UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT

A Stable Algorithm for Solving the Multifacility Location Problem Involving Euclidean Distances

P.H. Calamai
A.R. Conn

CS-80-17

March, 1980

A Stable Algorithm for Solving the Multifacility Location Problem Involving Euclidean Distances

P.H. Calamai

Systems Design
University of Waterloo

A.R.Conn

Computer Science
University of Waterloo

ABSTRACT

There is a rapidly growing interdisciplinary interest in the application of location models to real life problems. Unfortunately, the current methods used to solve the most popular minisum and minimax location problems are computationally inadequate. A more unified and numerically stable approach for solving these problems is proposed. Detailed analysis is done for the linearly constrained Euclidean distance minisum problem for facilities located in a plane. Preliminary computational experience suggests that this approach compares favourably with other methods.

A Stable Algorithm for Solving the Multifacility Location Problem Involving Euclidean Distances

P.H. Calamai

Systems Design
University of Waterloo

A.R.Conn

Computer Science
University of Waterloo

1. INTRODUCTION

Facility location problems are generally concerned with finding the optimal location of a set of new facilities in a network of existing facilities. Over the past few decades various forms of the problem have been developed and addressed in the literature (see Francis and Goldstein [11]). In this paper we present a stable method for solving the multifacility location problem involving Euclidean distances. We assume that the feasible region is connected and that the parameters are static with respect to time (for a discussion of the dynamic aspect of these problems see Wesolowsky and Truscott [17]). In addition, we assume that the problem is well formulated (Francis and Cabot [10] formally describe this and many other properties of the problem).

Various approaches have been used in an attempt to efficiently solve this location problem. One such method involves the use of a heuristic algorithm described by Vergin and Rogers [14]. Unlike their approach, most solution techniques quarantee optimality. Included among these are the convex programming approach described by Love [13], the hyperbolic approximation method used by Eyster, White and Wierwille [9], the pseudo-gradient technique described by Calamai and Charalambous [3] and the subgradient algorithm presented by Chatelon, Hearn and Lowe [5].

One property of this problems objective function that causes considerable difficulty is the fact that it is not everywhere differentiable. This non-differentiability occurs whenever any new facility coincides with any other new facility or with an existing facility. As a result, standard minimization techniques cannot be directly applied. We overcome this difficulty by projecting onto a particular affine space in which the current near zero terms stay unchanged and the remaining well-defined terms can be decreased. Then, when we think we are optimal, we perform a linear refinement step that makes the near zero terms exactly zero. Under mild and suitable conditions we show that this technique

This work is supported in part by NSERC Grant No. A8639

converges to the solution of our problem.

Another significant property of this technique is that it can be used for the linearly constrained problem. Here we replace the constrained problem by a sequence of unconstrained problems by using a penalty function due to Pietrzykowski [15] and Zangwill [18]. Using this method we are able to obtain a sequence of points (optimal to the unconstrained penalty function) which converge to the desired constrained optimum.

2. PROBLEM STATEMENT AND DEFINITIONS

The multifacility minisum problem involving costs associated with Euclidean distances between facilities in $\mathbf{R}^q$ can be stated as: Find the point $x^{*T} = \{x_1^{*T}, \cdots, x_n^{*T}\}$ in $\mathbf{R}^{qn}$ to minimize

$$f(x) = \sum_{1 \leqslant j < k \leqslant n} v_{jk} \| x_j - x_k \| + \sum_{j=1}^{n} \sum_{i=1}^{m} w_{ji} \| x_j - p_i \| \tag{2.1}$$

where

$n \triangleq$ number of new facilities (NFs) to be located.

$m \triangleq$ number of existing facilities (EFs).

$x_j^T = (x_{j1} \cdots x_{jq}) \triangleq$ vector location of NF_j in $\mathbf{R}^q$ $j = 1, \ldots, n$.

$p_i^T = (p_{i1} \cdots p_{iq}) \triangleq$ vector location of EF_i in $\mathbf{R}^q$ $i = 1, \ldots, m$.

$v_{jk} \triangleq$ nonnegative constant of proportionality relating the l_2 distance between NF_j and NF_k to the cost incurred, $1 \leqslant j < k \leqslant n$.

$w_{ji} \triangleq$ nonnegative constant of proportionality relating the l_2 distance between NF_j and EF_i to the cost incurred, $1 \leqslant j \leqslant n$, $1 \leqslant i \leqslant m$.

$\| x_j - x_k \| = (\sum_{c=1}^{q} | x_{jc} - x_{kc} |^2)^{1/2} \triangleq$ l_2 distance between NF_j and NF_k, $1 \leqslant j < k \leqslant n$.

$\| x_j - p_i \| = (\sum_{c=1}^{q} | x_{jc} - p_{ic} |^2)^{1/2} \triangleq$ l_2 distance between NF_j and EF_i, $1 \leqslant j \leqslant n$, $1 \leqslant i \leqslant m$.

In order to simplify the remainder of the discussion we set $q = 2$ (i.e. all facilities are in $\mathbf{R}^2$). Now if we let $\eta = \dfrac{n^2 - n}{2}$ and $\tau = \eta + mn$ and if we define the constants

$$\{\alpha_1, \ldots, \alpha_\tau\} = \{\underbrace{v_{12}, \ldots, v_{1n}}_{n-1}, \underbrace{v_{23}, \ldots, v_{2n}}_{n-2}, \ldots, \underbrace{v_{n-1n}}_{1}, \underbrace{w_{11}, \ldots, w_{1m}}_{m}, \ldots, \underbrace{w_{n1}, \ldots, w_{nm}}_{m}\},$$

$$\{\beta_1, \ldots, \beta_\tau\} = \{\underbrace{1, \ldots, 1}_{n-1}, \underbrace{2, \ldots, 2}_{n-2}, \ldots, \underbrace{n-1}_{1}, \underbrace{1, \ldots, 1}_{m}, \ldots, \underbrace{n, \ldots, n}_{m}\},$$

$$\{\gamma_1, \ldots, \gamma_\tau\} = \{\underbrace{2, 3, \ldots, n}_{n-1}, \underbrace{3, 4, \ldots, n}_{n-2}, \ldots, \underbrace{n}_{1}\},$$

and the index set $M = \{1, \ldots, \tau\}$ then (2.1) can be restated more conveniently as

$$\text{minimize } f(x) = \sum_{i \in M} \| A_i^T x - b_i \| \tag{2.2}$$

$$\text{where } A_i^T = \begin{cases} [0_1 \quad \alpha_i I \quad 0_2 \quad -\alpha_i I \quad 0_3] & i = 1, \ldots, \eta \\ [0_1 \quad \alpha_i I \quad 0_4] & i = \eta + 1, \ldots, \tau \end{cases}$$

$$b_i = \begin{cases} 0_5 & i = 1, \ldots, \eta \\ \alpha_i p_e & i = \eta + 1, \ldots, \tau \end{cases}$$

I is a 2 by 2 identity matrix,

$0_1, 0_2, 0_3, 0_4$ and 0_5 are zero matrices of dimension 2 by $2(\beta_i - 1)$, 2 by $2(\gamma_i - \beta_i - 1)$, 2 by $2(n - \gamma_i)$, 2 by $2(n - \beta_i)$ and 2 by 1 respectively,

$e = (i - \eta) \bmod m$ and $p_0 \equiv p_m$.

In a manner analogous to that of Bartels, Conn and Sinclair [2], define for any point x in $\mathbf{R}^{2n}$: the residual vectors

$$r_i(x) = A_i^T x - b_i \qquad \forall\, i \in M,$$

the index set

$$I_\epsilon(x^k) = \{ i \in M \mid \| r_i(x^k) \| \leqslant \alpha_i \epsilon \} = \{ i_1, \ldots, i_{t_k} \} \qquad \epsilon \geqslant 0,$$

the corresponding matrix and vector

$$\mathbf{A} = \mathbf{A}(x^k) = [A_{i_1}, \ldots, A_{i_{t_k}}] \text{ and } r(x^k)^T = [r_{i_1}(x^k)^T, \ldots, r_{i_{t_k}}(x^k)^T], \tag{2.3}$$

and the vector

$$\tilde{\nabla} f_k = \sum_{i \in M \setminus I_\epsilon(x^k)} \nabla(\| r_i(x^k) \|) = \sum_{i \in M \setminus I_\epsilon(x^k)} A_i r_i(x^k) / \| r_i(x^k) \| .$$

(Note that for $\epsilon = 0$ the index set $I_\epsilon(x^k)$ corresponds to those terms which are exactly *active* (binding) at the point x^k. Conn [7] gives an account of the necessity for considering , as we do, "*near-active*" terms .)

Finally define P_k as the orthogonal projector onto $S_k^\perp$ where S_k is the space spanned by the columns of $\mathbf{A}(x^k)$. *For the present it will be assumed that* $\mathbf{A}(x^k)$ *has full rank.*

3. OPTIMALITY CONDITIONS

For all points x close enough to x^k in $\mathbf{R}^{2n}$ we have

$$\begin{aligned} f(x) &= \sum_{i \in M \setminus I_\epsilon(x^k)} \| r_i(x) \| + \sum_{i \in I_\epsilon(x^k)} \| r_i(x) \| \\ &= h_1(x) + h_2(x) \end{aligned} \tag{3.1}$$

Assume $P_k \nabla h_1(x^k) \neq 0$ ($i.e. P_k \tilde{\nabla} f_k \neq 0$), then for $d^k = -P_k \tilde{\nabla} f_k$ we have

$$\begin{aligned} h_1(x^k+\lambda d^k) &= h_1(x^k) + \lambda(d^k)^T\nabla h_1(x^k) + 0(\lambda^2) \\ &= h_1(x^k) - \lambda\| P_k\tilde{\nabla} f_k\| + 0(\lambda^2) \end{aligned} \tag{3.2a}$$

and since $A_i^Td^k = 0 \quad \forall\, i \in I_\epsilon(x^k)$

$$h_2(x^k+\lambda d^k) = h_2(x^k) \tag{3.2b}$$

Combining equations (3.2a) and (3.2b) yields the result;

$$f(x^k+\lambda d^k) - f(x^k) = -\lambda\| P_k\tilde{\nabla} f_k\| + 0(\lambda^2). \tag{3.3}$$

Therefore for $d^k = -P_k\tilde{\nabla} f_k \neq 0$ there exists $\delta>0$ such that $f(x+\lambda d) < f(x)$ for all $0<\lambda<\delta$. Alternatively, if $P_k\nabla f_k = 0$ then *(under the linear independence assumption)* the corresponding point x^k is called a *(nondegenerate) dead point.* In this case ∇f_k can be expressed *(uniquely)* as a linear combination of the columns of $\mathbf{A}(x^k)$,

$$\tilde{\nabla} f_k = \mathbf{A}u = \sum_{i\in I_\epsilon(x^k)} A_i u_i \quad , \qquad u^T = [u_1^T \cdots u_{t_k}^T]. \tag{3.4}$$

Then,

$$\begin{aligned} h_1(x^k+\lambda d^k) &= h_1(x^k) + \lambda(d^k)^T\tilde{\nabla} f_k + 0(\lambda^2) \\ &= h_1(x^k) + \lambda \sum_{i\in I_\epsilon(x^k)} u_i^T A_i^T d^k + 0(\lambda^2) \end{aligned} \tag{3.5a}$$

and

$$\begin{aligned} h_2(x^k+\lambda d^k) &= h_2(x^k) + \lambda\left[\sum_{i\in I_0(x^k)} \| A_i^Td^k\| \right. \\ &\quad \left. + \sum_{i\in I_\epsilon(x^k)\setminus I_0(x^k)} \nabla^T(\| r_i(x^k)\|)d^k \right] + 0(\lambda^2) \\ &= h_2(x^k) + \lambda\left[\sum_{i\in I_0(x^k)} \| A_i^Td^k\| \right. \\ &\quad \left. + \sum_{i\in I_\epsilon(x^k)\setminus I_0(x^k)} \frac{r_i(x^k)^TA_i^Td^k}{\| r_i(x^k)\|} \right] + 0(\lambda^2). \end{aligned} \tag{3.5b}$$

Combining equations (3.5a) and (3.5b) yields the result;

$$\begin{aligned} f(x^k+\lambda d^k) - f(x^k) &= \lambda\left[\sum_{i\in I_0(x^k)} (u_i^TA_i^Td^k + \| A_i^Td^k\|) \right. \\ &\quad \left. + \sum_{i\in I_\epsilon(x^k)\setminus I_0(x^k)} (u_i^TA_i^Td^k + \frac{r_i(x^k)^TA_i^Td^k}{\| r_i(x^k)\|}) \right] + 0(\lambda^2). \end{aligned} \tag{3.6}$$

Now, if there exists an index $l \in I_\epsilon(x^k)$ such that $\| u_l\| > 1$ then take

$$d^k = -P_{k/l}A_l u_l, \tag{3.7}$$

where $P_{k/l}$ is the orthogonal projector onto $S_{k/l}^{\perp}$ and $S_{k/l}$ is the space spanned by the columns of $\mathbf{A}(x^k)$ with columns A_l deleted. Then, since $A_i^T d^k = 0$ for all $i \in I_\epsilon(x^k) - \{l\}$ and $A_l^T d^k = -A_l^T P_{k/l} A_l u_l = -\rho u_l$ where $\rho > 0$ (see Appendix), we have, for small $\lambda > 0$,

$$f(x^k + \lambda d^k) - f(x^k) = \begin{cases} -\lambda\rho\left[\| u_l\|^2 + \dfrac{r_l(x^k)^T u_l}{\| r_l(x^k)\|}\right] < 0 & l \in I_\epsilon(x^k)\backslash I_0(x^k) \quad (3.8a) \\ -\lambda\rho\| u_l\| (\| u_l\| - 1) < 0 & l \in I_0(x^k) \quad (3.8b) \end{cases}$$

Thus, if there exists $l \in I_\epsilon(x^k)$ such that $\| u_l\| > 1$ then, for $d = -P_{k/l}A_l u_l$, there exists a $\delta > 0$ such that $f(x + \lambda d) < f(x)$ for all $0 < \lambda < \delta$. What happens if x^k is a dead point and $\| u_i\| \leqslant 1 \;\; \forall i \in I_\epsilon(x^k)$?

CASE I: $I_\epsilon(x^k)\backslash I_0(x^k) = \varnothing$

Under these circumstances we have

$$f(x^k + \lambda d^k) - f(x^k) = \lambda \sum_{i \in I_0(x^k)} (u_i^T A_i^T d^k + \| A_i^T d^k\|) + 0(\lambda^2) \tag{3.9}$$

which is nonnegative for all d^k in $\mathbf{R}^{2n}$. Thus x^k is at least a weak local minimum and hence a global minimum of the convex function $f(x)$.

CASE II: $I_\epsilon(x^k)\backslash I_0(x^k) \neq \varnothing$

For any $l \in I_\epsilon(x^k)\backslash I_0(x^k)$ take $d^k = -P_{k/l}A_l r_l(x^k)$. For this choice we have $A_i^T d^k = 0 \;\; \forall i \in I_\epsilon(x^k) - \{l\}$ and expression (3.6) becomes

$$f(x^k + \lambda d^k) - f(x^k) = -\lambda\rho(u_l^T r_l(x^k) + \| r_l(x^k)\|) + 0(\lambda^2) \tag{3.10}$$

which is nonpositive (strictly negative unless $u_l = -\dfrac{r_l(x^k)}{\| r_l(x^k)\|}$).The next section explains a method for avoiding this case.

4. THE LINEAR REFINEMENT

If we are at the point x^k, and $\| P_k \tilde{\nabla} f_k \|$ is "*small*", then we may be approaching a dead point, say $\hat{x}$. We can then obtain estimates $\{\hat{u}_i\}$ of the "*dual*" variables $\{u_i\}$ by finding the least-squares solution to

$$\tilde{\nabla} f_k = \sum_{i \in I_\epsilon(x^k)} A_i u_i. \tag{4.1}$$

Then if $\| \hat{u}_i\| < 1 \;\; \forall i \in I_\epsilon(x^k)$ we may be near a local minimum (note that if $I_\epsilon(x^k)\backslash I_o(x^k) = \varnothing$ this is especially clear). We therefore wish to satisfy the near-active terms exactly. In [8], Conn and Pietrzykowski define a *vertical component*, based on linearizations, in an attempt to satisfy their "near-activities" exactly. Here we wish to find the solution to the already linear system of equations;

$$A_i^T(x^k - v^k) - b_i = 0 \quad \forall\, i \in I_\epsilon(x^k). \tag{4.2}$$

This is best accomplished by evaluating the least-squares solution to the following,

$$\mathbf{A}(x^k)^T v^k = r(x^k) \tag{4.3}$$

which is given, *algebraically at least*, by

$$v^k = (\mathbf{A}(x^k)^+)^T r(x^k) \tag{4.4}$$

where $\mathbf{A}(x^k)^+$ is the generalized (Moore-Penrose) inverse of $\mathbf{A}(x^k)$.

Note that once a refinement step is taken the set $I_\epsilon(x^k) \backslash I_0(x^k)$ is nullified and thus case II above need not be considered.

5. THE RESTRICTED GRADIENT

Either x^k is optimal or there exists a direction d^k such that, for small $\lambda > 0$,

$$f(x^k + \lambda d^k) - f(x^k) = \lambda (d^k)^T g \text{ with } (d^k)^T g < 0. \tag{5.1}$$

We take

$$g = \tilde{\nabla} f_k \text{ and } d^k = -P_k g \tag{5.2}$$

if x^k is not a dead point, and

$$g = \tilde{\nabla} f_k - A_l u_l / \| u_l \| \text{ and } d^k = -P_{k/l} A_l u_l \tag{5.3}$$

(where $l \in I_\epsilon(x^k)$ is an index giving $\| u_l \| > 1$) when x^k is a dead point.

As in reference [2] the vector g will be referred to as the *restricted gradient* of the function.

6. THE STEPSIZE

Suppose we are at a point $x \in \mathbf{R}^{2n}$ and a direction d has been chosen as described. Further, suppose our function is of the form

$$\Phi(x) = \sum_{i \in M} | \phi_i(x) | \tag{6.1}$$

where the functions $\phi_i(x)$ are all linear. Clearly the minimum of Φ , in the direction d , must be at a point $\bar{x}$ on d where, for at least one $i \in M \backslash I_\epsilon(x^k)$, we have $\phi_i(\bar{x}) = 0$. The points $\bar{x}$ are called "*breakpoints*". Therefore we determine the stepsize λ to the breakpoint that gives the minimum function value along d. In other words, *as long as the function is decreasing* (i.e. $d^T g(\bar{x}) < 0$) *we continue to move along d through successive breakpoints.*

To extend this idea to the case where the functions $\phi_i(x)$ are nonlinear is not difficult as long as we are: 1) content with estimating the location of breakpoints and, 2) willing to use a linear approximation in evaluating $d^T g(\bar{x})$.

In the first instance we wish to find the values of λ_i satisfying

$$A_i^T(x + \lambda_i d) - b_i = 0, \quad \forall\, i \in M \backslash I_\epsilon(x). \tag{6.2}$$

A good approximation to use, in the l_2 sense, comes from the equations:

$$\lambda_i = \frac{-(A_i^T d)^T r_i(x)}{\| A_i^T d \|^2}, \quad \forall\, i \in M \backslash I_\epsilon(x). \tag{6.3}$$

In the second instance we wish to find an expression for $d^T g(x + \lambda d)$ which is linear in λ. Using the first-order Taylor series approximation about the point x, we get

$$d^T g(x + \lambda d) = d^T g(x) + \lambda \sum_{i \in M \setminus I_\epsilon(x)} \left\{ \frac{\| A_i^T d \|^2}{\| r_i(x) \|} - \frac{[(A_i^T d)^T r_i(x)]^2}{\| r_i(x) \|^3} \right\} \tag{6.4}$$

With these ideas, the following stepsize algorithm (similar in concept to the one given in [4]) can be formulated:

```
1)  Set
        $\lambda_i \leftarrow \dfrac{-(A_i^T d)^T r_i(x)}{\| A_i^T d \|^2}, \quad \forall i \in M \setminus I_\epsilon(x)$
        $J_1 \leftarrow \{ i \in M \setminus I_\epsilon(x) \mid \lambda_i > 0 \}$
        $\Delta \leftarrow \sum_{i \in M \setminus I_\epsilon(x)} \left\{ \dfrac{\| A_i^T d \|^2}{\| r_i(x) \|} - \dfrac{[(A_i^T d)^T r_i(x)]^2}{\| r_i(x) \|^3} \right\}$
        $\hat{\lambda} \leftarrow -\dfrac{d^T g(x)}{\Delta}$
        $\nu \leftarrow 1, \; l_o \leftarrow 0, \; \lambda_o \leftarrow 0.$

2)  IF ($J_\nu = \emptyset$) THEN go to 5.
            ELSE:

3)  Determine $l_\nu \in J_\nu$ such that $\lambda_{l_\nu} \leqslant \lambda_i \quad \forall i \in J_\nu$.

4)  IF ($\lambda_{l_\nu} \geqslant \hat{\lambda}$). THEN go to 5.
            ELSE set $J_{\nu+1} \leftarrow J_\nu - \{l_\nu\}$
                     $\nu \leftarrow \nu + 1$   and go to 2.

5)  Set $\tilde{x} \leftarrow x + \lambda_{l_{\nu-1}} d$

6)  IF ($f(\tilde{x}) < f(x) - \delta$) THEN set $x \leftarrow \tilde{x}$ and return.
            ELSE use $x$ and $\tilde{x}$ in cubic interpolation
                 routine to get a new $x$, and return.
```

NOTES:

i) δ is some preassigned positive constant.

ii) the cubic line search used in step 6) need not be (and in our present implementation is not) exact.

7. THE MINIMIZATION ALGORITHM

```
1)  Choose any $x^1 \in \mathbf{R}^{2n}$ and set $k \leftarrow 1$.  Initialize $\epsilon_{ACT}, \epsilon_{ZP}, \epsilon_{VS}$ and $\epsilon$ .
2)  Evaluate all residuals. Identify all index sets, the vector $r(x^k)$ and the
    matrix $\mathbf{A}(x^k)$. Evaluate $\tilde{\nabla} f_k$ and construct $P_k$.
3)  Set DEL $\leftarrow$ .FALSE. and $d^k \leftarrow -P_k \tilde{\nabla} f_k$.
4)  Consider retaining same $\epsilon_{ACT}$ - active set:
            IF ($\| d^k \| \geqslant \epsilon_{ZP} \| \tilde{\nabla} f_k \|$ )  THEN go to 8.
                                                ELSE;
5)  Determine the "duals". Find the least-squares solution $\hat{u}$ to:
                        $\mathbf{A}u = \tilde{\nabla} f_k$.
6)  Decide whether or not to drop an activity:
            IF ($\| \hat{u}_i \| \leqslant 1 \quad \forall i \in I_{\epsilon_{ACT}}(x^k)$)   THEN go to 11.
                                                ELSE;
7)  Drop an activity:
        Choose $l \in I_{\epsilon_{ACT}}(x^k)$ with $\| \hat{u}_l \| > 1$ and set $d^k = -P_{k/l}A_l\hat{u}_l$
        Set DEL = .TRUE.
8)  Use the line search algorithm above to find $\tilde{x}^{k+1}$.
9)  Decide whether or not to do a linear refinement:
        IF ($\| d^k \| \geqslant \epsilon_{VS}$ .OR. DEL) THEN set $x^{k+1} \leftarrow \tilde{x}^{k+1}$
                                                   $k \leftarrow k+1$ and go to 2.
                                   ELSE;
10) Perform the linear refinement:
        Solve, in the least-squares sense, the system
                        $\mathbf{A}^T v^k = r(x^k)$
        IF ($f(\tilde{x}^{k+1} - v^k) < f(\tilde{x}^{k+1})$) THEN set  $x^{k+1} \leftarrow \tilde{x}^{k+1} - v^k$
                                                   $k \leftarrow k+1$ and go to 2.
                                       ELSE set    $x^{k+1} \leftarrow \tilde{x}^{k+1}$
                                                   $\epsilon_{ACT} \leftarrow \epsilon_{ACT}/10$
                                                   $k \leftarrow k+1$ and go to 2.
11) IF ($\| d^k \| > \epsilon$) THEN go to 8
                      ELSE IF ($r_k^\infty > \epsilon$)  THEN   set $\tilde{x}^{k+1} \leftarrow x^k$
                                                   and go to 10.
                                         ELSE    STOP!
```

NOTE: $r_k^\infty = \underset{i \in I_\epsilon(x^k)}{\text{maximum}}(\| r_i(x^k) \|)$

8. LINEAR CONSTRAINTS AND THE PENALTY FUNCTION

Consider the following constrained form of the problem,

$$\begin{aligned} \text{minimize} \quad f(x) &= \sum_{i\in M} \| A_i^T x - b_i \| \\ \text{such that } r_i(x) &= a_i^T x - b_i \geqslant 0 \qquad i = \tau+1, \ldots, \tau' \\ r_i(x) &= a_i^T x - b_i = 0 \qquad i = \tau'+1, \ldots, \tau'' \end{aligned} \tag{8.1}$$

We can transform this problem into the penalty function of Pietrzykowski [15] and solve the sequence of unconstrained problems,

$$\text{minimize}\, p(x,\mu) = \mu f(x) - \sum_{i\in LI} \min\,[0, r_i(x)] + \sum_{i\in LE} |\, r_i(x)| \tag{8.2}$$

where μ is a positive parameter, $LI = \{\tau+1, \ldots, \tau'\}$ and $LE = \{\tau'+1, \ldots, \tau''\}$. It has been shown [15] that under *mild and suitable conditions*, an exact minimum of (8.2) coincides with an exact minimum of (8.1) for all values of μ sufficiently small. This suggests the following outer algorithm:

```
1) Choose $\mu > 0$
2) Minimize $p(x, \mu)$ over $x$
3) IF (we are optimal) THEN STOP!
                        ELSE;
4) Set $\mu \leftarrow \mu/10$ and go to 2.
```

The minimization in step 2 can be performed using obvious extensions of the techniques already described (see [1]). Some of the details of these extensions will be given below.

For any x^k in $\mathbf{R}^{2n}$ define the index sets

$$\begin{aligned} I_\epsilon^{AI}(x^k) &= \{i \in LI \mid |\, r_i(x^k)| \leqslant \epsilon\} = \{i_{t_k+1}, \ldots, i_{t'_k}\} \\ I_\epsilon^{VI}(x^k) &= \{i \in LI \mid r_i(x^k) < -\epsilon\}, \end{aligned} \tag{8.3}$$

and $I_\epsilon^{AE}(x^k) = \{i \in LE \mid |\, r_i(x^k)| \leqslant \epsilon\} = \{i_{t'_k+1}, \ldots, i_{t''_k}\}$,

and the vector $\tilde{\nabla} p_k = \mu \tilde{\nabla} f_k - \sum_{i\in I_\epsilon^{VI}(x^k)} a_i + \sum_{i\in LE\setminus I_\epsilon^{AE}(x^k)} \text{sgn}\,[r_i(x^k)] a_i$.

Redefine the matrix $\mathbf{A}(x^k)$, the vector $r(x^k)$ and the scalar r_k^∞ as

$$\mathbf{A}(x^k) = [A_{i_1}, \ldots, A_{i_{t_k}}, a_{i_{t_k+1}}, \ldots, a_{i_{t'_k}}, a_{i_{t'_k+1}}, \ldots, a_{i_{t''_k}}]$$

$$r(x^k)^T = [r_{i_1}(x^k)^T, \ldots, r_{i_{t_k}}(x^k)^T, r_{i_{t_k+1}}(x^k), \ldots, r_{i_{t'_k}}(x^k), r_{i_{t'_k+1}}(x^k), \ldots, r_{i_{t''_k}}(x^k)],$$

and $r_k^\infty = \underset{j=1,\ldots,t''}{\text{maximum}}(\| r_{i_j}(x^k)\|)$.

Now if we replace $\tilde{\nabla} f_k$ by $\tilde{\nabla} p_k$ and $f(\cdot)$ by $p(\cdot, \mu)$ then our algorithm can be used to minimize $p(x, \mu)$ over x if we make the following changes to steps 6 and 7 ;

```
6) Decide whether or not to drop an activity:
    IF ( $\| \hat{u}_i \| \leqslant 1 \quad \forall i \in I_{\epsilon_{ACT}}(x^k)$  .AND.
         $| \hat{u}_i | \leqslant 1 \quad \forall i \in I_\epsilon^{AE}(x^k)$  .AND.
         $\hat{u}_i \geqslant 0 \quad \forall i \in I_\epsilon^{AI}(x^k)$ ) THEN go to 11.
                                      ELSE;
7) Drop an activity:
        Choose $l \in I_{\epsilon_{ACT}}(x^k)$ with $\| \hat{u}_j \| > 1$ and set $d^k = -P_{k/l} A_l \hat{u}_l$
    .OR. Choose $l \in I_\epsilon^{AI}(x^k)$ with $\hat{u}_l < 0$ and set $d^k = P_{k/l} a_l$.
        Set DEL ← .TRUE.
```

(It can be shown that if there exists an index $l \in I_\epsilon^{AI}(x^k)$ with $\hat{u}_l > 1$ or an index $l \in I_\epsilon^{AF}(x^k)$ with $| \hat{u}_l | > 1$ then the direction $d = -\text{sgn}(\hat{u}_l) P_{k/l} a_l$ is a descent direction. However, as pointed out by Bartels and Conn [1], constraint r_{j_l} will be violated at $x^k + \lambda d^k$ and, rather than do this, we take an alternative descent direction if one exists. Otherwise, we reduce μ in order to give more emphasis to the constraints.)

9. IMPLEMENTATION

Throughout this derivation it was assumed that the matrix $\mathbf{A}(x)$ was full rank in order to uniquely define the solution to various equations. This can no longer be accomplished when $\mathbf{A}(x)$ is rank deficient because of the inherent degeneracies encountered. Fortunately, this difficulty can be resolved in a straightforward manner computationally by using a method similar to Bartels et al. [2]. This involves treating all degeneracies as if they occurred due to the error introduced through the storage of data on a finite machine. Then the machine accuracy can be artificially reduced and the problem perturbed so that the degeneracies are resolved. Finally, after the perturbation, a unique solution can be attained which satisfies the original problem.

The full rank matrix $\mathbf{A}(x^k)$ can be factored, by forming a product of Givens transformations, into the form

$$\mathbf{A}(x^k) = Q^k \begin{bmatrix} R^k \\ 0 \end{bmatrix} = [Q_1^k \quad Q_2^k] \begin{bmatrix} R^k \\ 0 \end{bmatrix} \tag{9.1}$$

where Q^k is an orthonormal matrix and R^k is a nonsingular upper-triangular matrix. This numerically stable decomposition can easily be modified to accommodate the necessary additions and deletions of column vectors to the matrix $\mathbf{A}$ when it is updated (see [12]).

Since the columns of Q_2^k form an orthogonal basis for the space $S_k^\perp$ the projection matrix P_k can be computed as

$$P_k = (Q_2^k)(Q_2^k)^T \tag{9.2}$$

In addition, the vector $\hat{u}$ required in step 5) of the minimization algorithm (with ∇p_k replacing ∇f_k) can be obtained efficiently by solving

$$R^k\hat{u} = (Q_1^k)^T\tilde{\nabla}p_k \tag{9.3}$$

whereas, the vector v^k required in step 10) can easily be obtained by first solving

$$(R^k)^T\bar{v}^k = r(x^k) \tag{9.4}$$

and then forming

$$v^k = Q_1^k\bar{v}^k \tag{9.5}$$

10. PRELIMINARY COMPUTATIONAL EXPERIENCE

Six small problems were run on a Honeywell 6060 computer to compare the performance of this projection technique with other approaches. The problems used were chosen because of the availability of comparison data. The other solution methods considered are as follows:

i) Hyperbolic Approximation Procedure (HAP)

ii) Modified HAP (MHAP)

iii) The program of Calamai and Charalambous (MFLPVI).

A description of these methods and references to the data used in the six problems are given in [3]. A summary of the results is given below. The column labelled "NEW" refers to the projection algorithm.

#	MFLPVI	$\epsilon^{(0)} = 10^0$		$\epsilon^{(0)} = 10^{-4}$		NEW
		HAP	MHAP	HAP	MHAP	
1	564	1661	1381	2027	1407	42
2	148	647	546	4641	2281	23
3	63	87	70	770	197	10
4	31	45	45	45	45	17
5	223	142	114	1763	975	59
6	63	242	164	3743	1869	11
TOTAL	1092	2824	2320	12989	6744	162

TABLE 1: COMPARATIVE TEST RESULTS

Remarks:

a) The numbers in the table refer to the number of function evaluations.

b) No attempt was made to choose ideal parameters. For the results shown the choice was $\epsilon = 7.45\ E-9$, $\epsilon_{ACT} = 0.01$, $\epsilon_{VS} = 0.1$ and $\epsilon_{ZP} = 0.5$.

c) The results and the theory suggest that the new technique will work well with larger and/or more difficult problems.

11. CONCLUSIONS AND RECOMMENDATIONS

(a) The purpose of this paper is to provide a simple stable method for solving the multifacility location problem involving Euclidean distances; however, the same ideas can be, and perhaps should be, applied to location problems that use other distance measures (l_1 and l_∞ for example). This would provide a more unified approach to solving this class of problems.

(b) The method derived for handling linear constraints can be modified to handle any continuous functions mapping $\mathbf{R}^{2n} \rightarrow \mathbf{R}^1$ by considering local linearizations of those functions.

(c) Instead of this method that essentially uses steepest descent in a subspace, a more sophisticated approach would involve the use of second-order information to derive quasi-Newton steps in a subspace. Such a technique would undoubtedly lead to an improvement in the rate of convergence. See [6].

(d) For large problems the sparsity and structure of the matrix $\mathbf{A}(x)$ would have to be given attention. This would suggest the use of decomposition techniques as well as the use of methods that maintain sparsity in updating these decompositions whenever columns were added to or deleted from **A**.

(e) The stepsize algorithm outlined in this paper is not necessarily optimal. Many alternative schemes are available, including;

    1) the reordering of breakpoints based on the magnitude of the least-squares error in their estimation,

    2) the use of a more exact cubic line search either with or without the breakpoint analysis,

    3) the use of a more sophisticated steplength algorithm. See [14].

f) Step 7 of our algorithm results in the release of an index, associated with an "out-of-kilter" dual, from the activity set. In our implementation we drop the index (and thus the corresponding activity) associated with the first "out-of-kilter" dual encountered. This choice may not result in the optimal descent direction.

All of these ideas are currently being investigated.

ACKNOWLEDGEMENTS

The splendid typesetting was accomplished by Anne Trip-de-Roche and Brian Finch using the Photon typesetter and Troff.

12. BIBLIOGRAPHY

[1] Bartels, R.H. and Conn, A.R., *Linearly constrained discrete* l_1 problems, ACM TOMS, to appear.

[2] Bartels, R.H., Conn, A.R. and Sinclair, J.W., *Minimization techniques for piecewise differentiable functions: The l_1 solution to an overdetermined linear system*, SIAM J. Numer. Anal. 15, no. 2, 1978, 224-241.

[3] Calamai, P.H. and Charalambous, C., *Solving multifacility location problems involving Euclidean distances*, Naval Res. Log. Quart. (to appear).

[4] Charalambous, C. and Conn, A.R., *An efficient method to solve the minimax problem directly*, SIAM J. Numer. Anal. 15, No. 1, 1978, 162-187.

[5] Chatelon, J.A., Hearn, D.W. and Lowe, T.J., *A subgradient algorithm for certain minimax and minisum problems*, Math. Prog. 15, 1978, 130-145.

[6] Coleman, T.F. and Conn, A.R., *Nonlinear programming via an exact penalty function method: A global method*, Comp. Sci. Tech. Rep., University of Waterloo, Waterloo, Ontario, 1979.

[7] Conn, A.R., *Constrained optimization using a nondifferentiable penalty function*, SIAM J. Numer. Anal. 16, no. 4, 1973, 760-784.

[8] Conn, A.R. and Pietrzykowski, T., *A penalty function method converging directly to a constrained optimum*, SIAM J. Numer. Anal. 14, 1977, 348-375.

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

[10] Francis, R.L. and Cabot, A.V., *Properties of a multifacility location problem involving Euclidean distances*, Naval Res. Log. Quart. 19, 1972, 335-353.

[11] Francis, R.L. and Goldstein, J.M., *Location theory: A selective bibliography*, Opns. Res. 22, 1974, 400-410.

[12] Gill, P.E., Golub, G.H., Murray, W. and Saunders, M., *Methods for modifying matrix factorizations*, Math. Comp. 28, no. 126, 1974, 505-535.

[13] Love, R.F., *Locating facilities in three-dimensional space by convex programming*, Naval Res. Log. Quart. 16, 1969, 503-516.

[14] Murray, W. and Overton, M., *Steplength algorithms for minimizing a class of non-differentiable functions*, Stanford University Technical Report CS-78-679, 1978.

[15] Pietrzykowski, T., *An exact potential method for constrained maxima*, SIAM J. Numer. Anal. 6, 1969, 299-304.

[16] Vergin, R.C. and Rogers, J.D., *An algorithm and computational procedure for locating economic facilities*, Management Sci. 13, no. 6, 1967, 240-254.

[17] Wesolowsky, G.O. and Truscott, W.G., *Dynamic location of multiple facilities for a class of continuous problems*, AIIE Trans. 8, no. 1, 1976, 76-83.

[18] Zangwill, W.I., *Nonlinear programming via penalty functions*, Management Sci. 13, 1962, 344-350.

APPENDIX

Prove:

$$A_l^T P_{k/l} A_l = \rho I, \quad \rho > 0$$

where

$P_{k/l}$ is the orthogonal projector onto $S_{k/l}^{\perp}$ where $S_{k/l}$ is the space spanned by the columns of $\bar{\mathbf{A}}$. ($\bar{\mathbf{A}}$ is formed by deleting the columns of A_l from the full rank matrix $\mathbf{A}$.)

Proof: First we prove that $b^T W b = \xi > 0$ where b is any n-vector outside the span of B, B is any $n \times t$ full rank matrix and W is the $n \times n$ orthogonal projector onto $B^{\perp}$. Since B is full rank, there exists an $n \times n$ orthogonal matrix Q and a $t \times t$ nonsingular upper-triangular matrix R such that

$$B = [Q_1 Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R$$

where Q_1 denotes the first t columns and Q_2 denotes the remaining $n-t$ columns of Q. It can then be shown that $W = Q_2 Q_2^T$. Thus

$$b^T W b = (Q_2^T b)^T (Q_2^T b) = \xi > 0$$

NOTE: $b^T W b = 0 \;=>\; Q_2^T b = 0 \;=>$ there exists a $z \in \mathbf{R}^t$ *s.t.* $b = Q_1 z$

$=>$ $b = B\tilde{z}$ where $\tilde{z} = R^{-1} z$

$=>$ $b \in \text{span}(B)$

$=>$ contradiction.

Now define the transformation $T : \mathbf{R}^{\mu \times \nu} \rightarrow \mathbf{R}^{2\mu \times 2\nu}$ as follows:

$$T \begin{bmatrix} \beta_{11} & \cdots & \beta_{1\nu} \\ \vdots & & \vdots \\ \beta_{\mu 1} & \cdots & \beta_{\mu\nu} \end{bmatrix} = \begin{bmatrix} \beta_{11} I & \cdots & \beta_{1\nu} I \\ \vdots & & \vdots \\ \beta_{\mu 1} I & \cdots & \beta_{\mu\nu} I \end{bmatrix}$$

where $\beta_{ij} \in \mathbf{R}$ for $i = 1, \ldots, \mu$ and $j = 1, \ldots, \nu$ and I represents the standard 2×2 identity matrix. Obviously there exists a vector b and a matrix B such that $A_l = T(b)$ and $\bar{\mathbf{A}} = T(B)$.

Finally, if we note that whenever the product XY is defined then $T(XY) = T(X)T(Y)$ and similiarly, whenever the sum $X+Y$ is defined then $T(X+Y) = T(X) + T(Y)$.

Furthermore, if there is an upper-triangular matrix U then $T(U)$ is also upper-triangular and $T(I) = I$. Thus, if we apply this transformation to the foregoing proof we obtain

$$A_l^T P_{k/l} A_l = \rho I \quad \text{with } \rho > 0.$$

SIAM J. SCI. STAT. COMPUT.
Vol. 1, No. 4, December 1980

0196-5204/80/0104-0010 $01.00/0

A STABLE ALGORITHM FOR SOLVING THE MULTIFACILITY LOCATION PROBLEM INVOLVING EUCLIDEAN DISTANCES*

P. H. CALAMAI† AND A. R. CONN‡

Abstract. There is a rapidly growing interdisciplinary interest in the application of location models to real life problems. Unfortunately, the current methods used to solve the most popular minisum and minimax location problems are computationally inadequate. A more unified and numerically stable approach for solving these problems is proposed. Detailed analysis is done for the linearly constrained Euclidean distance minisum problem for facilities located in a plane. Preliminary computational experience suggests that this approach compares favourably with other methods.

Key words. continuous location problems, nonsmooth optimization, numerical linear algebra

1. Introduction. Facility location problems are generally concerned with finding the optimal location of a set of new facilities in a network of existing facilities. Over the past few decades various forms of the problem have been developed and addressed in the literature (see Francis and Goldstein [11]). In this paper we present a stable method for solving the multifacility location problem involving Euclidean distances. We assume that the feasible region is connected and that the parameters are static with respect to time (for a discussion of the dynamic aspect of these problems see Wesolowsky and Truscott [17]). In addition, we assume that the problem is well formulated (Francis and Cabot [10] formally describe this and many other properties of the problem).

Various approaches have been used in an attempt to efficiently solve this location problem. One such method involves the use of a heuristic algorithm described by Vergin and Rogers [14]. Unlike their approach, most solution techniques guarantee optimality. Included among these are the convex programming approach described by Love [13], the hyperbolic approximation method used by Eyster, White and Wierwille [9], the pseudo-gradient technique described by Calamai and Charalambous [3] and the subgradient algorithm presented by Chatelon, Hearn and Lowe [5].

One property of this problem that causes considerable difficulty is the fact that the objective function is not everywhere differentiable. This nondifferentiability occurs whenever any new facility coincides with any other new facility or with an existing facility. As a result, standard minimization techniques cannot be directly applied. We overcome this difficulty by projecting onto a particular affine space in which the current near zero terms stay unchanged and the remaining well-defined terms can be decreased. Then, when we think we are optimal, we perform a linear refinement step that makes the near zero terms exactly zero. Under mild and suitable conditions we indicate that this technique converges to the solution of our problem.

Another significant property of this technique is that it can be used for the linearly constrained problem. Here we replace the constrained problem by a sequence of unconstrained problems by using a penalty function due to Pietrzykowski

*Received by the editors March 24, 1980. This work was supported in part by NSERC Grant No. A8639. This paper was typeset at the University of Waterloo using the Troff software developed for the Unix operating system. Final copy was produced on a Photon Econosetter.

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

[15] and Zangwill [18]. Using this method we are able to obtain a sequence of points (optimal to the unconstrained penalty function) which converge to the desired constrained optimum.

2. Problem statement and definitions. The multifacility minisum problem involving costs associated with Euclidean distances between facilities in $\mathbf{R}^q$ can be stated as: Find the point $x^{*T} = \{x_1^{*T}, \ldots, x_n^{*T}\}$ in $\mathbf{R}^{qn}$ to minimize

$$f(x) = \sum_{1 \leqslant j < k \leqslant n} v_{jk} \| x_j - x_k \| + \sum_{j=1}^{n} \sum_{i=1}^{m} w_{ji} \| x_j - p_i \| , \tag{2.1}$$

where

$n \triangleq$ number of new facilities (NF's) to be located.

$m \triangleq$ number of existing facilities (EF's).

$x_j^T = (x_{j1} \cdots x_{jq}) \triangleq$ vector location of NF_j in $\mathbf{R}^q$, $j = 1, \ldots, n$.

$p_i^T = (p_{i1} \cdots p_{iq}) \triangleq$ vector location of EF_i in $\mathbf{R}^q$, $i = 1, \ldots, m$.

$v_{jk} \triangleq$ nonnegative constant of proportionality relating the l_2 distance between NF_j and NF_k to the cost incurred, $1 \leqslant j < k \leqslant n$.

$w_{ji} \triangleq$ nonnegative constant of proportionality relating the l_2 distance between NF_j and EF_i to the cost incurred, $1 \leqslant j \leqslant n$, $1 \leqslant i \leqslant m$.

$\| x_j - x_k \| = (\sum_{c=1}^{q} | x_{jc} - x_{kc} |^2)^{1/2} \triangleq l_2$ distance between NF_j and NF_k, $1 \leqslant j < k \leqslant n$.

$\| x_j - p_i \| = (\sum_{c=1}^{q} | x_{jc} - p_{ic} |^2)^{1/2} \triangleq l_2$ distance between NF_j and EF_i, $1 \leqslant j \leqslant n$, $1 \leqslant i \leqslant m$.

In order to simplify the remainder of the discussion we set $q = 2$ (i.e., all facilities are in $\mathbf{R}^2$). Now if we let $\eta = (n^2 - n)/2$ and $\tau = \eta + mn$, and if we define the constants

$$\begin{aligned}
\{\alpha_1, \ldots, \alpha_\tau\} &= \{\underbrace{v_{12}, \ldots, v_{1n}}_{n-1}, \underbrace{v_{23}, \ldots, v_{2n}}_{n-2}, \ldots, \underbrace{v_{n-1n}}_{1}, \underbrace{w_{11}, \ldots, w_{1m}}_{m}, \ldots, \underbrace{w_{n1}, \ldots, w_{nm}}_{m}\}, \\
\{\beta_1, \ldots, \beta_\tau\} &= \{\underbrace{1, \ldots, 1}_{n-1}, \quad \underbrace{2, \ldots, 2}_{n-2}, \quad \ldots, \underbrace{n-1}_{1}, \quad \underbrace{1, \ldots, 1}_{m}, \quad \ldots, \quad \underbrace{n, \ldots, n}_{m}\}, \\
\{\gamma_1, \ldots, \gamma_\tau\} &= \{\underbrace{2, 3, \ldots, n}_{n-1}, \quad \underbrace{3, 4, \ldots, n}_{n-2}, \quad \ldots, \quad \underbrace{n}_{1}\},
\end{aligned}$$

and the index set $M = \{1, \ldots, \tau\}$ then (2.1) can be restated more conveniently as

$$\text{minimize} f(x) = \sum_{i \in M} \| A_i^T x - b_i \| , \tag{2.2}$$

where the 2×2n matrix A_i^T and the 2×1 vector b_i are defined by

$$A_i^T = \begin{cases} [0_1 \quad \alpha_i I \quad 0_2 \quad -\alpha_i I \quad 0_3], & i = 1, \dots, \eta, \\ [0_1 \quad \alpha_i I \quad 0_4], & i = \eta + 1, \dots, \tau, \end{cases}$$

$$b_i = \begin{cases} 0_5, & i = 1, \dots, \eta, \\ \alpha_i p_e, & i = \eta + 1, \dots, \tau, \end{cases}$$

and

I is a 2×2 identity matrix,

$0_1, 0_2, 0_3, 0_4$ and 0_5 are zero matrices of dimension $2\times 2(\beta_i - 1)$, $2\times 2(\gamma_i - \beta_i - 1)$, $2\times 2(n - \gamma_i)$, $2\times 2(n - \beta_i)$ and 2×1 respectively,

$e = (i - \eta) \bmod m$ and $p_0 \equiv p_m$.

In a manner analogous to that of Bartels, Conn and Sinclair [2], define for any point x in $\mathbf{R}^{2n}$: the residual vectors

$$r_i(x) = A_i^T x - b_i \quad \forall\, i \in M,$$

the index set

$$I_\epsilon(x^k) = \{i \in M \mid \| r_i(x^k) \| \leqslant \epsilon f(x^{k-1})/\tau\} = \{i_1, \dots, i_{t_k}\}, \qquad \epsilon \geqslant 0,$$

the corresponding matrix and vector

$$\text{(2.3)} \quad \mathbf{A} = \mathbf{A}(x^k) = [A_{i_1}, \dots, A_{i_{t_k}}] \text{ and } r(x^k)^T = [r_{i_1}(x^k)^T, \dots, r_{i_{t_k}}(x^k)^T],$$

and the vector

$$\tilde{\nabla} f_k = \sum_{i \in M \setminus I_\epsilon(x^k)} \nabla(\| r_i(x^k) \|) = \sum_{i \in M \setminus I_\epsilon(x^k)} A_i r_i(x^k) / \| r_i(x^k) \| .$$

(Note that for $\epsilon = 0$ the index set $I_\epsilon(x^k)$ corresponds to those terms which are exactly *active* (binding) at the point x^k. Conn [7] gives an account of the necessity for considering, as we do, "*near-active*" terms.)

Finally, define P_k as the orthogonal projector onto $S_k^\perp$, where S_k is the space spanned by the columns of $\mathbf{A}(x^k)$. *For the present it will be assumed that* $\mathbf{A}(x^k)$ *has full rank.*

3. Optimality conditions. For all points x close enough to x^k in $\mathbf{R}^{2n}$ we have $I_\epsilon(x) = I_\epsilon(x^k)$. Furthermore,

$$\text{(3.1)} \quad \begin{aligned} f(x) &= \sum_{i \in M \setminus I_\epsilon(x^k)} \| r_i(x) \| + \sum_{i \in I_\epsilon(x^k)} \| r_i(x) \| \\ &= h_1(x) + h_2(x). \end{aligned}$$

Assume $P_k \nabla h_1(x^k) \neq 0$ (i.e., $P_k \tilde{\nabla} f_k \neq 0$); then for $d^k = -P_k \tilde{\nabla} f_k$ we have

$$\text{(3.2a)} \quad \begin{aligned} h_1(x^k + \lambda d^k) &= h_1(x^k) + \lambda (d^k)^T \nabla h_1(x^k) + O(\lambda^2) \\ &= h_1(x^k) - \lambda \| P_k \tilde{\nabla} f_k \|^2 + O(\lambda^2), \end{aligned}$$

and since $A_i^T d^k = 0$ for all $i \in I_\epsilon(x^k)$,

$$h_2(x^k+\lambda d^k) = h_2(x^k). \tag{3.2b}$$

Combining equations (3.2a) and (3.2b) yields the following result:

$$f(x^k+\lambda d^k)-f(x^k) = -\lambda\| P_k\tilde{\nabla} f_k\|^2+O(\lambda^2). \tag{3.3}$$

Therefore for $d^k = -P_k\tilde{\nabla} f_k \neq 0$ there exists $\delta>0$ such that $f(x+\lambda d)<f(x)$ for all $0<\lambda<\delta$. Alternatively, if $P_k\nabla f_k = 0$ then *(under the linear independence assumption)* the corresponding point x^k is called a *(nondegenerate) dead point.* In this case ∇f_k can be expressed *(uniquely)* as a linear combination of the columns of $\mathbf{A}(x^k)$,

$$\tilde{\nabla} f_k = \mathbf{A}u = \sum_{i\in I_\epsilon(x^k)} A_i u_i, \qquad u^T = [u_1^T \cdots u_{i_k}^T]. \tag{3.4}$$

Then, for any d^k,

$$\begin{aligned} h_1(x^k+\lambda d^k) &= h_1(x^k)+\lambda(d^k)^T\tilde{\nabla} f_k+O(\lambda^2) \\ &= h_1(x^k)+\lambda \sum_{i\in I_\epsilon(x^k)} u_i^T A_i^T d^k+O(\lambda^2), \end{aligned} \tag{3.5a}$$

and

$$\begin{aligned} h_2(x^k+\lambda d^k) &= h_2(x^k)+\lambda\left[\sum_{i\in I_0(x^k)} \| A_i^T d^k\| \right. \\ &\qquad \left. + \sum_{i\in I_\epsilon(x^k)\backslash I_0(x^k)} \nabla^T(\| r_i(x^k)\|)d^k\right]+O(\lambda^2) \\ &= h_2(x^k)+\lambda\left[\sum_{i\in I_0(x^k)} \| A_i^T d^k\| \right. \\ &\qquad \left. + \sum_{i\in I_\epsilon(x^k)\backslash I_0(x^k)} \frac{r_i(x^k)^T A_i^T d^k}{\| r_i(x^k)\|}\right]+O(\lambda^2). \end{aligned} \tag{3.5b}$$

Combining equations (3.5a) and (3.5b) yields the following result:

$$\begin{aligned} f(x^k+\lambda d^k)-f(x^k) &= \lambda\left[\sum_{i\in I_0(x^k)} (u_i^T A_i^T d^k+\| A_i^T d^k\|) \right. \\ &\qquad \left. + \sum_{i\in I_\epsilon(x^k)\backslash I_0(x^k)} \left(u_i^T A_i^T d^k+\frac{r_i(x^k)^T A_i^T d^k}{\| r_i(x^k)\|}\right)\right]+O(\lambda^2). \end{aligned} \tag{3.6}$$

Now, if there exists an index $l\in I_\epsilon(x^k)$ such that $\| u_l\| > 1$, then take

$$d^k = -P_{k/l}A_l u_l, \tag{3.7}$$

where $P_{k/l}$ is the orthogonal projector onto $S_{k/l}^{\perp}$ and $S_{k/l}$ is the space spanned by the columns of $\mathbf{A}(x^k)$ with columns A_l deleted. Then, since $A_i^T d^k = 0$ for all $i\in I_\epsilon(x^k)-\{l\}$ and $A_l^T d^k = -A_l^T P_{k/l}A_l u_l = -\rho u_l$ where $\rho>0$ (see Appendix), we have, for sufficiently small $\lambda > 0$,

$$f(x^k + \lambda d^k) - f(x^k) = \begin{cases} -\lambda\rho \left[\| u_l \|^2 + \dfrac{r_l(x^k)^T u_l}{\| r_l(x^k) \|} \right] < 0, & l \in I_\epsilon(x^k) \backslash I_0(x^k) \quad (3.8a) \\ -\lambda\rho \| u_l \| (\| u_l \| - 1) < 0, & l \in I_0(x^k) \quad (3.8b) \end{cases}$$

Thus, if there exists $l \in I_\epsilon(x^k)$ such that $\| u_l \| > 1$ then, for $d = -P_{k//l} A_l u_l$, there exists a $\delta > 0$ such that $f(x + \lambda d) < f(x)$ for all $0 < \lambda < \delta$. What happens if x^k is a dead point and $\| u_i \| \leqslant 1 \;\; \forall i \in I_\epsilon(x^k)$?

Case I. $I_\epsilon(x^k) \backslash I_0(x^k) = \varnothing$.

Under these circumstances we have

$$(3.9) \qquad f(x^k + \lambda d^k) - f(x^k) = \lambda \sum_{i \in I_0(x^k)} (u_i^T A_i^T d^k + \| A_i^T d^k \|) + O(\lambda^2),$$

which is nonnegative for all d^k in $\mathbf{R}^{2n}$. Thus x^k is at least a weak local minimum and hence a global minimum of the convex function $f(x)$.

Case II. $I_\epsilon(x^k) \backslash I_0(x^k) \neq \varnothing$.

For any $l \in I_\epsilon(x^k) \backslash I_0(x^k)$ take $d^k = -P_{k//l} A_l r_l(x^k)$. For this choice we have $A_i^T d^k = 0 \;\; \forall i \in I_\epsilon(x^k) - \{l\}$ and (3.6) becomes

$$(3.10) \qquad f(x^k + \lambda d^k) - f(x^k) = -\lambda\rho(u_l^T r_l(x^k) + \| r_l(x^k) \|) + O(\lambda^2),$$

which is nonpositive (strictly negative unless $u_l = -r_l(x^k) / \| r_l(x^k) \|$).The next section explains a method for avoiding this case.

4. The linear refinement. If we are at the point x^k, and $\| P_k \tilde{\nabla} f_k \|$ is "*small*", then we may be approaching a dead point, say $\hat{x}$. We can then obtain estimates $\{\hat{u}_i\}$ of the "*dual*" variables $\{u_i\}$ by finding the least-squares solution to

$$(4.1) \qquad \tilde{\nabla} f_k = \sum_{i \in I_\epsilon(x^k)} A_i u_i.$$

Then if $\| \hat{u}_i \| < 1 \;\; \forall i \in I_\epsilon(x^k)$ we may be near a local minimum (note that if $I_\epsilon(x^k) \backslash I_o(x^k) = \varnothing$ this is especially clear). We therefore wish to satisfy the near-active terms exactly. In [8], Conn and Pietrzykowski define a *vertical component*, based on linearizations, in an attempt to satisfy their "near-activities" exactly. Here we wish to find the solution to the already linear system of equations

$$(4.2) \qquad A_i^T(x^k - v^k) - b_i = 0 \quad \forall i \in I_\epsilon(x^k).$$

This is best accomplished by evaluating the least-squares solution of minimal norm to

$$(4.3) \qquad \mathbf{A}(x^k)^T v^k = r(x^k),$$

which is given, *algebraically at least*, by

$$(4.4) \qquad v^k = (\mathbf{A}(x^k)^+)^T r(x^k),$$

where $\mathbf{A}(x^k)^+$ is the generalized (Moore-Penrose) inverse of $\mathbf{A}(x^k)$. Note that once a refinement step is taken the set $I_\epsilon(x^k) \backslash I_0(x^k)$ is nullified and thus Case II above need not be considered.

5. Remarks on convergence. The optimality conditions (§3) along with the linear refinement (§4) show that either the point x^k is optimal or there exists a descent direction d^k such that, for small $\lambda > 0$,

$$f(x^k + \lambda d^k) - f(x^k) = \lambda (d^k)^T g \quad \text{with } (d^k)^T g < 0. \tag{5.1}$$

We take

$$d^k = -P_k \tilde{\nabla} f_k \quad \text{and} \quad g = \tilde{\nabla} f_k \tag{5.2}$$

if x^k is not a dead point, and

$$d^k = -P_{k/l} A_l u_l \quad \text{and} \quad g = \begin{cases} \tilde{\nabla} f_k - A_l u_l / \| u_l \|, & l \in I_0(x^k), \\ \tilde{\nabla} f_k + A_l r_l / \| r_l \|, & l \in I_\epsilon(x^k) \backslash I_0(x^k), \end{cases} \tag{5.3}$$

(where $l \in I_\epsilon(x^k)$ is an index giving $\| u_l \| > 1$) when x^k is a dead point. (As in [2], the vector g will be referred to as the *restricted gradient* of the function.)

Given a descent direction one is then able to determine a "sufficient" decrease in f as in [8, Prop. 1] (In practice we use the line search outlined in the next section.) One may thus analogously ([8, Thm. 3]) prove convergence. It is beyond the scope of this paper to give the theoretical details of these proofs. Furthermore, we are ultimately more interested in a second-order version and the subsequent convergence proofs.

6. The stepsize. Suppose we are at a point $x \in \mathbf{R}^{2n}$ and a direction d has been chosen as described. Further, suppose our function is of the form

$$\Phi(x) = \sum_{i \in M} \| \phi_i(x) \|, \tag{6.1}$$

where the functions $\phi_i(x)$ are all linear. Clearly a minimum of Φ, in the direction d, must be at a point $\bar{x}$ on d where, for at least one $i \in M \backslash I_\epsilon(x^k)$, we have $\phi_i(\bar{x}) = 0$. The points $\bar{x}$ are called "*breakpoints*". Therefore we determine the stepsize λ to the breakpoint that gives the minimum function value along d. In other words, *as long as the function is decreasing past the breakpoint we continue to move along d through successive breakpoints.*

To extend this idea to the case where the functions $\phi_i(x)$ are nonlinear is not difficult as long as we are: 1) content with estimating the location of breakpoints and, 2) willing to use a linear approximation in evaluating $d^T g(\bar{x})$.

In the first instance we wish to find the values of λ_i satisfying

$$A_i^T (x + \lambda_i d) - b_i = 0 \quad \forall i \in M \backslash I_\epsilon(x). \tag{6.2}$$

A good approximation to use, in the l_2 sense, comes from the equations:

$$\lambda_i = \frac{-(A_i^T d)^T r_i(x)}{\| A_i^T d \|^2} \quad \forall i \in M \backslash I_\epsilon(x). \tag{6.3}$$

In the second instance we wish to find an expression for $d^T g(x + \lambda d)$ which is linear in λ. Using the first-order Taylor series approximation about the point x, we get

$$d^T g(x + \lambda d) = d^T g(x) + \lambda \sum_{i \in M \backslash I_\epsilon(x)} \left\{ \frac{\| A_i^T d \|^2}{\| r_i(x) \|} - \frac{[(A_i^T d)^T r_i(x)]^2}{\| r_i(x) \|^3} \right\}. \tag{6.4}$$

With these ideas, the following stepsize algorithm (similar in concept to the one given in [4]) can be formulated:

```
1) Set:
        $\lambda_i \leftarrow \frac{-(A_i^T d)^T r_i(x)}{\| A_i^T d \|^2} \quad \forall i \in M \backslash I_\epsilon(x).$

        $J_1 \leftarrow \{ i \in M \backslash I_\epsilon(x) \mid \lambda_i > 0 \}.$

        $\Delta \leftarrow \sum_{i \in M \backslash I_\epsilon(x)} \left\{ \frac{\| A_i^T d \|^2}{\| r_i(x) \|} - \frac{[(A_i^T d)^T r_i(x)]^2}{\| r_i(x) \|^3} \right\}.$

        $\hat{\lambda} \leftarrow -\frac{d^T g(x)}{\Delta}.$

        $\nu \leftarrow 1,\ l_0 \leftarrow 0,\ \lambda_0 \leftarrow \hat{\lambda},\ \delta \leftarrow -\delta_0 d^T g(x).$

2) IF ($J_\nu = \emptyset$) THEN go to 5.
3) Determine $l_\nu \in J_\nu$ such that $\lambda_{l_\nu} \leqslant \lambda_i \quad \forall i \in J_\nu$.
4) IF ($\lambda_{l_\nu} \geqslant \hat{\lambda}$) THEN go to 5.
                     ELSE set $J_{\nu+1} \leftarrow J_\nu - \{l_\nu\}$
                              $\nu \leftarrow \nu + 1$ and go to 2.
5) Set $\tilde{x} \leftarrow x + \lambda_{l_{\nu-1}} d$.
6) IF ($f(\tilde{x}) < f(x) - \lambda_{l_{\nu-1}} \delta$) THEN set $x \leftarrow \tilde{x}$ and return.
                                   ELSE use $x$ and $\tilde{x}$ in a successive cubic
                                        interpolation routine to get a
                                        new $x$, and return.
```

Notes:

δ_0 is some preassigned positive constant.

The cubic line search used in step 6 need not be (and in our present implementation is not) exact.

7. The minimization algorithm.

1) Choose any $x^1 \in \mathbf{R}^{2n}$ and set $k \leftarrow 1$. Initialize $\epsilon_{\mathrm{ACT}}, \epsilon_{\mathrm{ZP1}}, \epsilon_{\mathrm{ZP2}}$ and ϵ. Set RSOL←.FALSE.

2) Evaluate all residuals. Identify all index sets, the vector $r(x^k)$ and the matrix $\mathbf{A}(x^k)$. Evaluate $\tilde{\nabla} f_k$ and construct P_k.

3) Set $d^k \leftarrow -P_k \tilde{\nabla} f_k$, and $g \leftarrow \tilde{\nabla} f_k$.

4) Consider retaining same ϵ_{ACT} - active set:

$$\text{IF } (\| d^k \| \geqslant \delta_1) \text{ THEN go to 10.}$$

5) Determine the "duals". Find the least-squares solution $\hat{u}$ to:

$$\mathbf{A} u = \tilde{\nabla} f_k.$$

6) Decide whether or not to drop an activity:

```
        IF ($\|\hat{u}_i\| \leqslant 1 \quad \forall i \in I_{\epsilon_{\mathrm{ACT}}}(x^k)$) THEN go to 8.
    7) Drop an activity:
        Choose $l \in I_{\epsilon_{\mathrm{ACT}}}(x^k)$ with $\|\hat{u}_l\| > 1$ and set $d^k \leftarrow -P_{k/l}A_l\hat{u}_l$.
        Set
            $g \leftarrow \begin{cases} \tilde{\nabla} f_k - A_l\hat{u}_l/\|\hat{u}_l\| & l \in I_0(x^k) \\ \tilde{\nabla} f_k + A_l r_l/\|r_l\| & l \in I_{\epsilon_{\mathrm{ACT}}}(x^k)\backslash I_0(x^k). \end{cases}$
        Go to 10.
    8) Consider checking for optimality:
        IF ($\|d^k\| > \delta_2$) THEN go to 10.
    9) Determine optimality or perform the linear refinement:
        IF (RSOL .OR. $r_k^\infty \leqslant \epsilon$) THEN STOP!
            ELSE Solve, in the least-squares sense, the system
                $\mathbf{A}^T v^k = r(x^k)$.
        IF ($f(x^k - v^k) < f(x^k)$) THEN set $x^{k+1} \leftarrow x^k - v^k$
                                           RSOL $\leftarrow$ .TRUE.
                                           $k \leftarrow k + 1$ and go to 2.
    10) Use the line search algorithm to find $x^{k+1}$. Set RSOL $\leftarrow$ .FALSE. ,
        $k \leftarrow k + 1$, and go to 2.
```

Notes:

$$r_k^\infty = \underset{i \in I_{\epsilon_{\mathrm{ACT}}}(x^k)}{\text{maximum}} (\|r_i(x^k)\|).$$

$$\delta_1 = \begin{cases} \epsilon_{\mathrm{ZP1}} f(x^k)/\tau, & I_{\epsilon_{\mathrm{ACT}}}(x^k) = \varnothing, \\ \epsilon_{\mathrm{ZP1}}\|g\|, & \text{otherwise.} \end{cases}$$

$$\delta_2 = \begin{cases} \epsilon_{\mathrm{ZP2}} f(x^k)/\tau, & I_{\epsilon_{\mathrm{ACT}}}(x^k) = \varnothing, \\ \epsilon_{\mathrm{ZP2}}\|g\|, & \text{otherwise.} \end{cases}$$

8. Linear constraints and the penalty function. Consider the following constrained form of the minisum multifacility location problem:

$$(8.1)\qquad \begin{aligned} \text{minimize } f(x) &= \sum_{i \in M} \|A_i^T x - b_i\| \\ \text{such that } r_i(x) &= a_i^T x - b_i \geqslant 0, \quad i = \tau + 1, \ldots, \tau', \\ r_i(x) &= a_i^T x - b_i = 0, \quad i = \tau' + 1, \ldots, \tau''. \end{aligned}$$

We can transform this problem into the penalty function of Pietrzykowski [15] and solve the sequence of unconstrained problems,

$$\text{(8.2)} \qquad \text{minimize } p(x,\mu) = \mu f(x) - \sum_{i \in LI} \min\,[0, r_i(x)] + \sum_{i \in LE} |\, r_i(x)|\,,$$

where μ is a positive parameter, $LI = \{\tau+1, \ldots, \tau'\}$ and $LE = \{\tau'+1, \ldots, \tau''\}$. It has been shown [15] that under *mild and suitable conditions*, an exact minimum of (8.2) coincides with an exact minimum of (8.1) for all values of μ sufficiently small. This suggests the following outer algorithm:

```
1) Choose $\mu > 0$.
2) Minimize $p(x, \mu)$ over $x$.
3) IF (we are optimal) THEN STOP!
4) Set $\mu \leftarrow \mu/10$ and go to 2.
```

The minimization in step 2 can be performed using obvious extensions of the techniques already described (see [1]). Some of the details of these extensions are now given.

For any x^k in $\mathbf{R}^{2n}$ define the index sets

$$\begin{aligned} I_\epsilon^{AI}(x^k) &= \{i \in LI \;\big|\; |\, r_i(x^k)| \leqslant \epsilon\} = \{i_{t_k+1}, \ldots, i_{t'_k}\}, \\ \text{(8.3)} \qquad I_\epsilon^{VI}(x^k) &= \{i \in LI \;\big|\; r_i(x^k) < -\epsilon\} \quad \text{and} \\ I_\epsilon^{AE}(x^k) &= \{i \in LE \;\big|\; |\, r_i(x^k)| \leqslant \epsilon\} = \{i_{t'_k+1}, \ldots, i_{t''_k}\}, \end{aligned}$$

and the vector

$$\tilde{\nabla} p_k = \mu \tilde{\nabla} f_k - \sum_{i \in I_\epsilon^{VI}(x^k)} a_i + \sum_{i \in LE \setminus I_\epsilon^{AE}(x^k)} \operatorname{sgn}\,[r_i(x^k)] a_i\,.$$

Redefine the matrix $\mathbf{A}(x^k)$, the vector $r(x^k)$ and the scalar r_k^∞ as

$$\mathbf{A}(x^k) = [A_{i_1}, \ldots, A_{i_{t_k}}, a_{i_{t_k+1}}, \ldots, a_{i_{t'_k}}, a_{i_{t'_k+1}}, \ldots, a_{i_{t''_k}}],$$

$$r(x^k)^T = [r_{i_1}(x^k)^T, \ldots, r_{i_{t_k}}(x^k)^T, r_{i_{t_k+1}}(x^k), \ldots, r_{i_{t'_k}}(x^k), r_{i_{t'_k+1}}(x^k), \ldots, r_{i_{t''_k}}(x^k)],$$

$$r_k^\infty = \underset{j=1,\ldots,t''_k}{\text{maximum}}(\| r_{i_j}(x^k) \|).$$

Now, if we replace $\tilde{\nabla} f_k$ by $\tilde{\nabla} p_k$ and $f(\cdot)$ by $p(\cdot, \mu)$, then our algorithm can be used to minimize $p(x, \mu)$ over x if we make the following changes to steps 6 and 7:

```
6) Decide whether or not to drop an activity:
        IF ($\| \hat{u}_i \| \leqslant 1 \;\; \forall i \in I_{\epsilon_{\mathrm{ACT}}}(x^k)$ .AND.
            $0 \leqslant \hat{u}_i \leqslant 1 \;\; \forall i \in I_\epsilon^{\mathrm{AI}}(x^k)$ ) THEN go to 8.
```

7) Drop an activity:

Choose $l \in I_{\epsilon_{\mathrm{ACT}}}(x^k)$ with $\| \hat{u}_j \| > 1$ and set $d^k = -P_{k/l}A_l\hat{u}_l$

.OR. Choose $l \in I_{\epsilon}^{\mathrm{AI}}(x^k)$ with $\hat{u}_l < 0$ and set $d^k = P_{k/l}a_l$.

$$\text{Set } g \leftarrow \begin{cases} \tilde{\nabla} f_k - A_l\hat{u}_l/\| \hat{u}_l \|, & l \in I_0(x^k), \\ \tilde{\nabla} f_k + A_l r_l/\| r_l \|, & l \in I_{\epsilon_{\mathrm{ACT}}}(x^k)\backslash I_0(x^k), \\ \tilde{\nabla} f_k, & l \in I_{\epsilon}^{\mathrm{AI}}, \end{cases}$$

go to 10.

(It can be shown that if there exists an index $l \in I_{\epsilon}^{\mathrm{AI}}(x^k)$ with $\hat{u}_l > 1$ or an index $l \in I_{\epsilon}^{\mathrm{AE}}(x^k)$ with $|\hat{u}_l| > 1$ then the direction $d = -\mathrm{sgn}(\hat{u}_l)P_{k/l}a_l$ is a descent direction for the penalty function. However, as pointed out by Bartels and Conn [1], constraint r_{j_l} will be violated at $x^k + \lambda d^k$ and, rather than do this, we take an alternative descent direction if one exists. Otherwise, we reduce μ in order to give more emphasis to the constraints.)

9. Implementation. Throughout this derivation it was assumed that the matrix $\mathbf{A}(x)$ was full rank in order to uniquely define the solution to various equations. This can no longer be accomplished when $\mathbf{A}(x)$ is rank deficient because of the inherent degeneracies encountered. Fortunately, this difficulty can be resolved in a straightforward manner computationally by using a method similar to Bartels et al. [2]. This involves treating all degeneracies as if they occurred due to the error introduced through the storage of data on a finite machine. Then the machine accuracy can be artificially reduced and the problem perturbed so that the degeneracies are resolved. Finally, after the perturbation, a unique solution can be attained which satisfies the original problem.

In particular, we do the following. After identifying the current active set, we randomly perturb all residuals associated with degenerate members of this set in such a way that those terms are no longer considered active. We then adjust the restricted gradient appropriately and proceed with the original algorithm. In this way, we either leave this degenerate neighbourhood, identify a solution among these perturbed vertices or enter a new degenerate neighbourhood. It should be noted that cycling cannot occur.

The full rank matrix $\mathbf{A}(x^k)$ can be factored, by forming a product of Givens transformations, into the form

$$\mathbf{A}(x^k) = Q^k \begin{bmatrix} R^k \\ 0 \end{bmatrix} = [Q_1^k \; Q_2^k] \begin{bmatrix} R^k \\ 0 \end{bmatrix}, \tag{9.1}$$

where Q^k is an orthonormal matrix and R^k is a nonsingular upper-triangular matrix. This numerically stable decomposition can easily be modified to accommodate the necessary additions and deletions of column vectors to the matrix $\mathbf{A}$ when it is updated (see [12]).

Since the columns of Q_2^k form an orthogonal basis for the space $S_k^{\perp}$, the projection matrix P_k can be computed as

$$P_k = (Q_2^k)(Q_2^k)^T, \tag{9.2}$$

In addition, the vector $\hat{u}$ required in step 5 of the minimization algorithm can be obtained efficiently by solving

$$R^k\hat{u} = (Q_1^k)^T\tilde{\nabla} f_k, \tag{9.3}$$

whereas the vector v^k required in step 10 can easily be obtained by first solving

$$(R^k)^T\bar{v}^k = r(x^k) \tag{9.4}$$

and then forming

$$v^k = Q_1^k\bar{v}^k. \tag{9.5}$$

10. Preliminary computational experience. Six small problems were run on a Honeywell 66/60 computer to compare the performance of this projection technique with other approaches. The problems used were chosen because of the availability of comparison data. The other solution methods considered are as follows:

i) Hyperbolic Approximation Procedure (HAP)

ii) Modified HAP (MHAP)

iii) The program of Calamai and Charalambous (MFLPVI)

A description of these methods and references to the data used in the six problems are given in [3]. A summary of the results is given below. The column labeled "NEW" refers to the projection algorithm.

Table 1
Comparative test results

#	MFLPVI	$\epsilon^{(0)} = 10^0$		$\epsilon^{(0)} = 10^{-4}$		NEW
		HAP	MHAP	HAP	MHAP	
1	77+487	1661	1381	2027	1407	64
2	34+114	647	546	4641	2281	17
3	16+47	87	70	770	197	8
4	15+16	45	45	45	45	17
5	40+183	142	114	1763	975	26
6	7+56	242	164	3743	1869	18
TOTAL	1092	2824	2320	12989	6774	150
OPS	112248	386592	318426	1647886	922390	77194

Remarks.

(a) Except for the last row, the figures in the table refer to the number of iterations required to reach the solution. Under the column heading MFLPV1, the first figure refers to the number of successful iterations whereas the second figure refers to the number of unsuccessful iterations.

(b) For our algorithm no attempt was made to choose ideal parameters. The choice, for the results shown, was $\epsilon_{ACT} = 0.1$, $\epsilon_{ZP1} = 0.2$, $\epsilon_{ZP2} = 0.05$ and $\epsilon = 7.45E-9$.

(c) The last row in the table gives an estimate of the total number of addition operations required by each of the techniques in solving the six problems. With the exception of the MFLPV1 approach approximately the same number of multiplications would be required. There are usually more addition than multiplication operations in MFLPV1 due to combinatorial situations which sometimes arise. (For these six problems there are approximately 17 percent more addition than multiplication operations in MFLPV1.)

(d) The following are formulas for evaluating the number of addition operations per iteration for the four techniques (with insignificant terms neglected).

$$\text{HAP and MHAP} \;\{\; 6\tau + 2n;$$

$$\text{MFLPV1} \begin{cases} v & \text{if unsuccessful,} \\ v + 3\tau + 4n + 2m & \text{if successful;} \end{cases}$$

$$\text{NEW} \begin{cases} w & \text{if §7 step 4 satisfied,} \\ w + 2t(2t+1) + 8nt & \text{if §7 step 6 satisfied,} \\ w + t(2t+1) + 8n(n-t) + 4(nt+1) & \text{if §7 step 6 not satisfied;} \end{cases}$$

where

$$w = 3\tau + 4(\tau - t) + 16(n-t)^2 + 8t^2 + 2n,$$

$$v = \begin{cases} 4\{\sum_{z \in Z} \sum_{k=2}^{z-1} k\binom{z}{k} + \sum_{k=2}^{z''} k\binom{z'}{k}\} & \text{for combinatorial case,} \\ \tau + 5n + 3m & \text{otherwise;} \end{cases}$$

and (see [3]):

$Z \triangleq$ set containing cardinalities of all clusters unsuccessfully tested,

$z' \triangleq$ cardinality of cluster successfully tested,

$z'' \triangleq$ cardinality of subset moved ($z'' \leqslant z' \leqslant n-1$).

(e) The results and the theory suggest that the new techniques will work well with larger and/or more difficult problems.

11. Conclusions and recommendations.

(a) The purpose of this paper is to provide a simple stable method for solving the multifacility location problem involving Euclidean distances; however, the same ideas can be, and perhaps should be, applied to location problems that use other distance measures (l_1 and l_∞ for example). This would provide a more unified approach to solving this class of problems.

(b) The method derived for handling linear constraints can be modified to handle any continuous functions mapping $\mathbb{R}^{2n} \to \mathbb{R}^1$ by considering local linearizations of those functions.

(c) Instead of this method that essentially uses steepest descent in a subspace, a more sophisticated approach would involve the use of second-order information to derive quasi-Newton steps in a subspace. Such a technique would undoubtedly lead to an improvement in the rate of convergence. See [6].

(d) For large problems the sparsity and structure of the matrix $\mathbf{A}(x)$ would have to be given attention. This would suggest the use of decomposition techniques as well as the use of methods that maintain sparsity in updating these decompositions whenever columns were added to or deleted from $\mathbf{A}$.

(e) The stepsize algorithm outlined in this paper is not necessarily optimal. Many alternative schemes are available, including:

1) the reordering of breakpoints based on the magnitude of the least-squares error in their estimation,

2) the use of a more exact cubic line search either with or without the breakpoint analysis,

3) the use of a more sophisticated steplength algorithm. See [14].

(f) Step 7 of our algorithm results in the release of an index, associated with an "out-of-kilter" dual, from the activity set. In our implementation we drop the index (and thus the corresponding activity) associated with the first "out-of-kilter" dual encountered. This choice may not result in the optimal descent direction.

All these ideas are currently being investigated.

Appendix. *Prove:*

$$A_l^T P_{k/l} A_l = \rho I, \qquad \rho > 0,$$

where $P_{k/l}$ is the orthogonal projector onto $S_{k/l}^{\perp}$ where $S_{k/l}$ is the space spanned by the columns of $\bar{\mathbf{A}}$. ($\bar{\mathbf{A}}$ is formed by deleting the columns of A_l from the full rank matrix $\mathbf{A}$.)

Proof. First we prove that $b^TWb = \xi > 0$, where b is any n-vector outside the span of B, B is any $n \times t$ full rank matrix and W is the $n \times n$ orthogonal projector onto $B^{\perp}$. Since B is full rank, there exists an $n \times n$ orthogonal matrix Q and a $t \times t$ nonsingular upper-triangular matrix R such that

$$B = [Q_1 Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_1 R,$$

where Q_1 denotes the first t columns and Q_2 denotes the remaining $n-t$ columns of Q. It can then be shown that $W = Q_2 Q_2^T$. Thus

$$b^TWb = (Q_2^Tb)^T(Q_2^Tb) = \xi > 0.$$

Note that $b^TWb = 0 \Rightarrow Q_2^Tb = 0 \Rightarrow$ there exists a $z \in \mathbf{R}^t$ s.t. $b = Q_1 z$

$\Rightarrow b = B\tilde{z}$ where $\tilde{z} = R^{-1}z$

$\Rightarrow b \in \operatorname{span}(B)$

$\Rightarrow$ contradiction.

Now define the transformation $T: \mathbf{R}^{\mu \times \nu} \to \mathbf{R}^{2\mu \times 2\nu}$ as follows:

$$T \begin{bmatrix} \beta_{11} & \cdots & \beta_{1\nu} \\ \vdots & & \vdots \\ \beta_{\mu 1} & \cdots & \beta_{\mu\nu} \end{bmatrix} = \begin{bmatrix} \beta_{11} I & \cdots & \beta_{1\nu} I \\ \vdots & & \vdots \\ \beta_{\mu 1} I & \cdots & \beta_{\mu\nu} I \end{bmatrix}$$

where $\beta_{ij} \in \mathbf{R}$ for $i=1, \ldots, \mu$ and $j=1, \ldots, \nu$ and I represents the standard 2X2 identity matrix. Obviously there exists a vector b and a matrix B such that $A_l = T(b)$ and $\overline{\mathbf{A}} = T(B)$.

Finally, we note that whenever the product XY is defined then $T(XY) = T(X)T(Y)$ and similiarly, whenever the sum $X+Y$ is defined then $T(X+Y) = T(X)+T(Y)$. Furthermore, if U is an upper-triangular matrix then $T(U)$ is also upper-triangular and $T(I) = I$. Thus, if we apply this transformation to the foregoing proof we obtain

$$A_l^T P_{k/l} A_l = \rho I \quad \text{with } \rho > 0.$$

Acknowledgement. The splendid typesetting was accomplished by Anne Trip-de-Roche and Brian Finch using the Photon Econosetter and Troff.

REFERENCES

[1] R. H. Bartels and A. R. Conn, *Linearly constrained discrete l_1 problems,* ACM TOMS, 6 (1980), pp. 594-608.

[2] R. H. Bartels, A. R. Conn, and J. W. Sinclair, *Minimization techniques for piecewise differentiable functions: The l_1 solution to an overdetermined linear system,* SIAM J. Numer. Anal., 15 (1978), pp. 224-241.

[3] P. H. Calamai, and C. Charalambous, *Solving multifacility location problems involving Euclidean distances,* Naval Res. Log. Quart., to appear.

[4] C. Charalambous and A. R. Conn, *An efficient method to solve the minimax problem directly,* SIAM J. Numer. Anal., 15 (1978), pp. 162-187.

[5] J. A. Chatelon, D. W. Hearn, and T. J. Lowe, *A subgradient algorithm for certain minimax and minisum problems,* Math. Prog., 15 (1978), pp. 130-145.

[6] T. F. Coleman and A. R. Conn, *Nonlinear programming via an exact penalty function method: A global method,* Comp. Sci. Tech. Rep. CS-80-31, University of Waterloo, Waterloo, Ontario, 1980 (to appear in Math. Prog.).

[7] A. R. Conn, *Constrained optimization using a nondifferentiable penalty function,* SIAM J. Numer. Anal., 10 (1973), pp. 760-784.

[8] A. R. Conn, and T. Pietrzykowski, *A penalty function method converging directly to a constrained optimum,* SIAM J. Numer. Anal., 14 (1977), pp. 348-375.

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

[10] R. L. Francis and A. V. Cabot, *Properties of a multifacility location problem involving Euclidean distances,* Naval Res. Log. Quart., 19 (1972), pp. 335-353.

[11] R. L. Francis and J. M. Goldstein, *Location theory: A selective bibliography,* Oper. Res., 22 (1974), pp. 400-410.

[12] P. E. Gill, G. H. Golub, W. Murray and M. Saunders, *Methods for modifying matrix factorizations,* Math. Comp., 28 (1974), pp. 505-535.

[13] R. F. Love, *Locating facilities in three-dimensional space by convex programming,* Naval Res. Log. Quart., 16 (1969), pp. 503-516.

[14] W. Murray and M. Overton, *Steplength algorithms for minimizing a class of non-differentiable functions,* Computing, 23 (1979), pp. 309-331.

[15] T. Pietrzykowski, *An exact potential method for constrained maxima,* SIAM J. Numer. Anal., 6 (1969), pp. 299-304.

[16] R. C. Vergin and J. D. Rogers, *An algorithm and computational procedure for locating economic facilities*, Management Sci., 13 (1967), pp. 240-254.

[17] G. O. Wesolowsky and W. G. Truscott, *Dynamic location of multiple facilities for a class of continuous problems*, AIIE Trans., 8 (1976), pp. 76-83.

[18] W. I. Zangwill, *Nonlinear programming via penalty functions*, Management Sci., 13 (1962), pp. 344-350.