Journal of the Korean
OR/ MS Society
Vol. 19, No. 1, April 1994

Optimal Design of a Covering Network†

Young-soo Myung*

Abstract

This paper considers the covering network design problem(CNDP). In the CNDP, an undirected graph is given where nodes correspond to potential facility sites and arcs to potential links connecting facilities. The objective of the CNDP is to identify the least cost connected subgraph whose nodes can cover the given demand points. The problem defines a demand point to be covered if some node in the selected graph is present within an appropriate distance from the demand point. We present an integer programming formulation for the problem and develop a dual-based solution procedure. The computational results for randomly generated test problems are also shown.

1. Introduction

Consider a situation where each demand point(node) can be served only by a facility that exists within some maximal coverage distance, say L, from the demand point. We say that a facility can 'cover' a demand point if that facility is within distance L from the demand point. We also say that a subset of facilities can cover all demand points if each demand point is covered by at least one facility in the set. In the covering network design problem(CNDP), demand points and potential facility sites are given and we are to decide where to locate a subset of facilities that can cover the demand point. Moreover, we must connect the established facilities for mutual interaction. Costs are involved when establishing facilities and links connecting them. Therefore, the purpose of the CNDP is to identify the least cost connected network spanning the established facilities that can cover the demand points.

The CNDP has many real world applications. The CNDP model fits various problems to de-

† This paper was supported by NON DIRECTED RESEARCH FUND, Korea Research Foundation, 1993.
* Department of Business Administration, Dankook University.

termine the location of regional service centers(e.g. public facilities, branches, distribution centers) which should be connected by establishing or building links (e.g. highways, communication links). For example, when a company tries to establish regional marketing centers that control the scattered sales offices and construct a communication network which interconnects the established centers, the company faces a CNDP.

Although the CNDP to our knowledge has not been dealt with in any other research, similar problems have been considered by several researchers. Covering location problems where connectivity condition among established facilities have been considered by Church and Schilling[4], Current and Storbeck[6] and Pirkul and Schilling[12]. Current and Schilling [5] also dealt with a covering location problem where the established facilities should be connected via a ring. Kim and Tcha[9] and Myung et al. [11] also considered a similar network design model with connectivity constraints.

In this paper, we present a dual-based solution approach for the CNDP. The next section considers integer programming formulations of the problem. Section 3 develops a dual-ascent procedure for calculating lower bounds to the optimal objective value and also shows how to construct a feasible solution for the problem. Section 4 gives the computational results for randomly generated test problems.

2. Model Formulations

Let K be the index set of all demand points and V the index set of potential facility sites. For each $k \in K$, we let $V_k \subseteq V$ be a subset of facility sites that exist within distance L from demand point k. We refer to the index of a link between two facilities sited at i and j as $\{i, j\}$ and let E be the index set of all potential links. Consider an undirected graph $G=(V, E)$ where V and E denote the same sets as defined, then the CNDP can also be defined as the problem of finding a subgraph $G'=(V', E')$ of G at minimum cost which satisfies the following conditions :

(i) V' covers K.

(ii) G' is connected.

We use the following notation. Given a graph $G=(V, E)$ and a set S of vertices, $\delta(S)$ represents the set of edges in E with exactly one endpoint in S while $E(S)$ represents the set of edges in E with both endpoints in S. The corresponding notions for a digraph $D=(N,$

A) are as follows. For a set $S \subseteq N$, $\delta^-(S)$ denotes the set of arcs $\{(i,j) \in A : i \notin S,\ j \in S\}$, $\delta^+(S) = \delta^-(N/S)$ and $A(S) = \{(i,j) : i \in S,\ j \in S\}$. For simplicity, we write $\delta^-(i)$ (resp. $\delta^+(i)$ or $\delta(i)$) instead of $\delta^-(\{i\})$ (resp. $\delta^+(\{i\})$ or $\delta(\{i\})$). If x is defined on the elements of a set M (typically M is an edge set E, an arc set A or a vertex set V) then we denote $\sum_{i \in N} x_i$ for $N \subseteq M$ by $x(N)$. The only exceptions are $\delta(\cdot)$, $\delta^-(\cdot)$, $\delta^+(\cdot)$, $E(\cdot)$ and $A(\cdot)$ which were defined previously.

The CNDP can be formulated as the following 0−1 integer programming problem :

$$
\begin{array}{llll}
(P1) & \text{Min} & \sum_{ij \in E} t_{ij} x_{ij} + \sum_{i \in V} F_i y_i & \\
 & s.t. & y(V_k) \geq 1, \quad k \in K & (1) \\
 & & x(\delta(S)) \geq y_i + y_j - 1, \quad i \in S \subset V \text{ and } j \notin S & (2) \\
 & & x_{ij} \leq y_i \text{ and } x_{ij} \leq y_j, \quad \{ij\} \in E & (3) \\
 & & y_i,\ x_{ij} \in \{0,1\}, \quad \{ij\} \in E,\ i \in V & (4)
\end{array}
$$

where

$$x_{ij} = \begin{cases} 1 & \text{if edge } \{ij\} \text{ is included in the selected subgraph} \\ 0 & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if node } i \text{ is included in the selected subgraph} \\ 0 & \text{otherwise} \end{cases}$$

t_{ij} = the nonnegative cost for establishing a link between facilities at site i and j

F_i = the nonnegative cost for establishing a facility at site i

The constraints (2) guarantee the existence of a path between any pair of selected nodes. The same types of constraints as (2) are also considered by Balas [1] for formulating the prize collecting traveling salesman problem. Without the constraints (3), a selected subgraph might contain a node not selected.

The LP relaxation of $(P1)$ is somewhat loose. For example, if all of y_i variables have value less than or equal to $\frac{1}{2}$ in an optimal solution, then the value of each x_{ij} can be 0. So we consider another 0−1 integer programming formulation for the CNDP which leads to a tighter relaxation than $(P1)$. For this purpose, we transform the CNDP into a degree constrained Steiner arborescence problem on a directed graph. Given a directed graph $D=(N,A)$ with $N = \{r\} \cup K \cup V$, the Steiner arborescence problem is the problem of finding a minimum cost tree spanning a given subset $\{r\} \cup K$ of nodes such that there exists a directed path from r, the root vertex, to every member of K.

Given an undirected graph $G=(V,E)$ and the index set of all demand points K, we construct a directed graph $D=(N,A)$ corresponding to G as follows :

(i) $N = \{r\} \cup V \cup K$

(ii) $A = \{(r,i) \mid i \in V_1\} \cup \{(i,k) \mid i \in V_k \text{ and } k \in K\} \cup \{(i,j) \mid i,j \in V \text{ and } i \neq j\}$.

And for each (directed) arc $(i,j) \in A$, we associate cost as follows :

$$c_{ri} = F_i \qquad \text{for all } i \in V_1$$

$$c_{ij} = t_{(ij)} + F_j \qquad \text{for all } i,j \in V;\ i \neq j$$

$$c_{ik} = 0 \qquad \text{for all } i \in V_k,\ k \in K.$$

Then the CNDP on an undirected graph $G=(V,E)$ can be transformed into a Steiner arborescence problem on a directed graph $D=(N,A)$ where the number of arcs incident to the root node should be no more than one. In an arborescence, a unique arc is directed into every node except the root node. Therefore, we can get rid of the node costs by adding F_i to the costs of all arcs entering i. Let T be an obtained arborescence from a transformed degree constrained steiner arborescence problem. From T, remove nodes $\{r\} \cup K$ and the arcs adjacent to them, and refer to T' as the resulting subgraph. Note that at least one node in each V_k is included in T' since T contains a directed path from r to each node $k \in K$. Moreover, T' is connected since T is connected and only one arc is adjacent to the root node in T. Then the underlying graph of T' is an optimal network of the CNDP. In this transformation, arcs (r, i) directing from the root node to each node in V_1 can be replaced by the arcs directing from r to every node of V_k for some $k \in K$.

The constrained Steiner arborescence formulation of the CNDP is as follows :

(P2) Min $\sum_{(i,j) \in A} c_{ij} w_{ij}$

s.t.

$$w(\delta^+(r)) \leq 1, \tag{5}$$

$$f^k(\delta^-(i)) - f^k(\delta^+(i)) = \begin{cases} -1, & i=r \\ 1, & i=k \\ 0, & \text{otherwise} \end{cases} \quad k \in K \tag{6}$$

$$f^k_{ij} \leq w_{ij}, \qquad (i,j) \in A,\ k \in K \tag{7}$$

$$f^k_{ij} \geq 0, \qquad (i,j) \in A,\ k \in K \tag{8}$$

$$w_{ij} \in \{0,\ 1\}, \qquad (i,j) \in A \tag{9}$$

where w_{ij} is a binary variable that indicates whether or not arc (i,j) is selected in the optimal solution and f^k_{ij} represents the flow of commodity k on arc (i,j).

As is usual in network design formulations, the linear programming (LP) relaxation of $(P2)$, gives the better optimal value than that of $(P1)$. Let $(LP1)$ and $(LP2)$ be the LP relaxations of $(P1)$ and $(P2)$, respectively, both of that are obtained by replacing the integrality restrictions on 0−1 variables with nonnegativity constraints. Let $v(\cdot)$ and $F(\cdot)$ denote the optimal objective value and the feasible region of problem $(\cdot)$, respectively, then the following holds :

Proposition 1 $v(LP2) \geq v(LP1)$.

Proof :

Let (f,w) be an optimal solution of $(LP2)$, and also let $x_e = w_{ij} + w_{ji}$ for all $\{i,j\} \in E$ and $y_i = w(\delta^-(i))$ for all $i \in V$. Then we can show that (x,y) is a feasible solution of $(LP1)$ with the same objective function value. For each $k \in K$, by(6) and the fact that $\delta^+(k) = \phi$, $f^k(\delta^-(k)) = 1$. And

$$y(V_k) = \sum_{i \in V_k} w(\delta^-(i)) \overset{(7)}{\geq} \sum_{i \in V_k} f^k(\delta^-(i)) \overset{(6)}{=} f^k(\delta^-(k)) = 1.$$

Therefore, (x,y) satisfies (1).

Now we show (x,y) also satisfies (2) and (3). For this purpose, we need the following fact.

Claim 1 *There always exists an optimal solution (f,w) that satisfies*

$$w(\delta^-(i)) \leq w(\delta^-(S)), \tag{10}$$

for all $i \in S \subseteq V$.

By the max−flow min−cut theorem, the projection of $F(LP2)$ onto the w variables can be expressed as (Maculan [10]) :

$$F_w = \{w : w(\delta^+(r)) \leq 1, \quad w(\delta^-(S)) \geq 1, \quad r \notin S \text{ and } S \cap K \neq \phi, \quad w_a \geq 0 \quad a \in A\}. \tag{11}$$

Let (f,w) be an optimal solution of $(LP2)$ such that w is a minimal member of F_w. There always exists such an optimal solution by the nonnegativity assumption of cost coefficients. Suppose that w violates the inequality (10) for some S and i. Among all such inequalities, choose the one for which $|S|$ is minimal. If $w_a=0$ for all $a\in(\delta^-(i)/\delta^-(S))$ then $w(\delta^-(i))=w(\delta^-(S))$ and this is a contradiction. Let $a=(j,i)\in(\delta^-(i)\setminus(\delta^-(S))$ (i.e. $j\in S$) with $w_a>0$. Since w_a cannot be decreased without violating one of the constraints defining F_w (by the minimality), there exists R with $r\notin R$, $R\cap K\neq\phi$, $a\in\delta^-(R)$ and $w(\delta^-(R))=1$. By submodularity of $w(\delta(.))$, we have $w(\delta^-(S)+w(\delta^-(R)\geq w(\delta^-(S\cup R))+w(\delta^-(S\cap R))$. Since $r\notin(S\cup R)$ and $(S\cup R)\cap K\neq\phi$, (11)says that $w(\delta^-(S\cup R))\geq 1=w(\delta^-(R))$. Therefore, $w(\delta^-(S)\geq 1=w(\delta^-(S\cap R))$. This implies that w also violates (10) for $S\cap R$ and i. Since $j\in S/R$, we have $|S\cap R|<|S|$ and this contradicts the minimality of S.

◆

Using this fact, we can show that for all $i\in S\subset V$ and $j\notin S$, the following holds :

$$\begin{aligned} x(\delta(S)) &= \sum_{h\in S}\sum_{l\in V/S}(w_{hl}+w_{lh}) \\ &= w(\delta^-(S)) - \sum_{l\in S} w_{rl} + w(\delta^-(V/S)) - \sum_{l\in V\setminus S} w_{rl} \\ &\overset{(10)}{\geq} w(\delta^-(i)) - \sum_{l\in S} w_{rl} + w(\delta^-(j)) - \sum_{l\in V\setminus S} w_{rl} \\ &\overset{(5)}{\geq} w(\delta^-(i)) + w(\delta^-(j)) - 1 \\ &= y_i + y_j - 1. \end{aligned}$$

In addition, $x_{ij}=w(\delta^-(i))+w(\delta^-(j))-w(\delta^-(\{i,j\}))$, and by the inequality (10), $w(\delta^-(i))\leq w(\delta^-(\{i,j\}))$ and $w(\delta^-(j))\leq w(\delta^-(\{i,j\}))$. Therefore, (x,y) also satisfies (3). ■

Without the constraint (5), $(P2)$ is the multicommodity flow formulation of the Steiner arborescence problem which was presented by Wong [13]. Using the special structure of the model, he developed a dual ascent procedure which finds dual feasible solutions for the LP relaxation of the problem. The obtained dual solutions provide lower bounds to the optimal solution value. Since $(P2)$ has almost the same structure as Wong's model, we can develop an algorithm for solving $(P2)$ by slightly modifying Wong's dual ascent procedure. The details of our algorithm will be given in the next section.

3. Solution Procedure

In this section, we present a dual ascent method that generates a lower bound on the optimal CNDP value, and also develop a heuristic that constructs a primal feasible solution for the CNDP. Our dual ascent algorithm is based on the degree constrained Steiner arborescence formulation, (*P2*), since its LP relaxation not only provides a tight lower bound but also has a nice structure based on which an efficient dual algorithm can be constructed.

3.1 A Dual Ascent algorithm

Here, we develop a dual ascent method that generates a feasible solution for the dual of the LP relaxation of (*P2*). The obtained dual feasible solutions provide the lower bounds for the CNDP. Several researchers [2,3,7,13] have proven that dual ascent algorithms are efficient for solving network design related models due to their special structures.

Consider the dual of the LP relaxation of (*P2*).

(*D*) Max $\sum_{k \in K} v_k^k - \gamma$

s.t. $v_j^k - v_i^k \leq u_{ij}^k, \quad (i,j) \in A,\ k \in K \qquad (12)$

$\sum_{k \in K} u_{ij}^k - I((i,j))\gamma \leq c_{ij}, \quad (i,j) \in A \qquad (13)$

$u_{ij}^k \geq 0, \quad (i,j) \in A,\ k \in K. \qquad (14)$

In this formulation, the dual variables, γ, v_i^k, and u_{ij}^k, respectively correspond to the constraints (5), (6), and (7). And $I((i,j))$ is an indicating function of an arc which is set equal to 1 if $i=r$ and 0, otherwise. In other words, $I(\cdot)$ shows whether an arc is adjacent from the root vertex r.

If γ is given for (*D*), the resulting problem becomes the dual of the LP relaxations for the Steiner arborescence problem which was dealt with by Wong [13]. Thus we can construct the v_i^k values by directly using his dual ascent algorithm. His procedure, which we call DAP1, is to increase v_k^k for some $k \in K$ while maintaining the feasibility. Here, we briefly present Wong's dual ascent algorithm. For the given values of the dual variables, let s_{ij} denote the slack in constraints(13) and Z_D be the dual objective value. And also let $A^* = \{(i,j) \in A \mid s_{ij} = 0\}$ and $G^* = (V, A^*)$. For each $k \in K$, we define $N(k)$ as the set of nodes that are connected to node k in G^*. $N(k)$ also contains k. For later use, we assume that the increase of v_k^k is restricted to a subset of demand points $K' \subseteq K$.

algorithm DAP1 ;

begin

while there exists $k \in K^+$ such that $r \notin N(k)$ do

begin

$d_i := \min\{s_{ij} | (ij) \in A, j \in N(k)\}$ for all $i \notin N(k)$;

$\Delta := \min\{d_i | i \notin N(k)\}$;

$i^* := \text{argmin } d_i$;

$s_{ij} := s_{ij} - \Delta$ for all $(ij) \in \delta^-(N(k))$;

$Z_D := Z_D + \Delta$;

$N(k) := N(k) \cup \{i^*\}$;

end ;

end ;

We initially set γ and v_k^k for all $k \in K$ equal to zero and perform the DAP1 with $K^+ = K$. After DAP1 terminates, we consider the increase of γ. The increase of γ decreases the dual objective value but creates slack s_{ri} for all $i \in \delta^+(r)$. So our strategy is then to increase γ only when the unit increase of such a variable causes a unit increase in at least one v_k^k for $k \in K$. Let $V^*_1 = \{i \in V_1 | s_{ri} = 0\}$ and $V^*_1(k) = V^*_1 \cap N(k)$ for all $k \in K$. Note that $V^*_1(k)$ is not empty for any $k \in K$ after completing the DAP1. Suppose that we have two distinct elements k_1 and k_2 of K. If $V^*_1(k_1) \cap V^*_1(k_2) = \phi$, then the unit increase of γ and the resulting unit increases of s_{ri} for all $i \in \delta^+(r)$ enable v_{k_1} and v_{k_2} to increase by a unit. Our algorithm selects such pair of elements in K, if any, and increases the corresponding dual variables. As a result, the dual objective value also increases. This procedure can be specified as follows :

algorithm DAP2 ;

begin

while there exists a pair of elements k_1 and k_2 in K such that

$V^*_1(k_1) \cap V^*_1(k_2) = \phi$ do

begin

$\Delta_1 := \min\{s_{ij} | (ij) \in \delta^-(N(k_1)) / \delta^+(r)\}$;

$\Delta_2 := \min\{s_{ij} | (ij) \in \delta^-(N(k_2)) / \delta^+(r)\}$;

$\Delta := \min(\Delta_1, \Delta_2)$;

$\gamma := \gamma + \Delta$;

$s_{ri} := s_{ri} + \Delta$ for all $i \in V_I$;

$Z_D := Z_D - \Delta$;

Update $N(k)$;

$K^+ = \{k_1, k_2\}$ and execute the DAP1 ;

$K^+ = K$ and execute the DAP1 ;

end ;

end ;

3.2 Heuristic Procedure

Here we develop a heuristic algorithm which finds a primal feasible solution for the CNDP. Our procedure is a basically add-drop heuristic. We select a node one by one under a certain sequence until the selected subset of nodes covers K. The sequence of selecting nodes are determined as follows : let V^* be a set of nodes selected so far, then for each node $i \notin V^*$, we calculate the two values, MC_i and MK_i. MC_i is the additional cost needed when establishing facility i in addition to V^*. This cost includes the cost of establishing a link to connect i to some node in V^* and the fixed cost to establish a facility at site i. MK_i is the number of customers which i can cover but no element of V^* can do. Then our procedure selects a node with the minimum MC_i/MK_i ratio and includes it into V^* until V^* covers K.

Then by constructing a minimum cost spanning tree with node set V^*, we can obtain a primal feasible solution and an upper bound for the optimal value. Moreover, the solution might be improved by deleting some element of V^*, so our procedure checks whether the deletion of some node in V^* decreases the cost and if so, we remove it from V^* as far as the remaining node set covers K.

4. Computational Results

Our solution procedure was coded in FORTRAN Ⅳ and implemented on an IBM 386DX (20MHz) personal computer. Test runs were made on a series of randomly generated problems. The underlying graph of each test problem is a complete graph. Note that network design problems on a complete graph are most difficult to solve. For each $k \in K$, the elements of V_k are randomly selected but the average $|V_k|/|V|$ ratio is specified before. We test the problems with different $|V_k|/|V|$ ratios.

〈Table 1〉 shows the computational results for each test problem with the different combination of K, V, and $|V_k|/|V|$ ratio. In 〈Table 1〉, the fourth column in each table shows the average lower bounds as a percentage of the upper bound. Our algorithm solved the large scale problems within the reasonable time and the quality of the obtained lower and upper bounds are also satisfiable.

〈Table 1〉 Results for Test Problems

Problem Size				
|V|	|K|	$\|V_k\|/\|V\|$(%)	LB/OPT(%)	CPU(sec)
5	5	0.2	100.0	0.05
	5	0.5	100.0	0.05
	5	0.7	100.0	0.06
	8	0.2	100.0	0.11
	8	0.5	100.0	0.06
	8	0.7	100.0	0.17
	10	0.2	100.0	0.11
	10	0.5	100.0	0.05
	10	0.7	100.0	0.10
10	5	0.2	100.0	0.06
	5	0.5	100.0	0.05
	5	0.7	100.0	0.06
	8	0.2	100.0	0.11
	8	0.5	100.0	0.05
	8	0.7	100.0	0.11
	10	0.2	100.0	0.11
	10	0.5	88.6	0.22
	10	0.7	100.0	0.27
20	10	0.2	95.5	0.33
	10	0.5	100.0	0.16
	10	0.7	100.0	0.22
	15	0.2	100.0	1.10
	15	0.5	91.7	0.99
	15	0.7	94.1	0.71
	20	0.2	92.7	2.85
	20	0.5	89.5	1.48
	20	0.7	100.0	0.44
30	10	0.2	96.3	0.55
	10	0.5	79.6	0.33
	10	0.7	95.8	0.33
	15	0.2	97.4	1.15
	15	0.5	86.0	1.43
	15	0.7	92.0	1.37
	30	0.2	91.7	4.78
	30	0.5	82.4	5.99
	30	0.7	100.0	1.65

[1] Balas, E., "The prize collecting traveling salesman problem", *Networks 19*(1989) 621－636.

[2] Balakrishnan, A., T. L. Magnanti and R.T. Wong, "A dual－ascent procedure for large－scale uncapacitated network design", *Operations Research 37*(1989) 716－740.

[3] Balakrishnan, A., T. L. Magnanti and P. Mirchandani, "A dual－based algorithm for multi－level network design", Working Paper, Operations Research Center, Massachusetts Institute of Technology, Cambridge. (1991)

[4] Church, R.L. and C.S. ReVelle, "The maximal covering location problem", *Papers of the Regional Science Association 32*(1974) 101－118.

[5] Current, J. and D.A. Schilling, "The covering salesman problem", *Transportation Science 23*(1989) 208－213.

[6] Current, J. and J. Storbeck, "Capacitated covering models", *Environment and Planning B15*(1986) 153－164.

[7] Erlenkotter, D., "A dual based procedure for uncapacitated facility location", *Operations Research 26*(1978) 992－1009.

[8] Goemans, M.X. and Y.-s. Myung, "A catalog of Steiner tree formulations", *Networks 23*(1993) 19－28.

[9] Kim, J.-g. and D.-w. Tcha, "Optimal design of a two-level hierarchical network with tree-star configuration", *Computers and Industrial Engineering 22* (1992) 273－281.

[10] Maculan, N., "The Steiner problem in graphs", *Annals of Discrete Mathematics, 31* (1987) 185－212.

[11] Myung, Y.-s., C.-h. Lee and D.-w. Tcha, "On the generalized minimum spanning tree problem", Working Paper (1993).

[12] Pirkul, H. and D.A. Schilling, "The maximal covering location problem with capacities on total workload", *Management Science 37* (1991) 233－248.

[13] Wong, R.T., "A dual ascent approach for Steiner tree problems on a directed graph", *Mathematical Programming 28*(1984) 271－287.