Journal of the Korean
OR/MS Society
Vol. 10, No. 1, Apr. 1985

— Research Paper —

Models Maximizing Covering Reliability

Seong-in Kim*
Young Sin Park*

Abstract

By introducing the concept of reliability of parallel systems the set covering models are modified to accomodate a covering probability and the effects of overlapped covering. A branch-and-bound algorithm is developed and illustrated by a numerical example. The procedure has been coded and its computational efficiency is studied.

1. Introduction

Numerous situations have been modeled as set covering problems. In a discrete facility location context, the problem of determining the number of locations of facilities to cover the set of customers can be formulated as a set covering model with two variations [1]. The total covering problem involves the determination of the minimum number of locations of facilities such that all customers are covered. The partial covering model involves the determination of the locations of a limited number of available facilities such that the maximum number of customers are covered.

Previously, in both models, an individual customer interacts with only one facility. Thus, being covered by multiple facilities is considered identical to being covered by only one. However, it may be the case that increasing benefits are accrued by multiple coverage. In the total covering model each customer is classified into only one of two categories; covered or not covered by a facility. In the generalized partial covering model the coverage may be interpreted as the proportion of covering. Alternatively, situations exist in which this is not the case, a customer's coverage by a facility being probabilistic.

In this paper, the total covering model and partial covering model are modified to incorporate these variations of probabilistic and multiple coverage by introducing the concept of reliability of parallel systems. Section 2 formulates the mathematical models for these variations. The model corresponding to the total covering model can be solved by an integer programming algorithm, whereas the model corresponding to the partial covering model requires a branch-and-bound algorithm which is presented and illustrated by a numerical

* Department of Industrial Engineering, Korea University

example in section 3. Finally, section 4 provides some computational results of the algorithm and discusses its efficiency.

2. Mathematical Formulation of the Problem

The following problem is the typical one which can be solved by the model to be developed. Assume that radars (facilities) are to be installed at a number of possible sites to detect air attacks emanating from any of several points (customers). Radars can be classified into several kinds according to their capacities, efficiencies, prices, etc. The coverage of a point by a radar may be probabilistic. Multiple radars may cover one point simultaneously. Finally, at one site more than one radar may be installed.

The following notations are used :

Constants

m : number of customers

r : number of kinds of facilities

n : number of sites

$w=r\times n$

s : maximum number of facilities to be located

y : maximum budget available

$$a_{ik} = \begin{cases} 1, & \text{if customer } i \text{ is covered by a facility located at site } k, \\ 0, & \text{otherwise,} \end{cases}$$

or measure of covering of customer i by a facility located at site k

p_{ik} : probability that customer i is covered by a facility located at site k

$\overset{*}{p}_{iu}=\max\ (p_{iu}, p_{i,u+1}, \cdots\cdots, p_w)$

p_{ijk} : probability that customer i is covered by a jth-kind facility located at site k

p_i : minimum probability with which customer i should be covered

c_k : cost of assigning a facility to site k

c_{jk} : cost of assigning a jth-kind facility to site k

t_j : maximum number of jth-kind facilities when whole budget is devoted

t : maximum number of wth-kind facilities when remaining budget with current solution is devoted

$\boldsymbol{x}_u$: w-component vector $(x_1, x_2, 0, \cdots, x_u, 0, \cdots\cdots, 0)$ where x_u is the last nonzero component

$\boldsymbol{e}_v$: w-component unit vector whose components are all 0 except for a single 1 in vth position

z_u : upper bound

$\overset{*}{z}_u$: minimum upper bound

z_L : lower bound

Decision variables

$$X_k = \begin{cases} 1, & \text{if a facility is located at site } k \\ 0, & \text{otherwise} \end{cases}$$

X_{jk} : number of jth-kind facilities located at site k

The total covering problem determining the minimum number of facilities required to cover a set of customers is formulated as

T1. minimize $z = \sum_{k=1}^{n} X_k$

subject to

$$\sum_{k=1}^{n} a_{ik} X_k \geq 1, \qquad i = 1, 2, \cdots, m,$$
$$X_k = 0 \text{ or } 1, \qquad k = 1, 2, \cdots, n. \qquad (1)$$

Since a_{ik} in this model has value of 1 or 0 depending on whether customer i is covered or not by any facility located at site k, this model does not consider the situations where the coverages are probabilistic and the facilities are not same in their capacities, efficiencies, prices, etc. Also the value 1 or 0 of X_k, depending on whether a facility is located at site k or not, means that the model does not allow multiple facilities to be installed at one site. These limitations can be solved by introducing the probability, $0 \leq p_{ijk} \leq 1$, which the customer i is covered by jth-kind facility located at site k and decision variable X_{jk} which is the number of jth-kind facilities at site k.

The constraints (1) imply that each of the m customers requires to be covered by at least one facility and that there is no difference between being covered by only one facility and being covered by multiple facilities. In case where these implications are not appropriate, the revised model introduces a measure of coverage. Using the concept of reliability of parallel systems [4], the probability that customer i is covered will be

$$1 - \prod_{j=1}^{r} \prod_{k=1}^{n} (1-p_{ijk})X_{jk}$$

which must be greater than or equal to p_i. Then model T1 can be modified as

T2. minimize $z = \sum_{j=1}^{r} \sum_{k=1}^{n} X_{jk}$

subject to

$$1-\prod_{j=1}^{r} \prod_{k=1}^{n} (1-p_{ijk})X_{jk} \geq p_i, \qquad i = 1, 2, \cdots, m, \qquad (2)$$
$$X_{ijk} : \text{integer}, \quad i = 1, 2, \cdots, m,$$
$$j = 1, 2, \cdots, r,$$
$$k = 1, 2, \cdots, n.$$

The partial covering problem in which s available facilities are assigned to sites in such a way that the maximum number of customers are covered is formulated as

P1. maximize $z = \sum_{i=1}^{m} \max_{j} a_{ij} X_j \qquad (3)$

subject to

$$\sum_{j=1}^{r} X_j \leq s,$$
$$X_j = 0 \text{ or } 1, \qquad j = 1, 2, \cdots, r.$$

This model also has the same limitations that probabilistic and multiple coverage, differences in facilities and multiple installation at one site are not considered.

Introducing the same concepts as in formulation of T2, model P1 can be modified as

P2. maximize $\sum_{i=1}^{m}[1-\prod_{j=1}^{r}\prod_{k=1}^{n}(1-p_{ijk})^{X_{jk}}]$ (4)

subject to

$$\sum_{j=1}^{r}\sum_{k=1}^{n} c_{jk}X_{jk} \leq y, \quad (5)$$

X_{jk} : integer, $j=1,2,\cdots,r,$

$k=1,2,\cdots,n.$

Thus model P2 maximizes the sum of coverage reliabilities within the budget available, y.

It should be noted that T1 and P1 are special cases of T2 and P2, respectively, where $p_{ijk}=p_{ik}=0$ or 1, $p_i=1$ for every i, and $X_{jk}=X_k$ for every j, $c_{jk}=1$ for every j and k.

3. A Branch-and-bound Algorithm

Taking logarithms of both sides of constraints in (2), model T2 becomes an integer program. Thus, many algorithm strategies for integer programming can be applied. For model P1, a heuristic algorithm was developed by Ignizio [2]. The algorithm can be applied to model P2 with alterations due to the difference between objective functions (3) and (4) [3]. However, a branch-and-bound algorithm will be developed.

The model P2 can be modified as following without loss of generality.

P3. minimize $\sum_{i=1}^{m}\prod_{l=1}^{w}(1-p_{il})^{X_l}$

subject to

$$\sum_{l=1}^{w} c_l X_l \leq y,$$

X_l : integer, $l=1,2,\cdots,w,$

where $w=r\times n$ and $c_1\geq c_2\geq\cdots\geq c_w$. Note that the decision variables are arranged in descending order of its corresponding costs.

In order to use the branch-and-bound technique describe the problem as a tree in which each node represents a subset of feasible solutions. A node denoted by a w-component vector, $\boldsymbol{x}_u=(x_1,x_2,\cdots,x_u,0,0,\cdots,0)$, where x_u is the last nonzero component, represents a subset of feasible solutions with $X_1=x_1$, $X_2=x_2,\cdots,X_{u-1}=x_{u-1}$. It should be noted that the vector $\boldsymbol{x}_u$ is itself a feasible solution and so the value of objective function at this solution denoted by z_u is an upper bound for the node. Thus the minimum z_u denoted by $\overset{*}{z}_u$ is an upper bound for the whole problem.

From a node of $\boldsymbol{x}_u=(x_1,x_2,\cdots,x_u,0,0,\cdots,0)$ a node of $\boldsymbol{x}_v$ can be created whose components are equal to $\boldsymbol{x}_u$ except that vth $(v\geq u)$ component is increased by one if $\boldsymbol{x}_v$ can be a feasible solution. That is, a new node can be expressed as

$$\boldsymbol{x}_v = \boldsymbol{x}_u + \boldsymbol{e}_v, \qquad v = u,\ u+1,\ \cdots,\ w,$$

where $\boldsymbol{e}_v$ is a unit vector whose components are all 0 except for a single 1 in the vth position.

To obtain a lower bound for a node of $\boldsymbol{x}_u$, some definitions are needed. Let t be the maximum value of x_w when the remaining budget with the feasible solution $\boldsymbol{x}_u$ is devoted wholly to x_w and $\overset{*}{p}_{iu}$ be the maximum value of probabilities that customer i is covered from jth-kind $(j \geq u)$ facility. That is,

$$t = [(y - \sum_{l=1}^{u} c_l x_l) / c_w]$$

where $[x]$ means largest integer smaller than or equal to x, and

$$\overset{*}{p}_{iu} = \max\ (p_{iu},\ p_{i,u+1},\ \cdots,\ p_{iw}).$$

Since t is the maximum value of $x_u + x_{u+1} + \cdots + x_w$ with the feasible solution $\boldsymbol{x}_u$ and $\overset{*}{p}_{iu} \geq p_{ij}$ $(j = u,\ u+1, \cdots,\ w)$, the value

$$z_L = \sum_{i=1}^{m} \prod_{l=1}^{w} (1-p_{il})^{x_{il}}\ (1-\overset{*}{p}_{iu})^t$$

can be a lower bound. The node having z_L greater than or equal to $\overset{*}{z}_u$ will be fathomed.

Now the algorithm can be stated;

Step 0. Start with a node of null vector representing the set of all feasible solutions with $z_u = \overset{*}{z}_u = m$ and $z_L = 0$.

Step 1. Let the node having least upper bound be $\boldsymbol{x}_u = (x_1,\ x_2,\ \cdots,\ x_u,\ 0, 0, \cdots 0)$. Create the node, $\boldsymbol{x}_v = \boldsymbol{x}_u + \boldsymbol{e}_v$, $(v = u,\ u+1,\ \cdots,\ w)$, if $\boldsymbol{x}_v$ can be a feasible solution.

Step 2. For each node calculate z_u and z_L. Revise $\overset{*}{z}_u$ if necessary. Store corresponding solution as the incumbent solution.

Step 3. Apply fathoming rule to all remaining nodes, i.e., fathom the node if $z_L \geq \overset{*}{z}_u$.

Step 4. If there is no remaining node, let the current incumbent solution as the optimal solution and stop. Otherwise, go to Step 1.

The algorithm is illustrated by the following 4-customer, 3-facility problem.

$$\text{minimize} \sum_{i=1}^{4} \prod_{l=1}^{3} (1-p_{il})^{x_l}$$

subject to

$$6x_1 + 4x_2 + 3x_3 \leq 12$$
$$X_j : \text{integer}, \qquad j = 1, 2, 3,$$

where p_{il}'s are as follows.

	facility j		
customer i	1	2	3
1	.85	.72	.55
2	.92	.65	.68
3	.75	.81	.72
4	.86	.74	.60

At first it is convenient to form $\overset{*}{p}_{iu}$ matrix. It is given in Table 1.

Table 1. $\overset{*}{p}_{iu}$ Matrix

customer i	u = 1	2	3
1	.85	.72	.55
2	.92	.68	.68
3	.81	.81	.72
4	.86	.74	.60

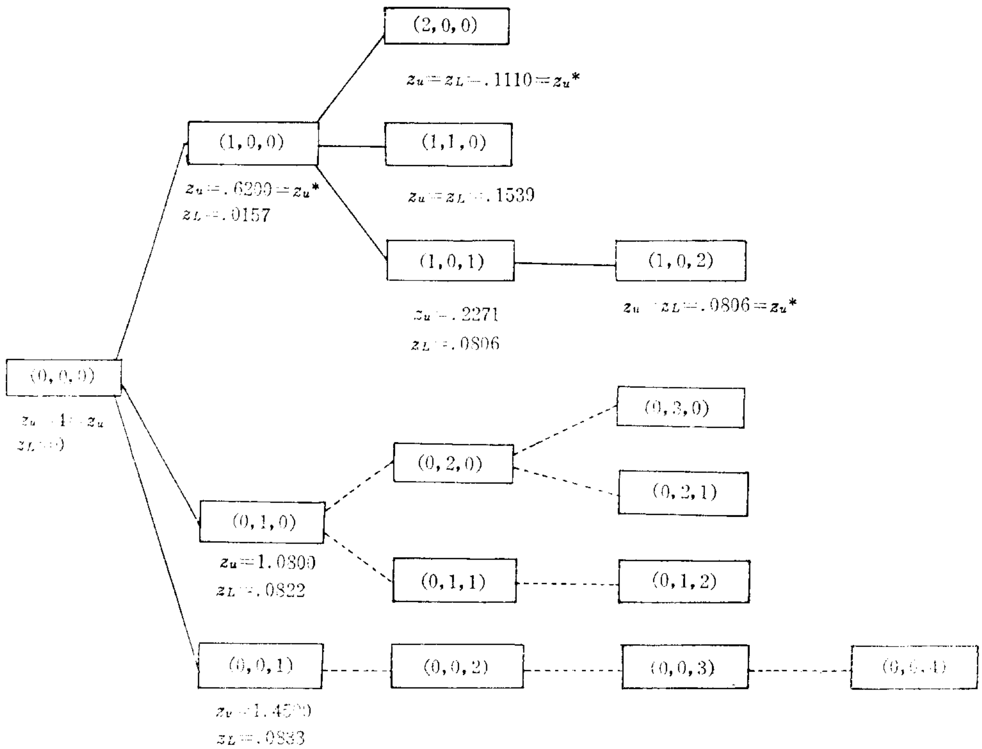

Figure 1. Tree with All Possible Nodes

Figure 1 shows the tree having all of the possible 16 nodes with the corresponding values of z_u, z_L and $\overset{*}{z}_u$. The nodes connected by dotted lines are not actually created. In this example 8 nodes are created in the solving procedure. The upper bound for node (0, 1, 0), as an example, is simply the value of the objective function at this feasible solution given by

$$z = (1-.72)+(1-.65)+(1-.81)+(1-.74) = 1.08.$$

With the remaining budget of $8 = 12-4$, x_3 whose cost is 3 can have the maximum value of

$$t = [8/3] = 2.$$

Thus the lower bound for this node is given by

$$z_L = (1-.72)^3+(1-.65)^2(1-.68)+(1-.81)^3+(1-.74)^3 = 0.0822.$$

The nodes connected by bold lines are those selected from which further nodes will be created. The optimal solution is found as (1, 0, 2) with objective value of 0.0806.

4. Computational Efficiency

To evaluate the efficiency of the algorithm numerical examples were generated in the following way. The number of customers, m, does not have any effect on the number of nodes created and has linear effect on computing time. Throughout the experiment it is fixed as 5. The cost, c_j, is given by

$$1 + (r-j) / (r-1).$$

From this cost formula it is known that the most expensive facility cost, $c_r = 2$, is double of the cheapest facility cost, $c_i = 1$. Then the cost c_{ij} is generated from a random variable uniformly distributed on interval $[c_j - 2(c_j - c_{j-1})/3,\ c_j + 2(c_{j+1} - c_j)/3]$. Note that one fourth of c_j and c_{j+1} can be overlapped. The covering probabilities, p_{ijk}'s, are generated in the same way as c_{ij}'s but they are normalized to be values in $[0,1]$. The budget available, y, is given by the sum of costs of lst-kind facilities, $\sum_{k=1}^{n} c_{1k}$. Thus at least one facility of lst-kind can be assigned to every site.

Table 2. Computational Results

$r \times n$	No. runs	mean No. of possible nodes (1)	No. of nodes created: max	mean(2)	(2)/(1)	min	computing time (second): max	mean	min
2×5	100	$2.89\cdot 10^{4}$	$5.94\ 10^{2}$	$3.19\ 10^{2}$	$(1.10\ 10^{0})$	163.10^{2}	*	*	*
2×7	100	$1.56\cdot 10^{6}$	$5.47\ 10^{3}$	$2.98\ 10^{3}$	$(1.91\ 10^{-1})$	$4.33\ 10^{2}$	4	2	*
2×10	50	$6.67\cdot 10^{8}$	$2.11\ 10^{5}$	$1.27\ 10^{5}$	$(3.16\ 10^{-2})$	$4.04\ 10^{4}$	190	104	28
3×5	100	$2.72\cdot 10^{5}$	$1.85\ 10^{3}$	$9.15\ 10^{2}$	$(3.36\ 10^{-1})$	$4.11\ 10^{2}$	1	*	*
3×7	100	$3.75\cdot 10^{7}$	$2.45\ 10^{4}$	$1.33\ 10^{4}$	$(3.55\ 10^{-2})$	$7.76\ 10^{3}$	17	7	4
3×10	50	$6.55\cdot 10^{10}$	$3.27\ 10^{5}$	$1.61\ 10^{5}$	$(2.46\ 10^{-4})$	$8.23\ 10^{4}$	340	112	37
5×5	100	$5.97\cdot 10^{6}$	$1.36\ 10^{4}$	$5.00\ 10^{3}$	$(8.40\ 10^{-2})$	$1.73\ 10^{3}$	6	2	1
5×7	50	$2.98\cdot 10^{9}$	$4.08\ 10^{5}$	$1.38\ 10^{5}$	$(4.63\ 10^{-3})$	$3.78\ 10^{4}$	348	118	32
7×5	100	$5.14\cdot 10^{7}$	$3.50\ 10^{4}$	$1.77\ 10^{4}$	$(3.44\ 10^{-2})$	$3.18\ 10^{3}$	14	7	1
7×7	50	$6.34\cdot 10^{10}$	$1.26\ 10^{6}$	$5.80\ 10^{5}$	$(9.16\ 10^{-4})$	$1.99\ 10^{5}$	828	381	177

* less than 1 second

The number of feasible solutions having $\sum_{k=1}^{n} X_{jk} = t_j \ (j = 1, 2, \cdots, r)$ is given by

$$\prod_{j=1}^{r} {}_{k+t_j-1}C_{t_j} \tag{6}$$

where ${}_aC_b$ is the combination of a things taken b at a time. From the formula, it can be seen that the number of kinds of facilities, r, and the number of sites, k, as well as budget available, y, have significant effect on the number of nodes created and so computing time. The numerical examples are generated for the combinations of selected values of $r = 2, 3, 5, 7$ and $n = 5, 7, 10, 15$. The computer code has been developed and used to obtain the computational experience in Table 2. The code is written in FORTRAN 77 and has been tested on IBM 4341/M11. The computing times exclude the time to read-in, set-up and printout.

The table shows that the ratio of mean of nodes created to mean number of possible nodes decreases as n increases. From this ratio, computing time and formula (6), it can be inferenced that problems of size up to 10×10, can be solved in 2 or 3 hours.

References

1. Francis, R. L. and White, J. A., *Facility layout and location*, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1974.
2. Ignizio, J. P., *A heuristic solution to generalized covering problems*, unpublished Ph. D. dissertation, Virginia Polytechnic Institute and State University, Blacksburg, VA., 1971.
3. Kim, S. I. and Kim, Y. H., "Set covering models with applications," (in Korean) *The sciences and technologies*, Vol. 24, Nov. 1983.
4. Shooman, M. L., *Probabilistic reliability : An engineering approach*, McGraw-Hill Book Co., Inc., New York, 1968.