic operations under certain conditions.

Key words. polynomial time, interior-point algorithm, minimizing a sum of Euclidean norms, Euclidean facilities location, shortest networks, Steiner minimum trees

AMS subject classifications. 68Q20, 68Q25, 90C25, 90C35

PII. S1052623495288362

1. Introduction.

The motivation to write this paper was to apply new techniques—polynomial time interior-point algorithms for convex programming—to solve two old problems: the Euclidean facilities location problem and the Steiner minimal tree (SMT) problem. The first problem, studied by researchers in location science, has applications in transportation and logistics. The second problem, studied by researchers in combinatorial optimization, has applications in communication networks. Both problems can be described as the minimization of a sum of Euclidean norms and they both trace back to an ancient problem studied by Fermat in the 17th century.

At the end of his celebrated essay on maxima and minima, in which he presented precalculus rules for finding tangents to a variety of curves, Fermat threw out this challenge: "Let he who does not approve of my method attempt the solution of the following problem: Given three points in the plane, find a fourth point such that the sum of its distances to the three given points is at minimum!" The solution to the original Fermat problem is either the Torricelli point—an interior point which opens an angle of 120° to each of the three sides of the triangle—or one of the given points whose inner angle is no less than 120°. This problem has been generalized into the Euclidean facilities location problem and the SMT problem.

The facilities location problem is one of locating N new facilities with respect

*Received by the editors June 28, 1995; accepted for publication April 30, 1996.
http://www.siam.org/journals/siopt/7-4/28836.html

†Department of Computer Science and Electrical Engineering, The University of Vermont, Burlington, VT 05405-0156 (xue@cs.uvm.edu). The research of this author was supported in part by the US Army Research Office grant DOD DAAH04-96-1-0233 and by the National Science Foundation grants NSF ASC-9409285 and NSF OSR-9350540.

‡Department of Management Sciences, The University of Iowa, Iowa City, IA 52242-1000 (yyye@dollar.biz.uiowa.edu). The research of this author was supported in part by NSF grant DDM-9207347 and the University of Iowa Business School Summer Grant.

to M existing facilities, the locations of which are known. The problem consists of finding locations of new facilities which will minimize a total cost function. This total cost function consists of a sum of costs directly proportional to the distances between the new facilities and costs directly proportional to the distances between new and existing facilities. If there is only one new facility ($N = 1$), the problem is called a Euclidean single facility location (ESFL) problem. If there is more than one new facility ($N \geq 2$), the problem is called a Euclidean multifacility location (EMFL) problem.

For the general ESFL problem, Weiszfeld [30] gave a simple closed form iterative algorithm in 1937. Later, it was proved by numerous authors [18, 24, 29] that the algorithm converges globally and, under certain conditions, linearly. Chandrasekaran and Tamir [5, 6] exhibited a solution to the strong separation problem associated with the ESFL problem which shows that an ϵ-*optimal solution* (i.e., a feasible solution whose absolute error in the objective function is within ϵ to the optimal objective function value) to the ESFL problem can be constructed in polynomial time using the ellipsoid method.

Miehle [21] was the first to propose an extension of the Weiszfeld algorithm for ESFL to solve EMFL problems. Ostresh [24] proved that Miehle's algorithm is a descending one. However, Miehle's algorithm may converge to a nonoptimal point; see [26, 31]. Eyster, White, and Wierwille [9] proposed a hyperboloid approximation procedure (HAP) for solving the perturbed EMFL problem. Rosen and Xue [31, 27] proved that the HAP always converges from any initial point. Calamai and Conn [3, 4] and Overton [25] proposed projected Newton algorithms for minimizing a sum of Euclidean norms and proved that the algorithms have quadratic rate of convergence provided the sequence of points generated by the algorithm converges to a strong minimizer. For more details, see the books by Francis, McGinnis, and White [11] and by Love, Morris, and Wesolowsky [19].

Recently, Xue, Rosen, and Pardalos [32] showed that the dual of the EMFL problem is the minimization of a linear function subject to linear and convex quadratic constraints and can therefore be solved by the interior-point techniques in polynomial time. den Hertog [8] (and see references therein) also presented a polynomial-time interior-point Newton barrier method for solving (2.1). More recently, Andersen [1] used the HAP idea [9] to smooth the objective function by introducing a perturbation $\epsilon > 0$ and applied a Newton barrier method to solving the problem. Andersen and Christiansen [2] and Conn and Overton [7] also proposed a primal–dual method based on the ϵ-perturbation and presented impressive computational results, although no complexity result is established for their method at this moment. None of the above formulations is in conic form.

The SMT problem [12, 20] is concerned with interconnecting a set of given points on the Euclidean plane with a shortest network. The shortest network is always a tree network and may contain some additional points called Steiner points. The SMT problem is NP-hard. Recently, there have been increased interests in the computation of a shortest interconnection network after the connections among the points (which is called a *topology*, to be defined in section 6.3) are specified. Hwang [14] proposed a linear-time algorithm for computing the shortest network under a full Steiner topology when the shortest network is a nondegenerate full Steiner tree. Hwang and Weng [15] proposed an $O(N^2)$ arithmetic operation algorithm for computing the shortest network under a Steiner topology when the shortest network is a tree whose vertex degrees are all less than or equal to 3. Smith [28] used an EMFL approach to compute

the shortest network under a given topology. His algorithm is essentially a first-order method.

In this paper, we first transform the basic problem of minimizing a sum of Euclidean norms into a standard convex programming problem in conic form and present an interior-point algorithm that can compute an ϵ-optimal solution in $O(\sqrt{m}(\log(\bar{c}/\epsilon)+\log m))$ iterations, where m is the number of norms in the summation and $\bar{c}$ is a constant that is not less than the Euclidean norm of any of the given vectors $c_i, i = 1, 2, \ldots, m$. We then study several applications of the basic problem and show improved computational complexity results wherever possible. In particular, we show that an ϵ-optimal solution to the shortest network under a given tree topology for a set of N points can be computed in $O(N\sqrt{N}(\log(\bar{c}/\epsilon)+\log N))$ arithmetic operations where $\bar{c}$ is the largest pairwise distance among the given points.

The rest of this paper is organized as follows. In section 2, we describe the basic problem of minimizing a sum of Euclidean norms. In section 3, the basic problem is transformed into a standard convex programming problem in conic form. In section 4, we present a primal–dual potential reduction algorithm for solving the problem. In section 5, we discuss the computational complexity and simplifications of the potential reduction algorithm. In section 6, we present applications to the ESFL problem, the EMFL problem, and the SMT problem. In section 7, we present some computational examples of SMT problems. We conclude this paper in section 8.

2. Minimizing a sum of Euclidean norms. Let $c_1, c_2, \ldots, c_m \in R^d$ be column vectors in the Euclidean d-space and $A_1, A_2, \ldots, A_m \in R^{n\times d}$ be n-by-d matrices with each having full column rank. We want to find a point $u \in R^n$ such that the following sum of Euclidean norms is minimized:

$$\begin{array}{ll} \min & \sum_{i=1}^m ||c_i - A_i^T u|| \\ \text{s.t.} & u \in R^n. \end{array} \tag{2.1}$$

It is clear that $u = 0$ is an optimal solution to (2.1) when all of the c_i are zero. Therefore, we will assume in the rest of this paper that not all of the c_i are zero. Problem (2.1) is a convex programming problem, but its objective function is not everywhere differentiable. Two special cases of this problem are the Euclidean facilities location problem and the SMT problem under a given topology.

We will call problem (2.1) the *basic problem* in the rest of our paper. This problem can be formulated as the maximization of a linear function subject to affine and convex cone constraints as follows:

$$\begin{array}{ll} \max & -\sum_{i=1}^m t_i \\ \text{s.t.} & t_1 \geq ||c_1 - A_1^T u||, \\ & t_2 \geq ||c_2 - A_2^T u||, \\ & \vdots \\ & t_m \geq ||c_m - A_m^T u||, \end{array} \tag{2.2}$$

where $t_i \in R, \quad i = 1, 2, \ldots, m$.

Problem (2.1) and problem (2.2) are equivalent in the following sense. If $(t_1; t_2; \cdots; t_m; u)$ is the optimal solution to (2.2), then u is the optimal solution to (2.1). If u is the optimal solution to (2.1), then $(t_1; t_2; \cdots; t_m; u)$ is the optimal solution to (2.2), where $t_i = ||c_i - A_i^T u||, \quad i = 1, 2, \ldots, m$ and $(t_1; t_2; \cdots; t_m; u)$ is an $(m+n)$-dimensional column vector whose first m elements are $t_i, \quad i = 1, 2, \ldots, m$ and whose last n elements are the elements of u.

In the rest of this paper, when we represent a large matrix with several small matrices, we will use semicolons ";" for column concatenation and commas "," for row concatenation. This notation also applies to vectors. We will use 0_n to represent an n-dimensional column vector whose elements are all zero. We will also use I_d to represent the d-by-d identity matrix.

3. Conic formulation. In this section, we will transform our basic problem (2.1) into a standard convex programming problem in conic form, where the cone and its associated barrier are *self-scaled* (or *homogeneous and self-dual*); see Nesterov and Nemirovskii [22], Nesterov and Todd [23], and Güler [13]. Because of the special constraints in problem (2.2), the cone of our choice is the second-order cone or the Lorentz cone. For definitions and theory about the second-order cone, self-scaled barriers, and related theory, see [22, 23, 13].

Let the cone be

$$K := \{(t; s) \in R^{d+1} : \ t \geq \|s\|\}.$$

Then its interior is

$$\text{int}K := \{(t; s) \in R^{d+1} : \ t > \|s\|\}.$$

Let

$$\delta(t; s) = \sqrt{t^2 - \|s\|^2},$$

and

$$f(t; s) = -\log \delta^2(t; s).$$

Then, for any $(t; s) \in \text{int}K$ we have

$$f'(t; s) = \frac{2}{\delta^2(t; s)} \begin{pmatrix} -t \\ s \end{pmatrix}$$

and

$$f''(t; s) = \frac{2}{\delta^2(t; s)} \begin{pmatrix} -1 & 0 \\ 0 & I_d \end{pmatrix} + \frac{4}{\delta^4(t; s)} \begin{pmatrix} t^2 & -ts^T \\ -ts & ss^T \end{pmatrix}, \tag{3.1}$$

which is positive definite. Its inverse is

$$(f''(t; s))^{-1} = \frac{\delta^2(t; s)}{2} \begin{pmatrix} -1 & 0 \\ 0 & I_d \end{pmatrix} + \begin{pmatrix} t^2 & ts^T \\ ts & ss^T \end{pmatrix}. \tag{3.2}$$

Also note that

$$(f''(t; s))^{-1} f'(t; s) = -(t; s). \tag{3.3}$$

Now let

$$\mathcal{B} = \begin{pmatrix} -1 \\ -1 \\ \vdots \\ -1 \\ 0_n \end{pmatrix} \in R^{m+n}, \quad \mathcal{C} = \begin{pmatrix} (0; c_1) \\ (0; c_2) \\ \vdots \\ (0; c_m) \end{pmatrix} \in R^{m+md},$$

and

$$\mathcal{A}^T = \begin{pmatrix} -1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & A_1^T \\ 0 & -1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & A_2^T \\ & & \ddots & & \\ 0 & 0 & \cdots & -1 & 0 \\ 0 & 0 & \cdots & 0 & A_m^T \end{pmatrix} \in R^{(m+md)\times(m+n)}.$$

Then, problem (2.1) or (2.2) can be written in the standard (dual) form

$$\begin{array}{ll} \max & \mathcal{B}^T(t_1;t_2;\cdots;t_m;u) \\ \text{s.t.} & \begin{pmatrix} (t_1;s_1) \\ (t_2;s_2) \\ \vdots \\ (t_m;s_m) \end{pmatrix} = \mathcal{C} - \mathcal{A}^T(t_1;t_2;\cdots;t_m;u), \\ & (t_i;s_i) \in K,\ i = 1,2,\ldots,m. \end{array} \tag{3.4}$$

Let $(\tau_1;x_1),(\tau_2;x_2),\ldots,(\tau_m;x_m) \in R^{d+1}$. Then its corresponding primal problem is

$$\begin{array}{ll} \min & \mathcal{C}^T((\tau_1;x_1);(\tau_2;x_2);\cdots;(\tau_m;x_m)) \\ \text{s.t.} & \mathcal{A}((\tau_1;x_1);(\tau_2;x_2);\cdots;(\tau_m;x_m)) = \mathcal{B}, \\ & (\tau_i;x_i) \in K,\ i = 1,2,\ldots,m. \end{array} \tag{3.5}$$

Thus, using $\mathcal{X} := ((\tau_1;x_1);(\tau_2;x_2);\cdots;(\tau_m;x_m))$, $\mathcal{S} := ((t_1;s_1);(t_2;s_2);\cdots;(t_m;s_m))$, $\mathcal{Y} := (t_1;t_2;\cdots;t_m;u)$, and $\mathcal{K} := K^m := K \times K \times \cdots \times K$, we can write the two problems (3.5) and (3.4) as

$$(P) \qquad \begin{array}{ll} \min & \mathcal{C}^T\mathcal{X} \\ \text{s.t.} & \mathcal{A}\mathcal{X} = \mathcal{B}, \\ & \mathcal{X} \in \mathcal{K} \end{array}$$

and

$$(D) \qquad \begin{array}{ll} \max & \mathcal{B}^T\mathcal{Y} \\ \text{s.t.} & \mathcal{S} = \mathcal{C} - \mathcal{A}^T\mathcal{Y}, \\ & \mathcal{S} \in \mathcal{K}. \end{array}$$

This is the pair of problems (P) and (D) in Nesterov and Nemirovskii [22] and Nesterov and Todd [23]. Since K is a convex self-dual and self-scaled cone with $\nu = 2$, $\mathcal{K}$ is a convex self-dual and self-scaled cone with $\nu = 2m$. Thus, we can use an interior-point algorithm to compute an ϵ-optimal solution of the problem in polynomial time.

Kojima [16] recently pointed out to us that problem (2.2) can be formulated as a positive semidefinite program:

$$\begin{array}{ll} \max & -\sum_{i=1}^m t_i \\ \text{s.t.} & \begin{pmatrix} t_i & (c_i - A_i^T u)^T \\ (c_i - A_i^T u) & t_i I_d \end{pmatrix} \text{ positive semidefinite for} \\ & i = 1,2,\ldots,m. \end{array} \tag{3.6}$$

However, as we will illustrate later, the complexity bound for solving positive semidefinite program (3.6) will be $\sqrt{d}$ factor higher than that for solving problem (3.4).

4. A primal–dual potential reduction algorithm. Let

$$F(\mathcal{X}) = \sum_{i=1}^{m} f(\tau_i; x_i) \quad \text{and} \quad F(\mathcal{S}) = \sum_{i=1}^{m} f(t_i; s_i). \tag{4.1}$$

A primal–dual potential function for the pair (P) and (D) is

$$\phi_\rho(\mathcal{X}, \mathcal{S}) := \rho \log(\langle \mathcal{X}, \mathcal{S} \rangle) + F(\mathcal{X}) + F(\mathcal{S}), \tag{4.2}$$

where $\rho = 2m + \gamma\sqrt{2m},\ \gamma \geq 1$. Note that

$$\langle \mathcal{X}, \mathcal{S} \rangle = \mathcal{X}^T \mathcal{S} = \mathcal{C}^T \mathcal{X} - \mathcal{B}^T \mathcal{Y}$$

and

$$\phi_{2m}(\mathcal{X}, \mathcal{S}) := 2m \log(\langle \mathcal{X}, \mathcal{S} \rangle) + F(\mathcal{X}) + F(\mathcal{S}) \geq 2m \log m. \tag{4.3}$$

The central trajectory for this pair is $\{\mathcal{X}(\mu), \mathcal{Y}(\mu), \mathcal{S}(\mu)\}$, for any given $\mu > 0$, such that $\mathcal{X} = \mathcal{X}(\mu)$ is primal feasible and $(\mathcal{Y}; \mathcal{S}) = (\mathcal{Y}(\mu); \mathcal{S}(\mu))$ is dual feasible, and

$$\begin{pmatrix} \tau_i \\ x_i \end{pmatrix} + \mu f'(t_i; s_i) = 0, \quad i = 1, 2, \ldots, m, \tag{4.4}$$

or

$$\begin{pmatrix} t_i \\ s_i \end{pmatrix} + \mu f'(\tau_i; x_i) = 0, \quad i = 1, 2, \ldots, m. \tag{4.5}$$

The main iteration of a potential reduction algorithm starts with a strictly feasible primal–dual pair $\mathcal{X}$ and $(\mathcal{Y}; \mathcal{S})$; i.e.,

$$\begin{aligned} &\mathcal{A}\mathcal{X} = \mathcal{B}, \qquad \mathcal{S} = \mathcal{C} - \mathcal{A}^T \mathcal{Y}, \\ &\mathcal{X} \in \text{int}\mathcal{K}, \ \text{ and } \ \mathcal{S} \in \text{int}\mathcal{K}. \end{aligned}$$

It computes a search direction $(d_\mathcal{X}, d_\mathcal{Y}, d_\mathcal{S})$ via solving a system of linear equations. After obtaining $(d_\mathcal{X}, d_\mathcal{Y}, d_\mathcal{S})$, a new strictly feasible primal–dual pair $\mathcal{X}^+$ and $(\mathcal{Y}^+; \mathcal{S}^+)$ is generated from

$$\mathcal{X}^+ = \mathcal{X} + \alpha d_\mathcal{X},\ \mathcal{Y}^+ = \mathcal{Y} + \beta d_\mathcal{Y},\ \mathcal{S}^+ = \mathcal{S} + \beta d_\mathcal{S},$$

for some step-sizes α and β, and

$$\phi_\rho(\mathcal{X}^+, \mathcal{S}^+) \leq \phi_\rho(\mathcal{X}, \mathcal{S}) - \Omega(1).$$

The search direction $(d_\mathcal{X}, d_\mathcal{Y}, d_\mathcal{S})$ is determined by the following equations.

$$\mathcal{A} d_\mathcal{X} = 0, \quad d_\mathcal{S} = -\mathcal{A}^T d_\mathcal{Y} \quad \text{(feasibility)} \tag{4.6}$$

and

$$d_\mathcal{X} + F''(\mathcal{S}) d_\mathcal{S} = -\frac{\rho}{\mathcal{X}^T \mathcal{S}} \mathcal{X} - F'(\mathcal{S}) \quad \text{(dual scaling)}, \tag{4.7}$$

or

$$d_\mathcal{S} + F''(\mathcal{X}) d_\mathcal{X} = -\frac{\rho}{\mathcal{X}^T \mathcal{S}} \mathcal{S} - F'(\mathcal{X}) \quad \text{(primal scaling)}, \tag{4.8}$$

or

$$d_{\mathcal{S}} + F''(\mathcal{Z})d_{\mathcal{X}} = -\frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{S} - F'(\mathcal{X}) \quad \text{(joint scaling)}, \tag{4.9}$$

where $\mathcal{Z}$ is chosen to satisfy

$$\mathcal{S} = F''(\mathcal{Z})\mathcal{X}. \tag{4.10}$$

(These directions were presented for linear and quadratic programming in Ye [33].) We will discuss each of these three cases in the next three subsections. The differences among the three algorithms are the computation of the search direction and their theoretical close-form step-sizes. All three generate an ϵ-optimal solution $(\mathcal{X}, \mathcal{Y}, \mathcal{S})$; i.e.,

$$\langle \mathcal{X}, \mathcal{S} \rangle \leq \epsilon$$

in a guaranteed $O(\gamma\sqrt{2m}\log(\langle \mathcal{X}^0, \mathcal{S}^0\rangle/\epsilon) + \phi_{2m}(\mathcal{X}^0, \mathcal{S}^0) - 2m\log m)$ iteration. (Note from (4.3) that $\phi_{2m}(\mathcal{X}^0, \mathcal{S}^0) - 2m\log m) \geq 0$.)

In practice, one usually finds the largest step-sizes $\bar{\alpha}$ and $\bar{\beta}$ such that

$$\mathcal{X} + \bar{\alpha}d_{\mathcal{X}} \in \mathcal{K}, \quad \text{and} \quad \mathcal{S} + \bar{\beta}d_{\mathcal{S}} \in \mathcal{K} \tag{4.11}$$

then takes $\alpha \in [0, \bar{\alpha}]$ and $\beta \in [0, \bar{\beta}]$, via a line search, to minimize $\phi_\rho(\mathcal{X}^+, \mathcal{S}^+)$, or simply chooses

$$\alpha = (0.5 \sim 0.99)\bar{\alpha} \quad \text{and} \quad \beta = (0.5 \sim 0.99)\bar{\beta} \tag{4.12}$$

as long as ϕ_ρ is reduced.

4.1. Dual scaling. The theoretical potential reduction algorithm using dual scaling can be described as follows.

ALGORITHM PDD.

{γ and Δ are fixed constants such that $\gamma \geq 1$, $0 < \Delta < 1$, and $\frac{\gamma(\gamma(1-\Delta)-\Delta)}{1+\gamma} > \frac{\Delta^2}{2(1-\Delta)^2}$}.

Step_1 Compute the search direction $(d_{\mathcal{X}}, d_{\mathcal{Y}}, d_{\mathcal{S}})$ using (4.6) and (4.7).

Step_2 Compute $\lambda = \sqrt{d_{\mathcal{S}}^T F''(\mathcal{S}) d_{\mathcal{S}}}$.

If $\lambda > \Delta$ then
  $\mathcal{X}^+ = \mathcal{X}$, (primal step-size $\alpha = 0$)
  $\mathcal{S}^+ = \mathcal{S} + \frac{1}{1+\lambda}d_{\mathcal{S}}$, (dual step-size $\beta = \frac{1}{1+\lambda}$)
else
  $\mathcal{X}^+ = \mathcal{X} + \frac{\langle \mathcal{S}, \mathcal{X}\rangle}{\rho}d_{\mathcal{X}}$, (primal step-size $\alpha = \frac{\langle \mathcal{S}, \mathcal{X}\rangle}{\rho}$)
  $\mathcal{S}^+ = \mathcal{S}$. (dual step-size $\beta = 0$)
endif

According to Nesterov and Nemirovskii [22], we have the following theorem.

THEOREM 4.1. *Starting from any strictly feasible primal solution $\mathcal{X}^0$ and strictly dual feasible solution $(\mathcal{Y}^0; \mathcal{S}^0)$, an ϵ-optimal solution to problem (2.2) can be obtained by repeated application of Algorithm PDD for at most $O(\gamma\sqrt{2m}\log(\langle \mathcal{X}^0, \mathcal{S}^0\rangle/\epsilon) + \phi_{2m}(\mathcal{X}^0, \mathcal{S}^0) - 2m\log m)$ iterations.* □

At first glance, it seems that the dimension of the system of linear equations defined by (4.6) and (4.7) is very large. However, the system is structured and its solution can be simplified.

Consider the dual-scaling form (4.7). Using $d_{\mathcal{S}} = -\mathcal{A}^T d_{\mathcal{Y}}$, we have

$$d_{\mathcal{X}} - F''(\mathcal{S})\mathcal{A}^T d_{\mathcal{Y}} = -\frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{X} - F'(\mathcal{S}).$$

Multiplying $\mathcal{A}$ on both sides and noting that $\mathcal{A}d_{\mathcal{X}} = 0$, we have

$$\mathcal{A}F''(\mathcal{S})\mathcal{A}^T d_{\mathcal{Y}} = \frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{A}\mathcal{X} + \mathcal{A}F'(\mathcal{S}),$$

or

$$\mathcal{A}F''(\mathcal{S})\mathcal{A}^T d_{\mathcal{Y}} = \frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{B} + \mathcal{A}F'(\mathcal{S}),$$

which is a least-squares problem where $\mathcal{A}$ is scaled to $\mathcal{A}(F''(\mathcal{S}))^{1/2}$.

Therefore, the search direction $d_{\mathcal{X}}, d_{\mathcal{Y}}, d_{\mathcal{S}}$ determined by dual scaling can be computed by solving the following system of linear equations:

$$\begin{aligned} \mathcal{A}F''(\mathcal{S})\mathcal{A}^T d_{\mathcal{Y}} &= \tfrac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{B} + \mathcal{A}F'(\mathcal{S}), \\ d_{\mathcal{X}} &= F''(\mathcal{S})\mathcal{A}^T d_{\mathcal{Y}} - \tfrac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{X} - F'(\mathcal{S}), \\ d_{\mathcal{S}} &= -\mathcal{A}^T d_{\mathcal{Y}}. \end{aligned} \tag{4.13}$$

4.2. Primal scaling. The theoretical potential reduction algorithm using primal-scaling can be described as follows.

ALGORITHM PDP.

{γ and Δ are fixed constants such that $\gamma \geq 1$, $0 < \Delta < 1$, and $\frac{\gamma(\gamma(1-\Delta)-\Delta)}{1+\gamma} > \frac{\Delta^2}{2(1-\Delta)^2}$}.

Step_1 Compute the search direction $(d_{\mathcal{X}}, d_{\mathcal{Y}}, d_{\mathcal{S}})$ using (4.6) and (4.8).

Step_2 Compute $\lambda = \sqrt{d_{\mathcal{X}}^T F''(\mathcal{X}) d_{\mathcal{X}}}$.

If $\lambda > \Delta$ then

$\mathcal{X}^+ = \mathcal{X} + \frac{1}{1+\lambda} d_{\mathcal{X}}$, (primal step-size $\alpha = \frac{1}{1+\lambda}$)

$\mathcal{S}^+ = \mathcal{S}$. (dual step-size $\beta = 0$)

else

$\mathcal{X}^+ = \mathcal{X}$, (primal step-size $\alpha = 0$)

$\mathcal{S}^+ = \mathcal{S} + \frac{\langle \mathcal{S}, \mathcal{X} \rangle}{\rho} d_{\mathcal{S}}$, (dual step-size $\beta = \frac{\langle \mathcal{S}, \mathcal{X} \rangle}{\rho}$)

endif

According to Nesterov and Nemirovskii [22], we have the following theorem.

THEOREM 4.2. *Starting from any strictly feasible primal solution $\mathcal{X}^0$ and strictly dual feasible solution $(\mathcal{Y}^0; \mathcal{S}^0)$, an ϵ-optimal solution to problem (2.2) can be obtained by repeated application of Algorithm PDP for at most $O(\gamma\sqrt{2m}\log(\langle \mathcal{X}^0, \mathcal{S}^0 \rangle/\epsilon) + \phi_{2m}(\mathcal{X}^0, \mathcal{S}^0) - 2m \log m)$ iterations.* □

As in the dual-scaling case, we can also simplify the system of linear equations defined by (4.6) and (4.8) as follows.

Consider the primal form. Using $d_{\mathcal{S}} = -\mathcal{A}^T d_{\mathcal{Y}}$, we have

$$-\mathcal{A}^T d_{\mathcal{Y}} + F''(\mathcal{X}) d_{\mathcal{X}} = -\frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{S} - F'(\mathcal{X})$$

or

$$\begin{aligned} d_{\mathcal{X}} - (F''(\mathcal{X}))^{-1}\mathcal{A}^T d_{\mathcal{Y}} &= -\frac{\rho}{\mathcal{X}^T\mathcal{S}}(F''(\mathcal{X}))^{-1}\mathcal{S} - (F''(\mathcal{X}))^{-1}F'(\mathcal{X}) \\ &= -\frac{\rho}{\mathcal{X}^T\mathcal{S}}(F''(\mathcal{X}))^{-1}\mathcal{S} + \mathcal{X}. \end{aligned}$$

Here we have used relation (3.3) implying

$$(F''(\mathcal{X}))^{-1}F'(\mathcal{X}) = -\mathcal{X}.$$

Also note that there is a close form for $(F''(\mathcal{X}))^{-1}$ given by (3.2). Multiplying $\mathcal{A}$ on both sides and noting $\mathcal{A}d_{\mathcal{X}} = 0$, we have

$$\mathcal{A}(F''(\mathcal{X}))^{-1}\mathcal{A}^T d_{\mathcal{Y}} = \frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{A}(F''(\mathcal{X}))^{-1}\mathcal{S} - \mathcal{A}\mathcal{X},$$

or

$$\mathcal{A}(F''(\mathcal{X}))^{-1}\mathcal{A}^T d_{\mathcal{Y}} = \frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{A}(F''(\mathcal{X}))^{-1}\mathcal{S} - \mathcal{B},$$

which again is a least-squares problem where $\mathcal{A}$ is scaled to $\mathcal{A}(F''(\mathcal{X}))^{-1/2}$.

Therefore, the search direction $d_{\mathcal{X}}, d_{\mathcal{Y}}, d_{\mathcal{S}}$ determined by primal scaling can be computed by solving the following system of linear equations:

$$\begin{aligned} \mathcal{A}(F''(\mathcal{X}))^{-1}\mathcal{A}^T d_{\mathcal{Y}} &= \tfrac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{A}(F''(\mathcal{X}))^{-1}\mathcal{S} - \mathcal{B}, \\ d_{\mathcal{X}} &= (F''(\mathcal{X}))^{-1}\mathcal{A}^T d_{\mathcal{Y}} - \tfrac{\rho}{\mathcal{X}^T\mathcal{S}}(F''(\mathcal{X}))^{-1}\mathcal{S} + \mathcal{X}, \\ d_{\mathcal{S}} &= -\mathcal{A}^T d_{\mathcal{Y}}. \end{aligned} \tag{4.14}$$

4.3. Joint scaling. The theoretical potential-reduction algorithm using primal–dual joint scaling generates the search direction from

$$d_{\mathcal{S}} + F''(\mathcal{Z})d_{\mathcal{X}} = -\frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{S} - F'(\mathcal{X}),$$

where $\mathcal{Z}$ is chosen to satisfy

$$\mathcal{S} = F''(\mathcal{Z})\mathcal{X}.$$

According to Nesterov and Todd [23], there is a unique $\mathcal{Z} := ((\kappa_1; z_1); \ldots; (\kappa_m; z_m))$ such that

$$(t_i; s_i) = f''(\kappa_i; z_i)(\tau_i; x_i), \quad i = 1, \ldots, m.$$

In fact, for any $(\tau; x) \in \text{int}K$ and $(t; s) \in \text{int}K$ we have a unique $(\kappa; z) \in \text{int}K$ with

$$(t; s) = f''(\kappa; z)(\tau; x),$$

where

$$\kappa = \zeta\tau + \eta t \quad \text{and} \quad z = \zeta x - \eta s,$$

where

$$\zeta = \frac{1}{\sqrt{\delta(\tau; x)\delta(t; s) + \tau t + x^T s}} \quad \text{and} \quad \eta = \zeta\frac{\delta(\tau; x)}{\delta(t; s)}.$$

One can verify that

$$\delta^2(\kappa; z) = \frac{2\delta(\tau; x)}{\delta(t; s)},$$

so that from (3.1) and (3.2)

$$f''(\kappa; z) = \frac{\delta(t;s)}{\delta(\tau;x)} \begin{pmatrix} -1 & 0 \\ 0 & I_d \end{pmatrix} + \frac{\delta^2(t;s)}{\delta^2(\tau;x)} \begin{pmatrix} \kappa^2 & -\kappa z^T \\ -\kappa z & zz^T \end{pmatrix},$$

and

$$(f''(\kappa; z))^{-1} = \frac{\delta(\tau;x)}{\delta(t;s)} \begin{pmatrix} -1 & 0 \\ 0 & I_d \end{pmatrix} + \begin{pmatrix} \kappa^2 & \kappa z^T \\ \kappa z & zz^T \end{pmatrix}.$$

The joint-scaling algorithm can be described as follows.

ALGORITHM PDJ.

Step_1 Compute the scaling point $\mathcal{Z} := ((\kappa_1; z_1); (\kappa_2; z_2); \ldots; (\kappa_m; z_m))$ from
$$\kappa = \zeta\tau + \eta t \quad \text{and} \quad z_i = \zeta_i x_i - \eta_i s_i, \quad i = 1, 2, \ldots, m$$
where
$$\zeta_i = \frac{1}{\sqrt{\delta(\tau_i;x_i)\delta(t_i;s_i) + \tau_i t_i + x_i^T s_i}} \quad \text{and} \quad \eta_i = \zeta_i \frac{\delta_i(\tau_i;x_i)}{\delta(t_i;s_i)}, \quad i = 1, 2, \ldots, m.$$

Step_2 Compute the search direction $(d_\mathcal{X}, d_\mathcal{Y}, d_\mathcal{S})$ using (4.6), (4.9), and (4.10).

Step_3 Let $\sigma(\mathcal{Z})$ be the largest primal feasible step-size form $\mathcal{X}$ along direction $\mathcal{Z}$.
Let $\sigma(d_\mathcal{X})$ be the largest primal feasible step-size form $\mathcal{X}$ along direction $d_\mathcal{X}$.
Let $\sigma(d_\mathcal{S})$ be the largest dual feasible step-size form $\mathcal{S}$ along direction $d_\mathcal{S}$.
Choose the joint step-size $\bar{\alpha}$ by
$$\bar{\alpha} = \min\{\tfrac{1}{\sigma(\mathcal{Z})^2 + \sigma(d_\mathcal{X})}, \quad \tfrac{1}{\sigma(\mathcal{Z})^2 + \sigma(d_\mathcal{S})}\}.$$

Step_4 Update the approximate solution by
$$\mathcal{X}^+ = \mathcal{X} + \bar{\alpha} d_\mathcal{X}, \quad \mathcal{S}^+ = \mathcal{S} + \bar{\alpha} d_\mathcal{S}, \quad \mathcal{Y}^+ = \mathcal{Y} + \bar{\alpha} d_\mathcal{Y}.$$

According to Nesterov and Todd [23], we have the following theorem.

THEOREM 4.3. *Starting from any strictly feasible primal solution $\mathcal{X}^0$ and strictly dual feasible solution $(\mathcal{Y}^0; \mathcal{S}^0)$, an ϵ-optimal solution to problem (2.2) can be obtained by repeated application of Algorithm PDJ for at most $O(\gamma\sqrt{2m}\log(\langle\mathcal{X}^0, \mathcal{S}^0\rangle/\epsilon) + \phi_{2m}(\mathcal{X}^0, \mathcal{S}^0) - 2m\log m)$ iterations.* □

As in the cases of dual scaling and primal scaling, we can simplify the system of linear equations defined by (4.6) and (4.9) as follows.

Using $d_\mathcal{S} = -\mathcal{A}^T d_\mathcal{Y}$, we have

$$-\mathcal{A}^T d_\mathcal{Y} + F''(\mathcal{Z}) d_\mathcal{X} = -\frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{S} - F'(\mathcal{X})$$

or

$$\begin{aligned} d_\mathcal{X} - (F''(\mathcal{Z}))^{-1}\mathcal{A}^T d_\mathcal{Y} &= -\frac{\rho}{\mathcal{X}^T\mathcal{S}}(F''(\mathcal{Z}))^{-1}\mathcal{S} - (F''(\mathcal{Z}))^{-1}F'(\mathcal{X}) \\ &= -\frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{X} - F'(\mathcal{S}). \end{aligned}$$

Here we have used relations

$$(F''(\mathcal{Z}))^{-1}\mathcal{S} = \mathcal{X}$$

and

$$(F''(\mathcal{Z}))^{-1}F'(\mathcal{X}) = F'(\mathcal{S}).$$

Multiplying $\mathcal{A}$ on both sides and noting $\mathcal{A}d_\mathcal{X} = 0$ and $\mathcal{A}\mathcal{X} = \mathcal{B}$, we have

$$\mathcal{A}(F''(\mathcal{Z}))^{-1}\mathcal{A}^T d_\mathcal{Y} = \frac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{B} + \mathcal{A}F'(\mathcal{S}),$$

which again is a least-squares problem where $\mathcal{A}$ is scaled to $\mathcal{A}(F''(\mathcal{Z}))^{-1/2}$.

Therefore, the search direction $d_{\mathcal{X}}, d_{\mathcal{Y}}, d_{\mathcal{S}}$ determined by joint scaling can be computed by solving the following system of linear equations:

$$\begin{aligned} \mathcal{A}(F''(\mathcal{Z}))^{-1}\mathcal{A}^T d_{\mathcal{Y}} &= \tfrac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{B} + \mathcal{A}F'(\mathcal{S}), \\ d_{\mathcal{X}} &= (F''(\mathcal{Z}))^{-1}\mathcal{A}^T d_{\mathcal{Y}} - \tfrac{\rho}{\mathcal{X}^T\mathcal{S}}\mathcal{X} - F'(\mathcal{S}), \\ d_{\mathcal{S}} &= -\mathcal{A}^T d_{\mathcal{Y}}. \end{aligned} \tag{4.15}$$

5. Complexity and implementation. As we have seen, the number of iterations required (as stated in Theorems 4.1–4.3) to compute an ϵ-optimal solution to problem (2.2) depends on the initial point $(\mathcal{X}^0, \mathcal{S}^0, \mathcal{Y}^0)$. In this section, we discuss initial point selection and other computational issues for solving problem (2.2) using the algorithms presented in section 3.

5.1. Initial point. The algorithms discussed in the previous section all require a pair of strictly primal–dual interior feasible solutions. In the following, we give one such pair.

Let

$$\bar{c} = \max_{1 \le i \le m} \|c_i\|,$$

and

$$u^0 = 0, \quad s_i^0 = c_i, \quad t_i^0 = \sqrt{\|c_i\|^2 + m\bar{c}^2}, \quad i = 1, 2, \ldots, m,$$

and

$$\tau_i^0 = 1, \quad x_i^0 = 0, \quad i = 1, 2, \ldots, m.$$

Then, one can verify that $\mathcal{X}$ is an interior feasible solution to (P) and $\mathcal{S}$ and $\mathcal{Y}$ form an interior feasible solution to (D). One can also verify that

$$\langle \mathcal{X}^0, \mathcal{S}^0 \rangle = (\mathcal{X}^0)^T \mathcal{S}^0 = \sum_{i=1}^m t_i^0 \tau_i^0 = \sum_{i=1}^m \sqrt{\|c_i\|^2 + m\bar{c}^2} \le \bar{c} m \sqrt{1+m}$$

and the initial value

$$\begin{aligned} \phi_{2m}(\mathcal{X}^0, \mathcal{S}^0) - 2m \log m &= 2m \log(\langle \mathcal{X}^0, \mathcal{S}^0 \rangle) + F(\mathcal{X}^0) + F(\mathcal{S}^0) - 2m \log m \\ &= 2m \log(\langle \mathcal{X}^0, \mathcal{S}^0 \rangle) + F(\mathcal{S}^0) - 2m \log m \\ &= 2m \log(\langle \mathcal{X}^0, \mathcal{S}^0 \rangle) - m \log(m\bar{c}^2) - 2m \log m \\ &\le 2m \log(m\sqrt{1+m}\bar{c}) - m \log(m\bar{c}^2) - 2m \log m \\ &= m \log(1+m) - m \log m \\ &= m \log(1 + 1/m) \\ &\le 1. \end{aligned}$$

With this initial point, we have the following corollary.

COROLLARY 5.1. *Let the initial feasible primal solution $\mathcal{X}^0$ and dual feasible solution $(\mathcal{Y}^0; \mathcal{S}^0)$ be given as above. Then, an ϵ-optimal solution to problem* (2.2) *can*

be obtained by the potential reduction algorithms in at most $O(\gamma\sqrt{m}(\log(\bar{c}/\epsilon)+\log m))$ *iterations, where*

$$\bar{c} = \max_{1\le i\le m} \|c_i\|. \qquad \square$$

Note that if positive semidefinite program (3.6) is solved, the iteration complexity bound will be $O(\gamma\sqrt{md}(\log(\bar{c}/\epsilon)+\log md))$, which is $\sqrt{d}$ higher than the bound given by the above corollary.

5.2. Search direction. At each step of the potential-reduction algorithm, we need to compute the search direction $d_{\mathcal{X}}$, $d_{\mathcal{S}}$, and $d_{\mathcal{Y}}$ by solving a system of linear equations. In what follows, we will show that this can be further simplified, taking advantage of the special structure of the problem.

Consider the search direction defined by dual scaling (4.7). For $i = 1, \dots, m$, it can be decomposed as

$$\begin{pmatrix} d_{\tau_i} \\ d_{x_i} \end{pmatrix} + \left(\frac{2}{\delta^2(t_i;s_i)} \begin{pmatrix} -1 & 0 \\ 0 & I_d \end{pmatrix} + \frac{4}{\delta^4(t_i;s_i)} \begin{pmatrix} (t_i)^2 & -t_i(s_i)^T \\ -t_i s_i & s_i(s_i)^T \end{pmatrix} \right) \begin{pmatrix} d_{t_i} \\ d_{s_i} \end{pmatrix}$$

$$= -\frac{\rho}{\mathcal{X}^T\mathcal{S}} \begin{pmatrix} \tau_i \\ x_i \end{pmatrix} - \frac{2}{\delta^2(t_i;s_i)} \begin{pmatrix} -t_i \\ s_i \end{pmatrix}. \tag{5.1}$$

Note that $s_i = c_i - A_i^T u$, $d_{s_i} = -A_i^T d_u$, $\tau_i = 1$, and $d_{\tau_i} = 0$ for $i = 1, \dots, m$. The system can be written as

$$\left(-\frac{2}{\delta^2(t_i;s_i)} + \frac{4(t_i)^2}{\delta^4(t_i;s_i)} \right) d_{t_i} + \frac{4t_i}{\delta^4(t_i;s_i)} (s_i)^T A_i^T d_u = -\frac{\rho}{\mathcal{X}^T\mathcal{S}} + \frac{2}{\delta^2(t_i;s_i)} t_i,$$

$$d_{x_i} - \frac{2}{\delta^2(t_i;s_i)} A_i^T d_u + \frac{4}{\delta^4(t_i;s_i)} (-t_i d_{t_i} s_i - s_i(s_i)^T A_i^T d_u) = -\frac{\rho}{\mathcal{X}^T\mathcal{S}} x_i - \frac{2}{\delta^2(t_i;s_i)} s_i.$$

From the first equation we have

$$d_{t_i} = \frac{\delta^2(t_i;s_i)t_i - \frac{\rho\delta^4(t_i;s_i)}{2\mathcal{X}^T\mathcal{S}} - 2t_i(s_i)^T A_i^T d_u}{2(t_i)^2 - \delta^2(t_i;s_i)}.$$

Substituting this relation into the second equation, we have

$$d_{x_i} + \frac{2}{\delta^2(t_i;s_i)} \left(\frac{2}{2(t_i)^2 - \delta^2(t_i;s_i)} s_i(s_i)^T - I_d \right) A_i^T d_u$$

$$= -\frac{\rho}{\mathcal{X}^T\mathcal{S}} x_i + \left(\frac{2(1 - \frac{\rho}{\mathcal{X}^T\mathcal{S}} t_i)}{2(t_i)^2 - \delta^2(t_i;s_i)} \right) s_i.$$

Moreover, since

$$\sum_{i=1}^m A_i x_i = 0, \qquad \sum_{i=1}^m A_i d_{x_i} = 0,$$

we have

$$\left(\sum_{i=1}^{m}\frac{2}{\delta^2(t_i;s_i)}\left(\frac{2}{2(t_i)^2-\delta^2(t_i;s_i)}A_i s_i (s_i)^T A_i^T - A_i A_i^T\right)\right) d_u$$

$$= \sum_{i=1}^{m}\left(\frac{2(1-\frac{\rho}{\mathcal{X}^T\mathcal{S}}t_i)}{2(t_i)^2-\delta^2(t_i;s_i)}\right) A_i s_i. \tag{5.2}$$

Note that the system for computing d_u may not have full rank. If that is the case, any feasible solution is acceptable.

It requires $O(mn^2d)$ operations to set up the system (5.2) for computing d_u. Solving the system requires $O(n^3)$ operations. Once d_u is computed, $O(mnd)$ operations are required to compute d_x and d_s. Therefore, the number of arithmetic operations in each iteration is bounded by $O(n^3 + mn^2d)$. The following theorem follows from Corollary 5.1 and the above analysis.

THEOREM 5.2. *Let the initial feasible primal solution $\mathcal{X}^0$ and dual feasible solution $(\mathcal{Y}^0;\mathcal{S}^0)$ be given as above. Then, an ϵ-optimal solution to problem* (2.1) *can be obtained by the potential reduction algorithms in at most $O(\gamma\sqrt{m}(\log(\bar{c}/\epsilon)+\log m))$ iterations, where*

$$\bar{c} = \max_{1\le i\le m}\|c_i\|,$$

and each iteration requires $O(n^3 + mn^2d)$ arithmetic operations. □

Note that if γ is chosen as a constant and the problem is normalized such that $\bar{c} = 1$, i.e., all of c_i is within the unit ball in R^d, then the iteration bound is $O(\sqrt{m}(\log(1/\epsilon)+\log m))$. We will further discuss this issue in following applications.

6. Applications. In this section, we will apply the algorithms presented in the previous sections to solve the ESFL problem, the EMFL problem, and the SMT problem under a given topology. We will also take advantage of the special structures of these special problems and obtain improved computational complexity results wherever possible.

6.1. The ESFL problem. Let $a_1, a_2, \ldots, a_M$ be M points in R^d, the d-dimensional Euclidean space. Let w_1, w_2, ..., w_M be M positive weights. Find a point $x \in R^d$ that will minimize

$$f(x) = \sum_{i=1}^{M} w_i||x - a_i||. \tag{6.1}$$

This is called the ESFL problem.

In the ESFL problem, $a_1, a_2, \ldots, a_M$ represent the respective locations of M clients in a given region and x represents the location of a prospective service center. $w_1, w_2, \ldots, w_M$ represent the respective amount of service requests of the clients to the service center. The ESFL problem is concerned with finding the location for the service center to minimize the sum of weighted Euclidean distances from the service center to each of the clients. For more information on this problem, see [17, 19].

The ESFL problem can be easily transformed into a special case of problem (2.1) where $m = M$, $n = d$ and $c_i = w_i a_i$, $A_i^T = w_i I_d$, $i = 1, 2, \ldots, M$. It follows from Theorem 5.1 that Theorem 6.1 holds.

THEOREM 6.1. *An ϵ-optimal solution to the ESFL problem* (6.1) *can be computed using any of our potential reduction algorithms in $O(\sqrt{M}(\log(\bar{c}/\epsilon)+\log M))$ iterations*

where $\bar{c} = \max_{1\le i\le m} \|w_i a_i\|$, *and each iteration requires* $O(d^3 + d^2M)$ *arithmetic operations.* □

6.2. The EMFL problem. Let $a_1, a_2, \ldots, a_M$ be M points in R^d, the d-dimensional Euclidean space. Let w_{ji}, $j = 1, 2, \ldots, N$, $i = 1, 2, \ldots, M$, and $v_{jk}, 1 \le j < k \le N$ be given nonnegative numbers. Find a point $x = (x_1; x_2; \ldots; x_N) \in R^{dN}$ that will minimize

$$f(x) = \sum_{j=1}^{N}\sum_{i=1}^{M} w_{ji}||x_j - a_i|| + \sum_{1\le j<k\le N} v_{jk}||x_j - x_k||. \tag{6.2}$$

This is the so-called EMFL problem. For ease of notation, we assume that $v_{jj} = 0$ for $j = 1, 2, \ldots, N$ and that $v_{jk} = v_{kj}$ for $1 \le k < j \le N$.

In the EMFL problem, $a_1, a_2, \ldots, a_M$ represent the locations of M existing facilities; $x_1, x_2, \ldots, x_N$ represent the locations of N new facilities; the objective function $f(x)$ is the sum of weighted Euclidean distances from each new facility to each existing facility and those between each pair of new facilities; and our goal is to find optimal locations for the new facilities, i.e., to minimize $f(x)$.

In problem (6.2), some of the weights w_{ji} and v_{jk} may be zero. Let m be the number of nonzero weights in (6.2). Then the EMFL problem (6.2) is the minimization of m Euclidean norms. Without loss of generality, we assume that for each $j \in \{1, 2, \ldots, N\}$ there exists a nonzero w_{ji} for some $i \in \{1, 2, \ldots, M\}$ or a nonzero v_{jk} for some $k \in \{1, 2, \ldots, N\}$.

To transform the EMFL problem (6.2) into an instance of problem (2.1), we simply do the following. Let $u = (x_1; x_2; \ldots; x_N)$. It is clear that $u \in R^n$ where $n = dN$. For each nonzero w_{ji}, there is a corresponding term of Euclidean norm $||c(w_{ji}) - A(w_{ji})^T u||$ where $c(w_{ji}) = w_{ji}a_i$, and $A(w_{ji})^T$ is a row of N blocks of d-by-d matrices whose jth block is $w_{ji}I_d$ and whose other blocks are all zero. For each nonzero v_{jk}, there is a corresponding term of Euclidean norm $||c(v_{jk}) - A(v_{jk})^T u||$ where $c(v_{jk}) = 0$, and $A(v_{jk})^T$ is a row of N blocks of d-by-d matrices whose jth and kth blocks are $-v_{jk}I_d$ and $v_{jk}I_d$, respectively, and whose other blocks are all zero.

Now it is clear that we have transformed the EMFL problem (6.2) into an instance of (2.1) where $n = dN$, and m is the number of nonzero weights w_{ji} and v_{jk}. Note that the system (5.2) can be set up with $O(md^2)$ operations. Therefore, it follows from Theorem 5.1 that we have the following theorem.

THEOREM 6.2. *An* ϵ*-optimal solution to the EMFL problem* (6.2) *can be computed using any of our algorithms in* $O(\sqrt{MN}(\log(\bar{c}/\epsilon) + \log(MN)))$ *iterations where* $\bar{c} = \max_{1\le j\le n \;\; 1\le i\le m} \|w_{ji}a_i\|$, *and each iteration requires* $O(d^3N^3 + MNd^2)$ *arithmetic operations.* □

6.3. The SMT problem. The *Euclidean SMT problem* is given by a set of points $P = \{p_1, p_2, \ldots, p_N\}$ in the Euclidean plane and asks for the shortest planar straight-line graph spanning P. The solution takes the form of a tree, called the *SMT*, that includes all the given points, called *regular points*, along with some extra vertices, called *Steiner points*. It is known that there are at most $N - 2$ Steiner points and the degree of each Steiner point is at most 3. See [12, 20] for details.

DEFINITION 6.3 (see [12, 14, 15]). *A* full Steiner topology *of point set* P *is a tree graph whose vertex set contains* P *and* $N - 2$ *Steiner points and that the degree of each vertex in* P *is exactly* 1 *and that the degree of each Steiner vertex is exactly* 3.

Computing an SMT for a given set of N points in the Euclidean plane is NP-hard. However, the problem of computing the shortest network under a given full Steiner

topology can be solved efficiently. Recently, there have been increased interests in this latter problem, and several algorithms have been proposed [14, 15, 28]. We will formulate this problem as a special case of problem (2.1).

Let $m = 2N - 3$, $d = 2$, and $n = 2N - 4$. Let $u \in R^{2N-4}$ represent the locations of the $N - 2$ Steiner points. Without loss of generality, we may order the edges in the given full Steiner topology in such a way that each of the first N edges connects a regular point to a Steiner point. For $i = 1, 2, \ldots, N$, c_i is p_{i_1} where i_1 is the index of the regular point on the ith edge; $A_i^T \in R^{2\times n}$ is a row of $N - 2$ 2-by-2 block matrices where only the i_2th block is I_2 and the rest are all zero, where i_2 is the index of the Steiner point on the ith edge. For $i = N + 1, N + 2, \ldots, m$, $c_i = 0$ and $A_i^T \in R^{2\times n}$ is a row of $N - 2$ 2-by-2 block matrices where the i_1st block is $-I_2$, the i_2nd block is I_2, and the rest of the blocks are all zero, where i_1 and i_2 are the indices of the two Steiner points on the ith edge. It is clear that we have transformed the problem of computing a shortest network under a full Steiner topology into an instance of (2.1), where $d = 2$, $n = 2N - 4$, and $m = 2N - 3$. Therefore, it can be solved efficiently using our interior-point algorithm.

Note that we can move the point set P so that its gravitational center is the origin. Therefore, the Euclidean norms of the regular points are bounded by the largest pairwise distance among the points in P which corresponds to the constant $\bar{c}$ in previous theorems. Furthermore, we will show in the following that the search direction can be computed in $O(N)$ arithmetic operations using a technique known as *Gaussian elimination on leaves of a tree* [28].

Since $A_i^T \in R^{2\times N}$ contains at most two nonzero 2-by-2 blocks, the system (5.2) can be set up in $O(N)$ operations. The left-hand-side matrix of (5.2) (call it H) consists of $(N - 2)$-by-$(N - 2)$ blocks of 2-by-2 matrices. The (i, j) block of H is nonzero only if there is an edge in the topology which connects the ith and the jth Steiner points. Now consider the tree spanning the $N - 2$ Steiner points. We may *delete* a leaf vertex ea in the tree as follows: let eb be the (unique) vertex in the tree that is connected to ea by an edge in the tree. We *delete* the vertex ea and the edge $(ea,\ eb)$ from the tree by choosing $H(2 * ea - 1,\ 2 * ea - 1)$ as the pivot element and eliminate the entries $H(2 * ea - 0,\ 2 * ea - 1)$, $H(2 * eb - 1,\ 2 * ea - 1)$, and $H(2 * eb - 0,\ 2 * ea - 1)$. Then use $H(2 * ea - 0,\ 2 * ea - 0)$ as the pivot element and eliminate the entries $H(2 * eb - 1,\ 2 * ea - 0)$ and $H(2 * eb - 0,\ 2 * ea - 0)$. All of this can be done in $O(1)$ operations and will not make a zero block nonzero. In other words, *deleting* a leaf vertex in the tree requires $O(1)$ operations. Therefore, Gaussian elimination on leaves of a tree requires $O(N)$ operations. In the reverse order, back substitution can be done in $O(N)$ operations, too. Therefore, we have Theorem 6.4.

THEOREM 6.4. *An ϵ-optimal solution to the shortest network under a given full Steiner topology of N regular points in the Euclidean plane can be computed using our potential-reduction algorithms in $O(\sqrt{N}(\log(\bar{c}/\epsilon) + \log N))$ iterations where $\bar{c}$ is the largest pairwise distance among the regular points and each iteration requires $O(N)$ arithmetic operations. Therefore, the computation of an ϵ-optimal solution requires $O(N\sqrt{N}(\log(\bar{c}/\epsilon) + \log N))$ arithmetic operations.* □

The problem of computing the shortest network under a full Steiner topology was first studied by Hwang [14], Hwang and Weng [15], and Smith [28]. Hwang [14] presented a linear time exact algorithm that can output the shortest network under a given full Steiner topology if there exists a nondegenerate SMT corresponding to that given topology and quits otherwise. Hwang and Weng [15] presented an $O(N^2)$ time graphical algorithm that can output the shortest network under a given full Steiner

topology if the shortest network under the given topology is a tree with maximum vertex degree 3 and quits otherwise. Our algorithm can always output an ϵ-optimal network under the given topology in $O(N\sqrt{N}(\log(\bar{c}/\epsilon)+\log N))$ operations where $\bar{c}$ is the largest pairwise distance among the given points. This resolves an open problem of [15].

7. Computational examples. We have implemented all three versions of our algorithm using MATLAB. From our *preliminary* implementation, it seems that the one using dual scaling is numerically simpler and stabler. Therefore, we implemented that algorithm for computing the shortest network under a tree topology in Fortran 77, using Gaussian elimination on leaves of the topology tree. In the following, we present some preliminary computational results on the shortest network problem. Extensive computational study of the algorithms will be given in a separate paper.

TABLE 7.1
The coordinates of the 10 *regular points in example* 1.

index	x-coordinate	y-coordinate	index	x-coordinate	y-coordinate
9	2.30946900	9.20821100	14	7.59815200	0.61583600
10	0.57736700	6.48093800	15	8.56812900	3.07917900
11	0.80831400	3.51906200	16	4.75750600	3.75366600
12	1.68591200	1.23167200	17	3.92609700	7.00879800
13	4.11085500	0.82111400	18	7.43649000	7.68328400

The program was run on a Silicon Graphics Indy workstation. In our implementation, we used $\gamma = 2m$ to take *long steps* instead of using the conservative theoretical parameter $\gamma = 1$. Also, we used 0.9 times the largest feasible step-size as the actual step-size rather than using the theoretical step-size or a line search. For our implementation, we index the Steiner points first, followed by the regular points.

TABLE 7.2
The tree topology for example 1.

edge-index	ea-index	eb-index	edge-index	ea-index	eb-index
1	9	7	10	18	8
2	10	1	11	5	6
3	11	2	12	6	4
4	12	3	13	4	3
5	13	4	14	3	2
6	14	5	15	2	1
7	15	5	16	1	7
8	16	6	17	7	8
9	17	8			

Our first example contains 10 regular points. The coordinates of the 10 regular points are given in Table 7.1. The tree topology is given in Table 7.2 where for each edge, indices of its two vertices are shown next to the index of the edge. This topology is the best topology obtained by a branch-and-bound algorithm. Therefore, the shortest network under this topology is actually the SMT for the given 10 regular points.

Our algorithm solves this problem to 10^{-8} in 0.045 seconds and a total of 23 iterations. Table 7.3 shows the computer output of this test run. The second column in Table 7.3 shows the cost of the current network (i.e., the sum of Euclidean norms in the current network). The third column shows the duality gap, which is an upper bound of the error in the cost of the current network to the cost of the optimal

TABLE 7.3
Output of our algorithm for example 1.

iteration	network-cost	duality-gap	pstep-max	dstep-max
1	67.4046273974	755.3696677104	28.8384401017	0.1356841643
2	55.4697474888	195.1824411479	3.4806842442	0.1032272123
3	27.4932167097	101.3496514586	0.4832738411	0.1638326757
4	27.0322340903	26.0099590874	0.1113384478	0.1119122550
5	26.1012759902	9.1871566968	0.0403664605	0.1191177601
6	25.6601657571	2.5422959471	0.0099090385	0.1017762723
7	25.4826595690	0.8430133790	0.0022051467	0.0996860187
8	25.3997342761	0.3156519930	0.0006569311	0.1251074829
9	25.3713549379	0.1277519346	0.0001667930	0.1478311776
10	25.3613423325	0.0715146951	0.0002153768	0.3040757280
11	25.3575447731	0.0263880927	0.0000626213	0.3321649420
12	25.3565967482	0.0128961955	0.0000577043	0.5297753443
13	25.3562709801	0.0021763160	0.0000098621	0.1682928011
14	25.3561300062	0.0004901604	0.0000020309	0.1142469220
15	25.3560841523	0.0001476805	0.0000006305	0.1145905399
16	25.3560721582	0.0000397667	0.0000001962	0.1007746545
17	25.3560692545	0.0000107249	0.0000000618	0.0897091127
18	25.3560681805	0.0000026435	0.0000000156	0.0732031761
19	25.3560678856	0.0000008525	0.0000000057	0.0818257275
20	25.3560678157	0.0000002438	0.0000000015	0.0789620241
21	25.3560677874	0.0000000757	0.0000000005	0.0834184991
22	25.3560677824	0.0000000206	0.0000000001	0.0763003478
23	25.3560677802	0.0000000065	0.0000000000	0.0825900991

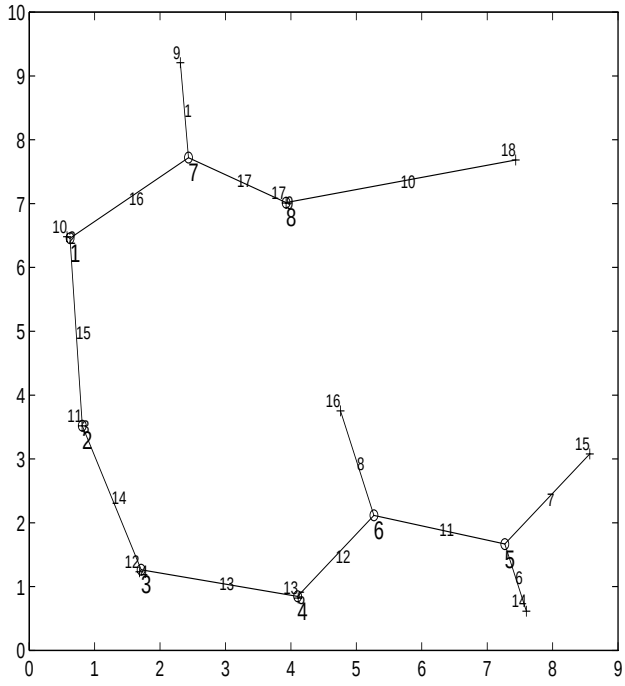

FIG. 7.1. *The shortest network for* 10 *regular points in example* 1.

TABLE 7.4
The topology and the coordinates of the four regular points in example 2.

point-index	x-coord	y-coord	point-index	x-coord	y-coord
3	−100.0	1.0	5	−100.0	−1.0
4	100.0	1.0	6	100.0	1.0
edge-index	ea-index	eb-index	edge-index	ea-index	eb-index
1	3	1	4	6	2
2	4	1	5	1	2
3	5	2			

TABLE 7.5
Output of our algorithm for example 2.

iteration	network-cost	duality-gap	pstep-max	dstep-max
1	40.1995024845	120.9404740550	0.8445579680	0.0492658697
2	41.3468150068	13.7530974274	0.0724695587	0.0339618552
3	40.2654764899	3.0068405143	0.0185627882	0.0299007875
4	40.2539043360	0.4948832387	0.0027833196	0.0183486445
5	40.2010107181	0.1731075550	0.0015791427	0.0255792520
6	40.2020816006	0.0325421036	0.0002280125	0.0173067376
7	40.2001533446	0.0115207903	0.0001157000	0.0248950063
8	40.1997465956	0.0022693804	0.0000146742	0.0171813134
9	40.1995061111	0.0008119015	0.0000084984	0.0253539092
10	40.1995148034	0.0001551869	0.0000010559	0.0173411905
11	40.1995055233	0.0000546467	0.0000005664	0.0248868315
12	40.1995036402	0.0000107280	0.0000000713	0.0171141485
13	40.1995024994	0.0000038458	0.0000000411	0.0253093321
14	40.1995025427	0.0000007393	0.0000000049	0.0173829301
15	40.1995024992	0.0000002599	0.0000000027	0.0249213172
16	40.1995024900	0.0000000508	0.0000000003	0.0170733567
17	40.1995024846	0.0000000183	0.0000000002	0.0252981953
18	40.1995024848	0.0000000035	0.0000000000	0.0174074280

(shortest) network. The last two columns show the largest primal and dual feasible step-sizes.

The final solution is shown in Figure 7.1, where regular points are labeled by "+" and Steiner points are labeled by "o." We can see from Figure 7.1 that the shortest network is degenerate [15] where five edges (each connecting a regular point to a Steiner point) shrink. This problem can be solved using the graphical method of [15] but is very difficult for algorithms like HAP [9]. For comparison, we have used HAP to solve the same problem by setting $\epsilon = 10^{-8}$ and using the locations of Steiner points generated by one step of our algorithm as the starting point for HAP. Because the problem is degenerate, HAP ran poorly compared with our algorithm. To get a solution as good as the one obtained using 10 iterations of our algorithm, HAP used 39.512 seconds and 248500 iterations. No matter how long we let it run, HAP failed to find a solution whose cost function is better than 25.3561402805 which can be obtained by our algorithm in 14 iterations.

Our second example has four regular points. The purpose of this example is to show that our algorithm can compute the shortest network under a tree topology where two Steiner points coincide. The algorithm in [15] will quit on this problem before it finds the shortest network. The coordinates of the four regular points and the tree topology in this example are given in Table 7.4. This topology is not the best topology. Therefore, the shortest network under this topology is not the SMT for the given four regular points.

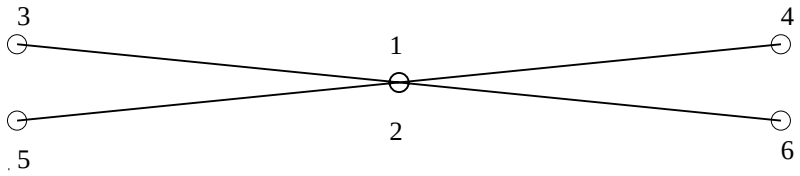

FIG. 7.2. *The shortest network for four regular points in example* 2.

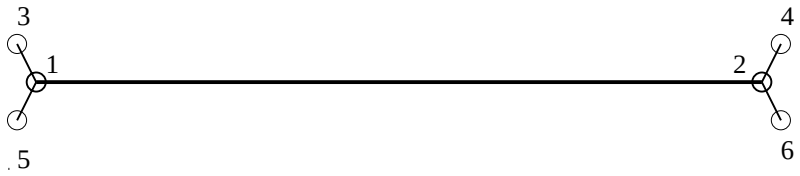

FIG. 7.3. *The SMT for four regular points in example* 2.

Our algorithm solves the second problem to 10^{-8} in 0.022 seconds and a total of 18 iterations. Table 7.5 shows the computer output of this test run. The shortest network under this topology has a cost of 40.1995 and is illustrated in Figure 7.2.

Figure 7.3 shows the SMT, which is the shortest network under a different topology. The corresponding cost is 23.4641. We would like to point out that the algorithms of [14] and [15] can both find the shortest network under this topology.

8. Conclusions. In this paper, we have transformed the problem of minimizing a sum of Euclidean norms into a standard convex programming problem in its dual conic form where the cone and its associated barrier are self-scaled [23]. We then presented an efficient primal–dual potential reduction algorithm for solving this problem. In applications, we have shown that computing an ϵ-optimal solution of the shortest network under a tree topology interconnecting N regular points requires only $O(N\sqrt{N}(\log(\bar{c}/\epsilon)+\log N))$ arithmetic operations, where $\bar{c}$ is the largest pairwise distance among the given point set.

When applied to compute the shortest network under a tree topology interconnecting N regular points, our algorithm does not suffer from degeneracies and it compares favorably with the $O(N^2)$ algorithm of [15] in both theoretical complexity and ease of implementation. When applied to EMFL problems, our algorithm compares favorably with the algorithm of [32] because our algorithm has a better complexity result and stores the locations of the new facilities in the dual variable u while the latter does not provide such information directly. Our implementation is only preliminary. Computational issues of our algorithm are under investigation and will be reported in another paper.

REFERENCES

[1] K. D. ANDERSEN, *An efficient Newton barrier method for minimizing a sum of Euclidean norms*, SIAM J. Optim., 6 (1996), pp. 74–95.

[2] K. D. ANDERSEN AND E. CHRISTIANSEN, *A Symmetric Primal–Dual Newton Method for Minimizing a Sum of Norms*, manuscript, Odense University, Denmark, 1995.

[3] P. H. CALAMAI AND A. R. CONN, *A second-order method for solving the continuous multifacility location problem*, in Numerical Analysis: Proceedings of the Ninth Biennial Conference, G. A. Watson, ed., Dundee, Scotland, Lecture Notes in Mathematics 912, Springer-Verlag, Berlin, 1982, pp. 1–25.

[4] P. H. Calamai and A. R. Conn, *A projected Newton method for l_p norm location problems*, Math. Programming, 38 (1987), pp. 75–109.
[5] R. Chandrasekaran and A. Tamir, *Open questions concerning Weiszfeld's algorithm for the Fermat–Weber location problem*, Math. Programming, 44 (1989), pp. 293–295.
[6] R. Chandrasekaran and A. Tamir, *Algebraic optimization: The Fermat–Weber location problem*, Math. Programming, 46 (1990), pp. 219–224.
[7] A. R. Conn and M. L. Overton, *A Primal–Dual Interior Point Method for Minimizing a Sum of Euclidean Distances*, 1995, manuscript.
[8] D. den Hertog, *Interior Point Approach to Linear, Quadratic and Convex Programming*, Kluwer Academic Publishers, Norwell, MA, 1994.
[9] J. W. Eyster, J. A. White, and W. W. Wierwille, *On solving multifacility location problems using a hyperboloid approximation procedure*, AIIE Transactions, 5 (1973), pp. 1–6.
[10] R. L. Francis and A. V. Cabot, *Properties of a multifacility location problem involving Euclidean distances*, Naval Res. Logist., 19 (1972), pp. 335–353.
[11] R. L. Francis, Leon F. McGinnis, Jr., and John A. White, *Facility Layout and Location: An Analytical Approach*, Prentice–Hall, Englewood Cliffs, NJ, 1991.
[12] E. N. Gilbert and H. O. Pollak, *Steiner minimal trees*, SIAM J. Appl. Math., 16 (1968), pp. 1–29.
[13] O. Güler, *Barrier Functions in Interior Point Methods*, manuscript, 1994.
[14] F. K. Hwang, *A linear time algorithm for full Steiner trees*, Oper. Res. Lett., 4 (1986), pp. 235–237.
[15] F. K. Hwang and J. F. Weng, *The shortest network under a given topology*, J. Algorithms, 13 (1992), pp. 468–488.
[16] M. Kojima, Tokyo Institute of Technology, Japan, 1995, private communication.
[17] H. W. Kuhn, *On a pair of dual nonlinear programs*, in Nonlinear Programming, J. Abadie, ed., North–Holland, Amsterdam, 1967, pp. 39–54.
[18] H. W. Kuhn, *A note on Fermat's problem*, Math. Programming, 4 (1973), pp. 98–107.
[19] R. F. Love, J. G. Morris, and G. O. Wesolowsky, *Facilities Location: Models & Methods*, North–Holland, Amsterdam, 1988.
[20] Z. A. Melzak, *On the problem of Steiner*, Canad. Math. Bull., 16 (1961), pp. 143–148.
[21] W. Miehle, *Link length minimization in networks*, Oper. Res., 6 (1958), pp. 232–243.
[22] Yu. E. Nesterov and A. Nemirovskii, *Interior Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia, 1994.
[23] Yu. E. Nesterov and M. J. Todd, *Self-Scaled Barriers and Interior-Point Methods for Convex Programming*, manuscript.
[24] L. M. Ostresh, *The multifacility location problem: Applications and descent theorems*, J. Regional Science, 17 (1977), pp. 409–419.
[25] M. L. Overton, *A quadratically convergent method for minimizing a sum of Euclidean norms*, Math. Programming, 27 (1983), pp. 34–63.
[26] J. B. Rosen and G. L. Xue, *On the convergence of Miehle's algorithm for the Euclidean multifacility location problem*, Oper. Res., 40 (1992), pp. 188–191.
[27] J. B. Rosen and G. L. Xue, *On the convergence of a hyperboloid approximation procedure for solving the perturbed Euclidean multifacility location problem*, Oper. Res., 41 (1993), pp. 1164–1171.
[28] W. D. Smith, *How to find Steiner minimal trees in Euclidean d-space*, Algorithmica, 7 (1992), pp. 137–177.
[29] C. Y. Wang et al., *On the convergence and rate of convergence of an iterative algorithm for the plant location problem*, Qufu Shiyun Xuebao, 2 (1975), pp. 14–25 (in Chinese).
[30] E. Weiszfeld, *Sur le point par lequel le somme des distances de n points donnes est minimum*, Tôhoku Math. J., 43 (1937), pp. 355–386.
[31] G. L. Xue, *Algorithms for Computing Extreme Points of Convex Hulls and the Euclidean Facilities Location Problem*, Ph.D. thesis, Computer Science Department, University of Minnesota, Minneapolis, MN, 1991.
[32] G. L. Xue, J. B. Rosen, and P. M. Pardalos, *A polynomial time dual algorithm for the Euclidean multifacility location problem*, in Proceedings of Second Conference on Integer Programming and Combinatorial Optimization, Pittsburgh, PA, 1992, pp. 227–236.
[33] Y. Ye, *Interior-point algorithms for quadratic programming*, in Recent Developments in Mathematical Programming, S. Kumar, ed., Gordon and Breach, New York, 1991, pp. 237–262.