Annals of Operations Research 40(1992)323–337

CAPACITATED EMERGENCY FACILITY SITING WITH MULTIPLE LEVELS OF BACKUP

Sridhar NARASIMHAN
School of Management, Georgia Institute of Technology, Atlanta, GA 30332-0520, USA

Hasan PIRKUL and David A. SCHILLING
College of Business, The Ohio State University, Columbus, OH 43210, USA

Abstract

In many service systems, the primary objective is to provide continuous service and/or service within a prespecified time interval. In the public sector, emergency service systems fit into this category. In the private sector, systems providing repair service to critical production facilities and computers constitute another example. In these systems, the concept of multiple service facilities providing backup to each other becomes an important element in the design process. In this paper, we study the capacitated facility siting problem with multiple levels of backup coverage. The problem is formulated as a mathematical program; an efficient solution procedure is developed and computational experiments are reported.

1. Introduction

Due to the nature of the service they provide, there are certain facility systems where a primary goal is to have the capability to provide that service at all times. In these cases, where the cost of not providing the service is sufficiently great and there is a significant likelihood the designated provider may be incapable, the system should be designed with redundancy to ensure continuous service. Clearly, in the public sector, emergency service systems fit into this category. One expects a call to 911 to bring an ambulance in a relatively short time. The inability to do so can be life threatening. In the private sector, a company guaranteeing computer repair services within twenty-four hours of a request risks losing customers if it fails to meet its claim. Facilities can be made unavailable by equipment breakdown, personnel problems, or other crises, but the most common reason is simply being occupied with another request for service. The more congested the system, the more likely this is to occur and, therefore, the greater the need for "backup" providers.

A base of literature is massing around the challenge of siting facility systems of critical services that meet the need for continuous service capability. Many researchers employ the concept of coverage in their backup service models because it ensures a worst-case lower bound on the quality of service delivered. These models include the location/simulation approach of Berlin [2], the multiple coverage-set covering formulation of Daskin and Stern [7], and the hierarchical maximal covering problem of Ruefli and Storbeck [26]. A brief review of backup service research appears in Hogan and ReVelle [17], who also develop a multiobjective formulation trading off primary and backup coverage. Each of the above are multiple coverage approaches which take no explicit consideration of facility workload. Weaver and Church [27] introduce the vector assignment p-median problem which allocates a fixed percentage of each site's demand to separate facilities. Again, however, without concern for a facility capacity.

It is in congested systems that one is most likely to find facilities operating at (or beyond) their workload limits. Thus, incorporating capacity restrictions in the siting model is a desirable enhancement. Several authors have included such limitations in the maximal covering location problems, but without addressing issues of backup service. Current and Storbeck [6] provide several foundations of the problem. Chung et al. [3] present a heuristic for solving the model with binary assignments, but only for relatively small problems. Pirkul and Schilling [24] develop another formulation of the capacitated maximal covering location problem and demonstrate an efficient heuristic based on a Lagrangian relaxation for large problems. Finally, Pirkul and Schilling [22] do employ facility capacities in the context of siting emergency facilities. In that paper, the authors consider a single level of backup service and do not utilize the concept of coverage.

The combination of facility capacities and backup service within the context of covering models has been examined in only one previous paper. Pirkul and Schilling [23] provide a formulation and solution approach for a capacitated facility model which seeks coverage from both a primary facility as well as a secondary facility. Only a single level of backup is considered, however. In highly congested systems and/or when service availability is hyper-critical, multiple levels of backup service will be needed. In this paper, we will generalize the work of Pirkul and Schilling [23] to permit any number of backup facilities.

2. Model development

In words, the model we examine here attempts to maximize the amount of demand covered by a fixed number of facilities such that all demand is assigned to a unique facility for each level of service and the total service provided by each facility does not exceed some specified service capacity.

Mathematically, the model can be stated as follows:

Problem P

$$Z_{\mathrm{P}} = \max \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ij}^k a_i^k x_{ij}^k \tag{1}$$

$$\text{subject to} \quad \sum_{j \in J} y_j = p, \tag{2}$$

$$\sum_{j \in J} x_{ij}^k = 1, \qquad \forall i \in I, \ \forall k \in K, \tag{3}$$

$$\sum_{k \in K} x_{ij}^k \le y_j, \qquad \forall i \in I, \ \forall j \in J, \tag{4}$$

$$\sum_{i \in I} \sum_{k \in K} a_i^k x_{ij}^k \le W_j, \qquad \forall j \in J, \tag{5}$$

$$y_j \in \{0, 1\}, \qquad \forall j \in J, \tag{6}$$

$$x_{ij}^k \in \{0, 1\}, \qquad \forall i \in I, \ \forall j \in J, \ \forall k \in K, \tag{7}$$

where

I = the index set of all demand points;
J = the index set of all potential facility sites;
K = the index set of all service levels;
a_i^k = the demand at point i for level k service;
W_j = the workload capacity for a facility at site j;
p = the number of facilities to be sited;
S^k = the maximum service distance or time for acceptable service at level k;
d_{ij} = the travel distance or time from j to i;

$$c_{ij}^k = \begin{cases} 1, & \text{if } d_{ij} \le S^k, \\ 0, & \text{otherwise;} \end{cases}$$

$$x_{ij}^k = \begin{cases} 1, & \text{if a facility at } j \text{ provides service of level } k \text{ to point } i, \\ 0, & \text{otherwise;} \end{cases}$$

$$y_j = \begin{cases} 1, & \text{if a facility is sited at } j, \\ 0, & \text{otherwise.} \end{cases}$$

This formulation adapts the "p-median" format of the Maximal Covering Location problem presented in Church and ReVelle [5], to include both facilities capacities and multiple service levels.

The "service level" corresponds to the order in which the facility providing that service is called. The first service level would be provided by the facility which initially receives the call for service (the primary service provider). The second service level is provided by the backup facility, i.e. the facility that is called when the primary provider is unavailable. The facility delivering the kth service level would be contacted if facilities assigned service levels 1 through $k-1$ are all busy. Clearly, the likelihood that a level k provider is called falls off quickly as k increases. The number of service levels desired in any particular problem instance can be set by the decision maker based on the criticality of providing service and the cost of providing that service. The model described here allows the decision maker to experiment with different levels of K and gauge their impact on system performance.

The constant a_i^k represents the demand at point i that is expected to be served by a level k provider. For example, a_i^1 would be the number of service calls met by the primary server and a_i^2 would be the number of calls met by the first backup facility. Thus, the total demand for service at point i would be $\sum_{k \in K} a_i^k$.

Determining the actual number of calls for service at each service level is not possible prior to the location–allocation decision, since facility availability depends on the number of calls the facility is assigned. Nevertheless, there are several alternative approaches to estimating a_i^k's. As suggested by Pirkul and Schilling [22], these values can be approximated with a simulation model, through examination of historical data, or by a worst-case estimate from the decision maker.

The objective function (1) maximizes the expected demand for each service level that is within the relevant worst-case distance from its assigned service provider. The effect of this objective is to maximize the total demand covered, without differential emphasis on whether that service coverage is provided by the primary server or by one of the backup facilities. The formulation could easily be modified, however, to value a given service level over the service levels which back it up. That is, the objective could be rewritten as

$$Z_P = \max \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} g_k c_{ij}^k a_i^k x_{ij}^k,$$

where g_k is the preference weight on service provided by level k. This form would be appropriate, for example, when the maximum service distances S^k increase with k so that the quality of service degrades as k increases. In such a case, g_k would be set to weight lower service levels more than higher ones.

Constraint (2) limits the number of facilities opened to be equal to p. Constraint set (3) stipulates that the demand at i for each service level must be assigned a provider. Thus, while only covered demand is counted in the objective, all demand is provided service – and contributes to some facility's workload.

In constraint set (4), two problem requirements are maintained. First, they ensure that demand is only assigned to a facility that is open. It also forces the different levels of demand at point i to be assigned to different facilities (i.e. x_{ij}^k and $x_{ij}^{k'}$, where $k \neq k'$, cannot both be 1). This requirement for separating the source of service for a given point follows from the recognition that a facility cannot provide backup to itself. If a facility is busy and cannot provide level k service to point i, it cannot be available at the next (backup) service level.

It is important to recognize that a facility services all demand assigned to it, not just demand within a worst-case travel distance. Therefore, constraint (5) establishes a capacity limit on the total demand a facility serves. This approach, while different from several other authors' work on the capacitated maximal covering problem [3,6] is more realistic for service settings where demand cannot be turned away. As discussed in Pirkul and Schilling [24], however, certain pathologies in the optimal demand allocation are likely. These pertain to the quality of service provided to demand that is uncovered. Service delivered to points within the worst-case travel distance is, by definition, satisfactory. There is no consideration given by the objective function to how close service providers are to points which are beyond the coverage distance. In fact, the model may assign demand to quite distant facilities even where closer (yet still not covering) facilities are available. Since the objective function does not consider them, uncovered demand points may be assigned to any available facility without regard to its proximity. There are any number of functions which address the service quality provided by these demand points. In this paper, we adopt the modification used in Pirkul and Schilling [24], where c_{ij}^k is replaced with υ_{ij}^k so that (1) is replaced by

$$Z_P = \max \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \upsilon_{ij}^k a_i^k x_{ij}^k, \tag{8}$$

where

$$\upsilon_{ij}^k = \begin{cases} 1 & \text{if } d_{ij} \le S^k, \\ \alpha \left[1 - \dfrac{(d_{ij} - S^k)}{(\max\limits_{ij} d_{ij} - S^k)} \right] & \text{if } d_{ij} > S^k, \end{cases} \tag{9}$$

and $0 \le \alpha \le 1$, where α is the weight given to uncovered service.

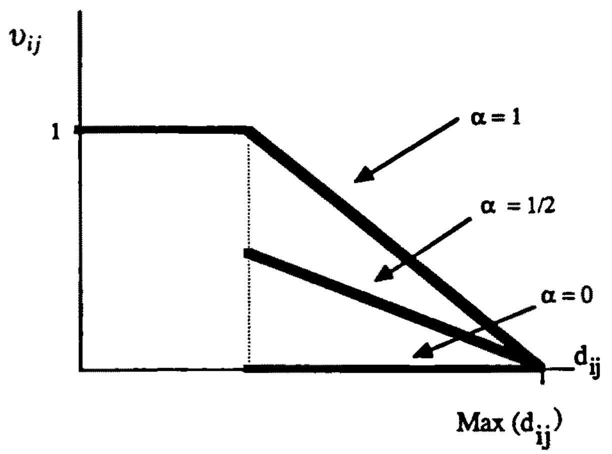

Fig. 1. Service function performance for three levels of α.

This modification serves to minimize the average weighted travel distance to uncovered points while also maximizing the demand which is covered. The effect of this definition of v_{ij} is perhaps more apparent in fig. 1, which shows how service quality changes as the response distance increases. For distances up to S, service quality is assumed unaffected by distance and valued at one. As d_{ij} increases above S units, service quality degrades. The multiplier α allows the decision maker to discount service beyond S even further. Note that when $\alpha = 0$, we have a formulation which places no value on uncovered demand [24]. In the next section, we explore an efficient solution procedure for this model.

3. Problem solution

The model presented in the previous section belongs to the NP-complete class of problems [11]. There are no algorithms specifically designed to solve this problem. Commercial integer programming codes can solve small instances of this model, but cannot handle realistic problem sizes. Development of an efficient solution procedure for this model is critically dependent on being able to exploit its spacial structure. We use a Langrangian relaxation approach to develop a heuristic solution procedure which is both efficient and effective.

The Lagrangian relaxation scheme has been successfully applied to many combinatorial optimization problems during the last decade or so. The use of generalized Langrange multipliers was first suggested by Everett [8]. The successful application of this relaxation to the traveling salesman problem by Held and Karp [15] led to its use in a number of problems. Various location problems [14,19], distributed computer system design problems [12,20], are only a few of the problems to which this relaxation has been applied. For surveys on Lagrangian relaxation, the reader is referred to Fisher [9,10]. We consider the following Lagrangian relaxation of problem **P** formed by multiplying the constraint set (3) by a vector of Lagrange multipliers λ_{ik}, and adding them to the objective function.

Problem L(λ)

$$Z_{\mathrm{L}}(\lambda) = \max \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} v_{ij}^k a_i^k x_{ij}^k + \sum_{i \in I} \sum_{k \in K} \lambda_{ik} \left(\sum_{j \in J} x_{ij}^k - 1 \right) \tag{10}$$

subject to constraints (2), (4), (5), (6), and (7).

Alternatively, problem L(λ) can be stated as the following set of subproblems tied together by constraint set (2):

For $j = 1, \ldots, |J|$,

$$Z_{\mathrm{L}}^j(\lambda) = \max \sum_{i \in I} \sum_{k \in K} (v_{ij}^k a_i^k + \lambda_{ik}) x_{ij}^k \tag{11}$$

$$\text{subject to} \quad \sum_{k \in K} x_{ij}^k \le y_j, \qquad \forall i \in I, \tag{12}$$

$$\sum_{i \in I} \sum_{k \in K} a_i^k x_{ij}^k \le W_j, \tag{13}$$

$$y_j \in \{0, 1\}, \tag{14}$$

$$x_{ij}^k \in \{0, 1\}, \qquad \forall i \in I, \ \forall k \in K. \tag{15}$$

In each subproblem, y_j is equal to either 0 or 1. If y_j is equal to 0, then all x_{ij}^k associated with that facility j are also 0 due to constraint set (12). If y_j is equal to 1, then the subproblem becomes a 0–1 multiple choice knapsack problem. In order to speed up the solution process, we relax the integrality constraint on x_{ij}^k (i.e. we now have $0 \le x_{ij}^k \le 1$), and the problem can now be solved using any one of the linear programming algorithms. This problem exhibits a special structure, however, that makes it possible to develop solution procedures that are significantly faster than simplex-based algorithms. This is the continuous multiple choice knapsack problem and can be solved using a procedure first proposed by Johnson and Padberg [18]. Note that by solving only this continuous version of the subproblem, we may not obtain a bound which is as tight as that obtained by solving the integer version, but we expect the results not to vary significantly since for each subproblem, at most two variables could be fractional (see [18]). Let $\overline{Z}_L(\lambda)$ denote the value of this continuous relaxation of problem $\mathrm{L}(\lambda)$. Note that by solving only the continuous version of the subproblems, we have

$$\overline{Z}_L(\lambda) \ge Z_L(\lambda) \ge Z_L(\lambda^*) \ge Z_P(x^*, y^*),$$

where $Z_L(\lambda^*) = \min_\lambda Z_L(\lambda)$ is the best bound using this relaxation, and $Z_P(x^*, y^*)$ is the optimal solution value to problem **P**. The tentative solution to a subproblem is obtained by setting y_j equal to 1 and keeping the solution of the linear program associated with this subproblem if the objective function value of this linear program is positive; otherwise, y_j is set at 0 and the solution of the subproblem is taken to be the zero vector. After all $|J|$ subproblems have been solved, then constraint set (2) is enforced, which restricts the number of facilities open to p. Obviously, those p facilities which correspond to subproblems with the highest objective function values are selected.

Excluding a few special cases, finding optimal multipliers is known to be a very difficult task. In practice a good, but not necessarily optimal, set of multipliers is often located by using either a subgradient optimization method or various multiplier adjustment methods known as ascent (descent) methods [1].

In this study, we use the subgradient optimization algorithm to derive bounds using problem $\mathrm{L}(\lambda)$. The subgradient method is an adaptation of the gradient method in which subgradients replace gradients. Given an initial multiplier vector λ^0, a sequence of multipliers is generated using the following rule:

$$\lambda_{ik}^{m+1} = \lambda_{ik}^{m} - t_m\left(\sum_{j \in J} (x_{ij}^k)^m - 1 \right), \qquad \forall k \in K, \ \forall i \in I,$$

where x^m is an optimal solution to problem $\mathrm{L}(\lambda^m)$ (the Lagrangian problem with multiplier vector (λ^m)), and t_m is a positive scalar step size. It was shown by Poljack [25] that $\limsup Z_{\mathrm{L}}(\lambda^m)$ converges to $Z_{\mathrm{L}}(\lambda^*)$ if $t_m \to 0$ and $\sum_{i=0}^{\infty} t_i \to \infty$. Since in general these conditions are very difficult to satisfy, this method is always used as a heuristic. We use the following step size that has been frequently used in practice:

$$t_m = \delta_m(Z_{\mathrm{L}}(\lambda^m) - Z_f) \Big/ \left(\sum_{i \in I} \sum_{k \in K} \left(\sum_{j \in J} (x_{ij}^k)^m - 1 \right)^2 \right)^{1/2},$$

where Z_f is a feasible solution value of problem **P** and δ_m is a scalar satisfying $0 \le \delta_m \le 2$. This scalar is set equal to 2 at the beginning of the algorithm and is halved whenever the bound does not improve in 20 consecutive iterations. In our implementation, the subgradient algorithm is terminated after 200 iterations or earlier if the gap between the dual (upper) bound and the best primal feasible solution value is within a user-specified tolerance.

A heuristic solution procedure (procedure F) for problem **P** is developed in conjunction with the Lagrangian relaxation presented in this section. This procedure attempts to generate a feasible solution after every iteration of the subgradient optimization algorithm. The best feasible solution is retained when the subgradient algorithm is terminated. We state procedure F.

PROCEDURE F

Step 1: Open the p facilities which were selected in the Lagrangian solution. Repeat steps 2 through 3 for level k coverage iterating from 1 through $|K|$.

Step 2: For level k, determine for every demand point how many facilities are within distance S^k.

Step 3: Attempt to assign those demand points which have only one facility within distance S^k. Assign first the demand with the largest population to its closest facility with sufficient remaining capacity, then assign the demand with second largest population, etc. Then attempt to assign (in decreasing order of population) those demand points which have two facilities within distance S^k by choosing that facility which has greater available capacity. Repeat this assignment procedure for demand points which have three facilities within distance S^k, and so on. It is possible that a demand point

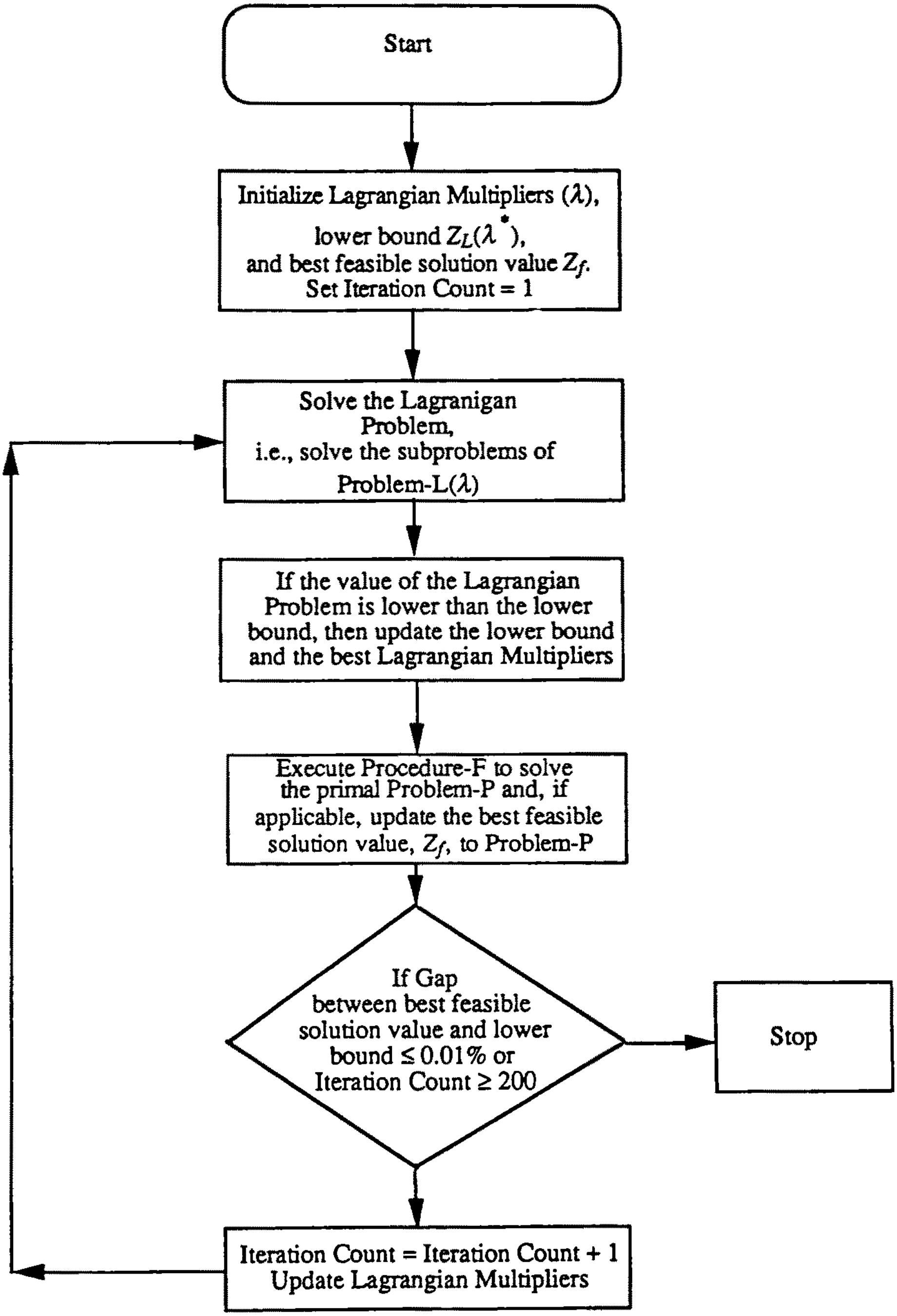

Fig. 2. Block diagram of overall solution procedure.

i may have to be assigned a facility for level k coverage which exists at a distance greater than S^k. For such cases, find the nearest open facility which has adequate capacity. Also note that the different levels of demand at point i are to be assigned to different facilities (i.e. x_{ij}^k and $x_{ij}^{k'}$, where $k \neq k'$, cannot both be 1).

Figure 2 is a block diagram which describes the overall solution procedure and indicates how the various components are tied together.

4. Computational experiments

In order to test the performance of procedure F, a set of computational experiments were performed. The subgradient optimization algorithm incorporating procedure F was coded in FORTRAN and experiments were performed using an IBM 3081-D computer running under MVS/SP 1.3.2.

The data used in these experiments were generated to conform to the primary, secondary, tertiary, etc., service levels of the model. The parameter K was fixed at either 3, 4, or 5. The total capacity requirement for each demand point i was drawn from a uniform distribution between 200 and 500. The capacity requirements a_{i1}, a_{i2}, a_{i3}, a_{i4}, and a_{i5}, were fixed, respectively, at 0.80, 0.10, 0.06, 0.03 and 0.01 times the total requirement. The remaining problem data were randomly generated. Data points representing facility sites and demand points were drawn from a uniform distribution over a rectangle with sides 50 and 100. The Euclidean distance d_{ij} between demand point i and potential facility j was used to define the objective function coefficients as follows. The minimum and maximum distances between any demand point and a facility ($D_{\min}$ and $D_{\max}$) were used to set the S^k parameters, S^k was set equal to $D_{\min} + M^k(D_{\max} - D_{\min})$, where the multipliers M^1, M^2, M^3, M^4, and M^5 were set equal to 0.2, 0.4, 0.6, 0.8, and 1.0, respectively, for the results reported in table 1, and set equal to 0.05, 0.10, 0.15, 0.20, and 0.25, respectively, for the results reported in table 2. Hence, table 2 reports on problems with much smaller coverage distances for all levels of service. The parameter α was set equal to 0.5. The maximum workload capacity of each facility W_j was set equal to $(\sum_{i \in I} \sum_{k \in K} a_{ik} \times 3/2)/p$. Even though we have chosen to generate facilities of equal size, the model is more general and can handle different facility capacities.

The results of the experiments are reported in tables 1 and 2 for problems with up to 200 demand points and 20 potential facility locations. A total of 150 problems were solved in each set. These problems were arranged into groups of 10, where all problems in a group were generated with the same structure in order to achieve a reasonable level of confidence about the performance of the procedure on that problem structure. The gap between the best (primal) feasible solution value and the best upper (Lagrangian) bound is used to judge the quality of the procedure. Note that this gap can never be less than the gap between the best feasible solution value and the (unknown) optimal solution values. The gaps are expressed as a

Table 1

Performance of solution procedure.

Data generation method[b]				Percentage gap[a] between the feasible solution and bound			Mean Computing time (IBM-3081 CPU secs)	Average percentage of demand points within				
M	N	K	p	Min	Mean	Max		S^1	S^2	S^3	S^4	S^5
100	10	3	5	0.59	1.93	2.62	21.1	72.5	97.3	100.0		
100	10	4	5	1.58	2.39	3.00	32.0	63.5	80.4	86.1	99.1	
100	10	5	5	0.41	1.51	2.60	45.5	71.2	96.4	98.1	98.6	100.0
100	20	3	10	0.00	0.00	0.01	21.7	90.7	100.0	100.0		
100	20	4	10	0.00	0.00	0.01	27.7	90.5	100.0	100.0	100.0	
100	20	5	10	0.00	0.07	0.36	49.3	99.2	100.0	100.0	100.0	100.0
200	15	3	10	0.00	0.01	0.04	80.3	83.8	99.6	100.0		
200	15	4	10	0.00	0.10	0.25	90.9	98.7	100.0	100.0	100.0	
200	15	5	10	0.00	0.02	0.04	95.3	98.1	100.0	100.0	100.0	100.0
200	20	3	10	0.00	0.15	0.35	56.9	97.4	100.0	100.0		
200	20	4	10	0.00	0.50	1.35	116.1	99.6	100.0	100.0	100.0	
200	20	5	10	0.02	0.43	2.28	170.2	97.9	98.5	100.0	100.0	100.0
200	30	3	15	0.00	0.01	0.04	90.5	99.4	99.6	100.0		
200	30	4	15	0.00	0.02	0.05	110.4	100.0	100.0	100.0	100.0	
200	30	5	15	0.00	0.01	0.02	182.0	97.7	98.3	100.0	100.0	100.0

[a] Percentage gap = (feasible solution value − upper bound)/upper bound × 100.

[b] M: number of demand points; N: number of potential facility locations; K: number of levels of coverage to be assigned for each demand point location; p: number of facilities to be sited. The multipliers M^1, M^2, M^3, M^4, M^5 were set equal to 0.2, 0.4, 0.6, 0.8, and 1.0, respectively.

percentage of the upper bound. For each group of problems, the minimum, mean, and maximum gap values are reported. The average computing times are also reported. It should be pointed out that these times are the times for the subgradient optimization procedure (since procedure F is an integral part of this procedure) and can be decreased or increased by changing the maximum iteration number permitted in the procedure. As mentioned in section 3, we used 200 iterations as a cutoff point. Only average computation times are reported, since the variance in computing times is not significant. Also mentioned are the percentages of demand points that were completely covered for the various levels.

Table 1 (and similarly table 2) should be interpreted as follows: Studying line 6, we know that the results in this line were obtained by solving 10 problems each with 100 demand points and 20 potential facility sites, with 5 levels of backup required, and 10 facilities to be opened. The minimum percentage gap between the feasible solution and the upper bound was 0% of the bound. The mean and maximum gaps were 0.07% and 0.36%, respectively. On the average, it took 49.3

Table 2

Performance of solution procedure.

Data generation method[b]				Percentage gap[a] between the feasible solution and bound			Mean Computing time (IBM-3081 CPU secs)	Percentage of demand points within				
M	N	K	p	Min	Mean	Max		S^1	S^2	S^3	S^4	S^5
100	10	3	5	2.14	3.36	4.45	22.5	12.6	13.7	13.4		
100	10	4	5	4.32	4.80	5.06	33.9	11.2	10.1	10.0	8.8	
100	10	5	5	3.15	4.74	6.63	46.0	12.0	10.0	12.0	15.6	16.2
100	20	3	10	1.64	3.37	4.70	32.9	17.4	18.0	20.8		
100	20	4	10	1.96	3.06	4.04	48.6	17.0	10.0	9.0	11.2	
100	20	5	10	1.24	2.00	2.75	65.3	19.7	14.3	13.0	10.3	9.7
200	15	3	10	0.89	2.16	2.87	63.0	16.0	5.0	5.5		
200	15	4	10	0.50	1.01	1.53	97.1	18.5	8.0	7.0	7.5	
200	15	5	10	1.12	2.61	5.20	127.6	15.5	15.0	20.5	25.5	21.0
200	20	3	10	0.97	2.54	3.61	84.1	21.0	8.0	9.5		
200	20	4	10	1.25	2.38	5.14	119.0	19.0	13.5	15.0	10.0	
200	20	5	10	0.56	2.27	3.24	170.6	14.5	11.0	9.0	4.5	2.5
200	30	3	15	1.34	2.13	3.21	105.1	24.6	25.5	32.2		
200	30	4	15	2.66	3.20	3.92	153.0	23.2	26.1	31.0	41.3	
200	30	5	15	2.04	3.13	4.02	199.7	25.5	25.0	24.6	33.8	45.2

[a] Percentage gap = (feasible solution value − upper bound)/upper bound × 100.

[b] M: number of demand points; N: number of potential facility locations; K: number of levels of coverage to be assigned for each demand point location; p: number of facilities to be sited. The multipliers M^1, M^2, M^3, M^4, M^5 were set equal to 0.05, 0.1, 0.15, 0.2, and 0.25, respectively.

seconds of CPU time to solve each problem in this set. Finally, on average, 99.2% of the demand points were covered at the primary level of service and 100% of the demand points were covered at the other four levels of service.

Table 2 reports on problems with the much smaller coverage distances. As might be expected, very high coverage at all levels of service is much more difficult to realize in these sets of problems since fewer complete coverage candidate facilities are available for the demand points. Also, the gaps between the feasible solutions and upper bounds, although small, are larger than those reported in table 1, with mean values ranging from 1.01% to 4.80%.

These results indicate that our procedure is effective in solving problems with a wide range of structures, with mean gaps less than or equal to 2.39% for the problems studied in table 1, and less than or equal to 4.80% for those studied in table 2. These are not significant gaps when we consider that the real gaps between the feasible and optimal solution values can indeed be significantly smaller than the gaps reported here.

5. An example

The model and solution procedure developed in this paper were applied to "real world" data obtained from a study one of the authors conducted in a major US city. (The data have been disguised to retain confidentiality.) The example discussed here is intended to serve as an illustration of the applicability of the model to practical-sized problems and not as a solution to any specific problem.

The decision setting for this example is identical to the one studied in Pirkul and Schilling [22]. It concerns identifying locations for fire stations such that all points in the city will receive k levels of coverage. Since only stations are being sited and not the type of equipment that they hold, we are assuming that all of the equipment in a station is dispatched to meet a call for service.

The demand network consists of 625 nodes covering an area of approximately 80 square miles. Thirty of these nodes were selected as potential facility sites. The selections were made such that a uniform distribution of possible locations was obtained. Total demand for the city is 21,000 runs per year. The determination of the level of demand to nodes was accomplished through a survey of actual fire response records. Each fire station was assumed to have a maximum workload of 2400 runs per year.

All demand nodes had their various levels of coverage as defined in the previous section. In table 3, we provide some sample results for a few problems for various values of the level of coverage when the number of facilities p that can be opened was set at 20. This example represents an extremely large mixed integer program with 93,780 variables and more than 20,000 functional constraints for the case where $|K| = 5$. Yet the heuristic was able to find solutions that would have been practically hard to obtain with commercial integer programming codes. Solution times with procedure F, while large, were still reasonable.

Table 3

Solution results for example problem.

Problem size[b]				Percentage	Computing time
M	N	K	p	gap[a]	(IBM-3081 CPU sec)
625	30	2	20	0.23	575
625	30	3	20	0.23	1027
625	30	4	20	0.22	1600
625	30	5	20	0.24	2200

[a] Percentage gap = (feasible solution value – upper bound)/upper bound × 100.
[b] M: number of demand points; N: number of potential facility locations; K: number of levels of coverage to be assigned for each demand point location: p: number of facilities to be sited. The multipliers M^1, M^2, M^3, M^4, M^5 were set equal to 0.2, 0.4, 0.6, 0.8, and 1.0, respectively.

6. Summary

In this paper, we have presented a model for the capacitated facility siting problem with multiple levels of backup coverage. This model differs from many other covering models in that the uncovered demand is forced to be assigned to a facility recognizing that all demand has to be served even if we are not able to meet our covering criteria. Also, the objective function is modified to reflect a measure of distance for those demand points that are assigned to facilities but are not covered due to the fact that the facility is outside the covering distance. This feature of the model prevents anomalies in that those demand points not covered are assigned to the nearest facility rather than to an arbitrary facility, as they would be in the traditional covering models. An effective solution procedure was developed in conjunction with a Lagrangian relaxation of the model and results of computational experiments were discussed. A large-scale example using real-world data was also solved using this solution procedure.

References

[1] M.S. Bazaara and J.J. Goode, A survey of various tactics for generating Lagrangrian multipliers in the context of Lagrangian duality, Eur. J. Oper. Res. 3(1979)322–338.
[2] G.N. Berlin, Facility location and vehicle allocation for provision of emergency service, Ph.D. Dissertation, The Johns Hopkins University (1972).
[3] C.H. Chung, D.A. Schilling and R. Carbone, The capacitated maximal covering problem: A heuristic, in: *Proc. 14th Annual Pittsburgh Conf. on Modeling and Simulation* (1983), pp. 1423–1428.
[4] R.L. Church and C.S. ReVelle, The maximal covering location problem, Papers of the Regional Sci. Assoc. 32(1974)101–118.
[5] R.L. Church and C.S. ReVelle, Theoretical and computational links between the *p*-median, location set-covering, and the maximal covering location problem, Geograph. Anal. 8(1976)406–415.
[6] J. Current and J. Storbeck, Capacitated covering models, Ohio State University Working Paper Series WPS 87-112 (Oct. 1987).
[7] M.S. Daskin and E.N. Stern, A hierarchical objective set covering model for emergency medical service deployment, Transp. Sci. 15(1981)137–152.
[8] H. Everett, Generalized Lagrange multipliers for solving problems of optimum allocation of resources, Oper. Res. 11(1963)399–417.
[9] M.L. Fisher, The Lagrangian relaxation method for solving integer programming problems, Manag. Sci. 27(1981)1–18.
[10] M.L. Fisher, An applications oriented guide to Lagrangian relaxation, Interfaces 15(1985)10–21.
[11] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979).
[12] B. Gavish and H. Pirkul, Computer and database location in distributed computer systems, IEEE Trans. Comput. C-35(1986)583–590.
[13] B. Gavish and H. Pirkul, Distributed computer system design for large decentralized organizations, *Proc. Int. Conf. on Information Systems*, San Diego (1986).
[14] A.M. Geoffrion and R. McBride, Lagrangian relaxation applied to capacitated facility location problems, AIIE Trans. 10(1978)40–47.
[15] M. Held and R.M. Karp, The traveling salesman problem and minimum spanning trees, Oper. Res. 18(1970)1138–1162.

[16] M. Held, P. Wolfe and H.B. Crowder, Validation of subgradient optimization, Math. Progr. 6(1974) 62–88.
[17] K. Hogan and C.S. ReVelle, Concepts and applications of backup coverage, Manag. Sci. 32(1986) 1434–1444.
[18] E.L. Johnson and M.W. Padberg, A note on the knapsack problem with special ordered sets, Oper. Res. Lett. 1(1981)18–22.
[19] R.M. Nauss and R.E. Markland, Theory and applications of an optimization procedure for lock box location analysis, Manag. Sci. 27(1981)855–865.
[20] H. Pirkul, Allocating users and their databases among processing sites of a distributed computer system, Eur. J. Oper. Res. 26(1986)401–411.
[21] H. Pirkul, Efficient algorithms for the capacitated concentrator location problem, Comput. Oper. Res. 14(1987)197–208.
[22] H. Pirkul and D. Schilling, The siting of emergency service facilities with workload capacities and backup services, Manag. Sci. 34(1988)896–908.
[23] H. Pirkul and D. Schilling, The capacitated maximal covering location problem with backup service, Ann. Oper. Res. 18(1989)141–154.
[24] H. Pirkul and D. Schilling, The maximal covering location problem with capacities on total service, Manag. Sci. 37(1991)233–248.
[25] B.T. Poljack, A general method of solving extremum problems, Sov. Math. Dokl. 8 (1967).
[26] T.W. Ruefli and J.E. Storbeck, Behaviorally linked location hierarchies, Environ. Planning B9(1982) 257–268.
[27] J.R. Weaver and R.L. Church, A median location model with nonclosest facility service, Transp. Sci. 19(1985)58–74.