A PROJECTION METHOD FOR THE UNCAPACITATED FACILITY LOCATION PROBLEM

A.R. Conn
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

G. Cornuéjols
Graduate School of Industrial Administration
Carnegie-Mellon University
Pittsburgh, Pennsylvania, USA

CS-87-19

Acknowledgement: This work was done mainly while the two authors were visiting ARTEMIS, University of Grenoble, France. Both are extremely grateful for the generous hospitality of their hosts and the financial support of CNRS. The work was supported in part by NSERC grant A8639 and NSF Grant 8601660.

Abstract: Several algorithms already exist for solving the uncapacitated facility location problem. The most efficient are based upon the solution of the strong linear programming relaxation. The dual of this relaxation has a condensed form which consists of minimizing a certain piecewise linear convex function. This paper presents a new method for solving the uncapacitated facility location problem based upon the exact solution of the condensed dual via orthogonal projections. The amount of work per iteration is of the same order as that of a simplex iteration for a linear program in m variables and constraints, where m is the number of clients. For comparison, the underlying linear programming dual has mn + m + n variables and mn + n constraints, where n is the number of potential locations for the facilities. The method is flexible as it can handle side constraints. In particular, when there is a duality gap, the linear programming formulation can be strengthened by adding cuts. Numerical results for some classical test problems are included.

1. Statement of the Problem

The uncapacitated facility location problem can be stated as follows. Suppose we have m clients indexed by $I = \{1, 2, \ldots, m\}$ and n potential sites for opening facilities indexed by $J = \{1, 2, \ldots, n\}$. We are given the profit c_{ij} that can be accrued from supplying all of client i's demand from a facility in location j, and the fixed cost $f_j \geq 0$ for setting up a facility in location j. The problem consists of selecting an optimal set of facility locations and assigning the clients to these facilities. Let y_{ij} represent the fraction of client i's demand supplied from facility j. Define

$$x_j = \begin{cases} 1 \text{ if facility } j \text{ is open,} \\ 0 \text{ otherwise.} \end{cases}$$

Then our model problem is

$$\text{Max} \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij} - \sum_{j \in J} f_j x_j \tag{1.1}$$

$$\sum_{j \in J} y_{ij} = 1 \qquad \text{for all } i \in I \tag{1.2}$$

$$x_j - y_{ij} \geq 0 \qquad \text{for all } i \in I,\ j \in J \tag{1.3}$$

$$y_{ij} \geq 0 \qquad \text{for all } i \in I,\ j \in J \tag{1.4}$$

$$x_j \in \{0,1\} \qquad \text{for all } j \in J. \tag{1.5}$$

The constraint (1.2) expresses the fact that all of client i's demand is supplied, and (1.3) expresses the fact that we can only supply the clients from open facilities.

This model has been extensively studied. See [7] for a recent survey of the literature. By relaxing the 0,1 restriction on x_j, we obtain the so-called *strong linear programming relaxation* of the uncapacitated facility location problem. More precisely, we replace (1.5) in the above formulation by

$$0 \leq x_j \leq 1 \quad \text{for all } j \in J. \tag{1.6}$$

This relaxation has been very effective in practice. It turns out that its solution is frequently integral for small and medium-size problems arising from applications (see, for example, ReVelle and Swain [19], Garfinkel, Neebe and Rao [11], Cornuejols, Fisher and Nemhauser [6], Erlenkotter[10], Mulvey and Crowder [17]). For large Euclidean problems, the existence of very small duality gaps (about 0.2%) is supported by a probabilistic analysis [1]. Much larger gaps arise under the uniform cost model but this model is not representative of real-world applications.

We also note that, because of the size of the strong linear programming relaxation (it has mn + n variables) and its special structure, it is not efficient to use the simplex method directly. The standard linear programming dual of problem (1.1)-(1.4),(1.6) can be written as

$$\text{Min} \quad \sum_{i \in I} u_i + \sum_{j \in J} t_j \tag{1.7}$$

$$u_i + w_{ij} \geq c_{ij} \qquad \text{for all } i \in I,\ j \in J \tag{1.8}$$

$$-\sum_{i \in I} w_{ij} + t_j \geq -f_j \qquad \text{for all } j \in J \tag{1.9}$$

$$w_{ij},\ t_j \geq 0 \qquad \text{for all } i \in I,\ j \in J. \tag{1.10}$$

This is a problem in mn + m + n variables, but it is possible to rewrite it in a condensed form in the light of the following observations.

We note from the form of (1.7) that, for any given u_i's, we would like to make the t_j's as small as possible. Thus, using (1.9) and (1.10), we require

$$t_j = \left[\sum_{i \in I} w_{ij} - f_j \right]^+, \tag{1.11}$$

where $a^+ \equiv \max(0,a)$. Consequently, we would like to make w_{ij} as small as possible. Constraints (1.8) and (1.10) imply that we should have

$$w_{ij} = (c_{ij} - u_i)^+. \tag{1.12}$$

It is now possible to replace the dual problem (1.7)-(1.10) above by the following *condensed dual*

$$\underset{u}{\text{Min}}\ F(u) = \sum_{i \in I} u_i + \sum_{j \in J} S_j^+(u) \tag{1.13}$$

where $$S_j(u) = \sum_{i \in I} (c_{ij} - u_i)^+ - f_j. \tag{1.14}$$

We remark that

(a) this transformation is well known. See, for example, Spielberg [21] and Erlenkotter [10].

(b) F(u) is a piecewise linear convex objective function.

(c) Problem (1.13) is an unconstrained optimization problem in n variables.

(d) There is an optimum solution of (1.13) such that $S_j(u) \leq 0$ for all $j \in J$. (To see this, note that the constraint $x_j \leq 1$ of (1.6) is superfluous in the strong linear programming relaxation since we have assumed $f_j \geq 0$. This shows that, in the dual, we can always set $t_j = 0$.)

In this paper we propose a method that minimizes F(u) directly as a piecewise linear function. Section 2 outlines the method and provides the theory. Section 3 contains the proof of the main theorem. Section 4 describes the algorithm, and Section 5 reports our computational experience. Section 6 describes connections between our method and Erlenkotter's. Finally, in Section 7, we discuss extensions of the method.

In the remainder of this section we provide additional background on the uncapacitated facility location problem. This problem is NP-hard. Therefore it is not surprising that most of the exact solution methods proposed in the literature resort to branch and bound. The success of such algorithms depends on the availability of a tight relaxation. The so-called *weak* linear programming relaxation is defined by replacing (1.3) by

$$\sum_{i \in I} y_{ij} \leq m x_j \quad \text{for all } j \in J. \qquad (1.3').$$

Although this relaxation is very easy to solve, its use within the context of branch and bound leads to large enumeration trees, even for relatively small problems (see Efroymson and Ray [9]). The strong linear programming relaxation (1.1)-(1.4),(1.6) on the other hand produces amazingly tight bounds, as we have noted already. Solving it is an interesting challenge as its structure can be exploited in many different ways. Garfinkel, Neebe and Rao [11] solved the strong linear programming relaxation by Dantzig-Wolfe decomposition, Schrage [20] devised a variable upper bound simplex algorithm to handle the constraints (1.3) while Morris [16] treated them as cutting planes to be incorporated as needed; finally Guignard and Spielberg [14] proposed to pivot only on unimodular bases. All these methods are variants of the primal simplex algorithm. For the purpose of branch and bound, however, there are advantages to solving the dual of the relaxation instead of the primal, as any dual feasible solution yields a valid bound. We have seen already that the dual has a condensed form (1.13). Erlenkotter [10] minimized this piecewise linear convex function using a descent heuristic. Narula, Ogbu and Samuelsson [18] and Cornuejols, Fisher and Nemhauser [6] used subgradient optimization. Both approaches quickly yield good dual solutions, are easy to program and well suited for branch and bound algorithms.

Recently some great successes have been achieved in the solution of combinatorial optimization problems by combining a cutting plane approach with branch and bound, see Grotschel and Padberg [12] for example. In order to generate cutting planes for the uncapacitated facility location problem, it is desirable to solve the strong linear programming relaxation to optimality. Since the primal has many more variables than the condensed dual -- mn + n

versus m --, it seems appropriate to solve the latter. Subgradient optimization can be very slow to achieve optimality and Erlenkotter's dual descent algorithm is a heuristic, so neither approach is well suited to solving the condensed dual optimally. The present paper proposes an algorithm to solve this condensed dual to optimality.

The best cutting planes are those that generate facets of the convex hull of the solutions to (1.2) - (1.5). This polytope is known as the uncapacitated facility location polytope, and its facets have been partially described by Guignard [13], Cho, Padberg and Rao [3] and Cornuejols and Thizy [8]. For example, the inequality

$$y_{r\ell} + y_{s\ell} + y_{sh} + y_{th} + y_{tk} + y_{rk} - x_\ell - x_h - x_k \leq 1 \qquad (1.15)$$

defines a facet of the uncapacitated facility location polytope for any ℓ, h, k ε J and r, s, t ε I such that $\ell \neq h \neq k$ and $r \neq s \neq t$. It cuts off fractional basic solutions of (1.1)-(1.4),(1.6) where all the variables in (1.15) take the value 1/2. Adding the constraint (1.15) to (1.1)-(1.4),(1.6) and taking the dual, we get

$$\text{Min} \sum_{i\varepsilon I} u_i + \sum_{j\varepsilon J} t_j + v$$

$$u_i + w_{ij} \geq c_{ij} \quad \text{for all } (i,j) \neq (r,\ell),(s,\ell),(s,h),(t,h),(t,k),(r,k)$$

$$u_i + w_{ij} + v \geq c_{ij} \quad \text{for } (i,j) = (r,\ell),(s,\ell),(s,h),(t,h),(t,k) \text{ or } (r,k)$$

$$-\Sigma w_{ij} + t_j \geq -f_j \quad \text{for all } j \neq \ell, h, k$$

$$-\Sigma w_{ij} + t_j - v \geq -f_j \quad \text{for } j = \ell, h \text{ or } k$$

$$w_{ij},\ t_j,\ v \geq 0.$$

Therefore the new condensed dual is

$$\underset{v\geq 0,u}{\text{Min}} \quad F(u,v) = \sum_{i\varepsilon I} u_i + \sum_{j\varepsilon J} S_j^+(u,v) + v \qquad (1.16)$$

where $$S_j(u,v) = \sum_{i\varepsilon I} (\bar{c}_{ij} - u_i)^+ - \bar{f}_j \qquad (1.17)$$

$$\text{and } \bar{c}_{ij} = \begin{cases} c_{ij} & \text{for } (i,j) \neq (r,\ell),(s,\ell),(s,h),(t,h),(t,k),(r,k) \\ c_{ij} - v & \text{for } (i,j) = (r,\ell),(s,\ell),(s,h),(t,h),(t,k) \text{ or } (r,k) \end{cases}$$

$$\bar{f}_j = \begin{cases} f_j & \text{for } j \neq \ell,\ h,\ k \\ f_j - v & \text{for } j = \ell,\ h \text{ or } k. \end{cases}$$

This example shows that the general form of the condensed dual is preserved when a cutting plane such as (1.15) is added. More generally, if p constraints are added to the primal formulation, the condensed dual has p new variables. Except for the nonnegativity of these variables, the new condensed dual is still the unconstrained minimization of a convex piecewise linear function. For the same reason, the potential extensions of the condensed dual include the capacitated facility location problem. We will not treat this latter extension in this paper, but both extensions further justify our interest in the condensed dual F(u).

2. Motivation and Theory

As we have already seen, we are initially concerned with the following optimization problem

$$\underset{u \in \mathbb{R}^m}{\text{Min}} \quad F(u) = \sum_{i=1}^{m} u_i + \sum_{j=1}^{n} S_j^+(u) \tag{2.1}$$

where $$S_j(u) = \sum_{i=1}^{m} (c_{ij} - u_i)^+ - f_j. \tag{2.2}$$

Clearly F(u) is a piecewise linear convex function.

It is nondifferentiable at all points u such that either

i) $S_j(u) = 0$ for some $j \in J = \{1, \ldots, n\}$ (2.3)

or ii) $c_{ij} = u_i$ for some $i \in I = \{1, \ldots, m\}$ with $S_j(u) > 0$. (2.4)

We call these activities (breakpoints) of type i and type ii, respectively.

We shall define the following index sets at a point u.

$$J^+(u) = \{j \in J : S_j(u) > 0\} \tag{2.5}$$

$$J^o(u) = \{j \in J : S_j(u) = 0\} \tag{2.6}$$

$$J^-(u) = \{j \in J : S_j(u) < 0\} \tag{2.7}$$

$$I_j^+(u) = \{i \in I : c_{ij} - u_i > 0\} \tag{2.8}$$

$$I_j^o(u) = \{i \in I : c_{ij} = u_i\} \tag{2.9}$$

$$I_j^-(u) = \{i \in I : c_{ij} - u_i < 0\} \tag{2.10}$$

$$I^o(u) = \bigcup_{j \in J^o(u) \cup J^+(u)} I_j^o(u) \tag{2.11}$$

$$J_i^+(u) = \{j \in J^+(u) : i \in I_j^o(u)\} \tag{2.12}$$

$$J_i^o(u) = \{j \in J^o(u) : i \in I_j^o(u)\}. \tag{2.13}$$

Whenever the underlying u is evident, we will write J^+ for $J^+(u)$, etc.

Using the above definitions

$$F(u) = \sum_{i \in I} u_i + \sum_{j \in J^+} S_j(u).$$

As we shall see, it is useful to define a "base gradient" for F(u). Thus we define

$$g(u) = e + \sum_{j \in J^+} \tilde{\nabla} S_j(u) \tag{2.14}$$

where $e = (1, \ldots, 1)^T \in \mathbb{R}^m$, and

$$\tilde{\nabla} S_j(u) = (s_1^j, \ldots, s_m^j)^T \in \mathbb{R}^m, \tag{2.15}$$

$$s_i^j = \begin{cases} -1 \text{ if } i \in I_j^+ \cup I_j^o \\ 0 \quad \text{otherwise.} \end{cases} \tag{2.16}$$

Notice that g(u) expresses exactly the first order change in F(u) along the direction d when, in this direction, the u_i's such that $i \in I_j^o$, $j \in J^+$ happen to decrease and, further, $S_j(u)$ remains zero for all $j \in J^o$. This statement,

and the modification necessary to express more general changes, is the basis of our method.

Let e_i denote the i^{th} unit vector in $\mathbb{R}^m$. We first state the following theorem.

Theorem 2.1. The point u^* solves problem (2.1) if and only if there exist scalars λ_j^* and μ_i^*, called the Lagrange dual variables, such that

$$g(u^*) = \sum_{j \in J^o(u^*)} \lambda_j^* \tilde{\nabla} S_j(u^*) - \sum_{i \in I^o(u^*)} \mu_i^* e_i \tag{2.17}$$

where

$$-1 \leq \lambda_j^* \leq 0 \qquad \text{for } j \in J^o(u^*) \tag{2.18}$$

$$0 \leq \mu_i^* \leq -\sum_{j \in J_i^o(u^*)} \lambda_j^* + |J_i^+(u^*)| \qquad \text{for } i \in I^o(u^*). \tag{2.19}$$

The theorem is constructive in the sense that whenever the conditions (2.17)-(2.19) are not satisfied, it is relatively straightforward to obtain a descent direction for F.

More specifically, the motivation is as follows. We first try to obtain a descent direction by projecting -g(u) orthogonally into a space such that $J^o(u) \cup I^o(u)$ does not change (the activities remain active). A descent direction is obtained when such a projection is nonzero. When the projection is zero, it follows that g(u) can be expressed entirely in terms of $\tilde{\nabla} S_j(u)$ for $j \in J^o(u)$ and e_i for $i \in I^o(u)$. Thus, we are able to consider the effect of dropping a *single* activity. This either determines a descent direction or establishes the required optimality conditions.

We will try to make this more concrete by considering a simple example. Consider $m = n = 3$,

$$(c_{ij}) = \begin{pmatrix} 2 & 4 & 1 \\ 3 & 1 & 4 \\ 4 & 4 & 5 \end{pmatrix}$$

and $(f_j) = (2\ 1\ 3)$.

Optimal u^*'s are $\begin{pmatrix}2\\1\\5\end{pmatrix}$, $\begin{pmatrix}3\\1\\5\end{pmatrix}$, $\begin{pmatrix}3\\1\\4\end{pmatrix}$, $\begin{pmatrix}3\\2\\4\end{pmatrix}$, $\begin{pmatrix}1\\2\\4\end{pmatrix}$, $\begin{pmatrix}2\\1\\4\end{pmatrix}$ or any convex combination of these vectors.

We take $u^0 = \begin{pmatrix}1\\2\\3\end{pmatrix}$ as our initial point. $F(u^0) = 11$ and $J^+(u^0) = \{1, 2, 3\}$. We have no activity of type i but one activity of type ii given by $u_1^0 = c_{13}$. The base gradient at u^0 is $g(u^0) = (-2,-1,-2)^T$. We maintain the activity of type ii by choosing our search direction in a space orthogonal to $(-1,0,0)^T$. Thus, we take our search direction to be $d^0 = -Pg(u^0)$ where P is the orthogonal projector onto the space orthogonal to the space spanned by the gradients of the activities. Thus in our example, $d^0 = (0,1,2)^T$.

We now descend as much as possible in this direction while minimizing the *entire* function F(u). This gives a step size of 1/3 and hence $u^1 = (1, 2+1/3, 3+2/3)^T$. For this point, $F(u^1) = 9+1/3$, $J^+(u^1) = \{2\}$, $J^0(u^1) = \{1, 3\}$, I_1^0 and I_2^0 are empty, and $I_3^0 = \{1\}$. This time $g(u^1) = (0,1,0)^T$ and we project orthogonal to $\tilde{\nabla} S_1(u^1) = (-1,-1,-1)^T$, $\tilde{\nabla} S_3(u^1) = (-1,-1,-1)^T$ and e_1, obtaining $d^1 = (0,-1,1)^T$, where without loss of generality d^1 has been suitably scaled.

The optimal step size for the line search is again 1/3 and thus $u^2 = (1,2,4)^T$. This gives $F(u^2) = 9$, $J^+(u^2) = \{2\}$, $J^0(u^2) = \{1,3\}$, $I_1^0 = \{3\}$, $I_2^0 = \{3\}$ and $I_3^0 = \{1\}$. Now $g(u^2) = (0,1,0)^T$ but the projection is zero. The point u^2 is a degenerate stationary point but in this particular case it is easy to choose a suitable basis. We choose $\tilde{\nabla} S_3(u^2)$, e_1 and e_3. Now

$$g(u^2) = \lambda_3 \tilde{\nabla} S_3(u^2) - \mu_1 e_1 - \mu_3 e_3$$

with $\lambda_3 = -1$ and $\mu_1 = \mu_3 = 1$. Clearly, condition (2.18) of Theorem 2.1 holds. To check condition (2.19) note that $J_1^0(u^2) = \{3\}$ and $J_1^+(u^2)$ is empty. So, for

$i = 1$, this condition reduces to $0 \leq \mu_1 \leq -\lambda_3$ which is satisfied. For $i = 3$ we have $J_3^0(u^2) = \{1\}$ and $J_3^+(u^2) = \{2\}$. In this case the condition (2.19) reduces to $0 \leq \mu_3 \leq |J_3^+(u^2)|$. Again the condition is satisfied.

Therefore, according to Theorem 2.1, the point u^2 is optimum for our example.

Given an optimum dual solution u^*, we can derive an optimum primal solution to the strong linear programming relaxation by using the complementary slackness conditions. Namely

$$j \in J^+(u^*) \quad \text{implies } x_j^* = 1 \tag{2.20}$$

$$j \in J^-(u^*) \quad \text{implies } x_j^* = 0 \tag{2.21}$$

$$i \in I_j^+(u^*) \quad \text{implies } y_{ij}^* = x_j^* \tag{2.22}$$

$$i \in I_j^-(u^*) \quad \text{implies } y_{ij}^* = 0. \tag{2.23}$$

The values of x_j^* for $j \in J^0(u^*)$ are the optimum Lagrange dual values λ_j^*, up to the sign. Namely

$$j \in J^0(u^*) \quad \text{implies } x_j^* = -\lambda_j^*. \tag{2.24}$$

The reason for (2.24) is that (2.22), (2.23), (1.2), (1.3) and (1.4) imply the existence of $0 \leq \alpha_{ij} \leq 1$ such that

$$\sum_{j \in J^0 : i \in I_j^+} x_j^* + \sum_{j \in J_i^0} \alpha_{ij} x_j^* + \sum_{j \in J_i^+} \alpha_{ij} = 1 \quad \text{for } i \notin \bigcup_{j \in J^+} I_j^+.$$

This is equivalent to (2.17)-(2.19).

In our example, these conditions yield

$x_1^* = 0$ (by 2.24)

$x_2^* = 1$ (by 2.20)

$x_3^* = 1$ (by 2.24)

$y_{12}^* = y_{23}^* = y_{33}^* = 1$ (by 2.22).

3. Proof of Theorem 2.1

In order to prove Theorem 2.1, we will require five lemmas. Algorithmically, each lemma will correspond to a means of determining a descent direction, when one exists.

Let P denote the orthogonal projector onto the space orthogonal to the space spanned by $\tilde{\nabla} S_j(u)$ for $j \in J^o(u)$ and e_i for $i \in I^o(u)$.

Lemma 3.1. If Pg(u) is nonzero, then $d = - Pg(u)$ is a descent direction for F(u).

Proof: It follows from the definition of P and the fact that, for a sufficiently small step size α, the index sets J^+, J^-, I_j^+ and I_j^- do not change, that

$$F(u+\alpha d) = F(u) + \alpha d^T g(u). \tag{3.1}$$

But

$$d^T g(u) = - g(u)^T P^T g(u) = - \| Pg(u) \|^2 < 0, \tag{3.2}$$

using the fact that P is an orthogonal projector ($P=P^T, P^2=P$) and the assumption that $Pg(u) \neq 0$.

Consequently $F(u+\alpha d) < F(u)$ for all sufficiently small positive α. □

Now suppose that $Pg(u) = 0$, or equivalently

$$g(u) = \sum_{j \in J^o(u)} \lambda_j \tilde{\nabla} S_j(u) - \sum_{i \in I^o(u)} \mu_i e_i. \tag{3.3}$$

It will be convenient to assume that the multipliers λ_j for $j \in J^o$ and μ_i for $i \in I^o$ are uniquely defined. This can be realized by methods such as perturbation techniques. Indeed, this is exactly analogous to the situation in linear programming. So, without loss of generality, we make the following assumption for the next four lemmas.

Assumption 3.2. The vectors $\tilde{\nabla} S_j(u)$ for $j \in J^o(u)$ and e_i for $i \in I^o(u)$ are linearly independent.

From a computational point of view, a perturbation technique is undesirable. Our algorithmic approach to degeneracy is presented in Section 4.

Lemma 3.3. If $\lambda_k > 0$ for some $k \in J^o$, then $d = - P^k g(u)$ is a descent direction for $F(u)$, where P^k denotes the orthogonal projector onto the space orthogonal to the space spanned by $\tilde{\nabla} S_j(u)$ for $j \in J^o \backslash \{k\}$ and e_i for $i \in I^o$.

Proof: We first note that, using (3.3) and the definition of P^k,

$$P^k g(u) = \lambda_k P^k \tilde{\nabla} S_k(u). \tag{3.4}$$

Thus $d^T \tilde{\nabla} S_k(u) = - \lambda_k \tilde{\nabla} S_k(u)^T P^k \tilde{\nabla} S_k(u)$

$$= - \lambda_k \| P^k \tilde{\nabla} S_k(u) \|^2 < 0,$$

where the inequality follows from the facts that $\lambda_k > 0$ (hypothesis) and $P^k \tilde{\nabla} S_k(u) \neq 0$ (Assumption 3.2).

Thus $S_k(u)$ descends in the direction d, i.e. it changes from an activity of type i to being strictly negative. All the other activities remain active. So, as for the proof of Lemma 3.1, we get $F(u+\alpha d) = F(u) + \alpha d^T g(u) < F(u)$, for all sufficiently small positive α. □

Lemma 3.4. If $\lambda_k < - 1$ for some $k \in J^o$, then $d = - P^k g(u)$ is a descent direction, where P^k is defined as in Lemma 3.3.

Proof: As in the proof of Lemma 3.3,

$$d^T \tilde{\nabla} S_k(u) = - \lambda_k \| P^k \tilde{\nabla} S_k(u) \|^2. \tag{3.5}$$

Hence $d^T \tilde{\nabla} S_k(u) > 0$ and, for small positive α,

$$F(u+\alpha d) = F(u) + \alpha d^T g(u+\alpha d)$$

$$= F(u) + \alpha d^T (g(u) + \tilde{\nabla} S_k(u)).$$

Since $d^T g(u) = \lambda_k \, d^T \tilde{\nabla} S_k(u)$,

$$F(u+\alpha d) = F(u) + \alpha(\lambda_k+1) \, d^T \tilde{\nabla} S_k(u). \tag{3.6}$$

Consequently, we will have descent if $\lambda_k < -1$. □

Lemma 3.5. If $\mu_k < 0$ for some $k \in I^o$, then $d = -Q^k g(u)$ is a descent direction for $F(u)$, where Q^k denotes the orthogonal projector onto the space orthogonal to the space spanned by $\tilde{\nabla} S_j(u)$ for $j \in J^o$ and e_i for $i \in I^o \backslash \{k\}$.

Proof: Using (3.3) and the definition of Q^k,

$$Q^k g(u) = -\mu_k \, Q^k e_k \tag{3.7}$$

Thus $d^T(-e_k) = -\mu_k \, \| \, Q^k e_k \, \|^2$.

Since $\mu_k < 0$ by hypothesis and $Q^k e_k \neq 0$ by Assumption 3.2, we have that u_k is decreasing in the direction d.

Now it follows from the definition of $\tilde{\nabla} S_j(u)$ that

$$g(u+\alpha d) = e + \sum_{j \in J^+(u+\alpha d)} \tilde{\nabla} S_j(u+\alpha d)$$

$$= e + \sum_{j \in J^+(u)} \tilde{\nabla} S_j(u) \qquad = g(u),$$

for sufficiently small positive α. Thus

$F(u+\alpha d) = F(u) + \alpha d^T g(u)$ for sufficiently small α and $d^T g(u) =$

$- \| \, Q^k g(u) \, \|^2 < 0$. □

Lemma 3.6. If $\mu_k > -\sum_{j \in J_k^o} \lambda_j + | \, J_k^+ \, |$ for some $k \in I^o$, then $d = -R^k g(u)$ is a descent direction for $F(u)$, where R^k denotes the orthogonal projector onto

the space orthogonal to the space spanned by $\tilde{\nabla} S_j(u) + e_k$ for $j \in J_k^o$, $\tilde{\nabla} S_j(u)$ for $j \in J^o \backslash J_k^o$, and e_i for $i \in I^o \backslash \{k\}$.

Proof: Using (3.3) and the definition of R^k

$$R^k g(u) = -(\mu_k + \sum_{j \in J_k^o} \lambda_j) R^k e_k. \tag{3.8}$$

Thus $d^T(-e_k) = -(\mu_k + \sum_{j \in J_k^o} \lambda_j) \| R^k e_k \|^2$.

Since $\mu_k + \sum_{j \in J_k^o} \lambda_j > 0$ by hypothesis and $R^k e_k \neq 0$ by Assumption 3.2, we obtain the property that u_k is increasing along d.

Thus, for α sufficiently small and positive,

$$\tilde{\nabla} S_j(u+\alpha d) = \begin{cases} \tilde{\nabla} S_j(u) + e_k & \text{if } k \in I_j^o(u) \\ \tilde{\nabla} S_j(u) & \text{otherwise.} \end{cases} \tag{3.9}$$

Therefore, $g(u+\alpha d) = e + \sum_{j \in J^+(u)} \tilde{\nabla} S_j(u) + \sum_{j \in J_k^+(u)} e_k$, i.e.

$$g(u+\alpha d) = g(u) + |J_k^+(u)| e_k. \tag{3.10}$$

Using (3.8) and (3.10), we obtain

$$R^k g(u+\alpha d) = -(\mu_k + \sum_{j \in J_k^o(u)} \lambda_j - |J_k^+(u)|) R^k e_k.$$

Let $\tau_k = \mu_k + \sum_{j \in J_k^o} \lambda_j - |J_k^+|$ and $\nu_k = \mu_k + \sum_{j \in J_k^o} \lambda_j$. Then

$$d^T g(u+\alpha d) = \nu_k (e_k)^T (R^k)^T g(u+\alpha d)$$

$$= -\nu_k \tau_k (e_k)^T R^k e_k = -\nu_k \tau_k \| R^k e_k \|^2.$$

By hypothesis $\tau_k > 0$ and $\nu_k > 0$. So $d^T g(u+\alpha d) < 0$ and therefore

$$F(u+\alpha d) = F(u) + \alpha d^T g(u+\alpha d) < F(u)$$

for sufficiently small positive α. □

Proof of Theorem 2.1.

It follows directly from Lemmas 3.1, 3.3, 3.4, 3.5 and 3.6 that the conditions (2.17) - (2.19) of Theorem 2.1 are necessary.

To prove that these conditions are sufficient, we assume that they hold and analyze the effect of dropping a single activity. We show that, in each case, $F(u^*)$ cannot decrease.

By the piecewise linearity of F and the fact that nondifferentiabilities correspond to activities of type i or ii, this implies that $F(u^*)$ cannot decrease in any direction.

First consider the effect of dropping $S_k(u^*)$ for $k \in J^o$. Thus $d = \sigma P^k g(u^*)$, where P^k is defined in Lemma 3.3. As in the proof of Lemma 3.3, $d^T \tilde{\nabla} S_k(u^*) = \sigma\lambda_k \| P^k \tilde{\nabla} S_k(u^*) \|^2$. If $\sigma > 0$, $\lambda_k \leq 0$ implies that, for sufficiently small α, $F(u^*+\alpha d) = F(u^*) + \alpha d^T g(u^*)$. Now $d^T g(u^*) = \sigma \| P^k g(u^*) \|^2 > 0$ implies that $F(u^*+\alpha d) \geq F(u^*)$. If $\sigma < 0$, $\lambda_k \leq 0$ implies, as in Lemma 3.4, that $F(u^*+\alpha d) = F(u^*) + \alpha(\lambda_k+1)d^T \tilde{\nabla} S_k(u^*)$. This gives non-descent since $\lambda_k \geq -1$.

Next we consider the effect of dropping e_k for $k \in I^o$. Thus $d = \sigma Q^k g(u^*)$, where Q^k is defined in Lemma 3.5 or $d = \sigma R^k g(u^*)$, where R^k is defined in Lemma 3.6.

We first consider $d = \sigma Q^k g(u^*)$. Then $d^T(-e_k) = \sigma\mu_k \| Q^k e_k \|^2$, and by hypothesis $\mu_k \geq 0$.

If $\sigma > 0$, then u_k^* decreases and, as in the proof of Lemma 3.5, $F(u^*+\alpha d) = F(u^*) + \alpha d^T g(u^*)$ for sufficiently small positive α. Now $F(u^*+\alpha d) \geq F(u^*)$ since $d^T g(u^*) = \sigma \| Q^k g(u^*) \|^2 > 0$.

$\sigma < 0$ implies that u_k^* increases. Consequently, if there exists $j \in J_k^o(u^*)$, we violate our condition that only one activity is dropped. However, if $J_k^o(u^*)$ is empty, Q^k is identically equal to R^k and $d = \sigma Q^k g(u^*) = \sigma R^k g(u^*)$. Thus, as in the proof of Lemma 3.6, omitting all terms involving $J_k^o(u^*)$, we obtain that

$F(u^*+\alpha d) = F(u^*) + \alpha d^T g(u^*+\alpha d)$ where $d^T g(u^*+\alpha d) = \sigma \mu_k \tau_k \| R^k e_k \|^2$ with $\tau_k = \mu_k - |J_k^+|$. Now, by hypothesis $0 \leq \mu_k \leq |J_k^+(u^*)|$ which implies $\tau_k \leq 0$ and thus $d^T g(u^*+\alpha d) \geq 0$.

It remains to consider $d = \sigma R^k g(u^*)$. If $\sigma < 0$, then directly from the proof of Lemma 3.6, we have $F(u^*+\alpha d) = F(u^*) - \alpha \nu_k \tau_k \| R^k e_k \|^2$ for any small positive α. If $\nu_k \geq 0$, then $d^T(-e_k) = \sigma \nu_k \| R^k e_k \|^2$ implies that u_k^* does not decrease along d and $\tau_k \leq 0$ (from (2.19)) implies that $F(u^*+\alpha d) \geq F(u^*)$. If $\nu_k < 0$, then u_k decreases along d. Therefore our condition that only one activity is dropped implies that $J_k^0(u^*)$ is empty. But then $\nu_k = \mu_k$ which with (2.19) implies $\nu_k \geq 0$, contradicting $\nu_k < 0$.

If $\sigma > 0$ then, as in the proof of Lemma 3.6, $R^k g(u^*) = - \nu_k R^k e_k$ and $d^T(-e_k) = \sigma \nu_k \| R^k e_k \|^2$. Now, if $\nu_k < 0$, u_k^* increases along d and $F(u^*+\alpha d) = F(u^*) + \alpha d^T g(u^*+\alpha d)$ with $d^T g(u^*+\alpha d) = - \sigma \nu_k e_k R^k g(u^*+\alpha d) = \sigma \nu_k \tau_k \| R^k e_k \|^2 \geq 0$ since $\tau_k \leq 0$. This implies $F(u^*+\alpha d) \geq F(u^*)$. Finally, if $\nu_k > 0$, u_k^* decreases along d which once again violates our condition that only one activity is dropped unless $J_k^0(u^*)$ is empty. Then $\nu_k = \mu_k$, $R^k = Q^k$ and $F(u^*+\alpha d) = F(u^*) + \alpha d^T g(u^*)$ from the proof of Lemma 3.5. But now $d^T g(u^*) = \sigma \| R^k g(u^*) \|^2 > 0$. □

4. The Algorithm

We now present a finite algorithm for solving problem (2.1). For simplicity of exposition, we first assume that nondegeneracy holds. Our approach to degeneracy is explained in the latter part of this section.

Minimization Algorithm

(1) Choose any $u^1 \in \mathbb{R}^m$ and set $k \leftarrow 1$.

(2) Identify $J^0(u^k)$, $J^+(u^k)$, $I_j^0(u^k)$, $I_j^+(u^k)$, $j \in J$.

(3) Compute $g(u^k)$.

(4) Compute $d^k = - Pg(u^k)$ where P is defined in Section 3.

(5) If $d^k \neq 0$, then go to (9).

(6) Compute the current estimate for the Lagrange dual variables λ_j^k, $j \in J^o(u^k)$ and μ_i^k, $i \in I^o(u^k)$ by solving

$$g(u^k) = \sum_{j \in J^o(u^k)} \lambda_j \tilde{\nabla} S_j(u^k) - \sum_{i \in I^o(u^k)} \mu_i e_i .$$

(7) Stop if $-1 \leq \lambda_j^k \leq 0$ for all $j \in J^o(u^k)$

and $0 \leq \mu_i^k \leq \sum_{j \in J_i^o(u^k)} \lambda_j^k + |J_i^+(u^k)|$ for all $i \in I^o(u^k)$.

(8) Choose one of the violated inequalities in (7) and drop the corresponding activity. Let $\ell \in J^o(u^k) \cup I^o(u^k)$ be its index. Define

$$d^k = \begin{cases} - P^\ell g(u^k) & \text{if } \lambda_\ell^k > 0 \text{ or if } \lambda_\ell^k < -1, \text{ where } P^\ell \text{ is defined in Lemma 3.3,} \\ - Q^\ell g(u^k) & \text{if } \mu_\ell^k < 0, \text{ where } Q^\ell \text{ is defined in Lemma 3.5,} \\ - R^\ell g(u^k) & \text{if } \mu_\ell^k > - \sum_{j \in J_\ell^g} \lambda_j^k + | J_\ell^+ |, \text{ where } R^\ell \text{ is defined in Lemma 3.6.} \end{cases}$$

where our preference is to choose d^k in the order $\lambda_\ell^k > 0$, $\mu_\ell^k < 0$, $\lambda_\ell^k < -1$, $\mu_\ell^k > - \sum_{j \in J_\ell^g} \lambda_j^k + |J_\ell^+|$, whenever there is a choice.

(9) Determine the step size α^k by solving $\min_{\alpha > 0} F(u^k + \alpha d^k)$ subject to keeping all activities active (except, of course, for the activity dropped in (8) when step (8) is performed in iteration k). This line search can be done by starting from u^k and moving from one breakpoint of F to the next, in the direction d^k, until either the value of F starts increasing or an active S_j becomes nonactive.

(10) Update and iterate.

$u^{k+1} = u^k + \alpha^k d^k$

$k \leftarrow k + 1$

Go to (2).

Finite Termination of the Algorithm

A point u such that $Pg(u) = 0$ is called a *stationary point.*

First, note that at most n iterations can occur before a stationary point is reached since, whenever we are not at a stationary point, the line search picks up at least one new activity while maintaining those satisfied at the beginning of the iteration. We next remark that there are only a finite number of stationary points because of the piecewise linear nature of the objective function -- i.e. the e_i's and $\tilde{\nabla} S_j$'s come from a finite collection, as does g(u). Finally, it is not possible to return to any given stationary point, since the objective function sequence $\{F(u^k)\}$ is monotonic decreasing. Thus termination occurs in a finite number of iterations.

Furthermore, the work required within each iteration is finite. In particular, in the line search (9), a breakpoint of type ii occurs when $u_i^k + \alpha d_i^k = c_{ij}$ for some $i \in I$ and $j \in J$. Given the current point and the sign of d_i^k, it is easy to find the next such breakpoint of type ii. A change in the sign of $S_j(u)$ between two consecutive breakpoints of type ii yields a breakpoint of type i since S_j is linear in that range. So breakpoints of type i are easy to find as well. Since there are only finitely many breakpoints along the line $u^k + \alpha d^k$, the line search is finite.

Remarks

(a) Although u^1 can be chosen arbitrarily in step (1), there are advantages in choosing either

(i) u_i^1 = second largest c_{ij} over $j \in J$,

or (ii) the best heuristic dual solution determined by Erlenkotter [10].

(b) The line search described in (9) can be modified to take into account the remark made in Section 1 that there always exists an optimal solution such that $S_j(u) \leq 0$ for all $j \in J$. Specifically, assume that $S_j(u^t) \leq 0$ has been maintained through iterations $t = 1, \ldots, k$. Perform the line search by moving from one breakpoint of F to the next until the first of the following events occurs:

(i) $F(u^k+\alpha d^k)$ starts increasing,

(ii) $S_j(u^k+\alpha d^k)$ becomes inactive for some $j \in J^o(u^k)$ [$J^o(u^k)\backslash\{\ell\}$ if u^k is stationary],

(iii) $S_j(u^k+\alpha d^k)$ becomes positive for some $j \in J(u^k)$.

We denote by LS1 the line search described in (9) and by LS2 the modified line search described here. Thus LS2 is obtained from LS1 by adding the stopping criterion (iii).

(c) We implemented an experimental code for the purpose of this article. We use QR factorizations that are updated for dropping and adding activities until there are a possible n activities, in which case we use LU factorizations with updating. This is adequate since our primary concern in this paper is with the number of iterations required to reach optimality. An ideal version of the algorithm would use genuine large sparse techniques.

Algorithmic Approach to Degeneracy

The difficulty arises in the degenerate case because the multipliers associated with (3.3) are then no longer uniquely defined. A unique solution can of course be determined if one chooses a basis $\tilde{\nabla} S_j(u)$, e_i from $j \in J^o(u)$, $i \in I^o(u)$, and sets all other multipliers to zero. We note that this is exactly what we did for the simple example given in Section 2. The problem is that, once activities are dropped in Lemmas 3.3, 3.4, 3.5, and 3.6, one has to verify the consequence to the dependent activities. Indeed, this is exactly analogous to the situation in linear programming and, from a theoretical point of view, can be handled by perturbation techniques. From a practical point of view, a perturbation technique is undesirable since one loses, in general, the underlying structure of the location problem. One may use an approach analogous to that of Busovaca [2], namely adding the conditions of Theorem 2.1 as explicit constraints. In other words, one recognizes optimality by solving

$$g(u) = \sum_{j \in J^o} \lambda_j \tilde{\nabla} S_j(u) - \sum_{i \in I^o} \mu_i e_i$$

subject to

$$-1 \leq \lambda_j \leq 0 \qquad \text{for } j \in J^o$$

$$0 \leq \mu_i \leq -\sum_{j \in J_i^o} \lambda_i + |J_i^+| \qquad \text{for } i \in I^o$$

as a constrained least squares problem in λ and μ. Moreover, if no solution exists, an optimal point has not been found and a descent direction can be readily constructed.

For the purpose of the present article, suffice it to say that

(i) degeneracy is a relatively common occurrence for uncapacitated facility location problems,

(ii) we were able to solve degenerate problems without any particular difficulties,

(iii) we outline in some detail one algorithmic approach to degeneracy that is relatively straightforward and effective in practice, given the special structure of location problems.

Suppose (3.3) is satisfied but Assumption 3.2 is not. Consequently, one may choose a basis among the activities such that

$$g(u) = \sum_{j\varepsilon J_B^o} \lambda_j \tilde{\nabla} S_j(u) - \sum_{i\varepsilon I_B^o} \mu_i e_i \qquad (4.1)$$

where the subscript B indicates that a basis is being chosen. Further, without loss of generality, we may always take $I_B^o = I^o$.

Now suppose $\lambda_k > 0$ for some $k \varepsilon J_B^o$ and consider $d = - P^k g(u)$ as for Lemma 3.3. Then

$$d^T \tilde{\nabla} S_k(u) = - \lambda_k \| P^k \tilde{\nabla} S_k(u) \|^2 < 0$$

and $$d^T g(u) = \lambda_k d^T \tilde{\nabla} S_k(u) < 0.$$

To find out whether d is a descent direction, we compute $d^T g(u+\alpha d)$ for small positive α. Thus it remains to consider $d^T \tilde{\nabla} S_h(u)$ for $h \varepsilon J^o \backslash J_B^o$. For any such h, we can write

$$d^T \tilde{\nabla} S_h(u) = \gamma_h d^T \tilde{\nabla} S_k(u) \qquad \text{for some } \gamma_h \varepsilon \mathbb{R}. \qquad (4.2)$$

Let $D^- = \{h \varepsilon J^o \backslash J_B^o : \gamma_h < 0\}$ and

$D^+ = \{h \varepsilon J^o \backslash J_B^o : \gamma_h > 0\}$. Now

$$\begin{aligned} d^T g(u+\alpha d) &= d^T g(u) + \sum_{h\varepsilon D^-} d^T \tilde{\nabla} S_h(u+\alpha d) \\ &= d^T g(u) + \sum_{h\varepsilon D^-} d^T \tilde{\nabla} S_h(u) \\ &= (\lambda_k + \sum_{h\varepsilon D^-} \gamma_h)\, d^T \tilde{\nabla} S_k(u). \end{aligned}$$

Therefore d is a descent direction if

$$\lambda_k + \sum_{h\varepsilon D^-} \gamma_h > 0. \tag{4.3}$$

Now suppose $\lambda_k < -1$ for some $k \varepsilon J_B^o$. Using the same direction d as above, we have $d^T \tilde{\nabla} S_k(u) > 0$. So, here, $d^T g(u+\alpha d) = d^T g(u) + d^T \tilde{\nabla} S_k(u) + \sum_{h\varepsilon D^+} d^T \tilde{\nabla} S_h(u)$. Therefore, we obtain that d is a descent direction if

$$\lambda_k + 1 + \sum_{h\varepsilon D^+} \gamma_h < 0. \tag{4.4}$$

Suppose $\mu_k < 0$ for some $k \varepsilon I_B^o$ and consider $d = -Q^k g(u)$ as for Lemma 3.5. Then

$$d^T(-e_k) = -\mu_k \| Q^k e_k \|^2 > 0. \tag{4.5}$$

Moreover,

$$g(u+\alpha d) = e + \sum_{j\varepsilon J^+(u+\alpha d)} \tilde{\nabla} S_j(u+\alpha d).$$

So, in order to know the sign of $d^T g(u+\alpha d)$ for small positive α we need to consider $d^T \tilde{\nabla} S_h(u)$ for $h \varepsilon J^o \backslash J_B^o$. We can write

$$d^T \tilde{\nabla} S_h(u) = \gamma_h \, d^T(-e_k) \qquad \text{for } h \varepsilon J^o \backslash J_B^o \tag{4.6}$$

Let $D^- = \{h \varepsilon J^o \backslash J_B^o : \gamma_h < 0\}$ and $D^+ = \{h \varepsilon J^o \backslash J_B^o : \gamma_h > 0\}$. Then

$$\begin{aligned} d^T g(u+\alpha d) &= d^T g(u) + \sum_{h\varepsilon D^+} d^T \tilde{\nabla} S_h(u+\alpha d) \\ &= (\mu_k + \sum_{h\varepsilon D^+} \gamma_h) \, d^T(-e_k). \end{aligned}$$

Therefore d is a descent direction if

$$\mu_k + \sum_{h\varepsilon D^+} \gamma_h < 0. \tag{4.7}$$

Finally, it remains to consider

$$\mu_k > -\sum_{j\varepsilon J_{kB}^o} \lambda_j + |J_k^+| \tag{4.8}$$

where $J_{kB}^o = J_k^o \cap J_B^o$. Let $d = -R^k g(u)$ as for Lemma 3.6. Then

$$d^T(-e_k) = -(\mu_k + \sum_{j\varepsilon J_{kB}^o} \lambda_j) \| R^k e_k \|^2 < 0.$$

Define γ_h for $h \in J^o \backslash J_B^o$ by

$$d^T(\tilde{\nabla} S_h(u)+e_k) = \gamma_h \, d^T(-e_k) \quad \text{if } h \in J_k^o \backslash J_B^o$$

$$d^T \tilde{\nabla} S_h(u) = \gamma_h \, d^T(-e_k) \quad \text{if } h \in (J^o \backslash J_k^o) \backslash J_B^o.$$

Let $D^- = \{h \in J^o \backslash J_B^o : \gamma_h < 0\}$ and $D^+ = \{h \in J^o \backslash J_B^o : \gamma_h > 0\}$ Now

$$d^T g(u+\alpha d) = d^T g(u) - |J_k^+(u)| d^T(-e_k) + \sum_{h \in D^-} d^T \tilde{\nabla} S_h(u+\alpha d)$$

$$= (\mu_k + \sum_{j \in J_{kB}^o} \lambda_j - |J_k^+| + \sum_{h \in D^-} \gamma_h) \, d^T(-e_k).$$

Therefore d is a descent direction if

$$\mu_k > - \sum_{j \in J_{kB}^o} \lambda_j + |J_k^+| - \sum_{h \in D^-} \gamma_h. \tag{4.9}$$

5. Numerical Results

The first set of results given are for a class of ten 33 × 33 problems where the c_{ij} values are taken from data for a traveling salesman problem [15]. This is a well-known test set considered representative. It was solved for example by Schrage [20] and Erlenkotter [10].

We give results for two different initial points

(a) $u_i = 0 \qquad i = 1, \ldots, 33$

(b) $u_i = c_{ik} \qquad i = 1, \ldots, 33$, where c_{ik} is the second largest entry for given i, and the two different line search algorithms LS1 and LS2. No results are given in the latter case for fixed charges 184 and 295 since not all the S_j's are initially positive.

Fixed Charge	# of iterations $u_o = 0$ LS1	LS2	$u_o = c_{ik}$ LS1	LS2	$F(u^*)$
184	3	3	10		- 6024
295		14	9		-8673
500	14	21	11	20	-11267
1000	10	17	9	12	-14832
1500	18	23	15	19	-17832
2000		40		31	-20346
2500	41	32	28	27	-22127
3000	30	23	14	22	-23474
4000	11	8	8	7	-25474
5000	5	7	4	5	-27474

We note that only the problem with fixed charge 2000 has a duality gap [the solution is - 20363]. Next we consider this problem in more detail. Using the complementary slackness conditions (2.20) - (2.24), we get

$x_3^* = x_7^* = x_8^* = x_{13}^* = x_{16}^* = 1/2,$

$x_{20}^* = x_{24}^* = 1$ and $x_j^* = 0$ for all the other j's.

The fractional y_{ij}^* are as follows.

$y_{1,3} = y_{2,3} = y_{3,3} = y_{4,3} = y_{5,3} = y_{6,3} = 1/2$

$y_{3,7} = y_{4,7} = y_{5,7} = y_{6,7} = y_{7,7} = y_{8,7} = y_{9,7} = y_{10,7} = 1/2$

$y_{2,8} = y_{7,8} = y_{8,8} = y_{9,8} = y_{10,8} = y_{11,8} = y_{12,8} = 1/2$

$y_{1,13} = y_{11,13} = y_{12,13} = y_{13,13} = y_{14,13} = y_{15,13} = y_{16,13} = y_{17,13} = 1/2$

$y_{13,16} = y_{14,16} = y_{15,16} = y_{16,16} = y_{17,16} = 1/2.$

The remaining $i \in I$ are assigned to either facility 20 or 24, whichever is closer.

Note that there are several cuts of the form (1.15) that cut off the above fractional solution (x^*,y^*). In fact there are 18 such cuts. Sixteen of these cuts involve the variables x_3, x_7 and x_8. The other two involve the variables x_3, x_8 and x_{13}. We show one cut of each type.

Cut 1 $y_{1,3} + y_{2,3} + y_{2,8} + y_{11,8} + y_{1,13} + y_{11,13} - x_3 - x_8 - x_{13} \leq 1$

Cut 2 $y_{2,3} + y_{4,3} + y_{4,7} + y_{9,7} + y_{2,8} + y_{9,8} - x_3 - x_7 - x_8 \leq 1$

Adding these two cuts to the formulation yields a modified condensed dual with two new dual variables, say v_1 associated with Cut 1 and v_2 associated with Cut 2 [See (1.16) and (1.17)]. Starting from the previous optimum dual solution u^*, we only needed 9 additional iterations to solve the modified condensed dual. We found an optimal dual solution (u^{**},v^{**}) with $v_1^{**} = 44$ and $v_2^{**} = 20$ but there are alternate optima, as is typical with problem (2.1). Using the complementary slackness conditions (5.1) - (5.5), we now have an integer primal optimum solution:

$x_7 = x_{13} = x_{20} = x_{24} = 1, \qquad x_j = 0$ otherwise.

This solution is unique. Instances where the dual formulation has alternate optima and the primal has a unique solution seem to be typical for the uncapacitated facility location problem. At any rate, we have observed it frequently whether it be with or without the addition of cuts.

We give another illustration of the cutting plane approach. By drawing the c_{ij}'s randomly from a uniform distribution, duality gaps are more likely to occur than when the c_{ij}'s satisfy the triangle inequality (such as in the above 33-city problem), [1]. Consider the following problem, where c_{ij} was drawn at random between 0 and 100 and where $f_j = 100$ for every j.

$$(c_{ij}) = \begin{pmatrix} 75 & 56 & 74 & 88 & 19 & 3 & 46 & 21 & 29 & 39 \\ 52 & 10 & 79 & 62 & 12 & 9 & 52 & 88 & 76 & 31 \\ 85 & 59 & 58 & 87 & 63 & 73 & 3 & 79 & 80 & 27 \\ 17 & 68 & 35 & 70 & 75 & 3 & 87 & 72 & 13 & 35 \\ 64 & 32 & 40 & 73 & 11 & 93 & 30 & 80 & 64 & 71 \\ 70 & 33 & 44 & 71 & 34 & 21 & 20 & 56 & 59 & 19 \\ 55 & 56 & 9 & 21 & 40 & 7 & 93 & 50 & 49 & 27 \\ 42 & 14 & 69 & 15 & 77 & 85 & 36 & 52 & 72 & 98 \\ 41 & 5 & 99 & 21 & 27 & 51 & 23 & 89 & 23 & 68 \\ 64 & 32 & 59 & 29 & 96 & 31 & 81 & 83 & 4 & 63 \end{pmatrix}$$

Solving problem (2.1) using the algorithm of Section 4 with line search algorithm LS1, we obtain the following optimum vector u, with value F(u) = 581 + 2/3.

$u_1 = 32 + 2/3$, $u_2 = 62$, $u_3 = 63$, $u_4 = 70$, $u_5 = 71$, $u_6 = 52+1/3$, $u_7 = 37$, $u_8 = 52$, $u_9 = 74+1/3$, and $u_{10} = 67+1/3$.

Using (5.1) - (5.5), we get the primal optimum solution

$x_3 = x_4 = x_7 = 1/3$, $x_8 = 2/3$ and $x_j = 0$ otherwise.

$y_{i,8} = 2/3$ for $i \neq 1$,

$y_{1,3} = y_{1,4} = y_{1,7} = y_{2,3} = y_{3,4} = y_{4,7} = y_{5,4} = y_{6,4} = y_{7,7} = y_{8,3} = y_{9,3} = y_{10,7} = 1/3$, $y_{ij} = 0$ otherwise.

It was shown in [3], [8], [13] that the following inequality defines a facet of the uncapacitated facility location polytope. Furthermore it cuts off the current fractional solution.

Cut 1 $y_{1,3} + y_{1,4} + y_{1,7} + y_{2,3} + y_{2,4} + y_{2,8} + y_{4,4} + y_{4,7} + y_{4,8} + y_{9,3} + y_{9,7} + y_{9,8} - x_3 - x_4 - x_7 - x_8 \leq 2.$

Adding it to the formulation and solving the new condensed dual yields an optimum solution value of 575. Going back to the primal through (5.1) - (5.5), we have

$x_3 = x_4 = x_7 = x_8 = 1/2$, $x_j = 0$ otherwise,

$y_{1,3} = y_{1,4} = y_{2,3} = y_{2,8} = y_{3,4} = y_{3,8} = y_{4,7} = y_{4,8} = y_{5,4} = y_{5,8} = y_{6,4} = y_{6,8} = y_{7,7} = y_{7,8} = y_{8,3} = y_{8,8} = y_{9,3} = y_{9,8} = y_{10,7} = y_{10,8} = 1/2$,

$y_{ij} = 0$ otherwise.

We added next

Cut 2 $y_{1,3} + y_{1,4} + y_{2,3} + y_{2,8} + y_{3,4} + y_{3,8} - x_3 - x_4 - x_8 \leq 1$

reducing the optimum solution value to 573.5, and then

Cut 3 $y_{1,3} + y_{1,4} + y_{3,4} + y_{3,8} + y_{9,3} + y_{9,8} - x_3 - x_4 - x_8 \leq 1.$

The optimum solution of the new condensed dual was

$u_1 = 21$, $u_2 = 52$, $u_3 = 62$, $u_4 = 48$, $u_5 = 66$, $u_6 = 53$, $u_7 = 50$, $u_8 = 49$, $u_9 = 40$, $u_{10} = 64$, $v_1 = 24$, $v_2 = 8$, $v_3 = 9$ where v_i is associated with Cut i, i = 1, 2, 3. The corresponding value is F(u,v) = 570. Now using (5.1)-(5.5) once more, we get

$$x_8 = 1, \quad y_{i8} = 1 \quad \text{for all i,}$$

$$x_j = 0 \text{ and } y_{ij} = 0 \quad \text{for } j \neq 8.$$

6. Relationship with Erlenkotter's Heuristic

A well-known heuristic approach to the condensed dual problem is that of Erlenkotter [10]. This method is simple and often very effective. We analyze it in the context of our proposed method.

Firstly, at all iterations of Erlenkotter's heuristic all S_j's are nonpositive. Consequently, $g(u^k) = e$ for all k.

We remark that the descent direction given by our algorithm amounts to steepest descent in the particular subspace defined by Lemmas 3.1, 3.3, 3.4, 3.5 and 3.6.

By contrast Erlenkotter's dual descent procedure corresponds to a coordinate-wise search until a new activity of type ii is found or until blocked by an activity of type i, repeating until no descent is so obtained. Consider, say, a search along $d = -e_k$. It is immediate that $d^T g(u) < 0$ and Erlenkotter's line search is effectively stopped whenever an $S_j \leq 0$ becomes positive. Since u_k is decreasing but all other u_i's remain fixed it is clear that all activities remain active except possibly an activity of type ii, given by $c_{kj} = u_k$, which one could consider to have been dropped. In the former case, we clearly

have a direction in the space defined by Lemma 3.1, although no longer in the steepest descent direction in general. In the latter case we have a direction of the type defined by Lemmas 3.5 or 3.6, but again, in general, no longer in the steepest descent direction.

Evidently, what the dual descent procedure lacks in sophistication it makes up for amply (at least for medium-size problems) in the simplicity of the computations. The dual adjustment procedure adds one level of complication to the choice of search direction when optimality cannot be reached via coordinate-wise search.

More particularly, suppose $c_{kj} > u_k$ for more than one $j \in J^o(u)$, $j_1, j_2, \ldots j_t$ say.

Now suppose we increase u_k. Consequently $S_{j_1} \ldots S_{j_t}$ that were active become negative. We can now attempt to decrease other u_j's that appear in $S_{j_1}, \ldots, S_{j_t}$. If more than one such u_j can be decreased unit for unit as u_k increases, say u_i $i = i_1, i_2, \ldots, i_s$, we gain and F(u) decreases.

Thus in effect we are searching in a direction

$$d = e_k - \sum_{p=1}^{s} e_{i_p}.$$

In other words, s + 1 activities of type ii are dropped, one by increasing u_k and the remaining s by decreasing u_{i_p}. In addition, several activities of type i may also be dropped in the search direction d.

7. Extensions

The essential ingredients of the method that we have presented are:

(i) F is a sum of nondifferentiable functions.

(ii) The combinatorial structure of the problem can be exploited.

See Calamai and Conn [4], and Conn [5] for additional related background.

We expect to see applications of these ideas to other structured linear programs that also have a piecewise linear condensed form. This is not an uncommon occurrence. In fact, the *primal* of the strong linear programming relaxation of the uncapacitated facility location problem itself has such a condensed form [7]. It can be written as

$$\max_{x \geq 0,\ \sum_{j \in J} x_j \geq 1} \sum_{i \in I} z_i(x) - \sum_{j \in J} f_j x_j$$

where $z_i(x)$ is the piecewise linear concave function defined by

$$z_i(x) = \min_{k \in J} \left(c_{ik} + \sum_{j \in J} (c_{ij} - c_{ik})^+ x_j \right).$$

Equivalently $z_i(x) = c_{ik} + \sum_{j \in J} (c_{ij} - c_{ik})^+ x_j$
where k is defined by

$$\sum_{j : c_{ij} > c_{ik}} x_j < 1 \leq \sum_{j : c_{ij} \geq c_{ik}} x_j.$$

In general, problems with fixed charges such as network design or lot-sizing problems have linear programming relaxations that may admit a condensed form.

The capacitated facility location problem is obtained by adding the capacity constraints

$$\sum_{i \in I} d_i y_{ij} \leq s_j x_j \qquad \text{for } j \in J \tag{7.1}$$

to the formulation (1.1)-(1.5). Here d_i represents the demand of client i and s_j the capacity of a facility at location j. The dual of the linear programming relaxation is

$$\text{Min} \quad \sum_{i \in I} u_i + \sum_{j \in J} t_j$$

$$u_i + w_{ij} + d_i v_j \geq c_{ij} \qquad \text{for all } i \in I,\ j \in J$$

$$-\sum_{i \in I} w_{ij} + t_j - s_j v_j \geq -f_j \qquad \text{for all } j \in J$$

$$w_{ij},\ t_j,\ v_j \geq 0 \qquad \text{for all } i \in I,\ j \in J.$$

This dual has a condensed form, namely

$$\underset{v \geq 0, u}{\text{Min}} \quad F(u,v) = \sum_{i \in I} u_i + \sum_{j \in J} S_j^+(u,v)$$

where $S_j(u,v) = \sum_{i \in I} (c_{ij} - u_i - d_i v_j)^+ + s_j v_j - f_j$.

As for the uncapacitated problem, $F(u,v)$ is piecewise linear and convex, and there is an optimum solution such that $S_j(u,v) \leq 0$ for all $j \in J$.

References

[1] S. Ahn, C. Cooper, G. Cornuejols and A. Frieze, "Probabilistic Analysis of a Relaxation for the k-Median Problem," Management Science Research Report 527, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh (1986), to appear in *Mathematics of Operations Research*.

[2] S. Busovaca, "Handling Degeneracy in a Nonlinear ℓ_1 Algorithm," Technical Report CS-85-34, Department of Computer Science, University of Waterloo, Waterloo, Canada (1985).

[3] D. C. Cho, M. W. Padberg and M. R. Rao, "On the Uncapacitated Plant Location Problem II: Facets and Lifting Theorems," *Mathematics of Operations Research* 8 (1983), 590-612.

[4] P. H. Calamai and A. R. Conn, "A Projected Newton Method for ℓ_p Norm Location Problems," to appear in *Mathematical Programming* (1987).

[5] A. R. Conn, "Nonlinear Programming, Exact Penalty Functions and Projection Techniques for Non-Smooth Functions," in "Numerical Optimization 1984," SIAM (1985), 1-25.

[6] G. Cornuejols, M. L. Fisher and G. L. Nemhauser, "Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms," *Management Science* 23 (1977), 789-810.

[7] G. Cornuejols, G. L. Nemhauser and L. A. Wolsey, "The Uncapacitated Facility Location Problem," Management Science Research Report 493,

Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh (1984), to appear in *Discrete Location Theory*, R. L. Francis and P. Mirchandani, eds., Wiley-Interscience.

[8] G. Cornuejols and J.-M. Thizy, "Some Facets of the Simple Plant Location Polytope," *Mathematical Programming* 23 (1982), 50-74.

[9] M. A. Efroymson and T. L. Ray, "A Branch and Bound Algorithm for Plant Location," *Operations Research* 14 (1966), 361-368.

[10] D. Erlenkotter, "A Dual-Based Procedure for Uncapacitated Facility Location," *Operations Research* 26 (1978), 992-1009.

[11] R. S. Garfinkel, A. W. Neebe and M. R. Rao, "An Algorithm for the M-median Plant Location Problem," *Transportation Science* 8 (1974), 217-236.

[12] M. Grotschel and M. W. Padberg, "Polyhedral Theory and Polyhedral Computations," in *The Traveling Salesman Problem*, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan and D. B. Shmoys, eds., John Wiley and Sons (1985), 251-360.

[13] M. Guignard, "Fractional Vertices, Cuts and Facets of the Simple Plant Location Problem," *Mathematical Programming Study* 12 (1980), 150-162.

[14] M. Guignard and K. Spielberg, "Algorithms for Exploiting the Structure of the Simple Plant Location Problem," *Annals of Discrete Mathematics* 1 (1977), 247-271.

[15] R. L. Karg and G. L. Thompson, "A Heuristic Approach to Solving Traveling Salesman Problems," *Management Science* 10 (1964), 225-248.

[16] J. G. Morris, "On the Extent to which Certain Fixed-Charge Depot Location Problems Can Be Solved by LP," *Journal of the Operational Research Society* 29 (1978), 71-76.

[17] J. M. Mulvey and H. L. Crowder, "Cluster Analysis: An Application of Lagrangian Relaxation," *Management Science* 25 (1979), 329-340.

[18] S. C. Narula, U. I. Ogbu and H. M. Samuelsson, "An Algorithm for the p-Median Problem," *Operations Research* 25 (1977), 709-713.

[19] C. S. ReVelle and R. S. Swain, "Central Facility Location," *Geographical Analysis* 2 (1970), 30-42.

[20] L. Schrage, "Implicit Representation of Variable Upper Bounds in Linear Programming," *Mathematical Programming Study* 4 (1975), 118-132.

[21] K. Spielberg, "Algorithms for the Simple Plant-Location Problem with Some Side Conditions," *Operations Research* 17 (1969), 85-111.