COMPARISON OF FOUR MODELS FOR DISPERSING FACILITIES*

ERHAN ERKUT , SUSAN NEUMAN

Department of Finance and Management Science, Faculty of Business, University of Alberta Edmonton, Alberta, T6G 2R6, CANADA

ABSTRACT

Given a set of candidate points in some space, this paper considers the problem of selecting a dispersed subset. Possible applications include the location of mutually undesirable facilities. Four different models of dispersal are considered. Mathematical programming formulations are given, and an implicit enumeration procedure, which solves all four problems in parallel, is developed. A simple heuristic is also developed which gives very good solutions. Computational experience with the four models is reported in detail, in which the solutions of the models are compared with an emphasis on similarities and differences. Possible extensions of the models are also considered, to incorporate minimum distance separation constraints and existing facility points.

Keywords : Location, dispersion, branch-and-bound, heuristics.

RÉSUMÉ

Etant donné un ensemble de points elligibles dans un espace quelconque, cet article considère le problème qui consiste à choisir un sous-ensemble dispersé. Les applications possibles de ce problème incluent la localisation de facilités mutuellement non-désirables. Quatre modèles différents de dispersion sont considérés. Des modèles de programmation mathématique sont formulés, et une procédure d'énumération implicite qui solutionne simultanément les quatre modèles est développé. Un heuristique simple, donnant de très bonnes solutions, est également développé. Les résultats expérimentaux de calcul avec les quatres modèles sont décrits en détail, et les solutions des modèles sont comparées en mettant une emphase particulière sur les similitudes et les différences. Des extensions possibles de ces modèles sont aussi considérées afin d'incorporer des contraintes d'éloignement minimum et des localisations existantes de facilités.

1. INTRODUCTION

In the past, researchers have tended to concentrate on the location of desirable facilities, such as hospitals and fire stations. In these location problems, the objective is usually one of minimizing some function of distance. Over the past decade, however, researchers have been turning to problems of locating undesirable facilities, such as nuclear power generators and landfill sites. Many models have been suggested, with various solution spaces and distance functions. For a thorough discussion of these models, the reader is referred to a recent survey paper by Erkut and Neuman (1989).

Of particular interest for this paper are problems involving mutually undesirable facilities. In these problems, there are no existing facilities or existing demand points. Therefore, the new facilities cannot be located relative to existing points; rather, they are located relative to each other. The objective is one of maximizing some function of the distances between these new facilities. Examples of where such an objective might be appropriate include defence installations such as missile silos. In the case of an attack, it is preferable to have the silos separated from each other, so that an attacker cannot disable more than one installation in a single strike. Another application, discussed by Kuby (1987), involves the location of franchises, where separation of facilities is desirable to minimize intra and/or interchain competition. One can also use such a model to locate radio transmitters, where the objective is to minimize interference. Other examples include facilities where some undesirable interactions occur between all facilities, and where the amount of interaction is inversely proportional to distance. In order to minimize the total number of interactions between all facilities, it is necessary to maximize the distances between all of them. Such a model might apply to dormitories at a University, cabins at a summer camp, or chairs during an

* Recd. Dec. 1988 Revd. Jan. 1990 Acc. Jan. 1990

examination. Applications from outside the location area include positioning new products in an attribute space, and selecting sets of parameters for scientific experiments, with a view to minimizing duplications.

In this paper we examine four different, but related, objective functions for locating mutually undesirable facilities. The intention is to determine whether there is much difference between the solutions generated by the four different models. If the differences are relatively minor, then little consideration is necessary in deciding which one to use for a particular application. However, if the differences are quite pronounced, then an understanding of the characteristics of the solutions becomes very important. This comparison of four models is inspired by a paper by Halpern and Maimon (1983), in which four models for locating a single desirable facility on tree networks are studied. The authors address the question "how much difference does it make which locational criterion is being optimized?". The locational criterion they compare are the median and center objectives, and dispersion and equity measures.

The main focus of our paper is on the comparison of the four different objective functions. Thus, the emphasis is on experimental results, rather than theory. Algorithm development is limited to quite simple, but efficient, solution techniques, which draw upon procedures used in earlier papers as well as some original work. In Section 2, mathematical programming formulations for the discrete versions of the four models are presented, followed by a short review of the relevant literature in Section 3. Optimal solution procedures, which make use of branch-and-bound techniques, are described in Section 4. Section 5 contains our computational results. We compare the solutions from the four models, emphasizing similarities and differences. We also experiment with a simple two-stage heuristic solution procedure. Section 6 contains a description of this heuristic, along with its experimental results. Finally, in Section 7 we discuss extensions to these problems, with a view to enhancing their applicability in real-world situations.

2. MODELS

The four models can be formulated in a general way, without reference to a specific solution space or distance metric. These formulations apply to the discrete version of the problems, where the locations for the new facilities are chosen from a finite set of candidate points in R^m. Let S be a set of points in R^m, and p the number of new facilities to be located, where $|S| \geq p \geq 2$. Let $d(i,j)$ be the distance between points i and j in S. The four problems require the selection of a solution set X, $|X| = p$, in order to maximize one of the following four functions:

MaxMinMin : $\max Z$ s.t. $Z \leq d(i,j) \quad \forall\, i,j \in X,\ i \neq j.$

MaxSumMin : $\max \sum_{i \in X} Z_i$ s.t. $Z_i \leq d(i,j) \quad \forall\, j \in X,\ i \neq j.$

MaxMinSum : $\max Z$ s.t. $Z \leq \sum_{j \in X} d(i,j) \quad \forall\, i \in X.$

MaxSumSum : $\max Z$ s.t. $Z = \sum_{i \in X} \sum_{j \in X} d(i,j).$

These four objective functions differ along two bases, which are represented by the terms "min" or "sum" in the second and third syllables of the function title. The first basis determines whether "worst-case" performance or total performance is of more importance. Using the min operator results in a more equitable solution, since the "worst-case" performance for any individual facility is maximized. Objectives which use the sum operator, however, are more concerned with total efficiency. They maximize overall system performance, possibly at the expense of individual facility performance. The second basis determines which interactions between new facilities are of interest. When the min operator is used, we consider only the *minimum* distance from each new facility to the other new facilities. (The two objectives which use the min operator, maxminmin and maxsummin, will be referred to as the "min" objectives.) With the sum operator, the *sum* of the distances from each new facility to all other new facilities is used. (The maxminsum and maxsumsum objectives will be referred to as the "sum" objectives.) The second basis conveys whether one is concerned with interactions between a new facility and all other new facilities or only with the closest other new facility.

The four problems of interest can be formulated as mixed-integer programming problems. Letting n represent the number of candidate points to choose from, p the number of candidate points to be chosen, d_{ij} the distance between candidate points i and j, and M a number larger than the largest distance between any two candidate points, the formulations for the four models can be given as follows, where x_i is set to 1 if the candidate point i is selected, and 0 otherwise:

$$
\begin{aligned}
\text{MaxMinMin} : \max Z & \\
\text{s.t.} \quad & Z \leq d_{ij} + M(2 - x_i - x_j), \quad 1 \leq i < j \leq n, \\
& \sum_{i=1}^{n} x_i = p, \\
& x_i = 0 \text{ or } 1, \quad 1 \leq i \leq n.
\end{aligned}
$$

$$
\begin{aligned}
\text{MaxSumMin} : \max \sum_{i=1}^{n} Z_i & \\
\text{s.t.} \quad & Zi \leq Mx_i, \quad 1 \leq i \leq n, \\
& Zi \leq d_{ij} + M(1 - x_j), \quad 1 \leq i < j \leq n, \\
& \sum_{i=1}^{n} x_i = p, \\
& x_i = 0 \text{ or } 1, \quad 1 \leq i \leq n.
\end{aligned}
$$

$$
\begin{aligned}
\text{MaxMinSum} : \max Z & \\
\text{s.t.} \quad & Z \leq \sum_{j=1}^{n} d_{ij}x_j + M(1 - x_i), \quad 1 \leq i \leq n, \\
& \sum_{i=1}^{n} x_i = p, \\
& x_i = 0 \text{ or } 1, \quad 1 \leq i \leq n.
\end{aligned}
$$

$$
\begin{aligned}
\text{MaxSumSum} : \max \sum_{i=1}^{n} Z_i & \\
\text{s.t.} \quad & Z_i \leq Mx_i, \quad 1 \leq i \leq n, \\
& Z_i \leq \sum_{j=1}^{n} d_{ij}x_j, \quad 1 \leq i \leq n, \\
& \sum_{i=1}^{n} x_i = p, \\
& x_i = 0 \text{ or } 1, \quad 1 \leq i \leq n.
\end{aligned}
$$

3. LITERATURE REVIEW

The maxminmin problem is most often referred to as the p-dispersion problem. Of the four models given, it is the one which has attracted the most attention in the literature, mainly due to its relationship to another well-studied problem, the p-center problem. Shier (1977) is the first to study this relationship. By extending early duality results, due to Meir and Moon (1975), he demonstrates both weak and strong duality results for the unweighted, continuous versions of these two problems on tree networks. Chandrasekaran and Tamir (1980, 1982) extend the duality results to the discrete problem, where new facility sites and potential centers are restricted to a subset of the tree.

Chandrasekaran and Daughety (1981) describe two similar polynomial algorithms for solving these problems on tree networks. Tansel *et al.* (1982) consider nonlinear versions of these problems on tree networks.

Moving away from tree networks, the following two papers deal with the p-dispersion problem without reference to any particular solution space. They accomplish this by making use of a matrix of point-to-point distances. Consequently, their solution techniques are limited to discrete problems, where the potential sites for new facilities are limited to a finite set of given points. In a modeling and application oriented paper, Kuby (1987) gives a mixed-integer programming formulation for the p-dispersion problem, similar to the one given in Section 2. The linear programming relaxation of this formulation produces fractional results so he solves the mixed-integer program using MPSX on an IBM 3090 mainframe. Run-times are 24 CPU seconds for $n = 25$, $p = 5$, and 53 seconds for $n = 25$, $p = 10$. Erkut (1990) also presents a similar (0-1) linear programming formulation, but solves it using a structure-exploiting branch-and-bound technique. On the average, problems with $n = 30$, $p = 6$ are solved in less than 14 seconds, on an AT-compatible micro, while problems with $n = 30$, $p = 9$ are solved in less than 40 seconds. In addition, Erkut demonstrates the NP-completeness of the p-dispersion problem, by reduction to the clique problem, and presents an efficient heuristic.

Kuby (1987) also studies the discrete maxsumsum problem using a mixed-integer programming formulation, which again yields fractional results when run in a relaxed form. Two sample problems were run using a software package, but optimality was not proven, despite being allotted 9 minutes of mainframe CPU time for the problem with $n = 25$, $p = 5$, and 14 minutes of CPU time for $n = 25$, $p = 10$. Erkut, Baptie and von Hohenbalken (1990) consider the discrete maxsumsum problem with both new and existing facilities. Their branch-and-bound algorithm, similar to the one for p-dispersion, gives optimal solutions to problems with $n = 25$, $p = 5$ in under 5 seconds and $n = 25$, $p = 10$ in less than 45 seconds, on average, on a micro. They give a (0-1) quadratic programming formulation for the problem, and present a three-stage heuristic which can be used to generate sharp lower bounds for the enumeration procedure.

Hansen and Moon (1988) describe an algorithm for the maxsumsum problem on tree networks, and mention that earlier results on quadratic knapsack problems can be used to solve the maxsumsum problem on general networks. They prove that the discrete version of the problem on general networks is strongly NP-complete, by reduction to the stable set problem.

One characteristic of the solutions to the maxsumsum problem is that the selected facilities are often quite close to each other, as opposed to the p-dispersion problem where the facilities are always separated by a minimum distance. Kuby (1987) gives a multicriteria formulation which combines the maxminmin and the maxsumsum objectives. The optimal maxminmin objective function value is used as a lower bound on the distances between selected facilities, while the maxsumsum objective is maximized. This has the effect of spreading out the selected facilities while still maximizing the sum of the distances between them, subject to the distance constraints. This multicriteria approach is in the spirit of Toregas *et al.* (1971), and Khumawala (1973, 1975), who analyzed the minimization of a single objective subject to constraints on other objectives. We explore Kuby's technique further, in Section 7, and expand on it to include distance constrained versions of the maxsummin and maxminsum objectives as well.

The maxsummin problem is referred to as the p-defence problem by Moon and Chaudhry (1984). To date, we know of no published research on this problem. The final problem, maxminsum, seems to have eluded researchers' attentions as well, as we find no mention of it in the location literature. One purpose of this paper is to determine the characteristics of the solutions to these two problems, and compare them against the more widely studied maxminmin and maxsumsum problems.

We note that the formulation given in Section 2 for the p-dispersion problem is similar to the formulation by Kuby (1987). In contrast, the other three formulations are original. In particular, the maxsumsum formulation uses $O(n)$ decision variables and the objective function is linear. The model given by Kuby (1987) has $O(n^2)$ decision variables, and the model by Hansen and Moon

(1988) has a quadratic objective function. To the extent of our knowledge, the other two problems have not been modeled in the literature.

4. OPTIMAL SOLUTION TECHNIQUES

The (0-1) programming formulations given in Section 2 suggest that implicit enumeration techniques can be used to solve these problems optimally. By exploiting the commonalities between these four problems, a single branch-and-bound program was developed which solves all four problems at once. If one is addressing only one of the four objectives, a more problem specific technique could be developed. For our purposes, a great deal of efficiency is derived from using common subroutines. As a further aid to efficiency, two $n \times n$ matrices are constructed. The first contains the point-to-point distances. It is sorted by the sum of the distances in a specific row (rowsum), where the site with the largest rowsum appears first. This ordering of the distance matrix was found to result in faster pruning for the maxsumsum and maxsummin problems. Those sites with large overall rowsums are likely to also have large partial rowsums, and are therefore more likely to be included in the solutions sets for these two problems. The pruning of the branch-and-bound trees for the two "min" problems did not respond as readily to a specific ordering of the distance matrix. Hence, the ordering by rowsum was used since it hastens the pruning process for at least two of the four objectives. After the ordering of the distance matrix is complete, a "sorted distance index" matrix is constructed. For each of the n sites, the distances to every other site are sorted from largest to smallest. The indices of the sites are then stored in the sorted distance index matrix, rather than the distances themselves. This matrix makes it easier to find the largest distances, relative to each site in the solution space.

Before the branch-and-bound procedure is started, a heuristic is used to generate good initial lower bounds for each objective. (This heuristic will be discussed in Section 6.) These lower bounds are updated whenever a larger objective function value is encountered at one of the tips of the branching tree. The branching method used is a depth first search, where upper bounds on the solutions are calculated at each node. If the upper bound for an objective is less than the corresponding lower bound, the branching tree is pruned for that objective. However, as long as at least one of the upper bounds calculated exceeds the corresponding lower bound, the search is continued down the branch. Only when all four upper bounds are less than the corresponding lower bounds is the branch abandoned.

The upper bounding techniques, used for the four objectives, are quite similar. In the following description, p is the number of facilities to be chosen (out of n), k is the number of facilities left to be chosen at any point in the procedure, and q is the number of sites already considered. Some of these will have been chosen, and some not. Therefore, $(p-k)$ is the number of facilities already chosen, and $(n-q)$ is the number of sites left to be considered for the remaining k facilities. A vector, X, of n entries is used to keep track of which sites have been considered at each node of the branching tree. Its entries appear in the same order as in the distance matrix. If a site's entry is 1 then it is chosen for inclusion in the current solution set. If it is 0 then it is one of the q sites considered up to that node, but it is not one of the $(p-k)$ sites chosen.

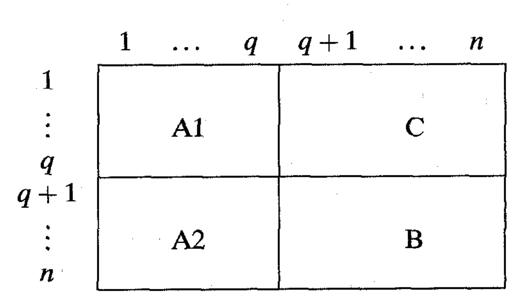


Figure 1. The Partitioning of the Distance Matrix, used in the Calculation of the Upper Bounds.

The upper bounds for the four objective functions are combinations of the largest possible distances for those sites which have already been chosen and those sites which have yet to be considered. In Figure 1, the distance matrix is divided into four areas. Rows (columns) 1 through q correspond to those sites already considered, and rows (columns) $(q+1)$ to n correspond to those sites not yet considered. Section A1 is that part of the distance matrix where the distance terms, to be included in the upper bounds, are known with certainty, since it contains the distances between the $(p-k)$ sites already chosen in the current solution set. Section C contains possible distances between the $(p-k)$ sites already chosen and the k sites left to be chosen from rows $(q+1)$ to n. The k largest values, in section C, for each of the $(p-k)$ rows, corresponding to selected sites, are used in the calculations of the upper bounds for these rows. Section A2 contains the possible distances between the k sites left to be chosen and the $(p-k)$ sites already chosen. Section B contains the possible distances between the k sites left to be chosen from rows $(q+1)$ to n. When calculating upper bounds using the sites in section B, it must be remembered that the sites' distances to themselves are included in this part of the distance matrix. Hence, of the k distances to be considered when choosing a specific site, $(k-1)$ of these are to other sites and one is to the site itself (this distance is zero). That is, only $(k-1)$ distances are of importance, as opposed to the k distances in section C.

The upper bounding technique is best described in terms of pseudocode. In the following description, the terms "currowmin" and "currowsum" refer to the minimum and the sum, respectively, of the distances between a specific site and the sites already chosen for inclusion in the current solution set (*i.e.* between a specific site and those sites with a 1 as their entry in the X vector). The upper bounding procedure is as follows:

For the first q rows which have $X=1$ do

> A1min := currowmin
> A1sum := currowsum
> Cmin := the minimum of {the k largest values in columns $q+1$ to n}
> Csum := the sum of {the k largest values in columns $q+1$ to n}
> min for the row := the minimum of {A1min, Cmin}
> sum for the row := the sum of {A1sum, Csum}

For the last $n-q$ rows do

> A2min := currowmin
> A2sum := currowsum
> Bmin := the minimum of {the $k-1$ largest values in columns $q+1$ to n}
> Bsum := the sum of {the $k-1$ largest values in columns $q+1$ to n}
> min for the row := the minimum of {A2min, Bmin}
> sum for the row := the sum of {A2sum, Bsum}

From the first q rows there are $p-k$ mins and $p-k$ sums.
From the last $n-q$ rows, take the k largest mins and the k largest sums.
This gives p mins and p sums.

The upper bound for	maxminmin	is the	minimum	of these p	mins.
	maxsummin		sum		mins.
	maxminsum		minimum		sums.
	maxsumsum		sum		sums.

When evaluating the objective function at a tip of the enumeration tree, if its value is found to be better than the best solution found so far then the lower bound is updated and the contents of the new solution set (entries in X) are stored. For the maxminmin objective, an additional step is required in that all solution sets which have the same objective function value as the current lower bound must be stored. Different solution sets can result in the same optimal objective function value because the objective function value is only defined by the distance between the two closest sites. Any other sites chosen in the solution set can be exchanged for sites not in the solution set, as long as the changes do not result in a new minimum distance between selected sites. Maxminmin is the only one of the four models which regularly results in multiple optima.

5. COMPUTATIONAL RESULTS AND COMPARISON

Our computational results are based on the generation of four different random problem sets. For each problem set, 25 (n) random points, in the unit square, were generated. Of these, 5, 10, 15 or 20 (p) points were selected as sites for new facilities, according to each of the four objectives. For each combination of n and p, 10 random problems were run, and the results were averaged over these 10 problems. Thus, in total, 40 random problems were generated. The code for the heuristic and the branch-and-bound technique were written in Turbo Pascal, and implemented on an AT-compatible microcomputer.

For each of the 40 random problems, the optimal objective function value and optimal solution set(s) are found for each of the four objectives. As mentioned earlier, all solution sets resulting in the optimal objective function value for the maxminmin objective are stored. (In earlier testing, we found one maxminmin problem, with $n = 25$ and $p = 10$, which had 594 alternate optima.)

Once these objective function values and solutions sets are found, a comparison is made between the four objectives, on a problem by problem basis. This comparison consists of using one objective's optimal solution set to determine the objective function values for the other three objectives. These three values, which in most cases are strictly less than their corresponding optimal objective function values, are then expressed as percentages of their optimal objective function values. All of these percentages are combined over the 10 problems in each of the four problem sets. Average percentages are reported, as well as the minimum and maximum percentages encountered. Tables 1 through 4 contain the results described above. The table entries represent the "average(minimum,maximum)" percentages over the 10 problems, and have been rounded to the closest integer value.

Since it can have multiple solution sets, a slightly more involved procedure is necessary for the maxminmin objective. For each problem, the maxminmin solution sets are taken one at a time, and are used to evaluate the other three objective functions. The results tabulated for each of the 10 problems are the average objective function value percentages, as well as the minimum and maximum percentages, rather than just one objective function value percentage as for the other three objectives.

Table 1. $n = 25$, $p = 5$ Percent of Optimal Objective Function Value -average (minimum, maximum)-

Solution Set(s) Used	Objective Function Evaluated			
	maxminmin	maxsummin	maxminsum	maxsumsum
maxminmin		95 (78,100)	81 (65,100)	90 (74,100)
maxsummin	95 (83,100)		87 (78,100)	95 (92,100)
maxminsum	38 (5,100)	60 (30, 95)		97 (93, 99)
maxsumsum	44 (5,100)	83 (71,100)	96 (91,100)	

The columns in these tables represent the degradation in an optimal objective function value as a result of using a solution set which is optimal for one of the other objectives. This kind of information is of importance when there is disagreement about which objective is most suitable for the specific problem at hand. If the two (or more) competing objectives are close substitutes for each other, then the choice of which solution set to use is relatively unimportant. However, as can be seen from these tables, the solution sets for some of the objectives are poor substitutes for others. The worst performance found is when the solution sets for the maxminsum and maxsumsum

Table 2. $n = 25$, $p = 10$ Percent of Optimal Objective Function Value
-average (minimum, maximum)-

Solution Set(s) Used	Objective Function Evaluated			
	maxminmin	maxsummin	maxminsum	maxsumsum
maxminmin		95 (87,100)	71 (63, 77)	85 (79, 91)
maxsummin	92 (82,100)		71 (65, 75)	87 (80, 91)
maxminsum	34 (7, 62)	59 (48, 68)		99 (96,100)
maxsumsum	28 (7, 47)	58 (48, 69)	99 (98,100)	

Table 3. $n = 25$, $p = 15$ Percent of Optimal Objective Function Value
-average (minimum, maximum)-

Solution Set(s) Used	Objective Function Evaluated			
	maxminmin	maxsummin	maxminsum	maxsumsum
maxminmin		93 (77,100)	76 (67, 85)	89 (81, 95)
maxsummin	87 (53,100)		74 (60, 84)	87 (77, 94)
maxminsum	32 (3, 48)	66 (49, 84)		99 (98,100)
maxsumsum	32 (3, 53)	65 (47, 79)	99 (98,100)	

Table 4. $n = 25$, $p = 20$ Percent of Optimal Objective Function Value
-average (minimum, maximum)-

Solution Set(s) Used	Objective Function Evaluated			
	maxminmin	maxsummin	maxminsum	maxsumsum
maxminmin		94 (87,100)	84 (78, 90)	93 (89, 97)
maxsummin	59 (19,100)		82 (77, 86)	91 (87, 95)
maxminsum	32 (2, 62)	76 (67, 84)		100 (99,100)
maxsumsum	32 (2, 62)	78 (72, 84)	100 (99,100)	

problems are applied to the maxminmin objective. The resulting objective function value is, on average, less than 35% optimal. In the worst cases, it is less than 3% optimal. The best substitution is achieved when the maxminsum solution set is applied to the maxsumsum objective, and vice versa. Even in the worst cases, the resulting objective function value is still better than 90% optimal. The quality of this substitution increases with increasing p.

Given these results, some suggestions can be made about suitable substitutions. For example, it appears that using either the maxminsum or maxsumsum objective will result in roughly the same solution set, especially for large or medium values of p. For small values of p, the same can be said for the two "min" objectives. This result for the "min" objectives does not hold, however, for large values of p. While applying the optimal maxminmin solution set to the maxsummin objective produces, on average, a near optimal solution, the reverse is not true.

For small values of p, the maxsummin solution set results in a solution which is near optimal, on average, for all four objectives. For large p, the maxminmin solution set produces the best overall performance. Hence, if is not clear which of the four objectives is the most appropriate one to use, and the value of p is quite small or quite large, then one of these two objectives could be applied to the problem, thereby achieving a reasonably good result. If a decision is being made between maxsumsum and a second objective, one could choose the second objective since all of the other objectives result in solution sets which are reasonably good substitutes (93% optimal, on average) for the maxsumsum solution set.

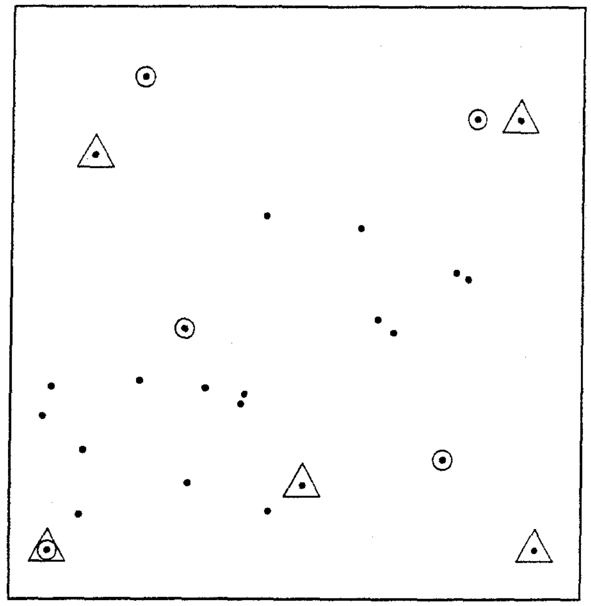


Figure 2a. For a problem with $n = 25$ and $p = 5$, the optimal solution for the maxsummin model (the triangles) and an optimal solution for the maxminmin model (the circles).

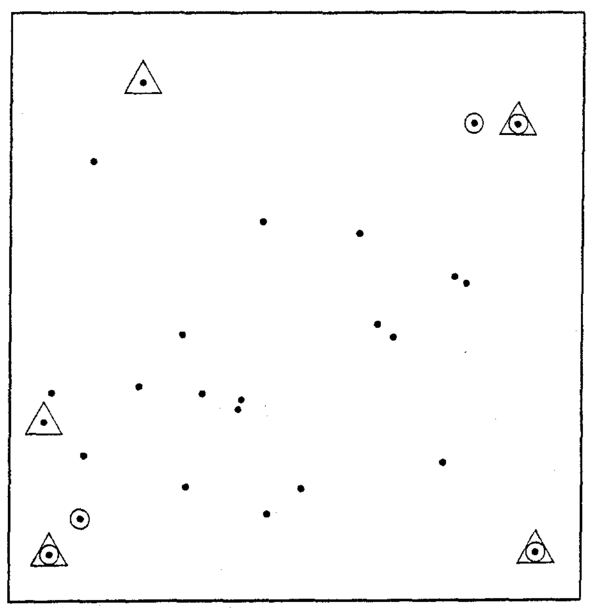


Figure 2b. The optimal solutions for the maxsumsum model (the triangles) and the maxminsum model (the circles), for the same problem as in Figure 2a.

The appropriateness of substitution between the four objectives is heavily influenced by the different spatial patterns of their optimal solution sets. The maxminmin objective is the only one

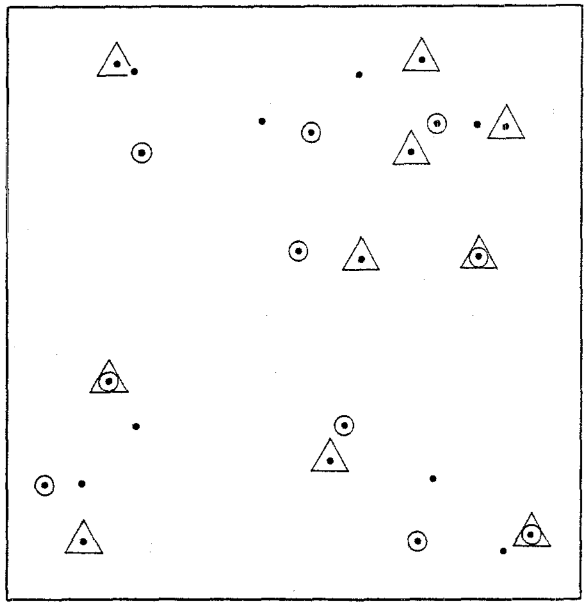


Figure 3a. For a problem with $n = 25$ and $p = 10$, the optimal solution of the maxsummin model (the triangles) and an optimal solution to the maxminmin model (the circles).

which guarantees consistent spacing between the chosen sites. However, for small and medium p the maxsummin objective achieves almost as good a minimum separation as maxminmin, but with the added advantage of increasing the distances between the non-binding pairs of points. In Figure 2a, although the solution sets differ by 4 points, the maxsummin solution is 98% optimal under the maxminmin objective. Also, the sites selected for the maxsummin solution set tend to be located closer to the edges of the solution space than the sites selected for the maxminmin solution set, as shown in Figures 2a and 3a. Thus, depending on the application and the size of p, the maxsummin objective might be preferable to the maxminmin objective.

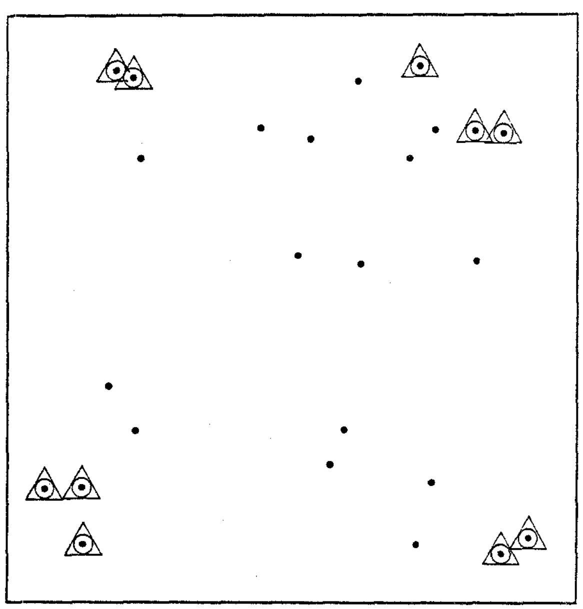


Figure 3b. The optimal solutions for the maxsumsum model (the triangles) and the maxminsum model (the circles), for the same problem as in Figure 3a.

The sites selected by the maxminsum and maxsumsum objectives also tend to be located close to the edges of the solution space. In fact, except for large p (where the majority of the sites must be chosen) the selected sites are rarely in the interior of the solution space, as illustrated by Figures 2b and 3b. This is because for each new facility, the sum of the distances to the other new facilities can usually be increased by choosing a site closer to the edge of the solution space. In contrast, the solution sets for the two "min" objectives almost always include interior points, as shown in Figure 3a.

Another major difference between the two pairs of objectives is that there can be a great deal of diversity between maxminmin and maxsummin solution sets, but there is rarely a lot of difference between the maxminsum and maxsumsum solution sets. Maxminmin problems usually have multiple optimal solutions sets, some of which may be quite similar to the maxsummin solution set. On the other hand, in our experiments, we found the maximum difference between the solution sets for the "min" objectives to be 4 out of 5 points (Figure 2a), and 7 out of 10 points (Figure 3a). In contrast, solution sets for the two "sum" objectives never differed by more than 2 points (for all values of p), and were frequently exactly the same, as in Figure 3b.

Facilities are much more likely to be located very close to each other if one of the two "sum" objectives is used rather than one of the two "min" objectives. This can be seen by contrasting the solution patterns in Figures 2a and 2b, and Figures 3a and 3b. This situation may be undesirable, especially in the case of mutually "noxious" facilities where there is real danger associated with having the facilities too close to each other. For these types of facilities, the maxminmin objective is probably the most appropriate, although the maxsummin objective can also be used if p is small relative to n.

Kuby (1987) points out instances involving mutually noxious facilities where the maxminmin objective might not be appropriate. One case is when the optimal maxminmin objective function value is less than the separation required to prevent the hazardous interaction between facilities. If this is the case, it makes little sense to evenly space the facilities at this ineffective distance. It may be more appropriate to cluster facilities into several groups, with larger separation distances between the groups, to lessen the effect of a chain reaction and preserve at least some of the facilities. None of the other three objectives are designed to do this explicitly, although the two "sum" objectives tend to produce solutions which have this characteristic, as illustrated in Figure 3b.

6. HEURISTIC

A simple heuristic for these problems is useful for a number of reasons. Firstly, run-times for the branch-and-bound procedures become excessive as n increases. In a related experiment with the maxminmin objective, the computational times for the branch-and-bound procedure grew with the square root of (n-choose-p) (Erkut, 1990). Hence, for a large problem, the computational requirements of an optimizing algorithm may be excessive. Even if the optimizing procedure is used, a heuristic is useful since it can provide good initial lower bounds for the branch-and-bound procedure.

The heuristic itself is a two step process, combining a straight forward greedy construction algorithm with a pairwise interchange improvement sequence. The greedy phase begins by choosing the two candidate sites with the largest distance between them. For $p = 2$, this is the optimal solution for all four problems. The algorithm then goes through another $(p-2)$ iterations. At each iteration, every facility which is not in the solution is considered for addition. The four objective functions are evaluated and the new facility which results in the largest objective function value for each of the four objectives is added to the solution set for that objective. This is continued until p facilities have been chosen for each objective.

The next stage in the heuristic is a pairwise interchange improvement step. For each of the four solution sets, facilities not included in the solution are considered for swapping with facilities which are in the solution. The best possible interchange is made, for each objective, and the process is repeated until no further improvements can be realized.

The heuristic was used to solve the same 40 problems as were solved by the optimal solution technique. The performance of the heuristic is very good, averaging from 97.81% optimal for the maxminmin objective to 99.99% optimal for the maxsumsum objective. Table 5 gives the average and the minimum percent optimality for this two-stage heuristic. The maximum percent optimality is 100% for all problem sets.

Table 5. Two-Stage Heuristic $n = 25$
Percent of Optimal Objective Function Value

Objective Function	$p = 5$	$p = 10$	$p = 15$	$p = 20$	Overall
maxminmin					
average	98.25	96.37	98.35	98.28	97.81
minimum	92.53	85.03	86.91	82.77	82.77
maxsummin					
average	98.80	99.06	99.22	99.47	99.14
minimum	95.15	96.30	94.38	97.57	94.38
maxminsum					
average	99.49	100.00	100.00	100.00	99.70
minimum	95.55	100.00	100.00	100.00	95.55
maxsumsum					
average	99.98	100.00	100.00	100.00	99.99
minimum	99.81	100.00	100.00	100.00	99.81

The greedy phase of the heuristic is, on average, quite good, especially for large values of p. Its performance ranges from a worst case of 71% optimal to a best case of 100% optimal. The interchange phase is able to make some improvements to the solutions found in the greedy phase. The improvements range from an average of 3.45% of optimal, for the maxminmin objective, to 0.01% of optimal, for the maxsumsum objective. The overall performance of the heuristic, using both phases, varies from 83% optimal to 100% optimal. Of the 160 problems solved (40 random problems with 4 objectives each), 129 were solved optimally, using this two-phase heuristic. Table 6 gives the breakdown of optimal solutions. The maxminsum and maxsumsum objectives have the best results, a finding supported by the values in Table 5. We note that the heuristic used in our test is a generic heuristic. By exploiting a specific problem structure, more effective heuristics can be developed for any of the four objectives.

Table 6. Heuristic Solutions $n = 25$
Number of Problems Solved Optimally

Objective	out of 10				out of 40
Function	$p = 5$	$p = 10$	$p = 15$	$p = 20$	Overall
maxminmin	7	4	8	9	28
maxsummin	6	5	6	7	24
maxminsum	8	10	10	10	38
maxsumsum	9	10	10	10	39
Overall	30	29	34	36	129

In an effort to explain the success of the heuristic on these combinatorial problems, we have enumerated all possible solutions to some of our problems. For a single representative problem with $n = 25$ and $p = 5$, the histograms in Figure 4 give the frequency of the objective function

```
Histogram of Max Min Min

  0% -   2%
  2% -   4% ********** 1771
  4% -   6%
  6% -   8%
  8% -  10% **************************** 4979
 10% -  12%
 12% -  14%
 14% -  16% ********* 1500
 16% -  18%
 18% -  20%
 20% -  22% ******** 1292
 22% -  24% ********* 1444
 24% -  26% ********* 1461
 26% -  28% ******************* 3261
 28% -  30% ************************************************** 8983
 30% -  32% ************** 2383
 32% -  34% **************** 2700
 34% -  36% **************** 2785
 36% -  38% ************* 2295
 38% -  40% ********** 1663
 40% -  42% ********** 1633
 42% -  44% ******* 1121
 44% -  46% *************** 2649
 46% -  48% ***** 765
 48% -  50% ******** 1349
 50% -  52% ******** 1418
 52% -  54% ********* 1562
 54% -  56% ****** 923
 56% -  58% **** 611
 58% -  60% *** 535
 60% -  62% ******* 1184
 62% -  64% ***** 793
 64% -  66% *** 360
 66% -  68% * 57
 68% -  70% * 148
 70% -  72% **** 543
 72% -  74% * 8
 74% -  76% * 91
 76% -  78% ** 238
 78% -  80% * 40
 80% -  82% ** 183
 82% -  84% * 64
 84% -  86% ** 218
 86% -  88% * 34
 88% -  90% * 22
 90% -  92% * 14
 92% -  94% * 1
 94% -  96% * 8
 96% -  98% * 22
 98% - 100% * 16
   Optimal  * 3
```

Figure 4a. The number of solutions versus solution quality (percent optimality) for a maxminmin problem with $n = 25$ and $p = 5$.

values (as a percentage of the optimum). The last three histograms have forms very similar to the probability density function of the normal distribution, whereas the histogram for the maxminmin objective (Figure 4a) does not fit this pattern. This histogram has several modal points and gaps, and exhibits a number of optimal solutions, with no apparent structure.

The worst solutions are between 2% and 14% optimal, for the four objectives. The average solution for the maxminmin problem is 34% optimal, whereas the averages are between 50% and 60% optimal for the other three objectives. Less than 0.005% of all solutions are better than 90% optimal. In light of this, the performance of the heuristic is quite surprising since for the problems with $n = 25$ and $p = 5$ in our experiment, the heuristic always found a solution which is better than 90% optimal.

Although they represent only a small percentage of all solutions, the number of good solutions for these problems is not very small. For example, for the maxsumsum problem of Figure 4d, 238 solutions are better than 90% optimal. For the maxminmin problem, 19 solutions are better than 98% optimal, with 3 alternate optima. The existence of several near-optimal solutions enhances the effectiveness of the heuristic.

```
Histogram of Max Sum Min

 0% -   2%
 2% -   4%
 4% -   6% * 1
 6% -   8% * 1
 8% -  10% * 11
10% -  12% * 16
12% -  14% * 19
14% -  16% * 43
16% -  18% ** 102
18% -  20% *** 178
20% -  22% **** 216
22% -  24% ***** 304
24% -  26% ******** 453
26% -  28% ************ 725
28% -  30% *************** 927
30% -  32% ****************** 1106
32% -  34% ********************** 1364
34% -  36% **************************** 1734
36% -  38% ******************************** 1930
38% -  40% ********************************** 2099
40% -  42% **************************************** 2487
42% -  44% ****************************************** 2605
44% -  46% ********************************************** 2834
46% -  48% ************************************************** 3082
48% -  50% *************************************************** 3161
50% -  52% **************************************************** 3198
52% -  54% ************************************************** 3071
54% -  56% ************************************************* 3012
56% -  58% ******************************************* 2648
58% -  60% ************************************** 2362
60% -  62% ************************************ 2204
62% -  64% ********************************* 2032
64% -  66% ****************************** 1852
66% -  68% *************************** 1638
68% -  70% ******************** 1267
70% -  72% ***************** 1081
72% -  74% ************** 879
74% -  76% *********** 669
76% -  78% ******** 501
78% -  80% ****** 352
80% -  82% ***** 271
82% -  84% **** 195
84% -  86% *** 176
86% -  88% ** 120
88% -  90% ** 86
90% -  92% * 50
92% -  94% * 43
94% -  96% * 17
96% -  98% * 6
98% - 100% * 1
  Optimal  * 1
```

Figure 4b. The number of solutions versus solution quality (percent optimality) for a maxsummin problem with $n = 25$ and $p = 5$.

For larger values of p, although the shape of the histograms are preserved, the lower ends and means are shifted upwards. For example, on a problem with $n = 25$ and $p = 20$, the worst solution, under any objective, is 68% optimal, with an average between 80% and 90% of the optimal. For another maxsumsum problem, the number of solutions which are 98% optimal or better is 1160. This shift in the quality of the solutions makes it easier to find near-optimal solutions for problems with large p, although it does not become any easier to find an optimal solution.

7. EXTENSIONS

In this section, we suggest some extensions to the four models, with a view to expanding their applicability. When considering extensions to these models, it is necessary to note the differences between them, in terms of the patterns of chosen sites. In general, the maxminmin and maxsummin objectives achieve a spatial separation of facilities by choosing candidate sites both at the edges and in the middle of the solution space. The maxminsum and maxsumsum objectives, however, tend to pick only those sites close to the boundaries of the solution space.

```
Histogram of Max Min Sum

 0% -   2%
 2% -   4%
 4% -   6%
 6% -   8%
 8% -  10%
10% -  12% * 1
12% -  14% * 2
14% -  16% * 5
16% -  18% * 10
18% -  20% * 25
20% -  22% * 64
22% -  24% * 67
24% -  26% ** 120
26% -  28% *** 196
28% -  30% **** 236
30% -  32% ****** 405
32% -  34% ******** 575
34% -  36% ************* 934
36% -  38% *************** 1076
38% -  40% ********************* 1471
40% -  42% ************************** 1859
42% -  44% ****************************** 2193
44% -  46% ************************************ 2635
46% -  48% ****************************************** 3026
48% -  50% ************************************************ 3441
50% -  52% ************************************************** 3600
52% -  54% *************************************************** 3662
54% -  56% ***************************************************** 3801
56% -  58% ************************************************** 3583
58% -  60% ************************************************* 3501
60% -  62% ********************************************* 3211
62% -  64% ************************************** 2756
64% -  66% ******************************** 2356
66% -  68% *************************** 1946
68% -  70% *********************** 1607
70% -  72% ***************** 1235
72% -  74% ************** 1001
74% -  76% ********** 756
76% -  78% ******** 549
78% -  80% ****** 405
80% -  82% **** 285
82% -  84% *** 202
84% -  86% ** 147
86% -  88% ** 96
88% -  90% * 42
90% -  92% * 23
92% -  94% * 13
94% -  96% * 7
96% -  98% * 2
98% - 100% * 2
   Optimal * 1
```

Figure 4c. The number of solutions versus solution quality (percent optimality) for a maxsumsum problem with $n = 25$ and $p = 5$.

These tendencies arise from the choice of which interactions between new facilities are to be considered. The "min" objectives are concerned with the closest other new facility, while the "sum" objectives deal with the total distances, or "average" distance, to other new facilities. Consequently, the "sum" objectives allow for two, or more, facilities to be located close to each other. The maxminmin objective achieves the best spatial separation between new facilities, but it disregards total separation in the process. Since its objective function value is only dependent on the two closest facilities, it is unconcerned with the locations of the non-binding facilities. The maxsummin and "sum" objectives, on the other hand, take every new facility into account.

Each of the four objectives has some shortcomings which may make it unsuitable for some applications. A multiobjective optimization approach may produce solution sets which correct for some of these shortcomings, while still achieving optimal objective function values close to those produced in the single objective approach. To determine whether this is the case, the same 40 random problems from the single objective computational experiments were run again, but with lower bounds on the allowable distances between new facilities. The optimal separation from the maxminmin problem is used as a lower bound. Hence, all of the resulting optimal constrained

```
Histogram of Max Sum Sum

 0% -   2%
 2% -   4%
 4% -   6%
 6% -   8%
 8% -  10%
10% -  12%
12% -  14% * 1
14% -  16% * 1
16% -  18% * 1
18% -  20% * 7
20% -  22% * 13
22% -  24% * 41
24% -  26% * 53
26% -  28% * 37
28% -  30% * 52
30% -  32% ** 109
32% -  34% ** 141
34% -  36% *** 198
36% -  38% *** 209
38% -  40% ***** 345
40% -  42% ******** 570
42% -  44% *********** 786
44% -  46% *************** 1089
46% -  48% ***************** 1255
48% -  50% ********************* 1568
50% -  52% **************************** 2036
52% -  54% ******************************** 2375
54% -  56% *********************************** 2585
56% -  58% ***************************************** 3030
58% -  60% ********************************************* 3349
60% -  62% ************************************************ 3553
62% -  64% *************************************************** 3742
64% -  66% ************************************************* 3625
66% -  68% ************************************************ 3587
68% -  70% *********************************************** 3470
70% -  72% ***************************************** 3061
72% -  74% *************************************** 2876
74% -  76% ********************************* 2442
76% -  78% *************************** 1977
78% -  80% ********************* 1540
80% -  82% *************** 1092
82% -  84% ************ 870
84% -  86% ******** 590
86% -  88% ****** 389
88% -  90% **** 227
90% -  92% ** 132
92% -  94% * 61
94% -  96% * 23
96% -  98% * 17
98% - 100% * 4
  Optimal  * 1
```

Figure 4d. The number of solutions versus solution quality (percent optimality) for a maxminsum problem with $n = 25$ and $p = 5$.

solution sets must match one of the original maxminmin solution sets. However, one solution set may prove to be optimal for one constrained objective, while a different solution set may be optimal for another. The maxminmin objective is the only one of the four objectives which can be used as a constraint in this manner, since it is the only one with multiple optima.

To implement this constrained optimization approach, the branch-and-bound program was modified so that nodes would be pruned if the resulting solution set included a point-to-point distance which was less than the optimal maxminmin separation. This constraint actually shortens computation times, since pruning of the branching tree is facilitated by the constraints, a finding also mentioned by Kuby (1987). The resulting constrained objective function values are compared with the original optimal objective function values, with the results reported in Table 7. For the computations given in this table, the optimal maxminmin objective function value (minimum separation) was known in advance. If this is not the case, the heuristic presented in Section 6 could be used to obtain a good estimate of this value, which would then be used as the lower bound on distances.

As would be expected, the maxsummin objective was the one least affected by the lower bounds

Table 7. Constrained Objectives
Percent of Optimal Objective Function Value

Objective Function	$p = 5$	$p = 10$	$p = 15$	$p = 20$	Overall
maxsummin					
average	97.63	98.55	97.31	96.68	97.54
minimum	90.63	96.59	86.79	91.54	86.79
maximum	100.00	100.00	100.00	100.00	100.00
maxminsum					
average	83.58	72.37	78.83	84.28	79.77
minimum	74.12	67.83	70.81	78.23	67.83
maximum	100.00	77.49	85.26	89.81	100.00
maxsumsum					
average	93.17	87.66	90.99	94.00	91.46
minimum	88.75	85.23	85.86	90.64	85.23
maximum	100.00	90.66	94.98	97.18	100.00

on distances between new facilities. The maxminsum objective was the most affected, with its constrained objective function value averaging only 80% of the unconstrained optimal. The performance for the maxsumsum objective falls between these two. Since the constrained versions of all three objectives have the same minimum separation, the choice of which of the three to use depends heavily on whether facilities interact with their closest neighbour, or with all other new facilities. Given that the maxsummin objective has the lowest degradation under the constraints, it might be the most appropriate choice.

The results in Tables 1 through 4 can be compared with those in Table 7 to see why the constrained multiobjective approach is useful. The first row, in Tables 1 through 4, contains the "average" percent of optimal achieved by using any of the maxminmin solution sets to evaluate the other three objective functions. Table 7 shows how these values can be improved by carefully selecting which maxminmin solution set to use with which objective. For example, in Table 3, with $p = 15$, the average percent of optimal for the maxsummin objective, when evaluated using all of the maxminmin solution sets, is 93.1%. By using the most appropriate maxminmin solution set, the average percentage increases by more than 4% to 97.3%. All of the figures in Table 7 represent an improvement over the corresponding figures in Tables 1 through 4.

A possible extension to this multiobjective approach would be to use a number smaller than the optimal maxminmin separation as the lower bound on distances. The characteristics of a certain application may be such that a minimum "safe" separation is all that is required, rather than the maximum possible value. In fact, a parametric analysis could be conducted to demonstrate how degradation in the optimal objective function value, for the maxminmin objective, affects the rate of improvement in the percent of optimal values for the other three constrained objectives. It may be the case that a slight decrease in the distance constraint would result in a large improvement in the optimal constrained objective function values for the other three objectives. With access to such information, it is much easier to evaluate the trade-offs between the conflicting goals of minimum separation and total system performance.

Another extension, which is useful when locating mutually undesirable facilities, involves the concept of existing facilities. In such a model, the distances between new and existing facilities must be taken into account, in addition to the distances between new facilities. With franchises, for example, existing facilities could represent current franchise locations and, perhaps, the locations of competitors. For noxious facilities, such as nuclear reactors, the existing facilities could depict population centers. This model could be extended further by including (possibly different) lower bounds on the allowable distances between new facilities and the allowable distances between new and existing facilities. We note that we can model existing facilities by extending the solution vector, X, to include new variables, x_i, which would be set equal to 1 permanently. With this modification, the branch-and-bound procedure we describe in this paper can be easily extended to take into account existing facilities.

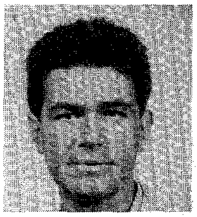
Given the growing number of mutually undesirable facilities in the world today, and the great variety of problem characteristics, it is important to have a number of different models which can be useful in the analysis of these location problems. The four models studied here allow for the consideration of a number of different factors, which might be included in a siting decision, and are easily extended to include various problem constraints. It is hoped that the results presented here will prove useful in deciding which model, or models, are most suitable for a given application.

ACKNOWLEDGEMENT

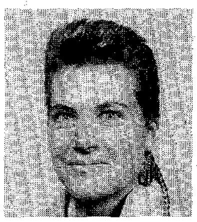
This research has been supported in part by the Natural Sciences and Engineering Research Council of Canada (grant number OGP 25481), and the Southam Fellowship and the Muir Fund of the Faculty of Business, University of Alberta.

REFERENCES

1. Chandrasekaran, R. and A. Daughety (1981), "Location on Tree Networks: p-Centre and n-Dispersion Problems," *Math. Opns. Res.*, Vol. 6, No. 1, pp.50-57.
2. Chandrasekaran, R. and A. Tamir (1980), "An $O(n \log p)^2$ Algorithm for the Continuous p-Center Problem on a Tree," *SIAM J. Algeb. Dis. Methods*, Vol. 1, pp. 370-375.
3. Chandrasekaran, R. and A. Tamir (1982), "Polynomially Bounded Algorithms for Locating p-Centers on a Tree," *Math. Prog.*, Vol. 22, pp. 304-315.
4. Erkut, E. (1990), "The Discrete p-Dispersion Problem," *Eur. J. Opnl. Res.* Vol. 46, pp. 48-60.
5. Erkut, E., T. Baptie and B. v. Hohenbalken (1990), "The Discrete p-Maxian Location Problem," *Comp. and Opns. Res.*, Vol. 17, pp. 51-61.
6. Erkut, E. and S. Neuman (1989), "Analytical Models for Locating Undesirable Facilities," *Eur. J. Opnl. Res.*, Vol. 40, pp. 275-291.
7. Halpern, J. and O. Maimon (1983), "Accord and Conflict among Several Objectives in Locational Decisions on Tree Networks," *Locational Analysis of Public Facilities*, J.-F. Thisse and H. G. Zoller (editors), North-Holland Publishing Company, pp. 301-314.
8. Hansen, P. and I. D. Moon (1988), "Dispersing Facilities on a Network," presented at the TIMS/ORSA meeting (Washington D.C., Apr. 1988).
9. Khumawala, B. M. (1973), "An Efficient Algorithm for the p-Median Problem with Maximum Distance Constraints," *Geogr. Analy.*, Vol. 5, pp. 309-321.
10. Khumawala, B. M. (1975), "Algorithm for the p-Median Problem with Maximum Distance Constraints: Extension and Reply," *Geogr. Analy.*, Vol. 7, pp. 91-95.
11. Kuby, M. J. (1987), "Programming Models for Facility Dispersion: The p-Dispersion and Maxisum Dispersion Problems," *Geogr. Analy.*, Vol. 19, pp. 315-329.
12. Meir, A. and J. W. Moon (1975), "Relations Between Packing and Covering Numbers of a Tree," *Pacific J. Math.*, Vol. 61, pp. 225-233.
13. Moon, I. D. and S. S. Chaudhry (1984), "An Analysis of Network Location Problems with Distance Constraints," *Mgmt. Sci.*, Vol. 30, No. 3, pp. 290-307.
14. Shier, D. R. (1977), "A Min-Max Theorem for p-Center Problems on a Tree," *Trans. Sci.*, Vol. 11, No. 3, pp. 243-252.
15. Tansel, B. C., R. L. Francis, T. J. Lowe and M. L. Chen (1982), "Duality and Distance Constraints for the Nonlinear p-Center Problem and Covering Problem on a Tree Network," *Opns. Res.*, Vol. 30, No. 4, pp. 725-744.
16. Toregas, C., R. Swain, C. ReVelle and L. Bergman (1971), "The Location of Emergency Service Facilities," *Opns. Res.*, Vol. 19, pp. 1363-1373.



Erhan Erkut is an Associate Professor in the Department of Finance and Management Science at the University of Alberta. He holds a B.Sc. in Industrial Engineering from Bogazici University in Istanbul, and a Ph.D. in Industrial Engineering from the University of Florida. Dr. Erkut has published in *Operations Research, Transportation Science, Interfaces, European Journal of Operational Research, Journal of Operational Research Society, Computers & Operations Research, Socio-Economic Planning Sciences* and *Environment and Planning B*. His current research interests include location of undesirable facilities and the routing of hazardous materials. He is a member of TIMS, COLA, ORSA, CORS, RSA, IIE and CAPICS.



Susan Neuman is a Research Assistant in the Department of Finance and Management Science at the University of Alberta. She holds a B.Sc. in Computing Science and an M.B.A. from the University of Alberta. In addition to her research on the location of undesirable facilities, she has also been involved in the development of a large-scale, mixed-integer programming model for coal transportation in China. Other projects include the design of multi-vehicle routing algorithms, to accommodate multiple time windows and vehicle capacity constraints.