Relaxation Method for the Solution of the Minimax Location-Allocation Problem in Euclidean Space

R. Chen
School of Physics and Astronomy, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel

G. Y. Handler
Leon Recanati Graduate School of Business Administration, Faculty of Management, Tel Aviv University, Tel Aviv 69978, Israel

A method previously devised for the solution of the p-center problem on a network has now been extended to solve the analogous minimax location-allocation problem in continuous space. The essence of the method is that we choose a subset of the n points to be served and consider the circles based on one, two, or three points. Using a set-covering algorithm we find a set of p such circles which cover the points in the relaxed problem (the one with $m < n$ points). If this is possible, we check whether the n original points are covered by the solution; if so, we have a feasible solution to the problem. We now delete the largest circle with radius r_p (which is currently an upper limit to the optimal solution) and try to find a better feasible solution. If we have a feasible solution to the relaxed problem which is not feasible to the original, we augment the relaxed problem by adding a point, preferably the one which is farthest from its nearest center. If we have a feasible solution to the original problem and we delete the largest circle and find that the relaxed problem cannot be covered by p circles, we conclude that the latest feasible solution to the original problem is optimal. An example of the solution of a problem with ten demand points and two and three service points is given in some detail. Computational data for problems of 30 demand points and 1–30 service points, and 100, 200, and 300 demand points and 1–3 service points are reported.

1. INTRODUCTION

In cases where the location of emergency facilities is considered, one usually utilizes the minimax criterion. This means that one attempts to locate the service facilities in such a way that the response time to the farthest "customer" will be minimal. In the context of the location of a single facility in two-dimensional Euclidean space, the problem can be formulated as

$$\min_{x,y} \max_{i} r_i, \tag{1}$$

where $r_i = [(x - a_i)^2 + (y - b_i)^2]^{1/2}$, (a_i, b_i) are the locations of the customers, $i = 1, \ldots, n$, and (x, y) is the variable point which is to be located optimally. Numerous authors (e.g., Hearn and Jesunathadas [12], Shamos and Hoey [17], Elzinga et al. [9]) solved this problem and its generalization,

$$\min_{x,y} \max_{i} w_i r_i, \tag{2}$$

Naval Research Logistics, Vol. 34, pp. 775–788 (1987)
 CCC 0028-1441/87/060775-14$04.00

where w_i is a weight attached to point i. Most of the methods developed for solving problem (1) (the extension of which will be considered in this paper) are "geometrical" in nature. The solution of the problem will be the center of the smallest circle enclosing n given points in the plane (Chrystal [3]). This can occur in one of two ways. The smallest circle can be determined by three demand points on its circumference or, alternatively, by two points which are on the two ends of a diameter. In the former case, the three points are the edges of an acute triangle (see Rademacher and Toeplitz [15]). The geometrical methods are based on a sophisticated search for the smallest enclosing circle among the circles built on subsets of two and three demand points.

Two kinds of extensions to this single-facility minimax problem can be considered. One is the multifacility minimax problem (Elzinga, Hearn, and Randolph [9]) in which one is interested in the optimal location of a number of *different* kinds of emergency stations (e.g., a fire station and a hospital); the solution as given in the literature is relatively simple, consisting of a repeated solution of the single-facility minimax problem. The more difficult extension is the location-allocation minimax problem in which a number of *identical* centers are to be located so that each demand point is served by the closest center. This problem can be written as (see Chen [2])

$$\min_{X_p C E^2} \max_i w_i \min_j r_{ij}, \tag{3}$$

where $X_p = \{x_j, y_j\}_{j=1}^{p}$, $r_{ij} = [(a_i - x_j)^2 + (b_i - y_j)^2]^{1/2}$ and $(x_j, y_j), j = 1, \ldots, p$ are the service points to be located optimally. The minimax location-allocation problem is the version applicable to the location of emergency stations of the rather well known minisum location-allocation problem (e.g., see Cooper [4], Eilon, Watson-Gandy, and Christofides [7], Ostresh [14], Scott [16]). In a recent paper, Chen [2] suggested a method which permits the solution of both the minisum and minimax location-allocation problems by using a differentiable approximation to the objective function and solving it by using nonlinear programming. This enabled the solution of very large problems, but the result was not necessarily optimal since local minima may have been reached. The minimax problem has been solved (see Handler and Mirchandani [11]) as a location-allocation problem, usually termed the p-center problem, on networks. This was first solved by Minieka [13], who devised a finite method which is rather inefficient for large problems. Handler and Mirchandani [11] improved on this by use of a relaxation approach. As described below, the present method is an adaptation of the method in [11], developed originally for network problems, to the present problem in continuous Euclidean two-dimensional space. As opposed to the method mentioned above [2], this yields optimal results for fairly large problems. However, the previous one is capable of solving much larger problems. A combination of the two methods is also considered in which the solution of the approximate method serves as a starting point for the optimal one.

A smaller number of investigators have recently tackled the present problem. Drezner [5] has developed a heuristic as well as an optimal method. The former can solve, not necessarily optimally, rather large problems, whereas the latter solves more limited problems. His examples contain results for problems with up to 60 demand points and 2 centers, 40 demand points and 4 centers or 30 demand points and 5 centers optimally. In another work, Drezner [6] offered a method which can optimally solve problems with a large number of demand points but only two service centers. Watson-Gandy [19] has also suggested an algorithm which can optimally solve problems with up to about 50 demand points and three centers in a reasonable time. Vijay [18] has very recently proposed an algorithm which appears to solve fairly large problems very efficiently. He presents computational results for problems with up to 100 demand points. Vijay's method, like Drezner's optimal algorithm [5], is similar to ours in that it solves a sequence of set-covering problems. The efficiency of his method is due to a geometrical technique whereby the number of columns in the set-covering matrices is significantly reduced. The algorithm described here is based on an entirely different idea. Using a relaxation approach the number of both rows and columns is significantly reduced. The relative efficiency of the method increases as a function of the number of demand points. As a thought for further research it seems to us that a hybrid version combining Vijay's method with the one described here might produce a highly efficient algorithm for very large-scale problems. Specifically, we would suggest using Vijay's algorithm to solve the reduced problems resulting from the relaxation process. The main advantage of the present method as compared to the others mentioned is the use of relaxation, which permits the solution of relatively large problems in most cases.

2. PROPERTIES OF THE SOLUTION

Let us consider first the properties of an optimal solution of the minimax location-allocation problem. Proceeding from our knowledge of the properties of the single-facility minimax problem and drawing an analogy from the p-center problem on networks, we can conclude that an optimal solution will consist of a set of p equal-radius circles which cover all the demand points. Usually, only one of these circles will be critical in the sense that two or three demand points will be on its circumference. There is much freedom in the exact position of the other circles and therefore in the location of all but one of the centers (see also [2]). The value of the solution is determined by the radius of this critical circle, whereas the radii of the other circles may vary in size below this critical value. Thus, the number of possible optimal solutions is usually infinite. The following theorem will help us to reduce the number of candidate optimal solutions to a finite number.

THEOREM: Among all the optimal solutions to the minimax problem of serving n demand points in Euclidean space by p service points, there is at least one in which all demand points are covered by critical circles, the largest of which has a radius r_p which is the value of the solution.

PROOF: Let us consider any optimal solution in which the n demand points are covered by p circles with radii r_p. Let us divide the n points into p disjoint subsets so that each demand point is covered by one of the circles with radius r_p. If a point is covered by more than one circle, we shall arbitrarily associate it with one of the covering circles. For each of the p subsets, let us consider the minimal covering circle. As known from the properties of the single-facility minimax solution, we obtain a circle based on one, two, or three demand points on the circumference. This yields p critical circles, the largest of which has a radius r_p. ∎

With the aid of this theorem, we can reduce the search to a finite number of critical circles. The number of critical circles to be considered is $\binom{n}{3} + \binom{n}{2} + n$, where $\binom{n}{3}$ is the number of circles determined by three points on their circumference, $\binom{n}{2}$ is the number of circles defined by two points determining the diameter, and n is the number of null circles, i.e., a service point at a demand point, the former serving solely the latter. The number of possible combinations to thus cover n points by p critical circles becomes very large indeed when n is large. The method described here shows how to reduce the number of combinations which are to be checked.

3. DESCRIPTION OF THE METHOD

The reduction in the number of candidate critical circles is achieved by utilizing the following ideas:

(i) If three demand points determine an obtuse triangle, we will disregard it, since a circle built on the two points farthest from one another will be smaller and will cover the three points [15].

(ii) In the following discussion we shall use the term "feasible solution" with respect to a set of given demand points and a set of given circles. If all the demand points are covered by the circles, the set of centers of the circles is said to be a feasible solution to the problem. The radius of the largest circle r_p is the value of the feasible solution. If we have a feasible solution to the problem with p service centers and we are looking for a better one, we will only consider circles with radii smaller than the current r_p. The fact that at each stage we can disregard any subset of two or three points determining a circle with radius larger than or equal to the current r_p reduces very substantially the number of subsets to be considered. Thus, for example, if we have an optimal solution for the $p - 1$-center problem, its value can serve as an upper limit to the value of r_p, and the number of candidate subsets is diminished. We can therefore solve the problems with $1,2,3, \ldots ,p$ service points sequentially. Alternatively, we can start with a known local minimum [2] and proceed by checking whether it is optimal or improve it. This permits us to find the p-center solution without going through the $1,2,3, \ldots ,p - 1$-center solutions. This method seems to be appropriate for the solution of relatively large problems.

(iii) As indicated previously, the essence of the method is the use of relaxations. Even with all the reductions mentioned above, the size of the problem may remain rather large. Similar to [11] we use the following strategy. We choose arbitrarily a subset of m demand points, $m < n$, and begin to solve the problem of optimally covering this subset by p critical circles. Once we find a feasible solution (if there exists one with a maximal radius smaller than the best solution known so far) to the relaxed (reduced) problem, we check whether it is a feasible solution to the full-size problem. This is done by checking the coverage by p circles with a radius of r_p, where r_p is the largest radius of the p critical circles covering the reduced problem. If there is no complete coverage of the full-size problem, we add a demand point to the relaxed point, preferably the one which is farthest away from its closest service center, and solve the enlarged relaxed problem. If the solution of the relaxed problem solves the full-size problem, we have an improved feasible solution to the latter and proceed by looking for a better one. We do this by reducing the current r_p to be the value of the latest feasible solution and choose again a relaxed problem and solve it. The procedure terminates when, for a given radius r_p, which is the best feasible solution known at the moment to the full-size problem, we cannot find a solution with value less than r_p to the reduced problem. It is obvious that if no solution can be found to the reduced problem under these circumstances, there will be no solution for the full-size one. Thus, the latest feasible solution to the n-points problem is necessarily the optimum.

An important point that has not been discussed so far is how to find a coverage to the reduced problem. For each critical circle considered we construct a vector of zeros and units having m terms, where m is the number of demand points in the reduced problem. In the ith place in the vector we insert unity if the ith point (out of the m) is covered by the critical circle and a zero otherwise. To each of these vectors we associate the coordinates of the center of the circle and its radius. Once we have this $0-1$ matrix we consider the unit set-covering problem, namely, the problem of finding the minimum number of columns (the vectors mentioned above) so that their sum does not include any zero. Suppose we find a solution with p columns; then we have a feasible solution to the reduced problem and the value of this solution is the largest of the radii associated with the covering columns. It is to be noted that the covering by p columns is not generally unique and hence the p-center solution is not necessarily optimal among all possible p-covers. However, the procedure is that once we find a feasible solution for the reduced problem, we do not have to check its being optimal, but rather test its feasibility to the full-size problem and proceed as explained above. In simple cases, such as the ten-demand-points example given below, the set-covering solution of the relaxed problem can usually be solved manually by inspection (see also [11] in the network problem). For larger problems, a computerized set-covering algorithm (see e.g., Garfinkel and Nemhauser [10] and Balas [1]) is required.

4. AN EXAMPLE

As a simple example we take the ten-demand-points problem defined in Table 1.

The solution found (by any of the established methods) for the single-facility minimax problem is a center at (45.4554, 51.7240) with a critical radius $r_1 =$ 46.5752. The critical points are (4,7,10). For the solution of two service points

Table 1. Ten demand points problem.

i	1	2	3	4	5	6	7	8	9	10
a_i	39	63	71	7	53	39	64	23	29	65
b_i	20	11	22	78	61	71	9	20	78	94

Table 2. Four points relaxation matrix.

Points \ Combination	1-4-10	1-4	1-7	1-10	4-7	4-10	7-10	1	4	7	10
1	1	1	1	1	1	0	1	1	0	0	0
4	1	1	0	0	1	1	0	0	1	0	0
7	0	0	1	0	1	0	1	0	0	1	0
10	1	0	0	1	0	1	1	0	0	0	1
r_p	40.3257	33.1210	13.6565	39.2173	44.7493	30.0832	42.5029	0	0	0	0
x_j	43.1414	23.0	51.5	52.0	35.5	36.0	64.5	39.0	7.0	64.0	65.0
y_j	60.1125	49.0	14.5	57.0	43.5	86.0	51.5	20.0	78.0	9.0	94.0

Table 3. Five points relaxation matrix.

Points \ Combination	1-4-10	4-5-10	1-4	1-5	1-7	1-10	4-5	4-7	4-10	5-7	5-10	7-10
1	1	0	1	1	1	1	1	1	0	1	0	1
4	1	1	1	0	0	0	1	1	1	0	0	0
5	1	1	1	1	0	1	1	1	0	1	1	1
7	0	0	0	0	1	0	0	1	0	1	0	1
10	1	1	0	0	0	1	0	0	1	0	1	1
r_p	40.3257	30.0836	33.1210	21.6622	13.6565	39.2173	24.5204	44.7493	30.0832	26.5754	17.5570	42.5029
x_i	43.1414	36.0418	23.0	46.0	51.5	52.0	30.0	35.5	36.0	58.5	59.0	64.5
y_i	60.1125	85.8484	49.0	40.5	14.5	57.0	69.5	43.5	86.0	35.0	77.5	51.5

we start from here where r_1 is the upper limit. Let us add arbitrarily point 1 and present the matrix shown in Table 2.

The 4 × 4 unity matrix on the right-hand side represents the null centers, each of which covers, obviously, only itself. In order to save space we shall not include it in the tables below, although its existence will be implicitly assumed. Let us take as a solution of the relaxed problem the third (1–7) and sixth (4–10) columns. The solution we take has a value (radius) $r_2 = 30.0832$. This value, along with the data on the centers, (51.5,14.5) and (36.0,86.0) is inserted into the other part of the program which checks feasibility of the full-size problem. We find that the solution of the reduced problem is not a feasible solution of the large one, and that the point farthest from its closest center is No. 5. We add this point to the above-mentioned four and return to the first part of the program. The limiting value for critical circles is still the best solution known so far, i.e., $r_1 = 46.5752$. We now have Table 3.

The second and fifth columns constitute a solution to the relaxed problem with $r_2 = 30.0836$. Checking this solution of centers at (36.0418,85.8484) and (51.5,14.5) shows that it is a solution of the original problem. We therefore delete all columns with $r_p \geq 30.0836$ and remain with the matrix shown in Table 4. Again, the 5 × 5 unity matrix is not explicitly shown. Columns 4 and 5 represent a cover for the relaxed problem. Checking the full-size problem shows that it is not a solution. We therefore have to add a point to the relaxed problem. The point found to be farthest from its closest center is No. 8. We add it to the current relaxed problem and obtain Table 5. The relaxed problem can now be covered by columns 1 and 6. Checking this solution with $r_2 = 30.0832$ shows that it is indeed a solution of the full-size problem, which is slightly better than the previous one. Now deleting column 6 leaves us with a matrix which has no cover with two columns. The latest solution, namely, centers at (48.4284,32.8695) and (36.0,86.0) with $r_2 = 30.0832$ is thus the optimal solution of the two-center problem.

Table 5 with column 6 omitted can be utilized for the solution of the three-center problem. A cover can be found by columns 1, 5, and 9 and the algorithm proceeds in the same way as before. The final results for $p = 1, \ldots, 10$ centers are summed up in Table 6.

Table 4. Updated five points relaxation matrix.

Points \ Combination	1-5	1-7	4-5	4-10	5-7	5-10
1	1	1	0	0	1	0
4	0	0	1	1	0	0
5	1	0	1	0	1	1
7	0	1	0	0	1	0
10	0	0	0	1	0	1
r_p	21.6622	13.6565	24.5204	30.0832	26.5754	17.5570
x_j	46.0	51.5	30.0	36.0	58.5	59.0
y_j	40.5	14.5	69.5	86.0	35.0	77.5

Table 5. Six points relaxation matrix.

Points \ Combination	5-7-8	1-5	5-7	1-8	4-5	4-10	5-7	5-8	5-10	7-8
1	1	1	1	1	0	0	1	1	0	1
4	0	0	0	0	1	1	0	0	0	0
5	1	1	0	0	1	0	1	1	1	0
7	1	0	1	0	0	0	1	0	0	1
8	1	0	0	1	0	0	0	1	0	1
10	0	0	0	0	0	1	0	0	1	0
r_p	28.4500	21.6622	13.6565	8.0	24.5204	30.0832	26.5754	25.4018	17.5770	21.2250
x_j	48.4284	46.0	51.5	31.0	30.0	36.0	58.5	38.0	59.0	43.5
y_j	32.8695	40.5	14.5	20.0	69.5	86.0	35.0	40.5	77.5	14.5

Table 6. Results of 1,2,3, . . . ,10 demand points problems.

p	r_p	Centers
1	46.5752	(45.4554,51.7240)
2	30.0832	(36.0,86.0),(51.5,14.5)
3	24.0208	(47.0,21.0),(59.0,77.5),(7.0,78.0)
4	17.557	(55.0,21.0),(31.0,20.0),(23.0,74.5),(59.0,77.5)
5	11.00	(31.0,20.0),(67.5,15.5),(18.0,78.0),(46.0,66.0),(65.0,94.0)
6	8.01	(31.0,20.0),(67.5,15.5),(34.0,74.5),(7.0,78.0),(53.0,61.0),(65.0,94.0)
7	7.39	(67.5,15.5),(34.0,74.5),(39.0,20.0),(7.0,78.0),(53.0,61.0),(23.0,20.0) (65.0,94.0)
8	6.11	(34.0,74.5),(39.0,20.0),(71.0,22.0),(7.0,78.0),(53.0,61.0),(64.0,9.0), (23.0,20.0),(65.0,94.0)
9	1.118	All demand points except for Nos. 2 and 7, and (63.5,10.0)
10	0	All demand points

5. COMPUTATIONAL EXPERIENCE

While the relaxation technique allows manual solution of small problems, larger problems require the use of a computerized algorithm. Preliminary computational experience that has been gained in the course of this research is described in this section.

Tables 7 and 8 contain a summary of results for six randomly generated problems with the number of demand points ranging from $n = 10$ to $n = 300$ and the number of service facilities between $p = 1$ and $p = 30$. In all, 45 p-center problems were solved using the CDC CYBER 170-855 computer.

For a given number n, the locations of the n demand points were randomly generated in a 100×100 square by choosing x and y coordinates from a uniform distribution with integral values from 1 to 100. The experiments were designed to provide some insight to problem complexity with respect to both input parameters $-p$ and n. Thus, Table 7 describes computational results for a 30-demand-points problem solved for all possible values of p. Table 8 provides results for six values of n, $n = 10,20,30,100,200,300$ where each is solved for $p = 1,2,3$. Before proceeding with a discussion of the results, we need to define the symbols appearing in the tables. The letters n,p refer to the number of demand points and the number of service facilities, respectively. CPf signifies the size (rows by columns) of the final set-covering problem solved in finding a given p-center solution. CPs denotes the number of set-covering problems solved for a given value of p. Cols is the maximum number of columns among the CPs set-covering problems for a given p, and Cuts is the total number of cuts generated while solving the CPs set-covering problems for a given p. Finally, Tcp is the total time in seconds to solve the CPs set-covering problems for a given p, and T is the total (incremental) time in seconds to solve the p-center problem for a given p.

We turn now to a discussion of the computational results in Tables 7 and 8. As expected, most of the computation time is taken up by the set-covering portion of the algorithm. An exception to this is the case of $n = 1$, where the set-covering problem degenerates into a trivial problem and furthermore, some overheads are incurred in setting up the problem. But clearly, as problem complexity increases, the major computational burden is due to solution of the

Table 7. Computational results for 30-demand-points problem. p, number of service facilities; CPf, size of final covering problem (rows × columns); CPs, number of covering problems; Cols, maximum number of columns; Cuts, total number of cuts in solving covering problems; Tcp, total time in seconds to solve the covering problems; and T, total incremental computation time in seconds (including Tcp).

p	CPf	CPs	Cols	Cuts	Tcp	T
1	5 × 15	4	16	1	0.007	0.660
2	12 × 113	18	122	31	1.614	2.034
3	14 × 88	23	120	24	0.930	1.713
4	15 × 67	19	85	85	4.796	5.251
5	17 × 75	16	78	27	1.050	1.507
6	19 × 74	21	88	37	2.798	3.602
7	20 × 79	16	80	27	1.916	2.522
8	21 × 66	23	78	66	9.843	11.048
9	23 × 74	19	79	58	8.657	9.741
10	22 × 64	14	64	163	46.061	46.627
11	24 × 55	19	65	19	2.058	3.252
12	25 × 56	16	63	16	2.124	3.089
13	24 × 43	15	47	0	0.508	1.378
14	23 × 38	12	40	0	0.382	1.020
15	26 × 46	13	47	3	0.728	1.567
16	27 × 47	13	47	0	0.578	1.497
17	28 × 45	16	51	2	1.208	2.522
18	27 × 42	11	43	0	0.561	1.399
19	27 × 36	12	42	0	0.698	1.732
20	29 × 39	11	40	0	0.673	1.689
21	30 × 40	11	41	0	0.740	1.849
22	30 × 38	11	40	0	0.799	1.989
23	30 × 36	9	38	0	0.673	1.652
24	27 × 30	4	30	0	0.268	0.630
25	30 × 33	7	36	0	0.591	1.420
26	29 × 32	4	32	0	0.333	0.772
27	30 × 33	4	33	0	0.365	0.850
28	30 × 31	5	33	0	0.492	1.172
29	30 × 30	3	30	0	0.302	0.705

respective set-covering problems. Our discussion will center around three categories—the set-covering algorithm, computational effort as a function of p, and computational effort as a function of n.

(i) The Set-Covering Algorithm

The set-covering algorithm used in this research is due to Bellmore and Ratliff as reported in Garfinkel and Nemhauser [10, Chap. 8]. Large computation times and memory requirements, when they occur, are due primarily to a large number of cuts. An interesting feature here is that the great majority of cuts were generated at the final covering problem for any value of p. It seems apparent from Table 7 that the size of problem that can be solved with the relaxation technique is essentially limited by the number of cuts generated by the set-covering algorithm. In this research effort we have not emphasized the computational efficiency of the set-covering algorithm. Two points are worth noting in this respect. First, the performance of the algorithm utilized here could be

Table 8. Computational results for several problems. n, number of demand points; p, number of service facilities: CPf, size of final covering problem (rows × columns); CPs, number of covering problems; Cols, maximum number of columns; Cuts, total number of cuts in solving covering problems; Tcp, total time in seconds to solve the covering problems; and T, total incremental computation time in seconds (including Tcp).

p	n	10	20	30	100	200	300
1	CPf	5 × 15	6 × 24	5 × 15	7 × 39	6 × 25	6 × 28
	CPs	4	5	4	6	5	5
	Cols	16	25	16	40	26	29
	Cuts	1	1	1	6	7	7
	Tcp	0.010	0.010	0.007	0.036	0.023	0.028
	T	0.902	0.701	0.660	0.746	0.747	0.804
2	CPf	8 × 37	12 × 101	12 × 113	12 × 115	14 × 134	15 × 190
	CPs	9	16	18	16	20	20
	Cols	49	101	122	126	134	194
	Cuts	3	96	31	36	26	96
	Tcp	0.073	10.843	1.614	1.608	1.138	19.741
	T	0.169	11.174	2.034	2.011	1.813	20.664
3	CPf	8 × 30	13 × 58	14 × 88	19 × 197	24 × 295	26 × 354
	CPs	9	11	23	27	26	40
	Cols	37	111	120	246	326	398
	Cuts	1	20	24	28	146	232
	Tcp	0.059	0.603	0.930	2.509	79.152	126.369
	T	1.224	1.038	1.713	4.426	83.689	132.77

substantially improved by sophisticated computer programming, for example, storage of the zero-one elements of the matrix in bits rather than in words. Second, the relatively outdated algorithm utilized here should be replaced by state-of-the-art techniques such as the recent one developed by Balas [1].

(ii) Computational Effort as a Function of p

Table 7 demonstrates how computational effort increases and then decreases as a function of p. An important factor influencing computational effort is the number of columns in the covering problems. Inherent in the relaxation method are two features which influence the number of columns as a function of p in opposite directions. As p increases, r_p decreases, thus reducing the number of columns, while the number of demand points in the relaxed set increases, thus increasing the number of columns. The net effect of these two factors appears to follow the pattern indicated in the table. As indicated earlier, the critical factor influencing computational effort is the number of cuts, and it appears from the table that this quantity is highly correlated with the number of columns.

(iii) Computational Effort as a Function of n

The advantages of the relaxation technique are clearly demonstrated in Table 8. Without the use of relaxations, the p-center problem quickly becomes unmanageable as n increases. For example, for $n = 100$ the size of the covering problem matrices would be 100 rows by 166,750 columns without relaxation. For $n = 300$ the corresponding size would be 300 by 4,500,250. In comparison, the relaxation algorithm yields very much smaller matrices, as indicated in the table. Furthermore, it appears that computational effort increases as a low-order polynomial function of n, though more computational experience is needed in order to validate this assertion.

The procedure discussed so far has been to get, for a given number of service points p, successively better solutions until the optimum is reached. Once this is achieved, the optimal solution for p service facilities is utilized as a first feasible solution for the problem with $p + 1$ facilities. Thus, in order to solve a problem with, say, ten service points, one has to go through the optimal solutions of 1,2,3, . . . ,9 service points, which may be very time consuming. This can be bypassed by the following procedure. Suppose we have a reasonably good feasible solution to the problem with p centers (see [2]). If the value r_p of this solution is relatively small, the number of columns eliminated will be large, and we will often be left with a manageable number of columns.

6. CONCLUSIONS

A relaxation method for the Euclidean p-center problem has been described. The method is capable of solving large-scale problems. Furthermore, the method offers the possibility for substantial improvement to solutions found by heuristic methods in very large problems that cannot be solved optimally.

Preliminary computational experience indicates that expected computational effort increases as a low-order polynomial function of the number of demand points.

Further research efforts are needed to consolidate the results attained thus far. In particular, the efficiency of the method can be substantially improved by use of an up-to-date set-covering algorithm and by more efficient computer programming. Finally, more extensive computational experiments are required to characterize the efficiency of the proposed method.

REFERENCES

[1] Balas, E., "Cutting Planes from Conditional Bounds: A New Approach to Set Covering," *Mathematical Programming*, **12**, 19–36 (1980).

[2] Chen, R., "Solution of Minisum and Minimax Location-Allocation Problems with Euclidean Distances," *Naval Research Logistics Quarterly*, **30**, 449–459 (1983).

[3] Chrystal, G., "On the Problem to Construct the Minimum Circle Enclosing n Given Points in the Plane," *Proceedings of the Edinburgh Mathematical Society*, **3**, 30–33 (1885).

[4] Cooper, L., "Heuristic Methods for Location-Allocation Problems," *SIAM Review*, **6**, 37–53 (1964).

[5] Drezner, Z., "The p-Center Problem—Heuristic and Optimal Algorithms," *Journal of Operational Research*, **35**, 741–748 (1984).

[6] Drezner, Z., "The Planar Two Center and Two Median Problems," *Transportation Science*, **18**, 351–361 (1984).

[7] Eilon, E., Watson-Gandy, C. D. T., and Christofides, N., *Distribution Management: Mathematical Modelling and Practical Analysis*, Griffin, London, 1971.

[8] Elzinga, J., and Hearn, D. W., "Geometrical Solutions for Some Minimax Location Problems," *Transportation Science*, **6**, 379–394 (1972).

[9] Elzinga, J., Hearn, D. W., and Randolph, W. D., "Minimax Multifacility Location with Euclidean Distances," *Transportation Science*, **10**, 321–336 (1976).
[10] Garfinkel, R. S., and Nemhauser, G. L., *Integer Programming*, Wiley, New York, 1972.
[11] Handler, G. Y., and Mirchandani, P. B., *Location on Networks: Theory and Algorithms*, MIT Press, Cambridge, MA, 1979.
[12] Hearn, D. W., and Jesunathadas, J., "Analysis and Extensions of Algorithms for Sylvester's Minimax Location Problem," Research Report No. 80-1, Industrial and Systems Engineering Department, University of Florida, Gainesville, FL, 1980.
[13] Minieka, E., "The *m* Center Problem," *SIAM Review*, **12**, 139–140 (1970).
[14] Ostresh, Jr., L. M., "Strong Heuristics for the Location-Allocation Problem," in *Proceedings of the International Symposium on Location Decisions, Banff, Alberta, 1978.*
[15] Rademacher, H., and Toeplitz, O., "The Spanning Circle of a Finite Set of Points," in *The Enjoyment of Mathematics*, Princeton University Press, Princeton, NJ, 1957, pp. 103–110.
[16] Scott, A. J., "Location-Allocation Systems: A Review," *Geographical Analysis*, **2**, 95–119 (1978).
[17] Shamos, M. I., and Hoey, D., "Closest Point Problems," in *Proceedings of the 16th Annual IEEE Symposium*, Foundations of Computer Science, 1975.
[18] Vijay, J., "An Algorithm for the *p*-Center Problem in the Plane," *Transportation Science*, **19**, 235–245 (1985).
[19] Watson-Gandy, C. D. T., "The Multi-Facility Min-Max Weber Problem," *European Journal of Operational Research*, **18**, 44–50 (1984).

Manuscript received June 28, 1983
Revised manuscript received November 1, 1985
Accepted December 17, 1986