A Projected Newton Method for l_p Norm Location Problems

P.H. Calamai
A.R. Conn
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

Technical Report CS-85-07
August 1985

A PROJECTED NEWTON METHOD FOR l_p NORM LOCATION PROBLEMS

P. H. CALAMAI† *AND A. R. CONN*‡

ABSTRACT

This paper is concerned with the numerical solution of continuous minisum multifacility location problems involving the l_p norm, where $1 < p < \infty$. This class of problems is potentially difficult to solve because the objective function is not everywhere differentiable. After developing conditions that characterize the minimum of the problems under consideration, a second-order algorithm is presented. This algorithm is based on the solution of a finite sequence of linearly constrained subproblems. Descent directions for these subproblems are obtained by projecting the Newton direction onto the corresponding constraint manifold. Univariate minimization is achieved via a specialized linesearch which recognizes the possibility of first derivative discontinuity (and second derivative unboundedness) at points along the search direction. The algorithm, motivated by Calamai [3] and Calamai and Conn [5,6] and related to methods recently described by Overton [32] and Dax [10], is shown to possess both global and quadratic convergence properties.

Degeneracy can complicate the numerical solution of the subproblems. This degeneracy is identified, and a method for handling this degeneracy is outlined.

An implementation of the algorithm, that exploits the intrinsic structure of the location problem formulation, is then described along with a discussion of numerical results.

Key Words
multifacility location problem, Fermat problem,
Weber problem, nonsmooth optimization, projection
techniques

†Department of Systems Design Engineering, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada and Mathematics and Computer Science Division, Argonne National Laboratories, Argonne, Illinois, U.S.A.

‡Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada.

This research was supported in part by Natural Sciences and Engineering Research Council of Canada grants A-5671 and A-8639 and by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under Contract W-31-109-Eng-38. This paper was typeset using software developed at Bell Laboratories and the University of California at Berkeley.

A PROJECTED NEWTON METHOD FOR l_p NORM LOCATION PROBLEMS

P. H. CALAMAI† *AND A. R. CONN*‡

1. Introduction. In this paper we examine a continuous minisum multifacility location problem involving l_p distances, where $1 < p < \infty$. For simplicity we consider the case where all facilities lie on the plane, $\mathbb{R}^2$. The method and results are readily extended to the more general case where facilities lie in a higher dimensional space. The problem can be stated as:

Find a point $x^{*T} = \{x_1^{*T}, \ldots, x_n^{*T}\}$ in $\mathbb{R}^{2n}$ that minimizes

$$f(x) = \sum_{1 \le j < k \le n} v_{jk} \| x_j - x_k \|_p + \sum_{j=1}^{n} \sum_{i=1}^{m} w_{ji} \| x_j - y_i \|_p \qquad \text{(P1)}$$

where

$n \;\Delta$ the number of new facilities (NF's) to be located,

$m \;\Delta$ the number of existing facilities (EF's),

$x_j \;\Delta$ the vector location of NF_j in $\mathbb{R}^2$, $j = 1, \ldots, n$,

$y_i \;\Delta$ the vector location of EF_i in $\mathbb{R}^2$, $i = 1, \ldots, m$,

$v_{jk} \;\Delta$ the nonnegative weight on the l_p distance between NF_j and NF_k, $1 \le j < k \le n$,

$w_{ji} \;\Delta$ the nonnegative weight on the l_p distance between NF_j and EF_i, $1 \le j \le n$, $1 \le i \le m$,

$\| x_j - x_k \|_p \;\Delta$ the l_p distance between NF_j and NF_k , $1 \le j < k \le n$,

$\| x_j - y_i \|_p \;\Delta$ the l_p distance between NF_j and EF_i , $1 \le j \le n$, $1 \le i \le m$.

The quantity of literature dealing with the algorithmic solution of problem P1 is monumental. The bibliographies of Francis and Goldstein [18] and Lea [26] and the books by Francis and White [19] and Eilon, Watson-Gandy and Christofides [16] attest to this.

The most popular algorithm for solving P1 when distances are Euclidean (when $p=2$) was devised by Weiszfeld [34]. For the single facility problem, where $n=1$ (also called the generalized Fermat problem and the Weber problem), he proposed an iterative fixed-step steepest descent algorithm that was

†Department of Systems Design Engineering, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada.

‡Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada.

This research was supported in part by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under Contract W-31-109-Eng-38 and by Natural Sciences and Engineering Research Council of Canada grants A-5671 and A-8639.

adapted by Kuhn and Kuenne [25] and Ostresh [31]. Various modifications to Weiszfeld's algorithm have been used to solve the Euclidean distance multifacility problem [31], the l_p distance multifacility problem [29] and the generalized Weber problem in Banach space [15]. Other algorithms proposed for solving these problems use a surrogate distance measure in place of the l_p distance measure. Examples include the convex programming procedure of Love [27], the hyperbolic approximation procedures of Wesolowsky and Love [35] and Eyster, White and Wierwille [17], a variant of Weiszfeld's method used by Morris and Verdini [29], and the trajectory optimization method of Drezner and Wesolowsky [14]. In addition, Vergin and Rogers [33] and Calamai and Charalambous [4] provide heuristic methods for solving location problems.

In this paper we extend the ideas introduced by Calamai and Conn [5,6] and Calamai [3] for solving P1. Optimality conditions, that avoid the qualifying conditions that were assumed in these previous papers (an independence assumption), are stated and proved. We then demonstrate how P1, a nonsmooth optimization problem, can be solved using optimization techniques for smooth functions without modifying the objective function to eliminate the gradient discontinuities. This is accomplished by successively minimizing f on linear manifolds on which it is locally smooth. A linesearch that computes all derivative discontinuities exactly, if any exist, and exploits this information is also presented. The resulting algorithm for solving P1 is shown to possess both global and quadratic convergence properties. One complication that sometimes arises in solving P1 occurs when groups of new facilities coincide. In such circumstances tests equivalent to those presented by Juel and Love [24] for the rectilinear problem (p=1 in P1) could be used to circumvent this difficulty. However these tests can be prohibitively expensive (as demonstrated by Juel and Love [24] and Dax [12]) because they are combinatorial in nature. The solution of a constrained least-squares subproblem, as demonstrated by Dax [11] for the Euclidean problem (p=2 in P1) and introduced by Busovača [2] for general nonlinear problems, will also verify optimality or provide a descent direction in such circumstances. Unfortunately the solution of this least-squares problem is nontrivial and involves techniques unrelated to the main procedure used in solving P1. In this paper we present a simple perturbation scheme, motivated by techniques for handling degeneracy in linear programming, that is unified in its approach, simple to implement and inexpensive to use. We prove that this technique finds a descent direction for the perturbed problem (if one exists) with very little computational effort and we demonstrate the effectiveness of this scheme in combination with the line search. Although no theoretical result is given, this technique has proven to be entirely successful on all problems tested. Finally we provide implementation details for our algorithm and demonstrate how the structure of these problems can be exploited to provide very efficient techniques for almost all of the linear algebra involved in the solution process. These techniques are developed by examining the incidence graph associated with clusters of facilities.

2. Problem Formulation and Duality Relationships. Following Calamai and Conn [6] we define the index set $M = \{1, \ldots, \tau\}$, where the set $\alpha = \{\alpha_1, \ldots, \alpha_\tau\}$ is in one-to-one correspondence with the set of *nonzero* weights v_{jk} and w_{ji}, and write P1 more conveniently as:

Find the point $x^{*T} = \{x_1^{*T}, \ldots, x_n^{*T}\}$ in $\mathbb{R}^{2n}$ that minimizes

$$f(x) = \sum_{l \in M} f_l(x) \qquad \text{(P2)}$$

where $f_l(x) = \| r_l \|_p$, $r_l = A_l^T x - b_l$ and where the $2n$ by 2 matrices A_l and the

2 by 1 vectors b_l satisfy

$$A_l = \alpha_l \, e(l) \otimes I_2,$$

$$b_l = \alpha_l \, y(l),$$

$$e(l) = \begin{cases} e_j - e_k & \text{when } \alpha_l \text{ corresponds to } v_{jk} \\ e_j & \text{when } \alpha_l \text{ corresponds to } w_{ji}, \end{cases}$$

$$y(l) = \begin{cases} 0 & \text{when } \alpha_l \text{ corresponds to } v_{jk} \\ y_i & \text{when } \alpha_l \text{ corresponds to } w_{ji}, \end{cases}$$

where e_j denotes the j^{th} column of the n by n identity matrix and $\otimes$ is the Kronecker product operator (see [23]).

Prior to stating optimality conditions for problem P2 it is convenient to introduce the concepts of a *dual norm* $\|\cdot\|_D$, corresponding to a primal norm $\|\cdot\|_P$, that satisfies

$$\|v\|_D = \max\{\, v^T w \mid \|w\|_P \le 1 \,\},$$

and a $\|\cdot\|_P$*-dual vector* ϑ, corresponding to a nonzero primal vector v, that satisfies

$$\|\vartheta\|_P = 1 \text{ and } v^T\vartheta = \|v\|_D.$$

For $\|\cdot\|_P = \|\cdot\|_p$, $1 < p < \infty$, and for q satisfying $p + q = pq$,

$$\|v\|_D = \|v\|_q$$

and, for $v \neq 0$, ϑ has components ϑ_i given by

$$\vartheta_i = \operatorname{sgn} v_i \left\{ \frac{|v_i|}{\|v\|_q} \right\}^{q-1}.$$

We assume that $p + q = pq$ for the remainder of this paper.

3. Optimality Conditions. The objective function f is everywhere convex but is nondifferentiable (nonsmooth) at all points x^k where any of the functions $f_l(x^k)$, $l \in M$, are zero. This occurs whenever two or more *interacting* facilities coincide.

If, for $\varepsilon \ge 0$, the set $M_\varepsilon(x^k)$ satisfies

$$M_\varepsilon(x^k) = \{ l \in M \mid f_l(x^k) \le \varepsilon \}$$

then, at the point x^k, the index set $M_0(x^k)$ identifies the functions f_l that are *active* (nondifferentiable) and the index set $M - M_0(x^k)$ identifies those functions f_l that are *inactive* (differentiable). Using these definitions we can divide the objective function f into two parts as follows:

$$f(x) = \sum_{l \in M - M_\varepsilon(x^k)} f_l(x) + \sum_{l \in M_\varepsilon(x^k)} f_l(x).$$

Theorem 3.1. *The point x^* solves P2 if and only if there exists vectors $u_l \in \mathbb{R}^2$, called the Lagrange vectors, such that*

$$\sum_{l \in M - M_0(x^*)} \nabla f_l(x^*) = \sum_{l \in M_0(x^*)} A_l u_l \tag{3.1}$$

and

$$\|u_l\|_q \le 1 \quad \forall\, l \in M_0(x^*). \tag{3.2}$$

Proof. For any convex function g, a point x^* is a global minimizer if and only if $0 \in \partial g(x^*)$, where $\partial g(x)$ denotes the subdifferential (the set of subgradients) of g at x. In our case the functions f_l, $l \in M$, are convex functions and $f = \sum_{l \in M} f_l$. Thus, (see [7]), $\partial f(x^*) = \sum_{l \in M} \partial f_l(x^*)$ where

$$\partial f_l(x^*) = \begin{cases} \{\, \nabla f_l(x^*) \,\} & \forall\, l \in M - M_0(x^*) \\ \{\, z \mid f_l(x^* + h) \ge f_l(x^*) + z^T h,\ \forall h \,\} & \forall\, l \in M_0(x^*) \end{cases}$$

or, equivalently,

$$\partial f_l(x^*) = \begin{cases} \{\, \nabla f_l(x^*) \,\} & \forall\, l \in M - M_0(x^*) \\ \{\, z \mid \|A_l^T h\|_p \ge z^T h,\ \forall h \,\} & \forall\, l \in M_0(x^*). \end{cases}$$

Since $A_l^T h = 0$ for all h in the nullspace of A_l^T, $z \in Range(A_l)$ or, equivalently,

$$\partial f_l(x^*) = \{\, A_l u_l \mid \|A_l^T h\|_p \ge u_l^T A_l^T h,\ \forall h \,\} \quad \forall\, l \in M_0(x^*).$$

Now $\|A_l^T h\|_p \|u_l\|_q \ge u_l^T A_l^T h$, $\forall\, h$ (Hölder) and if $h = A_l w_l$, where w_l is the solution to the full-rank system $A_l^T A_l w_l = \tilde{u}_l$ and $\tilde{u}_l$ is the $\|\cdot\|_p$- dual vector of u_l, then $\|A_l^T h\|_p = 1$ and $u_l^T A_l^T h = \|u_l\|_q$.
Consequently $\|A_l^T h\|_p \ge u_l^T A_l^T h$, $\forall h$, if and only if, $\|u_l\|_q \le 1$.
We therefore have

$$\partial f_l(x^*) = \begin{cases} \{\, \nabla f_l(x^*) \,\} & \forall\, l \in M - M_0(x^*) \\ \{\, A_l u_l \mid \|u_l\|_q \le 1 \,\} & \forall\, l \in M_0(x^*) \end{cases}$$

and the proof of the theorem is complete. ▪

Any point satisfying (3.1) will be called a *stationary point*. If a stationary point also satisfies (3.2) then it will be called a *minimizer*.

4. Motivation and Theory. Consider the following problem in which x^k is fixed and $\varepsilon \ge 0$:

$$\begin{array}{lll} \underset{x}{\text{minimize}} & \sum_{l \in M - M_\varepsilon(x^k)} f_l(x) & \\ \text{subject to} & r_l(x) = r_l(x^k) & \forall l \in M_\varepsilon(x^k). \end{array} \tag{4.1}$$

This objective function is differentiable at all points x in some *nonempty* neighborhood of x^k. Consequently, local first- and second-order methods exist for this linearly constrained problem. In addition, the solution x^* to P2 is a solution to this problem when $x^k = x^*$ and $M_\varepsilon(x^k) = M_0(x^*)$. It would therefore be beneficial if problem P2 could be posed, *without a priori knowledge of $M_0(x^*)$*, and solved via a sequence of subproblems of the form (4.1) in which $\{x^k\} \to x^*$ and $\{M_\varepsilon(x^k)\} \to M_0(x^*)$.

To aid further exposition define g $(=g_\varepsilon)$ and G $(=G_\varepsilon)$ so that in a neighborhood of the current point x^k,

$$g = \sum_{l \in M - M_\varepsilon(x^k)} \nabla f_l$$

and

$$G = \sum_{l \in M - M_\varepsilon(x^k)} \nabla^2 f_l$$

and define Z $(= Z(x^k))$ as an orthonormal matrix whose columns span the space $\{h \mid A_l^T h = 0, \ \forall l \in M_\varepsilon(x^k)\}$.

Case 1. If x^k is a nonstationary point then we can solve (4.1) *up to second-order terms* by solving

$$\begin{aligned} &\underset{h}{\text{minimize}} \quad h^T g(x^k) + \tfrac{1}{2}\, h^T G(x^k) h \\ &\text{subject to} \qquad A_l^T h = 0 \qquad \forall l \in M_\varepsilon(x^k). \end{aligned} \tag{4.2}$$

In other words, we can reduce $f(x^k)$ on the manifold defined by the constraints in (4.2).

The following lemma applies to problem P2 in this case:

Lemma 4.1. *If $Z^T g(x^k) \neq 0$ and if $Z^T G(x^k) Z$ is positive definite then the direction*

$$h^k = Z h_z^*,$$

where $h_z = h_z^*$ satisfies

$$Z^T G(x^k) Z h_z = -Z^T g(x^k),$$

solves (4.2) and is a local descent direction for P2 from the point x^k.

Proof. Since any direction h that satisfies the constraints in (4.2) can be written as Zh_z, for some vector h_z, problem (4.2) becomes

$$\underset{h_z}{\text{minimize}}\ h_z^T Z^T g(x^k) + \tfrac{1}{2}\, h_z^T Z^T G(x^k) Z h_z.$$

If $Z^T G(x^k) Z$ is positive definite the solution to this problem can be obtained by solving

$$Z^T G(x^k) Z h_z = -Z^T g(x^k).$$

In addition, if $Z^T G(x^k) Z$ is positive definite then

$$g(x^k)^T h^k = -h_z^{*T} Z^T G(x^k) Z h_z^* < 0$$

which completes the proof. ▪

The vector $Z^T g(x^k)$ and the matrix $Z^T G(x^k) Z$ are called respectively the *projected gradient* and the *projected Hessian* of f on $x^k \oplus \mathit{Range}(Z)$. In addition, the vector $h^k = Z h_z^*$ is the associated *projected Newton step* for f at the point x^k.

Case 2. If x^k is a stationary point then, as a consequence of (3.1) and the definitions of Z and g, $Z^T g(x^k) = 0$ $\forall \varepsilon \geq 0$. There therefore exists Lagrange vectors u_l, $l \in M_\varepsilon(x^k)$, such that

$$g(x^k) = \sum_{l \in M_\varepsilon(x^k)} A_l u_l. \tag{4.3}$$

We assume for the time being that the matrices A_l, $\forall l \in M_\varepsilon(x^k)$, are linearly independent but in §5 we demonstrate how this qualification is relaxed.

The following three possibilities exist:

Case 2.1. If $\|u_l\|_q \le 1$ and $f_l(x^k) = 0$, $\forall l \in M_\varepsilon(x^k)$, then $M_\varepsilon(x^k) = M_0(x^k)$, $g = g_0$ and x^k is a minimizer of P2 (as demonstrated in Theorem 3.1).

Case 2.2. If $\|u_l\|_q \le 1$ $\forall l \in M_\varepsilon(x^k)$ but $f_j(x^k) \ne 0, j \in M_\varepsilon(x^k)$, then $M_\varepsilon(x^k) \ne M_0(x^k)$. Since we may be on the correct linear manifold and in the neighborhood of a minimizer for P2 the point $x^k + v^k$, where the step v^k is the minimal norm solution of the full-rank system

$$r_l(x^k + v^k) = 0, \quad \forall l \in M_\varepsilon(x^k),$$

is considered (since $M_0(x^k + v^k) = M_\varepsilon(x^k)$). If $f(x^k + v^k) > f(x^k)$ this new point is rejected and our computations are refined (by, among other things, reducing ε).

Case 2.3. If $\|u_j\|_q > 1$, $j \in M_\varepsilon(x^k)$, where u_j is uniquely determined in (4.3), then $x^k \ne x^*$ if $M_\varepsilon(x^k) = M_0(x^k)$. To demonstrate how f can be reduced on a new linear manifold let Z_j $(= Z_j(x^k))$ be an orthonormal matrix whose columns span the space $\{h \mid A_l^T h = 0, \forall l \in M_\varepsilon(x^k) - \{j\}\}$. If we reconsider problem (4.1) with $M_\varepsilon(x^k) - \{j\}$ replacing $M_\varepsilon(x^k)$ then *locally* we wish to find a direction h such that, for $\lambda > 0$ sufficiently small,

$$\sum_{l \in M - M_\varepsilon(x^k) + \{j\}} \{f_l(x^k + \lambda h) - f_l(x^k)\} < 0 \tag{4.4}$$

and

$$A_l^T h = 0 \qquad \forall l \in M_\varepsilon(x^k) - \{j\}. \tag{4.5}$$

The following lemma applies to problem P2 in this case:

Lemma 4.2. *If $\|u_j\|_q > 1$, $j \in M_\varepsilon(x^k)$, where u_j is uniquely determined by (4.3) when the matrices A_l $l \in M_\varepsilon(x^k)$ are linearly independent, then the direction*

$$h_j^k = Z_j h_{z_j}, \tag{4.6}$$

with h_{z_j} chosen so that

$$A_j^T h_j^k = -\rho \tilde{u}_j, \; \rho > 0, \tag{4.7}$$

where $\tilde{u}_j$ is the $\|\cdot\|_p$-dual vector of u_j, satisfies (4.4) and (4.5) and is a local descent direction for P2 from the point x^k.

Proof. The definition of Z_j and the fact that $h_j^k \in Range(Z_j)$ guarantees that (4.5) is satisfied. In addition, using (4.3), (4.5), (4.7), and ignoring higher-order terms, we have

$$\begin{aligned}\sum_{l \in M - M_\varepsilon(x^k)} \{f_l(x^k + \lambda h_j^k) - f_l(x^k)\} &= \lambda (h_j^k)^T g(x^k) \\ &= \lambda (h_j^k)^T A_j u_j . \\ &= -\lambda \rho \tilde{u}_j^T u_j \\ &= -\lambda \rho \|u_j\|_q .\end{aligned}$$

Now explicit (but tedious) calculation shows that when $j \notin M_0(x^k)$ $\nabla f_j(x^k) = A_j \vec{r}_j$, where $\vec{r}_j$ is the $\|\cdot\|_q$-dual vector of r_j. Consequently, if we ignore higher-order terms when $j \notin M_0(x^k)$ and use the fact that $-\vec{u}_j^T \vec{r}_j \le \|\vec{u}_j\|_p \|\vec{r}_j\|_q = 1$, we have

$$f_j(x^k + \lambda h_j^k) - f_j(x^k) = \begin{cases} \lambda \|A_j^T h_j^k\|_p & j \in M_0(x^k) \\ \lambda (h_j^k)^T \nabla f_j(x^k) & j \notin M_0(x^k) \end{cases}$$

$$= \begin{cases} \lambda\rho \|\vec{u}_j\|_p & j \in M_0(x^k) \\ \lambda (h_j^k)^T A_j \vec{r}_j & j \notin M_0(x^k) \end{cases}$$

$$= \begin{cases} \lambda\rho & j \in M_0(x^k) \\ -\lambda\rho \vec{u}_j^T \vec{r}_j & j \notin M_0(x^k) \end{cases}$$

$$\le \lambda\rho \qquad j \in M_\varepsilon(x^k).$$

We therefore have

$$\sum_{l \in M - M_\varepsilon(x^k) + \{j\}} \{f_l(x^k + \lambda h_j^k) - f_l(x^k)\} \le \lambda\rho(1 - \|u_j\|_q) + O(\|\lambda h_j^k\|_2^2)$$

which completes the proof. ▪

It should be emphasized that this section in no way describes the numerical implementation of our method. These details are left for §9.

5. Degeneracy. When the matrices A_l, $l \in M_\varepsilon(x^k)$, are linearly dependent we call problem (4.1) a *degenerate* problem and the point x^k a *degenerate* point.

A difficulty arises with degenerate problems when a unique solution to (4.3) is sought; however, a unique solution can be obtained and the results of §4, Cases 2.1 and 2.2 can be used, if (4.3) is replaced with the problem of finding Lagrange vectors u_l, $l \in M_\varepsilon(x^k)$ such that

$$g(x^k) = \sum_{l \in M_\varepsilon(x^k)} A_l u_l \tag{5.1}$$

with

$$u_l = 0 \qquad \forall l \in M_\varepsilon(x^k) - \overline{M}_\varepsilon(x^k), \tag{5.2}$$

where the index set $\overline{M}_\varepsilon(x^k)$ is chosen from $M_\varepsilon(x^k)$ so that the matrices A_l, $l \in \overline{M}_\varepsilon(x^k)$, form a basis for the span of the matrices A_l, $l \in M_\varepsilon(x^k)$. However, if $A_j \in span\{A_l,\ l \in M_\varepsilon(x^k) - \{j\}\}$ then $Z_j^T A_j = 0$ and the direction h_j^k defined by (4.6) and (4.7) will not provide a local descent direction for P2 from the degenerate point x^k.

If we let Z_j ($= Z_j(x^k)$) be an orthonormal matrix whose columns span the space $\{h \mid A_l^T h = 0,\ \forall l \in \overline{M}_\varepsilon(x^k) - \{j\}\}$ then the following theorem suggests a method for handling degeneracy:

Theorem 5.1. *If we assume that*

(1) *the current point x^k is a degenerate stationary point,*

(2) *the Lagrange vectors* u_l, $l \in M_\varepsilon(x^k)$, *satisfy (5.1) and (5.2),*

(3) *there exists an index* $j \in M_\varepsilon(x^k)$ *such that* $\|u_j\|_q > 1$, *and*

(4) *the scalars* $\eta(l,i)$, $l \in \overline{M}_\varepsilon(x^k)$, *satisfy*

$$A_i = \sum_{l \in \overline{M}_\varepsilon(x^k)} \eta(l,i) A_l \qquad \forall i \in M_\varepsilon(x^k) - \overline{M}_\varepsilon(x^k)$$

then, if we perturb our problem by setting

$$b_i = A_i^T x^k - \zeta_i \vec{u}_j, \qquad \forall i \in N,$$

where $N = \{i \in M_\varepsilon(x^k) - \overline{M}_\varepsilon(x^k) \mid \eta(j,i) \neq 0\}$, $\vec{u}_j$ *equals the* $\|\cdot\|_p$*-dual vector of* u_j *and* $\zeta_i = \nu_i \, sgn\, \eta(j,i)$ *with* $\nu_i > \varepsilon$, *then*

(1) *the terms* f_i, $i \in N$, *are differentiable in some nonempty neighborhood of the point* x^k *and* $i \notin M_\varepsilon(x^k)$ *in the perturbed problem,*

(2) *if the vectors* μ_l, $l \in \overline{M}_\varepsilon(x^k)$, *are the Lagrange vectors at* x^k *for the perturbed problem then* $\|\mu_j\|_q \geq \|u_j\|_q > 1$, *and*

(3) *the direction* $h_j^k = Z_j h_{z_j}$, *where* h_{z_j} *is chosen so that* $A_j^T h_j^k = -\rho \vec{u}_j$, $\rho > 0$, *intersects each of the perturbed* $(2n-2)$*-dimensional hyperplanes defined by* $r_i(x) = 0$, $\forall i \in N$, *and is a descent direction for the perturbed problem.*

Proof.

Part 1

For the perturbed problem we have:

$$f_i(x^k) = \|r_i(x^k)\|_p = \|\zeta_i \vec{u}_j\|_p = |\nu_i| > \varepsilon, \qquad \forall i \in N,$$

and

$$\nabla f_i(x^k) = A_i \vec{r}_i = \nu_i \, sgn\, \eta(j,i) \, A_i u_j / \|u_j\|_q, \qquad \forall i \in N,$$

where $\vec{r}_i$ equals the $\|\cdot\|_q$-dual vector of r_i.

Part 2

For the perturbed problem we have:

$$\begin{aligned} g(x^k) &= g_u + \sum_{i \in N} \nabla f_i(x^k) \\ &= g_u + \sum_{i \in N} \nu_i \, sgn\, \eta(j,i) \, A_i u_j / \|u_j\|_q, \end{aligned}$$

where g_u corresponds to $g(x^k)$ for the unperturbed problem.
Therefore, using assumptions 2 and 4, we have

$$\begin{aligned} g(x^k) &= \sum_{l \in \overline{M}_\varepsilon(x^k)} A_l u_l + \left\{ \sum_{i \in N} \nu_i \, sgn\, \eta(j,i) \, \{ \sum_{l \in \overline{M}_\varepsilon(x^k)} \eta(l,i) A_l \} \, u_j / \|u_j\|_q \right\} \\ &= \sum_{l \in \overline{M}_\varepsilon(x^k)} A_l \mu_l, \end{aligned} \tag{5.3}$$

where $\mu_l = u_l + \{ \sum_{i \in N} \nu_i \, sgn\, \eta(j,i) \, \eta(l,i) \} \, u_j / \|u_j\|_q$, $\forall l \in \overline{M}_\varepsilon(x^k)$.

Consequently $\| \mu_j \|_q = \gamma \, \| u_j \|_q$ where $\gamma = 1 + \{ \sum_{i \in N} \nu_i \mid \eta(j,i) \mid \} / \| u_j \|_q \geq 1$.

Part 3

Using assumption 4 and the definitions of $r_i(x)$ and h_j^k we have

$$\begin{aligned} r_i(x^k + \lambda h_j^k) &= r_i(x^k) + \lambda A_i^T h_j^k \\ &= \zeta_i \vec{u}_j - \lambda \rho \eta(j,i) \vec{u}_j \\ &= 0, \text{ when } \lambda = \zeta_i / \rho \eta(j,i), \qquad \forall i \in N. \end{aligned}$$

Using (5.3) and the definition of h_j^k we have

$$\begin{aligned} \lambda g(x^k)^T h_j^k &= \lambda \sum_{l \in \overline{M}_\varepsilon(x^k)} \mu_l^T A_l^T h_j^k \\ &= \lambda \mu_j^T A_j^T h_j^k. \\ &= -\lambda \rho \mu_j^T \vec{u}_j \\ &= -\lambda \rho \gamma \| u_j \|_q . \end{aligned}$$

In addition, using the fact that $-\vec{u}_j^T \vec{r}_j \leq \| \vec{u}_j \|_p \| \vec{r}_j \|_q = 1$, where $\vec{r}_j$ is the $\| \cdot \|_q$-dual vector of r_j, ignoring higher-order terms when $j \notin M_0(x^k)$ and following the proof in lemma 4.2, we have

$$f_j(x^k + \lambda h_j^k) - f_j(x^k) \leq \lambda \rho \qquad j \in M_\varepsilon(x^k).$$

Therefore

$$f(x^k + \lambda h_j^k) - f(x^k) \leq \lambda \rho (1 - \gamma \| u_j \|_q) + O(\| \lambda h_j^k \|_2^2)$$

which completes the proof. ▪

One distinct advantage of this perturbation scheme is that a descent direction from the point x^k can be found without computing (for the perturbed problem) the restricted gradient g or the Lagrange vectors μ_l, $l \in \overline{M}_\varepsilon(x^k)$. Exact values for ζ_i, $\forall i \in N$, and additional advantages to this scheme are presented in §7.

6. The Minimization Algorithm. Ignoring the details of implementation (that will be described in §9) we now present a second-order algorithm for solving problem P2.

In this algorithm the user is responsible for setting six parameters: ε, Ψ, τ_f, τ_h, δ and δ_0. While there is no *a priori* optimal choice for these parameters a reasonable choice is often available.

The parameter ε controls the sets M_ε and $\overline{M}_\varepsilon$ which, in turn, determine the subspace in which the current minimization is performed. Recall (see §4.) that our procedure solves a sequence of subproblems of the form (4.1) such that $\{x^k\} \to x^*$ and $\{M_\varepsilon(x^k)\} \to M_0(x^*)$. In the initial stages (when we are most likely far from the solution x^*) a large value of ε (relative to ε=0) is appropriate since smaller values may inhibit the algorithms progress by admitting terms in the subproblems objective that have *near*-singularities in their gradients.

The user-supplied parameter, Ψ, is used to control the accuracy of the current subspace minimization. Since there is little justification for performing this minimization exactly (by setting $\Psi=0$) when far from the solution x^* a relatively large initial value is again appropriate.

Whereas ε and Ψ are dynamically adjusted by this algorithm when appropriate, the remaining four user-supplied parameters are not. These four parameters are used in tests for termination and descent and, for the *well-defined* problem, should reflect (within a few orders of magnitude) the machine precision. For ill-conditioned problems additional conditions and tests should be imposed to ensure convergence to the desired solution. [The choice $\Psi = \tau_h = \tau_f = 0$ corresponds to the situation described in §4.]

The parameter λ used in steps 9,10 and 11 of the following algorithm is determined via the line search described in section 7.

MINIMIZATION ALGORITHM

(1) Choose any $x^1 \in \mathbb{R}^{2n}$ and set $k \leftarrow 1$.

(2) Identify the index sets $M_\varepsilon(x^k)$ and $\overline{M}_\varepsilon(x^k)$.

(3) Compute the restricted gradient g and the restricted Hessian G.

(4) Compute the second-order descent direction h^k.

(5) [Branch if x^k lies outside the neighborhood of any stationary point]

If $\|h^k\|_2 > \Psi$ go to 11.

(6) Compute the Lagrange vectors $\{u_l\}$ by setting $u_l = 0$, $\forall l \in M_\varepsilon(x^k) - \overline{M}_\varepsilon(x^k)$, and solving

$$\underset{\{u_l\}}{\text{minimize}} \ \|g(x^k) - \sum_{l \in \overline{M}_\varepsilon(x^k)} A_l u_l \|_2.$$

(7) [Branch if any Lagrange vector is *out-of-kilter*]

If $\|u_j\|_q > 1$, $j \in M_\varepsilon(x^k)$ go to 10.

(8) [Stop if x^k is a minimizer of P2]

If $\|h^k\|_2 < \tau_h$ and $f_l(x^k) < \tau_f \ \ \forall l \in M_\varepsilon(x^k)$ then stop.

(9) [Attempt to find a stationary point on the current manifold]

Set $\bar{x} \leftarrow x^k + h^k + v^k$,

where v^k is the minimal norm solution to $f_l(x^k + v^k) = 0 \ \ \forall l \in \overline{M}_\varepsilon(x^k)$.

If $f(\bar{x}) - f(x^k) < -\delta_0(\|Z^T g(x^k)\|_2^2 + \sum_{l \in M_\varepsilon(x^k)} f_l(x^k)\,(1 - \|u_l\|_q)\,)$,

where $\delta_0 > 0$ is chosen to guarantee a sufficient decrease in f, then

set $x^{k+1} \leftarrow \bar{x}$, $k \leftarrow k + 1$ and go to 6;

otherwise [do a linesearch and refine activity and stationarity tolerances],

set $d \leftarrow h^k$, $x^{k+1} \leftarrow x^k + \lambda d$, $\varepsilon \leftarrow \frac{1}{2}\varepsilon$, $\Psi \leftarrow \frac{1}{2}\Psi$, $k \leftarrow k + 1$ and go to 2.

(10) [Attempt to reduce f on a new manifold (the *dropping step*)]

If $\|u_j\|_q > 1$, $j \in M_\varepsilon(x^k)$, and $\rho(1-\|u_j\|_q) < -\delta$,
where $\delta > 0$ is chosen to guarantee a sufficient decrease in f, then
identify the index set N,
perturb the residuals $r_i(x^k)$, $\forall i \in N$, and
set $d \leftarrow h_j^k$, $x^{k+1} \leftarrow x^k + \lambda d$, $k \leftarrow k + 1$ and go to 2;
otherwise [refine activity and stationarity tolerances],
set $\varepsilon \leftarrow \frac{1}{2}\varepsilon$, $\Psi \leftarrow \frac{1}{2}\Psi$, $k \leftarrow k + 1$ and go to 2.

(11) [Take a second-order step on the current manifold]

Set $d \leftarrow h^k$, $x^{k+1} \leftarrow x^k + \lambda d$, $k \leftarrow k + 1$ and go to 2.

The strategy suggested by this algorithm is simple and is based on limited numerical testing. Alternate strategies exist that would, no doubt, exhibit convergence properties similar to those possessed by this algorithm.

7. The Line Search. Calamai and Conn [6] and Overton [32] have both noted that any line search algorithm for problem P2 should recognize the possibility of first derivative discontinuities at steps λ where $f_l(\lambda) = f_l(x+\lambda d) = 0$ for some $l \in M$. (Overton also noted that second derivative unboundedness can occur in the neighborhood of these same points.) Fortunately, if the function $f(\lambda) = f(x+\lambda d)$ is nonsmooth the set $\{\lambda_l, l \in M\}$, where $\lambda = \lambda_l$ is the least-squares solution to

$$r_l(\lambda) = r_l(x+\lambda d) = 0, \tag{7.1}$$

includes all the breakpoints of $f(\lambda)$. In addition, the largest element of this set, say γ, is an upper-bound on λ^*, where $\lambda = \lambda^*$ minimizes $f(\lambda)$.

Our interval-reduction procedure uses these breakpoints to partition the interval of uncertainty (initially $[0,\gamma]$) into subintervals in which the function $f(\lambda)$ is smooth. As long as we are in one of these subintervals we apply safeguarded quadratic approximations to estimate λ^* (such techniques exhibit superlinear convergence on well-behaved functions). However, if an approximation of this sort takes us across subinterval boundaries then the resulting estimate is rejected (since the function we are approximating is no longer locally smooth). In this case we take the subinterval boundary crossed (a breakpoint) as the current (unsafeguarded) estimate. For example, if the "best" estimate of λ^* found so far, say β, lies in the subinterval $(\lambda_{l_k}, \lambda_{l_{k+1}})$, where λ_{l_k} and $\lambda_{l_{k+1}}$ are two consecutive breakpoints found using (7.1), then a quadratic approximation to λ^*, say ζ, is accepted if $\zeta \in (\lambda_{l_k}, \lambda_{l_{k+1}})$. However, if $\zeta \notin (\lambda_{l_k}, \lambda_{l_{k+1}})$ then we set the current estimate to λ_{l_k}, if $\lambda_{l_k} \in (\beta, \zeta)$, and to $\lambda_{l_{k+1}}$ otherwise. Safeguards are imposed on these estimates to ensure that the interval of uncertainty, (β,γ), is reduced at every iteration and to ensure that successive estimates are not numerically indistinguishable.

The criteria for terminating this process is based on our convergence requirements. To ensure that the objective function *decreases sufficiently* we insist that

$$f(0) - f(\lambda) \geq -\mu \, \lambda \, f'_+(0), \tag{7.2}$$

where μ is a preassigned scalar in the range $0<\mu\leq\frac{1}{2}$ and $f_+'(0)$ is the right first derivative of $f(0)$, and we control the accuracy of the line search by insisting that

$$\max\{f_-'(\lambda), 0, -f_+'(\lambda)\} \leq -\eta\, f_+'(0), \tag{7.3}$$

where η is a preassigned scalar in the range $\mu<\eta<1$ and $f_-'(\lambda)$ and $f_+'(\lambda)$ are, respectively, the left first and right first derivatives of $f(\lambda)$. However, if the current interval of uncertainty is sufficiently small these tests are ignored and the process is terminated.

The reader should note that test (7.3) differs from the (corresponding) test that is often employed in smooth univariate minimization (see, for example, [20,22,30]) only when λ is a breakpoint. The convexity of f guarantees that the comparison to $-\eta f_+'(0)$ in this test is made using the subdifferential of $f(\lambda)$ having minimal modulus. Elementary analysis shows that at least one point in the interval of uncertainty satisfies both (7.2) and (7.3).

The following notation is used in the line search algorithm that follows:

(i) $\lambda \in (\beta,\gamma)$ implies $\beta < \lambda < \gamma$ or $\gamma < \lambda < \beta$,

(ii) $\varepsilon_M \triangleq$ the smallest machine number satisfying $1 + \varepsilon_M > 1$.

LINE SEARCH ALGORITHM

Step 0: (Initialize)

$$\text{Let } \lambda_l = \begin{cases} -(A_l^T d)^T r_l(0) \,/\, \|A_l^T d\|_2^2 & \{\forall l \mid \|A_l^T d\|_2 \neq 0\} \\ -\infty & \text{otherwise.} \end{cases}$$

Define $\Lambda_0 := \{\lambda_l, l\in M \mid \lambda_l>0 \text{ and } f_l(\lambda_l)\leq\sqrt{\varepsilon_M}\}$.

Set $\beta\leftarrow 0$, $\delta\leftarrow\varepsilon_M f(0)|f_+'(0)|^{-1}$ and $\gamma\leftarrow\min\left\{\dfrac{2f(0)}{\sum\limits_{l\in M}\|A_l^T d\|_p}, \max\{\lambda_l > 0\}\right\}$.

Step 1: (Unsafeguarded)

Set $\zeta\leftarrow\beta - f_+'(\beta)/f''(\beta)$ and define $\Lambda(\beta,\zeta) := \{\lambda_l\in\Lambda_0 \mid \lambda_l\in(\beta,\zeta)\}$.

If $\Lambda(\beta,\zeta) \neq \phi$ set $\lambda\leftarrow\bar{\lambda}$, where $\bar{\lambda}$ $\underset{\lambda\in\Lambda(\beta,\zeta)}{\text{minimizes}}$ $|\beta-\lambda|$; otherwise, set $\lambda\leftarrow\zeta$.

Step 2: (Safeguarded)

If $\lambda \notin [\,\beta, \frac{1}{2}(\beta+\gamma)\,]$ then set $\lambda \leftarrow \frac{1}{2}(\beta+\gamma)$.

```
        If $|\lambda-\beta| < \delta$ and $\beta \neq 0$ then set $\lambda \leftarrow \beta + sgn(\lambda-\beta)\,\delta$.

Step 3:   ( Update the interval of uncertainty )

        If $f(\lambda) > f(\beta)$ then set $\gamma \leftarrow \lambda$ ;

        otherwise,

        if $(\lambda-\beta)\, f'_+(\lambda) > 0$ set $\gamma \leftarrow \beta$.

        Set $\beta \leftarrow \lambda$, $f(\beta) \leftarrow f(\lambda)$, $f'_+(\beta) \leftarrow f'_+(\lambda)$, $f'_-(\beta) \leftarrow f'_-(\lambda)$ and $f''(\beta) \leftarrow f''(\lambda)$.

Step 4:   ( Termination test )

        If [ ( $f(0) - f(\beta) \geq -\mu\, \beta\, f'_+(0)$  and  $\max\{f'_-(\beta), 0, -f'_+(\beta)\} \leq -\eta\, f'_+(0)$ )

            or  ( $|\beta-\gamma| \leq \delta$ and $\beta \neq 0$) ] then STOP with $\lambda = \beta$;

        otherwise go to step 1.
```

[In this algorithm the quantities $f'_+(\vartheta)$, $f'_-(\vartheta)$ and $f''(\vartheta)$ are given by,

$$f'_+(\vartheta) = \sum_{l \in M_1(\vartheta)} d^T A_l \vec{r}_l(\vartheta) + \sum_{l \in M_2(\vartheta)} \| A_l^T d \|_p ,$$

$$f'_-(\vartheta) = \sum_{l \in M_1(\vartheta)} d^T A_l \vec{r}_l(\vartheta) - \sum_{l \in M_2(\vartheta)} \| A_l^T d \|_p ,$$

and

$$f''(\vartheta) = (p-1) \sum_{l \in M_1(\vartheta)} \frac{d^T A_l \, (D_l - \vec{r}_l \vec{r}_l^{\,T}) A_l^T d}{f_l(\vartheta)} \quad ,$$

where $M_1(\vartheta) = \{l \in M \mid f_l(\vartheta) > 0\}$, $M_2(\vartheta) = M - M_1(\vartheta)$ and D_l $(=D_l(\vartheta))$ is the 2 by 2 matrix whose i-th diagonal component is the modulus of the i-th component of $\vec{r}_l$ $(=\vec{r}_l(\vartheta))$ raised to the power $((p-2)/(p-1))$ with $\vec{r}_l$ equaling the $\|\cdot\|_q$-dual vector of r_l (for $p = 2$, $D_l = I_2$).]

The following theorem demonstrates how the solution to (7.1) is particularly appropriate when any of the residual vectors r_l, $l \in M$, have just been perturbed using the scheme outlined in §5.

Theorem 7.1. *If* $d = h_j^k$ *and we assume (without loss of generality) that the set* N *and the scalars* ν_i *(both described in* §*5) satisfy*

$$N = \{1, \ldots, \sigma\} \text{ and } \nu_i = i \mid \eta(j,i) \mid \max\{\rho\delta, 2\pi\varepsilon\}, \quad \forall i \in N,$$

where the scalars $\eta(j,i)$, $i \in N$, *are defined as in assumption 4 of Theorem 5.1 and* $\pi = \max\{|\eta(j,l)|^{-1}, l \in N\}$, *then for the perturbed problem,*

(1) $f_i(\lambda_i) = 0, \ \forall i \in N,$

(2) $\lambda_i \geq \delta, \ \forall i \in N,$

(3) $|\lambda_i - \lambda_l| \geq \delta, \ \forall i,l \in N, \ i \neq l,$ and

(4) $f_i(\lambda) \leq \varepsilon, \ i \in N,$ *implies that* $f_l(\lambda) > \varepsilon, \ \forall l \in N - \{i\},$

where $\lambda_i = \zeta_i / \rho\eta(j,i) \ (= -(A_i^T h_j^k)^T r_i(0) / \, \| A_i^T h_j^k \|_2^2).$

Proof

Part 1 [$\forall i \in N$] - See part 3 of the proof of theorem 5.1.

Part 2 [$\forall i \in N$]

$$\lambda_i = \frac{\zeta_i}{\rho\eta(j,i)} = \frac{\nu_i \, sgn\, \eta(j,i)}{\rho\eta(j,i)} \geq i\delta \geq \delta.$$

Part 3 [$\forall i,l \in N, \ i \neq l$]

$$|\lambda_i - \lambda_l| = \left| \frac{\zeta_i}{\rho\eta(j,i)} - \frac{\zeta_l}{\rho\eta(j,l)} \right| \geq |i-l|\,\delta \geq \delta.$$

Part 4 [proof by contradiction]

If $f_i(\lambda) \leq \varepsilon$ and $f_l(\lambda) \leq \varepsilon, \ i,l \in N, \ i \neq l$ then

$$\left| \frac{\zeta_i}{\eta(j,i)} - \frac{\zeta_l}{\eta(j,l)} \right| \leq \varepsilon \left\{ \frac{1}{|\eta(j,i)|} + \frac{1}{|\eta(j,l)|} \right\} .$$

But for $i,l \in N, \ i \neq l,$

$$\left| \frac{\zeta_i}{\eta(j,i)} - \frac{\zeta_l}{\eta(j,l)} \right| = |i - l| \ \max\{\rho\delta, 2\pi\varepsilon\}$$

$$> \varepsilon \left\{ \frac{1}{|\eta(j,i)|} + \frac{1}{|\eta(j,l)|} \right\}$$

which completes the proof. ▪

In other words, for each of the perturbed $(2n-2)$-dimensional hyperplanes the step length to the hyperplane is easily computed, satisfies the lower bound of acceptability (as defined here), and is computationally unique in $f(\lambda)$ and with respect to possible inclusion in $M_\varepsilon(x+\lambda d)$. Each of these properties has proven beneficial in resolving degeneracy.

8. Convergence Properties. In this section we prove that our method exhibits a global convergence property and that, asymptotically, the method converges at a quadratic rate. Many of the results derived in this section follow directly from the analysis given in [8] and [9]; however, because of our problem's special structure, stronger results are obtained even though weaker assumptions are made.

Global Convergence

The following lemma demonstrates that there is a neighborhood of each minimizer in which the step "$h + v$" is successful.

Lemma 8.1. *If we assume that*

(1) x^* *is any strong minimizer,*

(2) *$M_{\varepsilon_k}(x^k) = M_0(x^*)$ for all k,*

(3) *the matrices A_l, $l \in M_0(x^*)$, are linearly independent, and*

(4) *there exists positive scalars Λ_1 and Λ_2 such that*

$$\Lambda_1 \| w \|_2^2 \le w^T Z^T G(x^k) Z w \le \Lambda_2 \| w \|_2^2 \quad \forall w \ne 0,$$

then there exists a positive constant Δ such that if $\| x^k - x^* \|_2 \le \Delta$ then

$$f(x^k + h^k + v^k) - f(x^k) \le -\delta_0 \{ \| Z^T g(x^k) \|_2^2 + \sum_{l \in M_{\varepsilon_k}(x^k)} f_l(x^k)(1 - \| u_l \|_q) \},$$

where h^k and v^k are defined as they were in §4 and δ_0 is some positive constant.

Proof. (To simplify notation the k superscripts and subscripts are dropped.)

Part 1 - Changes in f_l, $l \in M - M_\varepsilon(x)$.

$$\sum_{l \in M - M_\varepsilon(x)} \{ f_l(x+h+v) - f_l(x) \} = g(x)^T (h+v) + \tfrac{1}{2} (h+v)^T G(x) (h+v) + o(\| h+v \|_2^2)$$

But

$$g(x) = \sum_{l \in M_\varepsilon(x)} A_l u_l + Z w, \quad \text{for some vector } w,$$

$$h = - Z (Z^T G(x) Z)^{-1} Z^T g(x)$$

and

$$A_l^T v = - r_l(x) \qquad \forall l \in M_\varepsilon(x).$$

Therefore

$$\begin{aligned}\sum_{l \in M - M_\varepsilon(x)} \{ f_l(x+h+v) - f_l(x) \} &= \sum_{l \in M_\varepsilon(x)} u_l^T A_l^T v - g(x)^T Z (Z^T G(x) Z)^{-1} Z^T g(x) \\ &\quad + \tfrac{1}{2} h^T G(x) h + \tfrac{1}{2} v^T G(x) v + h^T G(x) v \\ &\quad + o(\| h+v \|_2^2) \\ &= - \sum_{l \in M_\varepsilon(x)} u_l^T r_l(x) - \tfrac{1}{2} g(x)^T Z (Z^T G(x) Z)^{-1} Z^T g(x) \\ &\quad + \tfrac{1}{2} v^T G(x) v + h^T G(x) v + o(\| h+v \|_2^2)\end{aligned}$$

Part 2 - Changes in f_l, $l \in M_\varepsilon(x)$.

$$\begin{aligned}\sum_{l \in M_\varepsilon(x)} \{ f_l(x+h+v) - f_l(x) \} &= \sum_{l \in M_\varepsilon(x)} \{ f_l(x+v) - f_l(x) \} \\ &= - \sum_{l \in M_\varepsilon(x)} f_l(x)\end{aligned}$$

Part 3 - Changes in f

$$\begin{aligned}f(x+h+v) - f(x) &= - \sum_{l \in M_\varepsilon(x)} u_l^T r_l(x) - \tfrac{1}{2} g(x)^T Z (Z^T G(x) Z)^{-1} Z^T g(x) \\ &\quad + \tfrac{1}{2} v^T G(x) v + h^T G(x) v - \sum_{l \in M_\varepsilon(x)} f_l(x)\end{aligned}$$

$$
\begin{aligned}
&+ o(\|h+v\|_2^2) \\
&\le -\tfrac{1}{2}\, g(x)^T Z(Z^T G(x) Z)^{-1} Z^T g(x) \\
&+ \tfrac{1}{2}\, v^T G(x) v + h^T G(x) v - \sum_{l \in M_\varepsilon(x)} f_l(x)(1 - \|u_l\|_q) \\
&+ o(\|h+v\|_2^2)
\end{aligned}
$$

But

$$v = -A(A^T A)^{-1} r(x)$$

where $M_\varepsilon(x) = \{l_1, \ldots, l_t\}$, $A = [A_{l_1} \cdots A_{l_t}]$ and $r(x)^T = [r_{l_1}(x)^T \cdots r_{l_t}(x)^T]$.

Therefore if we define $H_1(x) = (A^T A)^{-1} A^T H_2(x)$ and $H_2(x) = G(x) A (A^T A)^{-1}$ then

$$v^T G(x) v = w_1(x)^T r(x)$$

and

$$h^T G(x) v = w_2(x)^T r(x)$$

where $w_1(x) = H_1(x) r(x)$ and $w_2(x)^T = -h^T H_2(x)$.

Thus

$$
\begin{aligned}
f(x+h+v) - f(x) \le &-\tfrac{1}{2}\, g(x)^T Z(Z^T G(x) Z)^{-1} Z^T g(x) \\
&+ \{ \tfrac{1}{2}\, w_1(x) + w_2(x) \}^T r(x) - \sum_{l \in M_\varepsilon(x)} f_l(x)(1 - \|u_l\|_q) \\
&+ o(\|h+v\|_2^2).
\end{aligned}
$$

If Δ is sufficiently small then, by assumption 1, there exists a positive constant Λ_2 such that

$$g(x)^T Z(Z^T G(x) Z)^{-1} Z^T g(x) \ge \frac{2}{\Lambda_2}\, \|Z^T g(x)\|_2^2.$$

In addition, assumption 2 and the continuity of $w(x) = \frac{1}{2}\, w_1(x) + w_2(x)$ guarantee that $w(x) \to 0$ as $x \to x^*$. Therefore, if Δ is sufficiently small, there exists some $\bar{\delta}_0 > 0$ such that

$$f(x+h+v) - f(x) \le -\bar{\delta}_0 \{ \|Z^T g(x)\|_2^2 + \sum_{l \in M_\varepsilon(x)} f_l(x)(1 - \|u_l\|_q) \} + o(\|h+v\|_2^2)$$

and, since A, Z and $Z^T G(x) Z$ are bounded,

$$
\begin{aligned}
\|h+v\|_2^2 &= \|h\|_2^2 + \|v\|_2^2 \\
&\le L_1 \|Z^T g(x)\|_2^2 + L_2 \sum_{l \in M_\varepsilon(x)} f_l(x)(1 - \|u_l\|_q)
\end{aligned}
$$

where L_1, $L_2 > 0$.

Therefore, for Δ sufficiently small,

$$o(\|h+v\|_2^2) \le \frac{\bar{\delta}_0}{2} \{ \|Z^T g(x)\|_2^2 + \sum_{l \in M_\varepsilon(x)} f_l(x)(1 - \|u_l\|_q) \}$$

Hence, if $\delta_0 = \dfrac{\bar{\delta}_0}{2}$,

$$f(x+h+v)-f(x) \le -\delta_0 \{ \| Z^T g(x) \|_2^2 + \sum_{l \in M_\varepsilon(x)} f_l(x)(1 - \| u_l \|_q) \}$$

which completes the proof. ▪

The following theorem proves that the sequence of iterates $\{x^k\}$ converges to a strong minimizer via the steps $h^k + v^k$.

Theorem 8.1 *If we assume that*

(1) *the sequence of iterates $\{x^k\}$ is generated using the method described in §4 starting from any arbitrary initial point x^0,*

(2) *the matrices A_l, $l \in M_{\varepsilon_k}(x^k)$, are linearly independent,*

(3) *the line search described in §7 is used and the line search condition given by equation (7.2) is satisfied, and*

(4) *any minimizer is a strong minimizer*

then, for all δ_0 sufficiently small,

(1) $\varepsilon_k \not\to 0$,

(2) $\{x^k\} \to x^*$ *(a minimizer), and*

(3) *for k sufficiently large the step $h^k + v^k$ is successful.*

Proof.

Part 1 [We wish to prove that if there exists a positive scalar Δ such that $\| x^k - x \|_2 \le \Delta$ and $M_\varepsilon(x^k) = M_0(x)$, where x is any stationary point that is *not* a minimizer, then

$$\| u_j \|_q > 1 \text{ for some } j \in M_0(x)$$

and

$$(h_j^k)^T \bar{g} < -\delta, \ \ \delta > 0,$$

where u_j and h_j^k are defined as in §4 and $\bar{g} = g(x^k) - A_j u_j / \| u_j \|_q$]

For any particular stationary point x that is *not* a minimizer let Z_i be an orthonormal matrix whose columns span the space $\{h \mid A_l^T h = 0 \ \ \forall l \in M_0(x) - \{i\}\}$. The definition of x guarantees the existence of vectors μ_l, $l \in M_0(x)$, and an index $j \in M_0(x)$ such that

$$g_0(x) = \sum_{l \in M_0(x)} A_l \mu_l \ \text{ and } \ \| \mu_j \|_q > 1.$$

If

$$g = g_0(x) - A_j \mu_j / \| \mu_j \|_q$$

and

$$h = Z_j w,$$

where w is chosen so that $A_j^T h = -\rho \hat{\mu}_j$, $\rho > 0$, where $\hat{\mu}_j$ is the $\| \cdot \|_p$- dual vector of μ_j, then

$$\begin{aligned} h^T g &= -\rho \, (\, \| \mu_j \|_q - 1 \,) \\ &< 0. \end{aligned}$$

The convexity of f guarantees the existence of a positive scalar $\bar{\delta}$ such that $-h^T g > \bar{\delta}$ for all stationary points that are not minimizers. By continuity it follows that if $\|x^k - x\|_2 \le \Delta$ and $M_{\varepsilon_k}(x^k) = M_0(x)$ then $\|u_j\|_q > 1$, and $(h_j^k)^T \bar{g} < -\delta$ ($\delta \, \Delta \, \frac{\bar{\delta}}{2}$).

Part 2 [We wish to prove that the dropping step is successful only a finite number of times.]

(Proof by contradiction)

Without loss of generality we may assume that the dropping step is executed for all k. Then for all k there exists an index $j \in M_{\varepsilon_k}(x^k)$ such that $\|u_j\|_q > 1$. But $\bar{g}^T h_j^k < -\delta$ for the successful execution of the dropping step. The fact that the direction h_j^k is a descent direction at the point x^k along with the line search condition implies that $f(x^k) \to -\infty$, which is contradictory.

Part 3 [We wish to prove that the step $h^k + v^k$ is successful for all k sufficiently large, $\varepsilon_k \not\to 0$, and $\{x^k\} \to x^*$.]

a) Assuming $\varepsilon_k \to 0$ and (consequently) $\beta_k \to 0$.

If $Z^T g(x^{k_i}) \not\to 0$ for any subsequence $\{x^{k_i}\}$ then, for all k sufficiently large, $Z^T g(x^k) \ge \beta_k$. Thus β_k is not reduced and, for all k sufficiently large, $\beta_k \not\to 0$; which is contradictory. Therefore $Z^T g(x^{k_i}) \to 0$ for some subsequence $\{x^{k_i}\}$ with $x^{k_i} \to x$. But $\varepsilon_k \to 0$ and $x^{k_i} \to x$ implies that, for k_i sufficiently large, $M_{\varepsilon_{k_i}}(x^{k_i}) \subseteq M_0(x)$. Moreover, the linear independence assumption and the fact that $Z^T g(x^{k_i}) \to 0$ forces $M_{\varepsilon_{k_i}}(x^{k_i}) = M_0(x)$. Considering Parts 1 and 2, there must be one such subsequence $\{x^{k_i}\}$ that converges to a minimizer x^* with $M_{\varepsilon_{k_i}}(x^{k_i}) = M_0(x^*)$ for k_i sufficiently large. By Lemma 8.1, for k_i sufficiently large, the step $h^{k_i} + v^{k_i}$ is successful. It then follows that $\varepsilon_k \not\to 0$.

b) Assuming $\varepsilon_k \not\to 0$.

For k sufficiently large $\varepsilon_k = \varepsilon > 0$ and $\beta_k = \beta > 0$. It follows that there exists a subsequence $\{x^{k_i}\}$ such that $Z^T g(x^{k_i}) \to 0$, $x^{k_i} \to x$ and, for k_i sufficiently large, $M_{\varepsilon_{k_i}}(x^{k_i}) = M_0(x)$. (If $Z^T g(x^{k_i}) \not\to 0$ for any subsequence $\{x^{k_i}\}$ then, for all k sufficiently large, $\varepsilon_k = \varepsilon$, $\beta_k = \beta$ and $Z^T \nabla f_\varepsilon(x^k) \ge \beta$. The fact that the direction h^k is a descent direction at the point x^k along with the line search condition implies that $f(x^k) \to -\infty$; a contradiction.) Therefore, we have a convergent subsequence $\{x^{k_i}\} \to x$ and $Z^T g(x^{k_i}) \to 0$. Thus, for k_i sufficiently large, $\varepsilon_{k_i} = \varepsilon$, $\beta_{k_i} = \beta$ and $Z^T g(x^{k_i}) < \beta$. But $M_{\varepsilon_{k_i}}(x^{k_i}) = M_0(x^{k_i+1})$ for k_i sufficiently large (due to the boundedness of f the step $h^{k_i} + v^{k_i}$ must eventually be taken and $M_{\varepsilon_{k_i}}(x^{k_i}) = M_0(x^{k_i} + h^{k_i} + v^{k_i}) = M_0(x^{k_i+1})$). Thus, the linear independence assumption and the fact that $Z^T g(x^{k_i}) \to 0$ forces $M_{\varepsilon_{k_i}}(x^{k_i}) = M_0(x)$. Considering Parts 1 and 2 it follows that at least one subsequence $\{x^{k_i}\}$ converges to a minimizer x^* and $M_{\varepsilon_{k_i}}(x^{k_i}) = M_0(x^*)$ for k_i sufficiently large. But the step $h^{k_i} + v^{k_i}$ must be successful for k_i sufficiently large (by Lemma 8.1). ▪

Asymptotic Convergence Rate.

The following theorem proves that under weak conditions the final stages of the method described in §6 converges at a quadratic rate.

Theorem 8.2. *If we assume that*

(1) *the matrices A_l, $l \in M_{\varepsilon_k}(x^k)$, are linearly independent,*

(2) *there exists scalars* Λ_1 *and* Λ_2 $(0 < \Lambda_1 \le \Lambda_2)$ *such that*

$$\Lambda_1 \|w\|_2^2 \le w^T \, (Z^T G(x^k) Z) \, w \le \Lambda_2 \|w\|_2^2 \quad \forall k \quad \forall w,$$

(3) *the sequence of iterates* $\{x^k\} \to x^*$, *where* $x^{k+1} = x^k + h^k + v^k$ *(*h^k *and* v^k *are defined as they were in §3),* $M_{\varepsilon_k}(x^k) = M_0(x^*)$ $\forall k$ *and* x^* *is a strong minimizer,*

then

$$\limsup_{k \to \infty} \frac{\|x^{k+1} - x^*\|_2}{\|x^k - x^*\|_2^2} \le L, \quad L > 0.$$

Proof. Without loss of generality assume that $M_{\varepsilon_k}(x^k) = \{l_1, \ldots, l_s\}$, $s \le n$, and let $A = [A_{l_1} \cdots A_{l_s}]$ and $r(x^k)^T = [r_{l_1}(x^k)^T \cdots r_{l_s}(x^k)^T]$ $\forall k$.

Since the columns of A and Z span $\mathbb{R}^{2n}$ let the vectors y_A^k and y_Z^k satisfy

$$x^k - x^* = A \, y_A^k + Z \, y_Z^k \tag{8.1}$$

for all k.

Part 1 [We wish to prove that $y_A^k = 0$ $\forall k$.]

Using assumption 3 and the definitions of v^k and $r(x^k)$

$$\begin{aligned} x^{k+1} &= x^k + v^k + h^k \\ &= x^k - A \, (A^T A)^{-1} \, r(x^k) + h^k \\ &= x^k - A \, (A^T A)^{-1} \, A^T \, (x^k - x^*) + h^k \end{aligned}$$

Subtracting x^* from both sides and premultiplying by A^T yields

$$\begin{aligned} A^T \, (x^{k+1} - x^*) &= A^T \, (x^k - x^*) - A^T \, (x^k - x^*) \\ &= 0 \end{aligned}$$

and thus, using (8.1) and assumption 1 we obtain the desired result.

Part 2

Using assumption 3 and the definition of h^k

$$\begin{aligned} x^{k+1} &= x^k + h^k + v^k \\ &= x^k - Z \, (Z^T G(x^k) Z)^{-1} \, Z^T g(x^k) + v^k \end{aligned}$$

Since $Z^T g(x^k) = Z^T \{ g(x^*) + G(x) \, (x^k - x^*) \} = Z^T G(x) \, (x^k - x^*)$, where $x = x^* + \vartheta \, (x^k - x^*)$, $0 \le \vartheta \le 1$, we have

$$\begin{aligned} x^{k+1} &= x^k - Z \, (Z^T G(x^k) Z)^{-1} \, Z^T G(x) \, (x^k - x^*) + v^k \\ &= x^k - Z \, (Z^T G(x^k) Z)^{-1} \, Z^T \, [G(x) - G(x^k)] \, (x^k - x^*) \\ &\quad - Z \, (Z^T G(x^k) Z)^{-1} \, Z^T G(x^k) \, (x^k - x^*) + v^k \end{aligned}$$

Substituting (8.1) for the second occurrence of $(x^k - x^*)$ and using the result of Part 1 yields

$$x^{k+1} = x^k - Z \, (Z^T G(x^k) Z)^{-1} \, Z^T \, [G(x) - G(x^k)] \, (x^k - x^*) - Z \, y_Z^k + v^k$$

Subtracting x^* from both sides and premultiplying by Z^T gives

$$Z^T \, (x^{k+1} - x^*) = Z^T \, (x^k - x^*) - (Z^T G(x^k) Z)^{-1} \, Z^T \, [G(x) - G(x^k)] \, (x^k - x^*) - y_Z^k$$

and thus, using (8.1) and Part 1 again, we obtain

$$y_Z^{k+1} = -(Z^T G(x^k) Z)^{-1} Z^T [G(x) - G(x^k)] (x^k - x^*).$$

This result, along with assumption 2, proves that

$$\| y_Z^{k+1} \|_2 \leq L_1 \| G(x) - G(x^k) \|_2 \| x^k - x^* \|_2$$

for some $L_1 > 0$ and the Lipschitz continuity of G and the definition of x guarantee that for k sufficiently large

$$\| G(x) - G(x^k) \|_2 \leq \overline{L}_1 \| x^k - x^* \|_2$$

for some $\overline{L}_1 > 0$.

Therefore for k sufficiently large $\| y_Z^{k+1} \|_2 \leq L \| x^k - x^* \|_2^2$, $L = L_1 \overline{L}_1 > 0$.

Since $\| y_Z^{k+1} \|_2 = \| x^{k+1} - x^* \|_2$ our result follows. ▪

9. Implementation

In this section the numerical aspects of our algorithm are examined. Most of the topics discussed involve implementation procedures which could be dealt with using classical techniques (see, for example, [21]); however, the special structure of problem P1 can be exploited to provide much more efficient methods. Many of these methods are described using elementary graph theory. The reader who is unfamiliar with the associated terminology is referred to Deo [13] and Minieka [28].

For the purpose of the discussions that follow we assume that for $l \in M$ and e_j denoting the j-th column of I_n $(j=1,\dots,n)$, the scalars $j_l \in \{1,\dots,n\}$ and $k_l \in \{1,\dots,n+1\} - \{j_l\}$ are chosen so that

$$A_l = \alpha_l \, e(l) \otimes I_2 = \begin{cases} \alpha_l (e_{j_l} - e_{k_l}) \otimes I_2 & \text{when } \alpha_l \text{ corresponds to } v_{jk} \\ \\ \alpha_l e_{j_l} \otimes I_2 & \text{when } \alpha_l \text{ corresponds to } w_{ji} \end{cases}$$

where k_l is set to $n+1$ when α_l corresponds to w_{ji}. We also assume that when α_l corresponds to v_{jk} then the vector $e(l)$ corresponds to an edge E_l joining vertex V_{j_l} to vertex V_{k_l}. (This edge represents the interaction of new facility j_l with new facility k_l). Similarly, when α_l corresponds to w_{ji} the vector $e(l)$ corresponds to an edge E_l joining vertex V_{j_l} to vertex V_{n+1}. (This edge represents the interaction of new facility j_l with some (fixed) existing facility represented by vertex V_{n+1}).

a) Identifying $\overline{M}_\varepsilon$

Consider the digraph $G(V,E)$ consisting of the vertex set $V = \{V_1, \dots, V_{n+1}\}$ and edge set $E = \{E_l, l \in M_\varepsilon(x)\}$ and let the vertex set $\overline{V}$ contain all the isolated vertices of $G(V,E)$. [With respect to problem 4.1, the vertices in $\overline{V}$ correspond (in index) to new facilities whose movements are *currently* unconstrained and the vertices in each component of $G(V-\overline{V},E)$ correspond (in index) to new facilities whose movements are *currently* constrained.] The following lemma suggests one method for identifying $\overline{M}_\varepsilon(x)$:

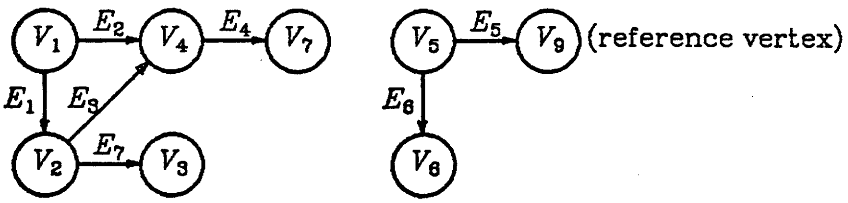
Lemma 9.1. *Suppose that the s edges in the set* $\overline{E} = \{E_{l_1} \cdots E_{l_s}\} \subset E$ *identify a spanning forest of* $G(V-\overline{V},E)$. *If* $\overline{A} = [e(l_1) \cdots e(l_s)]$ *then* $Rank(\overline{A}) = s$ *and* $A_l \in span\{A_{l_1} \cdots A_{l_s}\}$ *for all* $l \in M_\varepsilon(x)$.

Proof. We can think of $\bar{A}$ as the *reduced* incidence matrix of the digraph $G(V-\bar{V},\bar{E})$ having reference vertex V_{n+1}. It is a well known fact that $Rank(\bar{A}) = \tau - \kappa$, where τ and κ equal the number of vertices and components in the digraph respectively. But a spanning forest with s edges and κ components has $\tau = s + \kappa$ vertices. Therefore $Rank(\bar{A}) = s$.

Since the set $\bar{E}$ identifies a spanning forest of $G(V-\bar{V},E)$, adding edge $E_l \in E - \bar{E}$ to $G(V-\bar{V},\bar{E})$ does not change the number of vertices or components. In addition, the matrix $[\bar{A} \vdots e(l)]$ is the reduced incidence matrix of the digraph $G(V-\bar{V},\bar{E}+E_l)$ having reference vertex V_{n+1}. Consequently $Rank([\bar{A} \vdots e(l)]) = s$ and $e(l) \in span\{e(l_1), \ldots, e(l_s)\}$. Therefore since $A_l = \alpha_l \; e(l) \otimes I_2$ it follows that $A_l \in span\{A_{l_1}, \ldots, A_{l_s}\}$. ■

The following figure represents the graph $G(V,E)$ when n=8, $M_\varepsilon(x)$= {1,2,3,4,5,6,7}, $\{j_1, \ldots, j_7\}$={1,1,2,4,5,5,2}, and $\{k_1, \ldots, k_7\}$={2,4,4,7,9,6,3}:

FIGURE 9.1



In this example $\bar{V} = \{V_8\}$ and if we eliminate V_8 from $G(V,E)$ we are left with $G(V-\bar{V},E)$. If we then eliminate either edge E_1 or edge E_2 or edge E_3 from $G(V-\bar{V},E)$ we are left with one of the three possible choices for $G(V-\bar{V},\bar{E})$. Each of these representations defines a spanning forest on $G(V-\bar{V},E)$ and the indices of the edges remaining in $G(V-\bar{V},\bar{E})$ identify (the corresponding) $\bar{M}_\varepsilon(x)$.

Lemma 9.1 demonstrates that edges in $\bar{E}$ that identify a spanning forest of $G(V-\bar{V},E)$ also identify (the indices of) $\bar{M}_\varepsilon(x)$. A process for finding a spanning forest of $G(V-\bar{V},E)$ is simply stated:

At each stage a new edge $E_l \in E$ is examined to see if either or both of its end vertices appear in any tree formed so far. One of the following four mutually exclusive actions is taken:

(a) if both vertices are in the same tree then edge E_l is discarded,

(b) if neither vertex is in any tree then a new tree is started with edge E_l,

(c) if only one of the two vertices is in a tree, edge E_l is added to that tree, and

(d) if both vertices are in different trees, edge E_l is added and the two trees are joined to make one.

The following algorithm finds $\bar{M}_\varepsilon$ (= $\bar{M}_\varepsilon(x)$) and constructs a vector T (= $T(x)$) of length $n+1$ such that $t_{n+1} = -1$ and

$$t_i = \begin{cases} -1 & \text{if } V_i \text{ is in the same tree as } V_{n+1} \\ 0 & \text{if } V_i \text{ is isolated} \\ c & \text{if } V_i \text{ is in the tree labeled } c \quad (1 \le c \le \kappa). \end{cases}$$

Here κ equals the number of components in $G(V-\overline{V},E)$ if $V_{n+1} \in \overline{V}$ and equals one less than the number of components in $G(V-\overline{V},E)$ if $V_{n+1} \notin \overline{V}$. [The significance of these values with respect to the constraints of subproblem 4.1 is as follows: if $t_i=-1$ the location of new facility i is currently fixed because of this facilities proximity and interaction with some existing facility whereas if $t_i=0$ the movement of new facility i is currently unconstrained, and if $t_i=c$ $(1 \leq c \leq \kappa)$ the movement of new facility i is constrained to coincide with the movement of all new facilities in the set $\{j \mid t_j=c\}$ due to the current proximity and interaction of these facilities. For the example previously described and illustrated in Figure 9.1 we have $T = \{t_1, \ldots, t_9\} = \{1,1,1,1,-1,-1,1,0,-1\}$.]

Step 3 in the following algorithm corresponds to the four actions previously discussed:

ALGORITHM 1

```
(1)  Set $\underline{M} \leftarrow M_\varepsilon$, $\overline{M}_\varepsilon \leftarrow \phi$, $\kappa \leftarrow 0$, $i \leftarrow 0$, $t_j \leftarrow 0$ $(j=1,\ldots,n)$ and $t_{n+1} = -1$.
(2)  If $\underline{M} = \phi$ or $i = n+1-\kappa$ set $s \leftarrow i$ and STOP;
     otherwise,
     choose any index $l \in \underline{M}$, set $\underline{M} \leftarrow \underline{M}-\{l\}$, $j \leftarrow j_l$ and $k \leftarrow k_l$.
(3a) If $t_j = t_k \neq 0$ go to 2; otherwise,
(3b) if $t_j = t_k = 0$ set $\kappa \leftarrow \kappa+1$, $t_j \leftarrow \kappa$, $t_k \leftarrow \kappa$ and go to 4; otherwise,
(3c) if $t_j t_k = 0$ set $t_j \leftarrow t_k$ if $t_j = 0$ else set $t_k \leftarrow t_j$ and go to 4; otherwise,
(3d) if $t_j \neq t_k$ set $\underline{\kappa} \leftarrow \min\{t_j,t_k\}$ and $\overline{\kappa} \leftarrow \max\{t_j,t_k\}$.
             [merge]   for $l=1,\ldots,n$ if $t_l = \overline{\kappa}$ set $t_l \leftarrow \underline{\kappa}$
            [relabel]  for $l=1,\ldots,n$ if $t_l = \kappa$ set $t_l \leftarrow \overline{\kappa}$
     Set $\kappa \leftarrow \kappa-1$ and go to 4.
(4)  Set $i \leftarrow i+1$, $l_i \leftarrow l$, $\overline{M}_\varepsilon \leftarrow \overline{M}_\varepsilon+\{l_i\}$ and go to 2.
```

The reader should note that this is a *greedy* algorithm since each edge in the set E is examined, at most, once (step (2)). In addition, the time bound for the execution of all steps except (3d) is proportional to the number of edges examined. Aho et al. [1] give details of two algorithms called **Union** and **Find** that can perform step (3d) very efficiently (in almost linear time). The reader should also note that identifying M_ε $(= M_\varepsilon(x))$ in step (1) is trivial and that at the completion of this algorithm $t_{j_l} = t_{k_l} \neq 0$ for all $l \in \overline{M}_\varepsilon$.

b) Maintaining $\overline{M}_\varepsilon$ and T.

If the value of ε is reduced we use algorithm 1 to recompute $\overline{M}_\varepsilon$ and T. If the value of ε is not reduced there are three possible situations that can occur in progressing from the point x^k to the point x^{k+1}:

(1) $M_\varepsilon(x^{k+1}) = M_\varepsilon(x^k)$,

(2) $M_\varepsilon(x^{k+1}) = M_\varepsilon(x^k) + L$, for some nonempty index set L,

(3) an index is dropped from $\overline{M}_\varepsilon(x^k)$.

In the first situation we make no change to $\overline{M}_\varepsilon$ or T. In the second situation we simply repeat steps (2) through (4) of Algorithm 1 with $\underline{M}$ and i initialized to $\underline{L}$ and s respectively. In the last situation the process of dropping an index from $\overline{M}_\varepsilon$ is equivalent to removing an edge from $G(V-\overline{V},\overline{E})$ - the spanning forest of $G(V-\overline{V},E)$. The following algorithm updates $\overline{M}_\varepsilon$ and T assuming edge E_l (index l) is to be dropped:

ALGORITHM 2

```
(1) Set $\overline{M}_\varepsilon \leftarrow \overline{M}_\varepsilon - \{l\}$, $\underline{M} \leftarrow \{i \in \overline{M}_\varepsilon \mid t_{j_i} = t_{j_l}\}$, $\kappa \leftarrow \kappa - 1$ when $t_{j_l} \neq -1$, $i \leftarrow s-1$, and $t_k = 0$
    when $t_k = t_{j_l}$ $(k = 1, \ldots, n)$.
(2) Same as (2) of Algorithm 1.
(3) Same as (3a) through (3d) of Algorithm 1.
(4) Go to 2.
```

For the remainder of this section we assume that $\overline{M}_\varepsilon = \{l_1, \ldots, l_s\}$, $\bar{A} = [e(l_1) \ldots e(l_s)]$, $T = [t_1 \cdots t_{n+1}]$ and $\kappa = \max\{t_i \in T\}$.

c) Using Z

The next lemma demonstrates how Z, an orthonormal matrix whose columns span the space $\{h \mid A_l^T h = 0, \forall l \in M_\varepsilon(x)\}$, could be constructed from a matrix $\bar{Z}$ using the information in T. This matrix $\bar{Z}$ is associated with $G(V-\bar{V}, \bar{E})$ in the following manner:

(a) there is a zero row of $\bar{Z}$ corresponding (in index) to each vertex of $G(V-\bar{V}, \bar{E})$ in the same tree as vertex V_{n+1} when $V_{n+1} \notin \bar{V}$,

(b) there is a unique column of $\bar{Z}$ for each vertex in $\bar{V}$ (other than V_{n+1} if $V_{n+1} \in \bar{V}$) and the only nonzero entry in any such column is in the row corresponding (in index) to the associated vertex in $\bar{V}$, and

(c) there is a unique column of $\bar{Z}$ for each tree in $G(V-\bar{V}, \bar{E})$ (except the one containing V_{n+1} when $V_{n+1} \notin \bar{V}$) and the nonzero entries in any such column are in the rows corresponding (in index) to the vertices in the associated tree.

The actions taken in step 3 of the following lemma's algorithm correspond to these three cases. In the statement of this algorithm $\bar{e}_j$ denotes the j^{th} row of I_{n-s} and n_c equals the cardinality of the set $\{\, i \mid t_i = c \,\}$ (the number of vertices in the tree labeled c).

Lemma 9.2. *If we construct an orthonormal matrix* $\bar{Z} = \begin{bmatrix} \bar{z}_{1\bullet} \\ \vdots \\ \bar{z}_{n\bullet} \end{bmatrix}$ *as follows:*

ALGORITHM 3

```
(1) Set $i \leftarrow 0$ and $k \leftarrow \kappa$.
(2) If $i = n$ then STOP; otherwise set $i \leftarrow i + 1$.
(3a) If $t_i = -1$, set $\bar{z}_{i\bullet} \leftarrow 0$ and go to 2; otherwise,
(3b) if $t_i = 0$, set $k \leftarrow k + 1$, $\bar{z}_{i\bullet} \leftarrow \bar{e}_k$ and go to 2; otherwise,
(3c) if $t_i = c$, $1 \le c \le \kappa$, set $\bar{z}_{i\bullet} \leftarrow \sqrt{1/n_c}\ \bar{e}_c$ and go to 2,
```

then $Z = \bar{Z} \otimes I_2$ *is an orthogonal matrix whose columns span the space* $\{h \mid A_l^T h = 0, \forall l \in M_\varepsilon(x)\}$.

Proof.

Part 1 [We first prove that $\overline{A}^T\overline{Z}=0_{s\times(n-s)}$.]

Since $\overline{A} = [e(l_1)\cdots e(l_s)]$ we must prove that $e(l)^T\overline{Z} = 0$ for all $l\in\overline{M}_\varepsilon$. Now for $l\in\overline{M}_\varepsilon$, $j=j_l$ and $k=k_l$, we have $t_j=t_k\neq 0$ and therefore (by construction) $\overline{z}_{j\bullet}=\overline{z}_{k\bullet}$. Consequently, $e(l)^T\overline{Z} = 0$ for all $l\in\overline{M}_\varepsilon(x)$.

Part 2 [We now prove that $\overline{Z}^T\overline{Z}=I_{n-s}$.]

Since $\overline{Z}$ has (at most) one nonzero entry per row we have $\overline{z}_{ki}\overline{z}_{kj}=0$ $(k=1,\ldots,n)$ when $i\neq j$. Therefore $[\overline{Z}^T\overline{Z}]_{ij} = 0$ when $i\neq j$. In addition, when $c\in\{1,\ldots,\kappa\}$ there are n_c nonzero entries in column c all equaling $\sqrt{1/n_c}$ and, when $c\in\{\kappa+1,\ldots,s\}$, there is one nonzero entry in column c equaling 1. Therefore $[\overline{Z}^T\overline{Z}]_{ii} = 1$ for $i=1,\ldots,s$.

Part 3 [We finally prove that $Z=\overline{Z}\otimes I_2$ has the stated properties.]

If $Z = \overline{Z}\otimes I_2$ then, for $l\in\overline{M}_\varepsilon(x)$,

$$\begin{aligned} A_l^T Z &= \alpha_l(e(l)\otimes I_2)^T(\overline{Z}\otimes I_2) \\ &= \alpha_l(e(l)^T\overline{Z})\otimes I_2 \\ &= 0, \end{aligned}$$

and

$$\begin{aligned} Z^T Z &= (\overline{Z}^T\overline{Z})\otimes I_2 \\ &= I_{n-s}\otimes I_2 \\ &= I_{2(n-s)}. \end{aligned}$$

Therefore $Z = \overline{Z}\otimes I_2$ is an orthogonal matrix whose columns span the space $\{h\,|\,A_l^T h = 0, \forall l\in M_\varepsilon(x)\}$. ▪

For the example previously described (and illustrated in Figure 9.1) $\overline{Z}^T$ would equal

$$\overline{Z}^T = \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 0 & 0 & \frac{1}{\sqrt{5}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

It should be emphasized however that the matrix Z (or, for that matter, Z_j) should never be computed (or stored) since the products Z^Tw and Zy are easily computed (for any vectors w and y) without forming Z.

If $w^T = [w_1^T\cdots w_n^T]$ with $w_i\in R^2$ $(i=1,\ldots,n)$ and $y^T = [y_1^T\cdots y_{n-s}^T]$ with $y_i\in R^2$ $(i=1,\ldots,n-s)$ then the following algorithms demonstrate how these products are formed:

ALGORITHM 4a [forming the product $w=Zy$]

```
(1)  Set $i\leftarrow 0$ and $k\leftarrow\kappa$.
(2)  If $i=n$ then STOP; otherwise set $i\leftarrow i+1$.
(3a) If $t_i=-1$, set $w_i\leftarrow 0$ and go to 2; otherwise,
(3b) if $t_i=0$, set $k\leftarrow k+1$, $w_i\leftarrow y_k$ and go to 2; otherwise,
```

(3c) if $t_i = c$, $1 \le c \le \kappa$, set $w_i \leftarrow y_c / \sqrt{n_c}$ and go to 2.

ALGORITHM 4b [forming the product $y = Z^T w$]

```
(1)  Set $i \leftarrow 0$, $k \leftarrow \kappa$ and $y \leftarrow 0$.
(2)  If $i = n$ then STOP; otherwise set $i \leftarrow i+1$.
(3a) If $t_i = -1$ then go to 2; otherwise,
(3b) if $t_i = 0$, set $k \leftarrow k+1$, $y_k \leftarrow y_k + w_i$ and go to 2; otherwise,
(3c) if $t_i = c$, $1 \le c \le \kappa$, set $y_c \leftarrow y_c + w_i / \sqrt{n_c}$ and go to 2.
```

Both these computations require approximately $2n$ operations *in the worst case*.

d) Computing the Lagrange vectors

Since the matrix $I - ZZ^T$ is an orthogonal projector onto the space spanned by the columns of A_l, $l \in \overline{M}_\varepsilon$, the solution to the system

$$\sum_{l \in \overline{M}_\varepsilon} A_l u_l = c\,, \tag{9.1}$$

where $c = (I - ZZ^T)\, g$, is unique and satisfies our requirement for the Lagrange vectors (ie. minimizes $\| g - \sum_{l \in \overline{M}_\varepsilon} A_l u_l \|_2$).

Fortunately, the solution to (9.1) can be found very efficiently. For example, if edge E_2 were deleted (from Figure 9.1) in constructing $\overline{G} = G(V - \overline{V}, \overline{E})$ from $G(V - \overline{V}, E)$ we would have $\overline{M}_\varepsilon = \{1,3,4,5,6,7\}$ and (9.1) could be written as

$$\begin{bmatrix} \alpha_1 I_2 & 0 & 0 & 0 & 0 & 0 \\ -\alpha_1 I_2 & \alpha_3 I_2 & 0 & 0 & 0 & \alpha_7 I_2 \\ 0 & 0 & 0 & 0 & 0 & -\alpha_7 I_2 \\ 0 & -\alpha_3 I_2 & \alpha_4 I_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_5 I_2 & \alpha_6 I_2 & 0 \\ 0 & 0 & 0 & 0 & -\alpha_6 I_2 & 0 \\ 0 & 0 & -\alpha_4 I_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \end{bmatrix}$$

It is evident that this system can (for example) be solved sequentially for u_1, u_4, u_6, u_5, u_7, and u_3 using row blocks 1,7,6,5,3 and 2 (or 4) respectively. These row blocks correspond (in index) to a pendant vertex of the edges E_1, E_4, E_6, E_5, E_7, and E_3 in the graphs $G_1 \blacktriangle G$, $G_4 \blacktriangle G_1 - E_1$, $G_6 \blacktriangle G_4 - E_4$, $G_5 \blacktriangle G_6 - E_6$, $G_7 \blacktriangle G_5 - E_5$ and $G_3 \blacktriangle G_7 - E_7$ respectively.

Stated more generally, consider the spanning forest described by $G \blacktriangle G(V - \overline{V}, \overline{E})$ where the edges $\overline{E} = \{E_{l_1}, \ldots, E_{l_s}\}$ are directly associated with the vectors $e(l)$, $l \in \{l_1, \ldots, l_s\}$ $(= \overline{M}_\varepsilon)$. Since $\overline{G}$ describes a spanning forest there are at least two pendant vertices in G. Suppose edge $E_i \in \overline{E}$ is incident on a pendant vertex other than V_{n+1}. Then (by definition) vertex V_{j_i} or vertex V_{k_i} is a pendant vertex. Now, without loss of generality, suppose vertex V_{j_i} $(\ne V_{n+1})$ is the pendant vertex. Since edge E_i is the only edge in $\overline{E}$ incident on vertex V_{j_i} the vector $e(i)$ is the only vector in the set $\{e(l), l\in\overline{M}_\varepsilon\}$ with a nonzero in row j_i. Consequently A_i $(=\alpha_i e(i) \otimes I_2)$ is the only matrix in the set $\{A_l, l\in\overline{M}_\varepsilon\}$ with nonzero entries in *row block* j_i (where each block consists of 2 rows). Since these nonzeros occur as diagonal entries (of row block j_i) we can easily compute the vector u_i. This leaves the following reduced system to solve:

$$\sum_{l\in\overline{M}_\varepsilon-\{i\}} A_l u_l = c - A_i u_i. \tag{9.2}$$

Now if we delete vertex V_{j_i} (and edge E_i) from G we are left with a graph that describes a (reduced) spanning forest. In other words, this process can be repeated to compute all of the Lagrange vectors.

If we assume, without loss of generality, that the indices $l_1, \dots, l_s$ of $\overline{M}_\varepsilon$ are ordered so that $d(E_{l_1}) \le d(E_{l_2}) \le \cdots \le d(E_{l_s})$, where the degree of edge E_l, denoted $d(E_l)$, is defined as follows:

$$d(E_l) = \begin{cases} d(V_{j_l}) & \text{if } V_{k_l} = V_{n+1} \\ \min\{d(V_{j_l}), d(V_{k_l})\} & \text{otherwise,} \end{cases}$$

with $d(V_{j_l})$ and $d(V_{k_l})$ equaling the degree of vertices V_{j_l} and V_{k_l} respectively, then the following algorithm can be used to solve (9.1):

ALGORITHM 5

```
(1) Set $i \leftarrow 0$ and $c \leftarrow (I-ZZ^T)\,g$ with $c^T = [c_1^T \cdots c_n^T]$ and $c_i \in R^2$ $(i=1,\dots,n)$.
(2) If $i = s$ then STOP;
    otherwise,
    set $i \leftarrow i+1$, $l \leftarrow l_i$, $j \leftarrow j_l$ and $k \leftarrow k_l$. .
(3) If $k = n+1$ or $d(V_j) \le d(V_k)$ set $u_l \leftarrow c_j/\alpha_l$, $c_k \leftarrow c_k + c_j$ and go to 2;
    otherwise,
    set $u_l \leftarrow -c_k/\alpha_l$, $c_j \leftarrow c_j + c_k$ and go to 2.
```

This algorithm corresponds to a (leaf-by-leaf) traversal of the spanning forest described by G. The number of operations used in this algorithm is, in the worst case, equal to $2(s+2n)$.

e) Computing the refinement step

In our minimization algorithm we sometimes take a *refinement step* $x+v$ where v is the minimal norm solution to

$$r_l(x+v) = 0, \qquad \forall l \in \overline{M}_\varepsilon(x). \tag{9.3}$$

Fortunately, this solution can be found very easily. For example, if edge E_2 were deleted (from Figure 9.1) in constructing $G=G(V-\overline{V},\overline{E})$ from $G(V-\overline{V},E)$ we would have $\overline{M}_\varepsilon = \{1,3,4,5,6,7\}$ and (9.3) could be written as

$$\begin{bmatrix} \alpha_1 I_2 & 0 & 0 & 0 & 0 & 0 \\ -\alpha_1 I_2 & \alpha_3 I_2 & 0 & 0 & 0 & \alpha_7 I_2 \\ 0 & 0 & 0 & 0 & 0 & -\alpha_7 I_2 \\ 0 & -\alpha_3 I_2 & \alpha_4 I_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_5 I_2 & \alpha_6 I_2 & 0 \\ 0 & 0 & 0 & 0 & -\alpha_6 I_2 & 0 \\ 0 & 0 & -\alpha_4 I_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \begin{bmatrix} (x+v)_1 \\ (x+v)_2 \\ (x+v)_3 \\ (x+v)_4 \\ (x+v)_5 \\ (x+v)_6 \\ (x+v)_7 \\ (x+v)_8 \end{bmatrix} - \begin{bmatrix} b_1 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $b_1=b_3=b_4=b_6=b_7=0$. Since v_8 is completely unconstrained in this system we would clearly set $(v+x)_8 = x_8$. This component corresponds (in index) to the only vertex in $\bar{V}$. In addition, the equations involving v_5 and v_6 yield $(x+v)_5 = (x+v)_6 = b_5/\alpha_5$. These components correspond (in index) to the vertices in the component (tree) of G that includes vertex V_{n+1}. Finally, the equations involving the remaining components can be written as $(x+v)_1 = (x+v)_2 = (x+v)_3 = (x+v)_4 = (x+v)_7$. But the solution to these equations, that produces the minimal norm in v, is obtained by minimizing $\sum_{l\in L}(c-x_l)^2$ where $L=\{1,2,3,4,7\}$ and $(x-v)_l = c$ $\forall l \in L$. We therefore set $(x+v)_l = \frac{1}{5}\sum_{l\in L} x_l$ for all $l\in L$. These components correspond (in index) to the vertices in the remaining component (tree) of G. The ideas illustrated by this example are now made formal.

The following lemma shows that the refinement step $x+v$ can be written as

$$\begin{aligned} x+v &= x - (I-ZZ^T)(x-w) \\ &= w + ZZ^T(x-w) \end{aligned}$$

where $w\in R^{2n}$ is any solution to the full-rank system

$$A_l^T w = b_l, \qquad \forall l \in \overline{M}_\varepsilon(x). \tag{9.4}$$

Lemma 9.3. *If $w\in R^{2n}$ satisfies (9.4) then the minimal norm solution to (9.3) is given by $v = -(I-ZZ^T)(x-w)$.*

Proof. For $v = -(I-ZZ^T)(x-w)$ and $l\in\overline{M}_\varepsilon(x)$ we have

$$\begin{aligned} r_l(x+v) &= A_l^T(x+v)-b_l \\ &= A_l^T w - b_l \\ &= 0. \end{aligned}$$

In addition, the minimal norm solution of (9.3) is also the minimal norm solution of the following full-rank system:

$$A_l^T v = -A_l^T(x-w), \qquad \forall l\in\overline{M}_\varepsilon(x), \tag{9.5}$$

where w satisfies (9.4). But any solution v to (9.3) can be written as $v = v_Z + v_A$ where v_Z lies entirely in the space spanned by the columns of Z and v_A lies entirely in the space spanned by the columns of A_l, $\forall l\in\overline{M}_\varepsilon(x)$. Since $I-ZZ^T$ is an orthogonal projector onto the space spanned by the columns of A_l, $l\in\overline{M}_\varepsilon(x)$, the minimal norm solution to (9.5), and consequently (9.3), is given by $v = v_A = -(I-ZZ^T)(x-w)$ where w satisfies (9.4). ▪

In the next lemma we prove that there exists a unique solution to (9.4), $\bar{w} \in R^{2n}$ with $Z^T\bar{w} = 0$, which allows us to compute the refinement step as

$$x + v = \bar{w} + \bar{x}$$

where $\bar{x} = ZZ^Tx$. In the statement of this lemma T_l denotes the set $\{i \,|\, t_i = l\}$ ($l=-1,\ldots,\kappa$), n_c denotes the cardinality of T_c ($c=1,\ldots,\kappa$) and $\bar{l}_i$ is chosen from $\bar{M}_\varepsilon(x)$ so that $E_{\bar{l}_i}$ is the edge of G incident on vertex V_{n+1} in the (unique) path from vertex V_i to V_{n+1} when $i \in T_{-1}$.

Lemma 9.4. *If we construct the vector $\bar{w}$, having components $\bar{w}_i \in R^2$ ($i=1,\ldots,n$), and the vector $\bar{x}$, having components $\bar{x}_i \in R_2$ ($i=1,\ldots,n$), using the following algorithm:*

ALGORITHM 6

```
(1) Set $\bar{w} \leftarrow 0$ and $\bar{x} \leftarrow 0$.
(2) If $V_{n+1} \in \bar{V}$ go to 3;
    otherwise,
    set $\bar{w}_i \leftarrow b_{\bar{l}_i} / \alpha_{\bar{l}_i}$ for all $i \in T_{-1}$.
(3) For $c = 1,\ldots,\kappa$, set $\bar{x}_i \leftarrow n_c^{-1} \sum_{j \in T_c} x_j$ for all $i \in T_c$.
(4) Set $\bar{x}_i = x_i$ for all $i \in T_0$.
```

then the vector $\bar{w}$ satisfies (9.4) and lies entirely in the space spanned by the columns of A_l, $l \in \bar{M}_\varepsilon(x)$, the vector $\bar{x}$ equals ZZ^Tx, and

$$(x+v)_i = \begin{cases} \bar{w}_i & i \in T_{-1} \\ \bar{x}_i & \text{otherwise.} \end{cases}$$

Proof.

Part 1 [We first show that $\bar{w}$ satisfies (9.4).]

Assume that edge E_l, $l \in \bar{M}_\varepsilon(x)$, *is not* incident on vertex V_{n+1} in G. Then, by construction, $k_l \neq n+1$, $b_l = 0$ and $\bar{w}_{j_l} = \bar{w}_{k_l}$. Consequently, $A_l^T\bar{w} = \alpha_l(e(l) \otimes I_2)^T\bar{w} = \alpha_l(\bar{w}_{j_l} - \bar{w}_{k_l}) = b_l$ (= 0). On the other hand, if edge E_l, $l \in \bar{M}_\varepsilon(x)$, *is* incident on vertex V_{n+1} in G then, by construction, $k_l = n+1$, $\bar{w}_{j_l} = b_l / \alpha_l$ and $A_l^T\bar{w} = \alpha_l(e(l) \otimes I_2)^T\bar{w} = \alpha_l \bar{w}_{j_l} = b_l$.

Part 2 [We prove the second proposition by showing that $\bar{w}^TZ = 0$.]

Using the representation of Z given in lemma 9.2 we have

$$\bar{w}^TZ = \sum_{i=1}^{n} \bar{w}_i^T(\bar{z}_{i\bullet} \otimes I_2) = 0,$$

since $\bar{w}_i = 0$ when $i \notin T_{-1}$ and $\bar{z}_{i\bullet} = 0$ when $i \in T_{-1}$.

Part 3 [We now prove that $\bar{x}=ZZ^Tx$.]

The construction of $\bar{Z}$ given in lemma 9.2 guarantees that

$$\bar{z}_{i\bullet}\bar{z}_{j\bullet}^T = \begin{cases} 1 & \text{if } i\in T_0 \text{ and } j=i \\ n_c^{-1} & \text{if } i\in T_c \text{ and } j\in T_c \text{ with } 1\le c\le\kappa \\ 0 & \text{otherwise.} \end{cases}$$

Consequently, if $[ZZ^Tx]_i \in R^2$ represents the i-th component vector of ZZ^Tx $(i=1,\ldots,n)$, then

$$[ZZ^Tx]_i = \bar{z}_{i\bullet}(\bar{Z}^T \otimes I_2)\, x = \begin{cases} x_i & \text{if } i\in T_0 \\ n_c^{-1}\sum\limits_{j\in T_c} x_j & \text{if } i\in T_c \text{ with } 1\le c\le\kappa \\ 0 & \text{if } i\in T_{-1}. \end{cases}$$

Therefore $[ZZ^Tx]_i = \bar{x}_i$ $(i=1,\ldots,n)$.

Part 4

For $w = \bar{w}$ and $\bar{x} = ZZ^Tx$ we have (as a consequence of lemma 9.3)

$$\begin{aligned} x+v &= \bar{w} + ZZ^T(x-\bar{w}) \\ &= \bar{w} + \bar{x} \end{aligned}$$

and the final proposition follows from the disjoint property of the sets T_l $(l=-1,\ldots,\kappa)$. ▪

The construction of $\bar{w}$ and $\bar{x}$ given in algorithm 6 can be carried out using a (preorder or depth-first) traversal of G (using root V_{n+1} if $V_{n+1}\notin\bar{V}$). Since this traversal involves visiting each vertex in $V-\bar{V}$ once, computing $x+v$ requires approximately $2n$ operations.

f) Computing the projected Newton direction

The direction $h = Zh_z^*$, where $h_z = h_z^*$ satisfies

$$Z^TGZ = -Z^Tg,$$

is sometimes required in solving problem P1. Unfortunately, this system of equations may be ill-conditioned even when the projected Hessian, Z^TGZ, is positive definite. (The convexity of f guarantees positive semidefiniteness.) We therefore use a numerically stable modified Cholesky factorization (see [21]) of the projected Hessian and solve (via forward and backward substitution)

$$LDL^Th_z = -Z^Tg$$

where $LDL^T = Z^TGZ + \bar{D}$, L is a lower-triangular matrix, D is a diagonal matrix and $\bar{D}$ is a diagonal matrix with diagonal elements equaling zero if the projected Hessian is *sufficiently* positive definite.

The solution of these equations is the major computational expense in our algorithm. Properties of the projected Hessian suggest that it may be possible to improve this situation by solving the original system of equations via a truncated linear preconditioned conjugate-gradient technique.

10. Numerical Results

This algorithm has been implemented in ANSI FORTRAN on a Vax - 11/780 using single precision arithmetic throughout. The following values were used:

machine constants -	$\varepsilon_M = 7.45\times10^{-9}$	
activity and stationarity -	$\varepsilon = 10^{-1}$	$\Psi = 10^{-2}$
descent -	$\delta = 10^{-5}$	$\delta_0 = 10^{-5}$
line search -	$\eta = 9\times10^{-1}$	$\mu = 1\times10^{-3}$
termination -	$\tau_h = 5\times10^{-6}$	$\tau_f = 5\times10^{-6}$

The results of fourteen test problems are presented here. The data for these problems appears in Tables 1a and 1b. (The reader should examine the given references for further information on these problems.)

Table 1a : Data for test problems 1 through 6.

#	Source	n	m	p	x^1	y	$[v_{jk}]$	$[w_{ji}]$
1	[19] Ex. 5.23	5	3	2	(0,0) (0,0) (0,0) (0,0) (0,0)	(1,0) (2,0) (3,0)	$\begin{bmatrix} 2 & 2 & 2 & 2 \\ & 20 & 1 & 0 \\ & & 0 & 0 \\ & & & 40 \\ & & & \end{bmatrix}$	$\begin{bmatrix} 10 & 0 & 0 \\ 4 & 1 & 4 \\ 4 & 1 & 4 \\ 4 & 5 & 4 \\ 4 & 5 & 4 \end{bmatrix}$
2	[19] Ex. 5.7	2	3	2	(0,0) (0,0)	(8,15) (10,20) (30,10)	8	$\begin{bmatrix} 8 & 3 & 5 \\ 0 & 7 & 2 \end{bmatrix}$
3	[19] Ex. 5.6	2	3	2	(0,0) (0,0)	(3,4) (8,7) (15,2)	3	$\begin{bmatrix} 2 & 6 & 0 \\ 4 & 5 & 1 \end{bmatrix}$
4	[4] Problem 5	2	5	2	(0,0) (0,0)	(0,0) (2,4) (6,2) (6,10) (8,8)	2	$\begin{bmatrix} 4 & 2 & 3 & 0 & 0 \\ 0 & 2 & 1 & 3 & 2 \end{bmatrix}$
5	[4] Problem 5	9	5	2	(0,0) (0,0) (6,10) (1,3) (6,10) (8,8) (2,4) (2,4) (6,10)	(0,0) (2,4) (6,2) (6,10) (8,8)	1, $1 \le j < k \le 9$	1, $\substack{1 \le j \le 9 \\ 1 \le i \le 5}$
6	[4] Problem 6	2	3	2	(5,15) (5,15)	(2,5) (10,20) (10,10)	1.5	$\begin{bmatrix} 0.16 & 0.56 & 0.16 \\ 0.16 & 0.56 & 0.16 \end{bmatrix}$

Table 1b : Data for test problems 7 through 14.

#	Source	n	m	p	x^1	y	$[v_{jk}]$	$[w_{ji}]$
7	[32] Problem 5	5	9	2	(0,0) (2,−1) (3,−1) (4,−1) (5,−1)	(0,0) (2,4) (6,2) (6,10) (8,8) (7,7) (0,1) (0,2) (0,3)	$\left[\begin{array}{ccccc} & 1 & 1 & 1 & 1 \\ & & 1 & 10^{-2} & 10^{-1} \\ & & & 10^{-2} & 10^{-1} \\ & & & & 10^{-1} \\ & & & & \end{array}\right]$	$\left[\begin{array}{ccccccccc} 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \end{array}\right]$
8	[32] Problem 1a	1	3	2	(3,2)	(−1,0) (0,1) (1,0)	NA	[1 2 1]
9	[32] Problem 1b	1	3	2	$(1,10^{-6})$	(−1,0) (0,1) (1,0)	NA	[1 2 1]
10	[32] Problem 1c	1	3	2	$(1.000001,-10^{-6})$	(−1,0) (0,1) (1,0)	NA	[1 2 1]
11	[32] Problem 1d	1	3	2	$(1.001,-10^{-3})$	(−1,0) (0,1) (1,0)	NA	[1 2 1]
12	[32] Problem 2	1	3	2	(3,2)	(−1,0) (0,1) (1,0)	NA	[1 1 1]
13	[32] Problem 3	1	3	2	(3,2)	(−1,0) (0,1) (1,0)	NA	[1 1.414 1]
14	[32] Problem 4	1	3	2	(3,2)	(−1,0) (0,1) (1,0)	NA	[1 1.415 1]

The results of the test runs are summarized in Tables 2a and 2b. The figures in these tables refer to the number of iterations required to reach the solution.

Table 2a compares the performance, on problems 1 through 6, of: this algorithm (**NEW**), the hyperboloid approximation procedure (**HAP**) proposed by Eyster, White and Wierwille [17], a modified HAP (**MHAP**) that results when alterations suggested by Ostresh [31] are applied to HAP, and the projected Newton method (**PNM**) proposed by Calamai and Conn [6]. Table 2b compares the performance, on problems 7 through 14, of: this algorithm (**NEW**), the projected Newton method (**PNM**), and a method described by Overton [32] that is closely related to the method outlined here.

The effectiveness of the proposed degeneracy handling scheme is illustrated by comparing the performance of PNM and NEW on problems 1 through 7. (No degenerate iterates are encountered in solving problems 8 through 14 since $n = 1$.) The former algorithm uses random perturbations to resolve degeneracies but is much like the latter in all other respects.

Table 2a : Test results for problems 1 through 6.

#	$\varepsilon_h = 1$		PNM	NEW
	HAP	MHAP		
1	1661	1381	17	12
2	647	546	6	3
3	87	70	4	4
4	45	45	12	9
5	142	114	29	27
6	242	164	6	3
TOTAL	2824	2320	74	58

Key:

HAP refers to the Hyperboloid Approximation Procedure developed by Eyster, White and Wierwille [17].

MHAP refers to a modified HAP suggested by Ostresh [31].

PNM refers to the Projected Newton Method outlined by Calamai and Conn [6].

NEW refers to the method proposed here.

ε_h refers to the initial hyperbolic constant used in HAP and MHAP.

Table 2b : Test results for problems 7 through 14.

#	Overton	PNM	NEW
7	29	49	27
8	6	6	6
9	7	4	4
10	7	5	5
11	4	5	5
12	7	9	9
13	12	8	10
14	8	8	7
TOTAL	80	94	73

Key:

OVERTON refers to the method proposed by Overton [32] for minimizing a sum of Euclidean norms.

PNM refers to the Projected Newton Method outlined by Calamai and Conn [6].

NEW refers to the method proposed here.

Tables 3a, 3b and 3c give further details on the performance of our perturbation scheme on problems 1, 5 and 7. Each row in these tables, except the last row of Table 3c, presents the situation at a degenerate iterate. One important feature illustrated by these results is that the cardinality of N is always (much) smaller than the cardinality of $M_\varepsilon - \overline{M}_\varepsilon$. This means that, compared with the random perturbation scheme outlined in [6], only a few dependent terms need be perturbed using our scheme. Another feature illustrated by these results is that the number of degenerate points encountered is not directly related to the presence or degree of degeneracy at the solution x^*.

Table 3a : Performance of perturbation scheme (Problem 1)

Iteration #	$\lvert M_\varepsilon \rvert$	$\lvert \overline{M}_\varepsilon \rvert$	$\lvert N \rvert$	t_μ
1	12	5	4	5
2	8	5	0	4
4	11	5	4	2
6	7	5	1	4
7	7	5	2	2
8	5	5	0	1
9	5	5	0	1
10	5	5	0	1
12	9	5		

Key:

$\lvert S \rvert$ **refers to the cardinality of set S.**

t_μ **equals the number of out-of-kilter Lagrange multipliers.**

Note: This problem is degenerate at $x^{(1)}$ **and** x^***.**

Table 3b : Performance of perturbation scheme (Problem 5)

Iteration #	$\lvert M_\varepsilon \rvert$	$\lvert \overline{M}_\varepsilon \rvert$	$\lvert N \rvert$	t_μ
4	13	8	2	8
11	19	8	1	8
16	37	9	7	9
19	29	8	0	1
26	36	8	0	0
27	36	8		

Key: (see Table 3a)

Note: This problem is highly degenerate at x^***.**

Table 3c : Performance of perturbation scheme (Problem 7)

Iteration #	$\lvert M_\varepsilon \rvert$	$\lvert \overline{M}_\varepsilon \rvert$	$\lvert N \rvert$	t_μ
1	11	5	3	5
2	8	5	0	5
5	7	4	0	4
14	10	4	3	3
17	7	4	2	3
18	5	4	0	3
19	5	4	1	2
20	5	4	1	1
21	4	4	0	2
22	3	3	0	1
24	4	4	0	1
27	2	2		

Key: (see Table 3a)

Note: This problem is degenerate at $x^{(1)}$**.**

The reader should be made aware that there is little difficulty in modifying the approach presented here to solve linearly constrained location problems and mixed norm location problems. Details of these extensions can be found in [3].

11. Acknowledgments

The authors would like to extend their thanks to Michael Overton for many invigorating and enlightening exchanges, Jorge Moré for his helpful suggestions after several careful readings of preliminary drafts of this manuscript, and Gene Golub for his more than passing interest in this topic and its applications.

12. Bibliography

[1] Aho, A. V., Hopcroft, J. E., and Ullman, J. D., *The design and analysis of computer algorithms*, Addison-Wesley, Mass., 1974.

[2] Busovača, S., "Handling degeneracy in a nonlinear l_1 algorithm", PhD. thesis, University of Waterloo, Waterloo, Ontario, Canada, (1985).

[3] Calamai, P.H., "On numerical methods for continuous location problems", PhD. thesis, University of Waterloo, Waterloo, Ontario, Canada, (1983).

[4] Calamai, P.H., and Charalambous, C., "Solving multifacility location problems involving Euclidean distances", Naval Res. Logist. Quart., 27, 609-620, (1980).

[5] Calamai, P.H., and Conn, A.R., "A stable algorithm for solving the multifacility location problem involving Euclidean distances", SIAM J. Sci. Stat. Comput., 1, 512-525, (1980).

[6] Calamai, P.H., and Conn, A.R., "A second-order method for solving the continuous multifacility location problem", in Numerical Analysis, Proceedings, Dundee 1981, Ed. by G. A. Watson. Lecture Notes in Math,. 912, 1-25, Springer-Verlag, (1982).

[7] Clarke, F. *Optimization and nonsmooth analysis*, Wiley, New York, (1983).

[8] Coleman, T.F., and Conn, A.R., "Nonlinear programming via an exact penalty function: Global analysis", Math. Programming, 24, 137-161, (1982).

[9] Coleman, T.F., and Conn, A.R., "Nonlinear programming via an exact penalty function: Asymptotic analysis", Math. Programming, 24, 123-136, (1982).

[10] Dax, A., "The use of Newton's method for solving Euclidean multifaciity location problems", Tech. Report, Hydrological Service, Jerusalem, (1983).

[11] Dax, A., "A note on optimality conditions for the Euclidean multifacility location problem", submitted for publication in Mathematical Prog., (1985).

[12] Dax, A., "An efficient algorithm for solving the rectilinear multifacility location problem", Tech. Report, Hydrological Service, Jerusalem, (1985).

[13] Deo, N., *Graph theory with application to engineering and computer science*, Prentice-Hall, New Jersey, 1974.

[14] Drezner, A., and Wesolowsky, G.O., "A trajectory method for the optimization of the multi-facility location problem", Management Sci., 24, 1507-1514, (1978).

[15] Eckhardt, U., "Weber's problem and Weiszfeld's algorithm in general spaces", Math. Programming, 18, 186-196, (1980).