The Conditional p-Center Problem in the Plane

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

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

An algorithm is given for the conditional p-center problem, namely, the optimal location of one or more additional facilities in a region with given demand points and one or more preexisting facilities. The solution dealt with here involves the minimax criterion and Euclidean distances in two-dimensional space. The method used is a generalization to the present conditional case of a relaxation method previously developed for the unconditional p-center problems. Interestingly, its worst-case complexity is identical to that of the unconditional version, and in practice, the conditional algorithm is more efficient. Some test problems with up to 200 demand points have been solved. © 1993 John Wiley & Sons, Inc.

Location-allocation problems are those in which a number of service facilities, usually (as here) assumed identical, are to be optimally located to serve a number of given demand points. Our concern here is with the case in which the underlying universe for location and travel is the two-dimensional Euclidean space. The two major versions of the problem studied so far are the minisum problem in which the minimand is the sum of weighted distances of the demand points to their closest service facilities, and the minimax problem in which one wishes to minimize the maximum distance from any demand point to its closest center. The minisum problem was first suggested and treated by Cooper [6], and later dealt with by several investigators [3, 11, 18, 21]. The subject of this article, on the other hand, is related to the minimax problem, known also as the p-center problem. Common applications for this model include emergency services and communication centers. It has been discussed by a number of authors in recent years [3, 8, 9, 15, 22, 23]. In a previous work by the present authors [4], a relaxation method for the minimax problem has been found to work well for fairly large problems.

The location problems mentioned above deal with the simultaneous optimal location of a number of identical (uncapacitated) service centers, and the allocation of each of the demand points to its closest center. A problem which may be of no less practical significance is the conditional location problem where a

Naval Research Logistics, Vol. 40, pp. 117–127 (1993)

CCC 0894-069X/93/010117-11

number of facilities are to be optimally located, given a set of existing facilities. This kind of problem is of practical importance since it is quite likely that a decision maker, while considering addition of new service facilities to a number of existing centers, may not wish to consider altering the locations of the given ones due to the high cost involved.

To the best of our knowledge, the first reference to this "new center" problem was by Lin [14] (see also [13, pp. 156 and 208]), although a previous paper by Poulton and Kanafani [19] had dealt with the related subject of the location of off-airport terminals. The term "conditional" for this type of problem was coined by Minieka [16]. Lin and Minieka deal with graph problems and with just one additional center. Drezner [10] discussed the solution of the conditional p-center problem (with any metric or on a network), and showed that it can be solved by solving O (log n) unconditional p-center problems.

Here we propose an algorithm for the conditional minimax problem in two-dimensional Euclidean space (E^2). This problem has been addressed by Chen [5], who gave a solution method based on applying nonlinear programming to a differentiable approximation to the objective function. This yields good though not necessarily optimal results. In the present work we describe an algorithm yielding optimal solutions. The complexity of this algorithm is identical to that of its predecssor [4] for the unconditional problem, and in practice, it is usually more efficient, a distinct advantage over the method suggested in [10]. Before proceeding with the description of the algorithm, we present mathematical formulations of the unconditional and conditional p-center problems.

The conditional problem in E^2 is that of determining the set of p centers X_p^* and the p radius r_p^* given by

$$r_p^* = \min_{X_p \subset E^2} \max_{i=1,\ldots,n} \{ \min_{j=1,\ldots,p+m} w_{ij}[(a_i - x_j)^2 + (b_i - y_j)^2]^{1/2}\}, \qquad \text{(C)}$$

where (a_i, b_i), $i = 1, \ldots, n$ are the given demand points, $X_p = \{(x_j, y_j)\}_{j=1}^p$ are the service centers to be optimally located, and there exist m fixed centers denoted by $\{(x_j, y_j)\}_{j=p+1}^{p+m}$. The w_{ij}'s are the positive weights associated with the demand and supply pairs, the distinctive characteristics of which might require dispatch of different types of vehicles incurring different costs. However, we shall follow the usual practice of minimax problems of accepting equal weights (unity) as realistic. The meaning of (C) is that $\min_j$ selects for each demand point a closest service facility, $\max_i$ takes the largest (weighted) distance of the demand points to the service centers, and $\min_{X_p \subset E^2}$ minimizes the maximum distance on $x_1, y_1, \ldots, x_p, y_p$. This formulation generalizes the unconditional case in which $m = 0$.

1. THE RELAXATION METHOD FOR THE UNCONDITIONAL PROBLEM

The method reported in this article for conditional location is an extension of a previous work by the same authors for the unconditional problem [4]. This, in turn, is an adaptation of a relaxtion method given by Handler and Mirchandani [13] for the analogous problem in networks. Let us first define a *critical circle* (boundary plus interior) as one which either has three (or more) demand points on its circumference, or has a pair of demand points at the ends of a diameter, or is a degenerate circle consisting of a single demand point. A cornerstone in the development of the algorithm by Chen and Handler [4] is the following well-known theorem. Among all the optimal solutions to the p-center problem there is at least one in which all the demand points are covered by critical circles, the largest of which has a radius r_p^* which constitutes the value of the solution. Using this theorem, one can reduce the search for an optimum location set from an infinite number of possibilities to the finite number of centers of critical circles. The number of critical circles to be considered is $\binom{n}{3} + \binom{n}{2} + n$. Each of these circles is a *candidate circle* which may possibly be included in feasible and optimal solutions.

For problems with anything but very small n, the number of candidate solutions (i.e., p sets of critical circles) is very large; a number of ways have been utilized to reduce this number to a manageable size. These are described in the previous paper [4], and are briefly repeated here, followed by a formal presentation of the algorithm now given for the first time.

1. While considering the set of three points as candidate to form a critical circle, a set forming an obtuse triangle should be discarded.

2. If all the demand points in a problem are covered by a given set of critical circles, the set of centers of these circles is said to be a *feasible solution* to the problem with value equal to the radius of the largest critical circle. If we have a feasible solution to the problem with p service centers and we are looking for a better one, we need not (and will not) consider critical circles with radii equal to or larger than the value of the current feasible solution. This can reduce very substantially the number of subsets to be considered. If we have, for example, an optimal solution for the p-1-center problem, its value can serve as an upper bound on the value of r_p^* and the number of candidate subsets is thus diminished. We can therefore solve the problems with 1, 2, 3, . . . , p service facilities sequentially. Results of this way of solving the problems are shown in Tables 1-5. Alternatively, we can start with a known local minimum (see [3]) and proceed by checking whether it is optimal or improve upon it. This method is demonstrated in Table 6.

3. The essence of the method is the use of relaxations. To start off we assume given an upper bound r_p on the optimal value of the p radius r_p^*, together with a corresponding feasible location set which we denote X_p. We choose a "relaxed" subset R of k demand points, $k < n$ (we found the initial choice $k = \max\{3, p\}$ to be useful) and identify all critical circles generated by these k demand points and having radii less than r_p. (The number of these circles is initially $\binom{k}{3} + \binom{k}{2} + k$, but can immediately be reduced using the criteria given earlier.) Next we search for a p cover of R from these candidate circles by working on a minimum-cardinality set-covering problem (see [4]). It is to be noted that we do not have to solve the relaxation problem to optimality; any feasible solution (one with cardinality $\leq p$) suffices. We shall refer to this process as *feasibility-solving* the covering problem for R; it terminates either with a feasible solution, or with an optimal solution of cardinality $> p$. In the latter case, the procedure terminates and the incumbent quantities r_p, X_p are the value and location for the full-size problem. Suppose now that a p cover of R is found. [Note that the

set-covering algorithm may yield a z cover where $z < p$. By adding $(p - z)$ points arbitrarily, we can convert this to a p cover.] Denote the set of centers of this set of critical circles by $\hat{X}_p$.

We now check whether this solution to the relaxed problem constitutes an improved solution to the full-size problem. To do this we compute the quantity $l(\hat{X}_p) = d(y^0, \hat{X}_p) = \max_{y \in N} d(y, \hat{X}_p)$, where $d(y, \hat{X}_p) = \min_{x \in \hat{X}_p} d(y, x)$, N is the set of n demand points defined in (1), y is any element of N, and $d(y, x)$ is the straight-line distance between y and x in E^2. Thus, $l(\hat{X}_p)$ is the distance from the set $\hat{X}_p$ to the farthest demand point y^0. If $l(\hat{X}_p) \geq r_p$ then $\hat{X}_p$ is not an improved solution to the full-size problem. In this case we add to R the demand point y^0, enlarge the set of candidate circles to include the critical circles of radius $< r_p$ passing through y^0, and again search for a p cover of R continuing as before. If, on the other hand, $l(\hat{X}_p) < r_p$ then an improved solution to the full-size problem is at hand. We can thus update the incumbent solution by setting $r_p = l(\hat{X}_p)$ and $X_p = \hat{X}_p$. We are now ready to repeat the whole procedure of searching for an improved solution by choosing again an arbitrary set R and continuing as before. Note that an alternative is to retain the current set R but eliminate from the current set of candidate circles those with radii greater than or equal to r_p, and again search for a p cover of R.

For the sake of clarity we summarize the basic steps of the procedure in a formal algorithm. In addition to the symbols defined above, we define the covering set problem CP(R:r_p) composed of candidate circles generated by points in R, with radii less than r_p, denote by z the cardinality of the (optimal or feasible) solution found for CP(R;r_p), and by $\hat{X}_z$ the corresponding z set of locations (centers of candidate circles). We note again that it suffices to find a feasible solution to the CP.

Algorithm for Unconditional *p* Center

0. Given initial feasible solution X_p, with value r_p.
1. Choose $R \subseteq N$.
2. Set up and feasibility-solve CP(R:r_p) $\rightarrow z, \hat{X}_z$. If $z > p$ then $X_p^* = X_p$, $r_p^* = r_p$ is an optimal solution.
3. If $l(\hat{X}_z) = d(y^0, \hat{X}_z) > r_p$, and y^0 to R and go to 2. Otherwise, let $X_p = \hat{X}_z$, $r_p = l(\hat{X}_z)$ and go to 1. If two or more points tie for the role y^0 of "worst uncovered," choose one arbitrarily.

The arbitrariness in Step 1's choice, here and in the conditional case algorithm below, may be surprising. But these choices only initialize (or reinitialize) R at a relatively small size; that choice is augmented (in Step 3) in a systematic and heuristically efficient way, so that when the critical stages of the procedure are reached, the relaxed problems are very far from being arbitrary.

2. THE RELAXATION METHOD FOR THE CONDITIONAL PROBLEM

The procedure of solving the conditional problem is a modified version of the previous algorithm. Indeed the new algorithm comprises the same four steps, with two modifications which may be viewed as generalizing the unconditional algorithm. We shall discuss these modifications and then present the generalized algorithm.

Let F denote the set of m fixed facilities. Consider first a passage through Step 1, in which we choose a new set of demand points R. In the conditional problem we must select, for R, demand points that are not within a distance r_p (the value of the incumbent solution) from the set F of fixed points, for, a demand point initially covered by F will not properly guide the improvement of the current solution. Of course, as the procedure unfolds, the value of r_p decreases and the set of demand points from which to select R will grow.

The second modification occurs in step 3, where the quantity $l(\hat{X}_z)$ must be revised to read $l(\hat{X}_z/F)$ defined as $l(\hat{X}_z \cup F) = \max_{y \in N} d(y, \hat{X}_z \cup F)$. In this manner we can determine the demand point farthest from the set of fixed and variable centers. We are now ready to formulate the procedure.

Algorithm for Conditional p Center

0. Given initial feasible solution X_p, with value of r_p.
1. Choose $R \subseteq \{y \in N: d(y, F) > r_p\}$.
2. Set up and feasibility-solve CP$(R:r_p) \to z, \hat{X}_z$. If $z > p$ then $X_p^* = X_p$, $r_p^* = r_p$ is an optimal solution.
3. If $l(\hat{X}_z|F) = d(y^0, \hat{X}_z \cup F) > r_p$, add y^0 to R and go to 2. Otherwise, let $X_p = \hat{X}_z$, $r_p = l(\hat{X}_z|F)$ and go to 1. If two or more points tie for the role y^0 of "worst uncovered," choose one arbitrarily.

It is evident from the foregoing discussion that computational complexity of the conditional algorithm is, at worst, identical to that for the unconditional case, while in practice, we may expect the algorithm to be increasingly efficient as fixed facilities are added. This is borne out by our computational experience described in the next section (Table 3).

3. COMPUTATIONAL RESULTS

Problems with 30, 40, 50, 100, and 200 demand points distributed at random over different regions have been solved. The regions include a 100 × 100 square with three fixed service centers located at (10, 10), (10, 90), (90, 10) as well as various rectangles and ellipses with different eccentricities. In the latter cases, the number of fixed service centers varied between 3 and 10, and their locations over the given region were determined by the same random number generator utilized to determine the locations of the demand points. It may be stated at the outset that no significant variation has been found in the performance of the method under different positions of the demand points (with a possible exception noted in the discussion below.)

Table 1 gives the results for problems on the above-mentioned square. The runs reported in this table were performed on the Cyber 180-990 /DO CDC computer under the NOS/VE operating system at Tel-Aviv University. The demand points were chosen in such a way that the 30 points are a subset of the 40 demand points, which in turn are a subset of the 50 demand points, and so on. This choice has been made since it will aid us in demonstrating a way for

Table 1. r_p^* values (and CPU times in secs) of representative problems with demand points distributed on a 100 × 100 square. Three fixed centers are at (10, 10), (10, 90), (90, 10).

No. of new centers	30 demand points	40 demand points	50 demand points	100 demand points	200 demand points
1	35.36(0.015)	36.46(0.02)	36.46(0.025)	39.12(0.10)	41.23(0.25)
2	29.84(0.20)	29.84(0.21)	29.84(0.42)	31.55(0.85)	33.47(1.37)
3	23.02(0.28)	23.02(0.28)	25.49(2.24)	30.15(2.88)	30.15(2.75)
4	19.24(0.34)	22.36(0.41)	22.69(*)	26.30(6.53)	26.31(2.32)
5	19.23(0.13)	19.70(1.37)		20.88(*)	24.04(23.2)
6	17.31(1.56)	19.24(7.39)			23.17(*)

*Indicates that the solution has not been proven to be optimal.

evaluating lower and upper bounds to the solution of a problem which cannot be sovled to optimality.

In the problems with 50 demand points and up, the solution procedure was terminated when the set-covering program utilized reached 200 "cuts." To solve the set-covering problem we used the algorithm due to Bellmore and Ratliff as reported by Garfinkel and Nemhauser [12]. The program solves this integer programming problem by producing "cuts," the number of which is usually very small (0 or 1 in most cases). However, rather capriciously, the number of cuts needed may sometimes be very large (say, over 200), which consumes a lot of computer time and memory, prompting termination of the process. (A possible improvement upon this point is discussed below.) Thus, all the last entries in the 50-, 100-, and 200-point problems are merely feasible solutions which are not necessarily optimal. (As observed by Chen and Handler [4], the same difficulty occurs also in the unconditional problem.)

Let us, however, consider the feasible solution with $r_4 = 22.69$ attained for 50 demand points and four variable centers. This feasible solution certainly provides an upper bound to the value of the solution. On the other hand, the optimal solution for the subproblem with 40 demand points yields a lower bound ($r_4 = 22.36$) to the 50-point problem, since the 40 demand points are a subset of the 50 points, as indicated above. Therefore, although we could not solve the 50-demand-point problem to the end in this way, we know that its solution has $22.36 \leq r_4^* \leq 22.69$ and therefore that the feasible solution with $r_4 = 22.69$ is certain to be at most 1.5% from the optimum. The conclusion is that in cases where one has a feasible solution in hand, but optimality cannot be reached, a possible strategy may be to solve to optimality a problem with a subset of the demand points and thus have both upper and lower bounds to the solution.

An interesting idea has been suggested* to deal with this situation not so common but troublesome, of having very many cuts. Due to the somewhat unpredictable nature of the set-covering subroutines, it may very well happen that at a given stage of the process, we get very many cuts and therefore cannot reach a solution in reasonable time. It is possible, however, that if we keep the incumbent r_p and choose an entirely different initial relaxation R, the difficulty may be overcome. We have therefore introduced a new parameter M such that

*We would like to thank the anonymous referee who suggested this improvement of the method.

Table 2. Run times for a problem with 50 demand points, three fixed centers, and different values of the cutoff parameter *M*.

M	New centers	CPU time (secs)
40	4	>213
50	10	54.9 ± 1.9
60	10	61.3 ± 2.7
80	10	83.3 ± 16.4
100	10	94.9 ± 21.9
300	10	196.0 ± 87.0

if the number of cuts at a certain stage exceeds *M*, the procedure is restarted with a different relaxation. For choosing this relaxation problem, different random numbers were used for the serial numbers of the initial demand points. The results of such a series of runs is shown in Table 2. In this case, the 50 demand points are still distributed over a square with the three fixed demand points as before. While setting $M = 40$, the solution process reached optimal results only for three centers, and a feasible solution for the four-center case, all in over 200 CPU seconds.

This was repeated with any initial choice of $k = 3$ points as R in the initial relaxation problem. With $M = 50$ we were able to go all the way to 10 new centers in just over 50 seconds. From this point on, each reported problem was solved five times, starting from different sets R in the initial relaxation problem chosen at random. The run times are given as averages ± standard deviation. With the larger values of *M*, Table 2 shows that the computation time increases, demonstrating the advantage of choosing the "appropriate" value of *M* (see discussion below).

An interesting point to note is that for a given number of demand points, the problem is easier to solve with a larger number of fixed centers. This can be understood intuitively, since, for a given value of r_p, some demand points are covered by the fixed centers. Therefore, the more fixed centers, the fewer uncovered demand points, so the remaining problem to be solved at each stage is smaller. Table 3 sums up the results for problems with 50 demand points randomly distributed over the 100×100 square with a variable number, between 3 and 10, of fixed centers and 10 new service centers. The cutoff parameter was kept at $M = 50$. The run times are seen to decrease dramatically, from an average of 328 seconds with three fixed centers to 12 seconds for 10 fixed points. For the sake of comparison, the unconditional problem with the same demand points was run with $M = 50$. In 557 CPU seconds, only seven centers were located optimally and a feasible solution was reached for the eight center problem.

Table 3. Run times for problems with 50 demand points distributed on a square, ten variable centers, and different number of fixed centers, $M = 50$.

Fixed centers	3	4	5	6	7	8	9	10
Average CPU time (sec)	328.4	149.9	135.4	140.3	38.6	34.4	16.2	12.0
Standard deviation (sec)	10.6	4.7	4.8	8.0	3.9	6.4	7.8	4.3

Table 4. Results for problems with demand points distributed on rectangles.

Demand points	B	Fixed points	M	New Cent.	CPU time (sec)
30	0.1	3	35	10	4.6 ± 0.4
30	0.25	3	35	10	11.9 ± 2.6
30	0.5	3	35	10	8.3 ± 0.8
30	1.0	3	35	10	6.3 ± 1.7
40	0.1	4	45	10	29.7 ± 11.7
40	0.25	4	45	10	28.3 ± 3.3
40	0.5	4	45	7+	>165
40	1.0	4	45	10	64.1 ± 3.6
40	2.0	4	45	10	30.6 ± 4.3

In order to check the possible dependence of the performance of the algorithm on the shape of the region over which the demand points are distributed, rectangular and elliptical regions were also considered. The fixed centers were again located randomly within the chosen region.

Table 4 presents the results for such problems with 30 and 40 demand points, and $M = 35$ and $M = 45$, respectively. The parameter B is the ratio between the two dimensions of the rectangle. All the problems were solved in "reasonable" time, except for the one with 40 demand points, four fixed centers and $B = 0.5$. Here, after 165 seconds, the optimal solution for seven variable centers was reached, with but a feasible solution for eight variable centers. Using different initial points in the relaxation problem did not remedy this difficulty. However, changing M to 60 yielded solutions to the 10 new centers problem in 39.6 ± 4.2 seconds. Although variations are seen in the run times, up to a factor of nearly 3 in the average run time for the 30-demand-points problems and over 5 in the 40-points case, no specific pattern could be recognized in the results. Similar behavior is seen in Table 5 for a case of problems over elliptical regions. Here, the parameter B is the ratio between the major and minor axes (the eccentricity is defined as $e = \sqrt{1 - B^2}$).

The procedure of solving for p centers by building up successively from $p =$ 1 up is attractive for purposes of sensitivity analysis. However, when this is not needed, one can bypass the tedious process of going through the optimal solutions of $1, 2, \ldots, p - 1$ new centers by generating an intial estimate for r_p and proceeding with the algorithm. One possibility is to guess an initial value of r_p. Of course r_p may be too small, and in this case, the result of "no cover for p centers" emerges. If, on the other hand, the chosen r_p is larger than the

Table 5. Results of different conditional problems with demand points distributed in ellipses.

Demand points	B	Fixed Centers	M	New centers	CPU Time (secs)
30	0.1	3	35	10	11.2 ± 4.6
30	0.25	3	35	10	23.5 ± 4.8
30	0.5	3	35	10	22.1 ± 4.8
30	1.0	3	35	10	23.5 ± 8.6
40	0.1	4	45	10	11.9 ± 2.1
40	0.25	4	45	10	16.1 ± 1.9
40	0.5	4	45	10	31.1 ± 9.0
40	1.0	4	45	10	19.6 ± 4.6

Table 6. Run times for a problem with 50 demand points, three fixed centers, and six variable centers, $M = 55$.

Initial radius*	41.30	35.50	27.61	23.92	22.79	20.51	19.72
CPU time (sec)	7.97	7.69	7.47	7.26	6.94	4.60	2.67
Standard deviation (sec)	4.42	4.04	9.59	4.60	3.92	2.11	2.11

*Initial radius obtained as a local solution in the heuristic method [5].

optimal solution, it is quite obvious that the computation time will be shorter if the initial value of r_p is closer to the optimum.

Alternatively, one can use a nonoptimal method, such as that given in [5] to generate an initial upper bound on r_p. In order to demonstrate this alternative, we have solved a problem with three fixed and six variable centers, 50 demand points, and $M = 55$, starting with the radii from various nonoptimal solutions found by the previous method. The value of the optimal solution of this problem is $r_p^* = 18.67$. The run times are given in Table 6, and are seen to decrease gradually as the initial radius decreases.

4. CONCLUSIONS

The conditional p-center problem in two-dimensional space has been dealt with. A relaxation method previously suggested for the unconditional version has been adapted to the conditional one. The computational experience, summed up in Tables 1–6, indicates some of the strong points as well as the weaknesses of the method. The particular set-covering procedure utilized, like many other integer programming methods, is rather capricious. Thus, in the initial version of the program in which the procedure was stopped when one of its set-covering problems reached 200 cuts, it did not solve to optimality (cf. Table 1) a four-center problem with 50 demand points and three fixed centers, although the initial 0-1 matrix involved is merely of size 13×57. In the larger problem with 100 demand points, the four-variable-center problem was optimally solved and the "too many cuts" situation occurred only with five variable centers when the set-covering matrix was of size 129×344. Furthermore, in the still larger problem with 200 demand points, the five new centers were optimally located. A partial remedy for such "explosion of cuts" has been offered above, involving changing the choice of relaxation problem once the number of cuts exceeds M, but keeping the current value of r_p. As shown in Table 2 this may help significantly, and, indeed, larger problems could be solved using this improvement. However, even this medicine should be prescribed with caution. As shown, taking too small a value for M may not allow the set-covering process to terminate, restarting it prematurely with a new relaxation. On the other hand, taking too large a value of M may permit excessive consumption of the computer time in treating individual set-covering problems, which can impede the procedure from getting the optimal solution. The task of finding an optimal value of M seems complicated, since it certainly depends on the numbers of demand points and fixed and variable centers, as well as their locations. As a rule of thumb, we found that for three fixed service centers, $M \sim n$ is quite useful. However, although changing the relaxation as described helps in solving larger problems, the unpredictable nature of the performance still remains. It is to be noted that the same trick of changing the relaxation problem when too many cuts occur, can be employed also in the unconditional problem (this has been mentioned above in association with the unconditional problem with 10 new points, $M = 50$ and the same demand points as in Table 3.

As pointed out above, no clear correlation has been found between the distribution of the demand points or the fixed service centers, and the computational effort needed to solve the problem. A possible exception to this observation may be the cases in which some fixed centers are clustered and can be viewed approximately as a single center. Since, as shown in Table 3, the solution process is less time consuming when more fixed centers exist, clustering is expected to increase the computation time for the solution. As another general observation, one can say that the introduction of a restricting value for the cutoff parameter M reduces substantially the standard deviation of the run times needed for solving the same problem with different initial choices of the relaxation problem.

The fact that larger conditional problems can be solved than the unconditional ones with the same demand points deserves further discussion. The main reason for this fact is that at each stage, a number of demand points are already covered by F, so that the size of the remaining problem to be covered (by relaxations) is smaller. Also, the existence of the fixed centers reduces, sometimes quite substantially, the magnitude of r_p at each stage. Therefore, more pairs and triplets of demand points are excluded while setting up the relaxation problem, further reducing (sometimes very appreciably) the size of the set-covering problem involved. Comparison with the computational experience in [4] for the unconditional case shows that on the average, the size of the solvable problems is indeed larger now. Comparing the present algorithm with that suggested by Drezner [10], one should remember that in his case, in order to solve a conditional p-center problem, the unconditional procedure should be repeated a number of times ($O(\log n)$). The present method generally requires less computational effort than a single unconditional problem, and in the worst case has the same complexity. This comparison is made between a p-center unconditional problem and a conditional one with m fixed centers and the same values of p and n.

Table 6 has demonstrated the potential for using a combined approach of starting with a feasible solution generated by a nonoptimal solution method such as that in [5], followed by the last stages of the present one. This approach has not decreased the compuation time very dramatically, since in many cases, it is the final stages which are time consuming. Still, it may be helpful for large-scale problems.

We would like to point out that the set-covering algorithm employed here has been superseded by more efficient algorithms [1, 2, 7, 17, 20]. Using a set-covering program based on the modern theory of facets may well increase the potential of the algorithm.

Finally, we believe that ideas in the present work may be generalized and carried over to p-center problems with any metric or on a network, and to various related methods. In particular, returning to the original definition of r_p^* given in (C), the algorithm extends with minor modifications to the weighted case where $w_{ij} = w_j$. However, for non uniform demand weights w_{ij}, a different concept is required to that of critical circles.

REFERENCES

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

[2] Balas, E., and Ng, S.M., "On the Set Covering Polytope:II. Lifting the Facets with Coefficients in {0, 1, 2}," *Mathematical Programming*, **45**, 1–20 (1989).

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

[4] Chen, R., and Handler, G.Y., "A Relaxation Method for the Solution of the Minimax Location-Allocation Problems in Euclidean Space," *Naval Research Logistics*, **34**, 775–788 (1987).

[5] Chen, R., "Conditional Minisum and Minimax Location-Allocation Problems in Euclidean Space," *Transportation Science*, **23**, 157–160 (1988).

[6] Cooper, L., "Location-Allocation Problems," *Operations Research*, **1**, 331–341 (1963).

[7] Cornuejols, G., and Sassano, A., "On the 0,1 Facets of the Set Covering Polytope," *Mathematical Programming*, **43**, 45–55 (1989).

[8] Drezner, Z., "The *P*-Centre Problem—Heuristic and Optimal Algorithms," *Journal of the Operational Research Society*, **35**, 741–748 (1984).

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

[10] Drezner, Z. "Conditional p-Center Problems," *Transportation Science*, **23**, 51–53 (1989).

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

[12] Garfinkel, R.S., and Nemhauser, G.L., *Integer Programming,* Wiley, New York, 1972.

[13] Handler, G.Y., and Mirchandani, P.B., *Location on Networks: Theory and Algorithms*, The MIT Press, Cambridge, MA, 1979.

[14] Lin, C.C., "A Note about the New Emergency Facility Insertion in an Undirected Connected Graph," Sixth Annual Pitts. Conference on Modelling and Simulation, 1975.

[15] Love, R.F., and Morris, J.G., "Computation Procedure for the Exact Solution of Location-Allocation Problems with Rectangular Distances," *Naval Research Logistics Quarterly*, **22**, 441–453 (1975).

[16] Minieka, E., "Conditional Centers and Medians on a Graph," *Networks*, **22**, 265–272 (1980).

[17] Nobili, P., and Sassano, A., "Facets and Lifting Procedures for the Set Covering Polytope," *Mathematical Programming*, **45**, 111–137 (1989).

[18] Ostresh, L.M., Jr., "An Efficient Algorithm for Solving the Two Center Location-Allocation Problem," *Journal of Regional Science*, **15**, 209–216 (1975).

[19] Poulton, M.C., and Kanafani, A., "The Application of Location Models to Off-Airport Terminals," *Transportation Science*, **9**, 224–247 (1975).

[20] Sassano, A., "On The Facial Structure of the Set Covering Polytope," *Mathematical Programming*, **44**, 181–202 (1989).

[21] Scott, A.J., "Location-Allocation Systems: A Review," *Geographical Analysis*, **II**(2), 95–119 (1970).

[22] Vijay, J. "An Algorithm for the p-Center Problem in the Plane," *Transportation Science*, **19**, 235–245 (1985).

[23] Watson-Gandy, C.D.T., "The Multi-Facility Min-Max Weber Problem," *European Journal of Operations Research*, **18**, 44–50 (1984).

Manuscript received June 15, 1989
Revised manuscript received June 22, 1992
Accepted June 29, 1992