공학 연구논문집 제24권 제1호 pp. 27~46, 울산대학교
Journal of Engineering Research Vol. 24, No. 1, pp. 27~46, 1993.
University of Ulsan.

Two-Level Hierarchical Network Design Problem with Surviability Constraints*

Jae-gyun Kim* and Dong-hyun Lee**

Abstract

This paper deals with the topological design problem of a hierarchical two-level network with two-connected survivability constraints. As a means to widen the real-world applicability over the existing network design studies, a backbone node not opened is allowed to be included in the backbone tree for transhipment purpose. The problem is modelled as a mixed 0-1 linear programming, whose special structure is exploited for the development of a dual-based heuristic procedure. The effectiveness of the solution procedure is well demonstrated by the computaional experiments conducted with a variety of problems ranging up to 50 backbone nodes and 200 demand points.

(Keywords: Survivable Network Design, Mixed 0-1 Integer Programming, Dual-based Heuristic.)

생존제약식을 갖는 2계위 망의 설계

김재균 · 이동현

〈요 약〉

본 논문은 생존제약식을 갖는 2계위 망의 설계 문제를 다루었다. 생존제약식을 갖는 망의

* 산업공학과
** 울산전문대학 공업경영과 강사
* 본 논문은 1992년도 울산대학교 실험연구비의 지원에 의하여 이루어졌음.

설계문제는 기간망의 각 마디에서 중심마디까지의 연결되는 경로가 최소한 2개 이상이 되도록 기간망을 설계하는 것으로 기간망의 가지를 전송용량이 매우 큰 光電線으로 설치할 때 발생되는 문제이다. 본 논문에서는 이 문제를 0-1 정수계획모형으로 모형화 하였다. 또한 모형의 雙對構造의 특수성을 이용하여 이 문제를 매우 효과적으로 풀 수 있는 雙對基盤 探索法(Dual-based Heuristic)을 개발하였고 기간망의 크기가 50개이고 수요마디의 수가 200개 정도까지의 약 100여개의 문제에 대한 컴퓨터 계산결과를 통하여 해법의 효율성을 입증하였다.

1. Introduction

Fiber optic technology is rapidly becoming one of the major components of future communication network. This transmission medium is cost effective, reliable, and provides nearly unlimited capacity. The high capacity of fiber facilities results in much more sparse network designs with larger amount of traffic carried by each link. In such a case, survivability from node and/or link failure has become an important issue. Therefore, the unique characteristics of this technology is needed to new planning methods.

The major issue of the fiber optic network design problem with survivability is how to build cost-effective network that is immune to unusual but catastrophic link failure such as cable cut. The most common means to build a survivable network is to make link diverse path between certain nodes, which provides for protection against any single link failure between these two nodes. Clearly, higher level of redundant connectivity results in greater network survivability and greater overall network cost. Therefore, there exists tradeoffs between survivability and the total building cost. This leads to the problem of designing a minimum-cost network which meets certain required connectivity constraints.

In this paper, we address the survivable network design problem of a two-level hierarchical structure. We focus our interest to the following two points. First, the ring architecture among other network structures with survivability is giving popularity up to date. But the ring systems sacrifice link efficiency in the interest of survivability. Hence, there is open to question whether only the ring topology gives the most cost-effective and efficient structure when the survivable network is planned. Therefore, for constructing a cost-effective survivable network, we propose a method of the total cost minimization subject to the connectivity constraints rather than the confinement of the ring structure. Note that the resulting topology is at least a more cost-effective st-ructure than ring system since the method can generate the single ring according to the given cost structure.

Secondly, we attempt to propose a method which deals with both the design of a backbone network and that of local networks in a unified frame. However, because of the complexity included, it is almost impossible to formulate a generic mathematical programming model for the design of all types of networks. In this paper, to make this attempt feasible, our attention is confined to the design of two-level hierarchical networks with following redundant structure: the embedded backbone network has 'bi-Steiner' structure which is two-connected between the node 1 and all backbone nodes to be opened and lower level networks have a star type. We refer to this problem as two-connected two-level hierarchical network (TTHN) design problem. A typical example is depicted in Figure 1. With such structural redundancy, the network may provide very dependable services to users.

The problem is to design a two-connected hierarchical network at the minimum cost under the assumptions that the location of user nodes and the potential sites of backbone nodes are given, and the capacity of each backbone node is unlimited. Thus the number and location of backbone nodes, the assignment of each user node to an opened backbone node, and the establishment of two-connected backbone network should be determined.

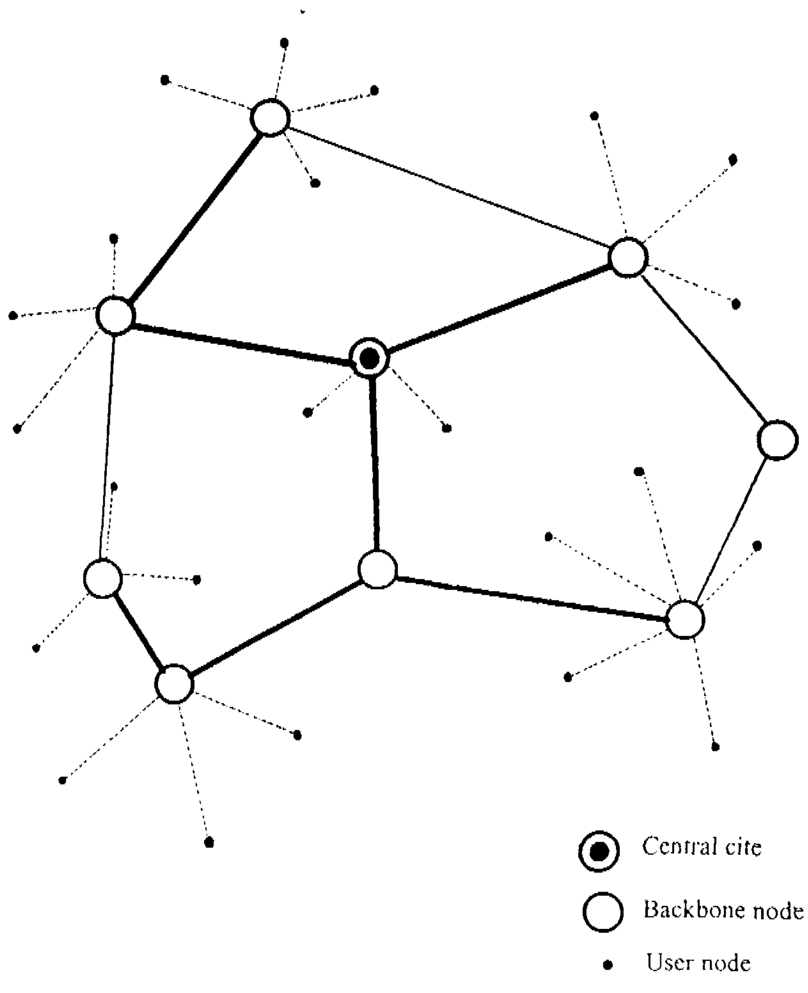


Figure 1. A Centralized Two-Connected Two-Level Hierarchical Network

Section 2 shows the mixed 0-1 linear programming model for the design problem. A solution procedure is presented in Section 3. We first notice that the structure of the linear programming dual of a candidate problem has very special form. The problem can be decomposed into the duals of the uncapacitated facility location problem (UFLP) and the 'bi-Steiner' tree problem. Based on this observation, a dual-based procedure is developed to generate lower bounds and the associated upper bounds. Section 4 is for the computational experiments conducted with number of fairly large test problems. Finally some concluding remarks are given.

2. Model Formulation

Let I be a set of candidate sites for backbone nodes, J a set of user nodes. Node 1 is the special node in I that should be included in the backbone network. Let $K=I\backslash\{1\}$ represent the set of commodities. Also, let $P=\{1,2\}$ denotes the set of path-type indicating primary (p=1) and secondary path (p=2). For each commodity k, we assume that one unit of flow should be shipped from node 1 to node k through each path-types respectively when node k is established as a backbone node. Define E to be the set of undirected links such that $E=\{(i,j)\,|\,i,j\in I,\ i<j,\ i\neq j\}$. It is assumed that each node pairs between node 1 and other nodes $i\in I\backslash\{1\}$ are two-connected.

The design problem can then be formulated as the following mixed 0-1 integer program, denoted by (P):

Min

$$Z_1 = \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{(i,l)\in E} d_{il}y_{il} + \sum_{i\in I} f_i y_i \qquad (1)$$

s.t.

$$\sum_{i\in I} x_{ij} = 1, \qquad \forall j\in J, \qquad (2)$$

$$x_{ij} \le y_i, \qquad \forall i\in I, j\in J, \qquad (3)$$

$$\sum_{l\in I} z_{il}^{pk} - \sum_{l\in I} z_{li}^{pk} = \begin{cases} y_k, & i=1 \\ -y_k, & i=k \\ 0, & \text{otherwise,} \end{cases} \qquad \forall k\in K, p\in P, \qquad (4)$$

$$z_{il}^{1k} + z_{il}^{2k} \le y_{il}, \qquad \forall (i,l)\in E, k\in K, \qquad (5)$$

$$z_{li}^{1k} + z_{li}^{2k} \le y_{il}, \qquad \forall (i,l)\in E, k\in K, \qquad (6)$$

$$z_{il}^{pk} \ge 0, \qquad \forall (i,l)\in E, k\in K, p\in P, \qquad (7)$$

$$x_{ij} \in \{0,1\}, \qquad \forall i\in I, j\in J, \qquad (8)$$

$$y_i \in \{0,1\} \qquad \forall i\in I, \qquad (9)$$

$$y_{il} \in \{0,1\}, \qquad \forall (i,l)\in E, \qquad (10)$$

where

x_{ij} : the binary variable that shows whether ($x_{il} = 1$) or not ($x_{ij} = 0$) user node j is attached to backbone node i,

y_i : the binary variable that denotes whether ($y_i = 1$) or not ($y_i = 0$) backbone node i is open,

y_{il} : the binary variable that indicates whether ($y_{il} = 1$) or not ($y_{il} = 0$) arc (i,l) is installed as an arc of backbone network,

z_{il}^{pk} : the nonnegative variable denoting the amount of flow of commodity k on the undirected arc (i,l), which is included in the path-type p.

In this formulation, three types of nonnegative cost parameters have been defined: f_i for establishing backbone node $i \in I$, d_{il} for installing link $(i,l) \in E$, and c_{ij} for connecting user node $j \in J$ to established backbone node $i \in I$. Furthermore, we have assumed that the capacities of all nodes and links are unlimited.

Constraints (2) and (3) are for design of the lower-level network. Demand of a user node has to be met by open backbone nodes, which in effect enforce a user node to be connected to an open backbone node.

Constraints (4),(5) and (6) are for backbone network design. Constraints (4) dictates that one unit of commodity k be routed from node 1 to node k for each path-type p only when node k is open. The forcing constraint (5) and (6) indicate that flow is allowed only on open arcs. Moreover, if the flow of a path-type for each commodity k has gone through to a direction on an arbitrary edge, the flow of other path-type cannot be gone through to the same direction on the edge. Note that these constraints always ensure that there exists at least two edges on an arbitrary cutset, separating between node 1 and node $k \in K$ to be opened. Thereby two-connectivity constraint from node 1 to all backbone nodes to be opened is obtained from the forcing constraints (5) and (6).

We can further tighten the formulation by replacing the forcing constraints (5) and (6) with the following constraints

$$z_{il}^{1k} + z_{il}^{2k} + z_{li}^{1k} + z_{li}^{2k} \le y_{il} \qquad \forall (i,l) \in E,\ k \in K. \tag{11}$$

which means that if the flow of a path-type for each commodity k is gone through, then the flow of the other path-type cannot be gone through for both directions when an edge has been established. Note that the resulting problem formulation $\widehat{P}$ will not affect the value of the optimal solution of the design model. However the resulting formulation will be at least as tight as the original formulation because there is a solution which is feasible in P, but infeasible in $\widehat{P}$. Consider, for example, the 4-node network shown in Figure 4.2 with d_{12}=10, d_{13}=15, d_{23}=2, d_{24}=10, d_{34}=4. Suppose that the node 4 is open. Then a primary and secondary paths in P_1 are given as 1→2→3→4 and 1→3→2→4 respectively and its total costs is 41, while the paths are given as 1→3→4 and 1→2→4 respectively and its total cost is 39. The edge (2,3) is included in the solution of P_2 but it is not included in tat of P_2. Therefore the feasible solution in P is infeasible in $\widehat{P}$.

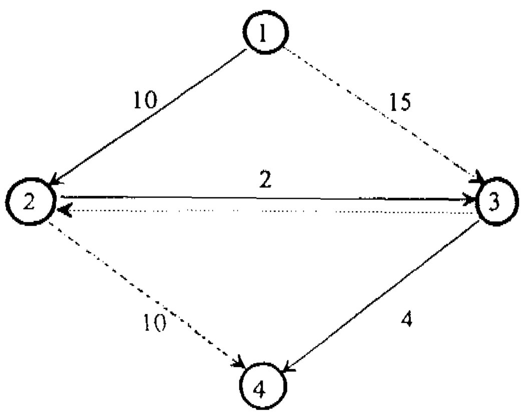
Though the bound can be tightened by using the $\widehat{P}$ instead of the P, the special structure similar to the case THTNTS design problem cannot be found in the linear programming dual of the $\widehat{P}$. This requires a new procedure different from that of paper for finding lower bound. Hence, we will discuss the solution method for the original problem P.

3. Dual-Based Solution Procedure

3.1 Condensed Dual Problem

Consider the LP relaxation of (P) where all the 0-1 variables are relaxed into nonnegative variables. For this relaxation we include the following constraint on each y_{il} variables

$$y_{il} \le 1 \qquad \forall (i, l) \in E. \tag{12}$$



Primal Path : 1→ 2→ 3→ 4

Secondary Path : 1→ 3→ 2→ 4

Total cost = 41

Optimal sol. = 39

Figure 2. Example for the effect of the forcing constraints

Now we have the following dual of the LP relaxation of (P) with (12), denoted by (D):

$$\max \quad Z_2 = \sum_{j\in J} v_j - \sum_{(i,l)\in E} u_{il} \tag{13}$$

$$\text{s.t.} \quad v_j - w_{ij} \le c_{ij}, \qquad \forall i \in I, j \in J, \tag{14}$$

$$\sum_{j\in J} w_{ij} - \sum_{p\in P} h_i^{pi} \le f_i, \qquad \forall i \in I, \tag{15}$$

$$\sum_{k\in K} q_{il}^k + \sum_{k\in K} q_{li}^k - u_{il} \le d_{il}, \qquad \forall (i,l) \in E, \tag{16}$$

$$h_l^{pk} - h_i^{pk} \le q_{il}^k, \qquad \forall k \in K, (i,l) \in E, p \in P, \tag{17}$$

$$h_i^{pk} - h_l^{pk} \le q_{li}^k, \qquad \forall k \in K, (i,l) \in E, p \in P, \tag{18}$$

$$w_{ij},\ q_{il}^k, \text{ and } u_{il} \ge 0,$$

$$v_j,\ h_i^{pk} \quad \text{unrestricted.}$$

Dual variables v_j and w_{ij} correspond to constraints (2) and (3), respectively. h_l^{pk} ($l\in I$ and $k\in K$) is the dual variable associated with the flow conservation equation (4) for commodity k at node l for each path-type p, while q_{il}^k and q_{li}^k for $k\in K$ and $(i, l)\in E$ correspond to the forcing constraints (5) and (6), respectively. The dual variable h_k^{pk} is interpreted as the length of shortest path from the root (node 1) to node k (commodity k) when we use the are length q_{il}^k and the path type p[1]. For each commodity $k\in K$, one of the flow conservation equation (4) is redundant; hence, we have arbitrary set the dual variable h_1^{pk} equal to zero.

The variable q_{il}^k acts as an intermediary which transforms the 'edge resource' associated with the constraint (16) into the 'node resource' associated ith the constraint (15) through the constraints (17) and (18). That is, the d_{il}' s are decreased according to the increase of the q_{il}^k's and q_{li}^k' s, while the q_{il}^k' s are consumed to increase of h_i^{pk}' s. After all, the h_k^{pk} will be increased according to the consumption of d_{il}.

Note that, for any edge (i, l), the variable q_{il}^k' s in the constraint (17) and (18) are commonly used for each path type p. From this fact, we can set the variable h_i^{pk} as follows:

$$h_i^{2k} \equiv h_i^{1k} \equiv h_i^k \qquad \forall i \in I, k \in K,$$

which means that the length of shortest path of the secondary path-type from the node i to the node k can be approximated as the that of the primary path-type. Then, by using term $2h_i^i$ instead of the term $\Sigma_{p\in P}\ h_i^{pi}$, the constraint (18) can be rewritten as follows:

$$\sum_{j\in J} w_{ij} - 2h_i^i \le f_i, \quad \forall i \in I. \tag{19}$$

For any feasible choice of the dual variables v_j and h_i^{pk}, we can set the dual variables, w_{ij} and q_{il}^k, at the following lowest possible value:

$$w_{ij} = \max\{0, v_j - c_{ij}\}, \qquad \forall i \in I, j \in J \tag{20}$$

$$\begin{aligned} q^k_{il} &= \max\{0, h^{1k}_l - h^{1k}_i, \; h^{2k}_l - h^{2k}_i\} \\ &= \max\{0, h^k_l - h^k_i\} \qquad \forall (i, l) \in E, k \in K, \end{aligned} \tag{21}$$

$$\begin{aligned} q^k_{li} &= \max\{0, h^{1k}_i - h^{1k}_l, \; h^{2k}_i - h^{2k}_l\} \\ &= \max\{0, h^k_i - h^k_l\} \qquad \forall (i, l) \in E, k \in K, \end{aligned} \tag{22}$$

by which the feasibility is preserved and the objective value remains unchanged.

Substituting (20),(21), and (22) for (14) and (19) respectively, we have the condensed expression of the dual problem. The condensed dual form, denoted by CDP, is described as follows:

(CDP)

$$\max \quad \overline{Z}_2 = \sum_{j \in J} v_j - \sum_{(i,l) \in E} u_{il}$$

s.t.

$$\sum_{j \in J} \max\{0, v_j - c_{ij}\} - 2h^i_i \le f_i, \quad \forall i \in I, \tag{23}$$

$$\sum_{k \in K} \max\{0, h^k_l - h^k_i\} + \sum_{k \in K} \max\{0, h^k_i - h^k_l\} - u_{il} \le d_{il}, \quad \forall (i, l) \in E. \tag{24}$$

Note that the dual problem *CDP* has the following structural properties: (a) the *CDP* has nothing to do with the path type and includes only three types of variables v_j, u_{il} and h^k_i which have an interrelation each other. (b) The variables v_j appear only in the constraint (23), suggesting that v_j are desired to be set as high as possible. But the increase of a v_j is restricted by other v_j' s competing for the common resource associated with the constraint (24) via the change of h^k_i . (c) The increase of the u_{il}' s indirectly affects to the increase of the v_j' s via the change of h^k_i . The increase of a unit of u_{il} yields two units of increase of h^k_i on the constraint (23), but directly decreases in the dual objective value $\overline{Z}_2$ like the same amounts.

Hovever, we can manipulate the variable u_{il} such that the $\overline{Z}_2$ preserves a monotonically increasing property.

From the constraints (23) and (24), we have defined the following two slack variables, one for a node i and the other for an arc (i, l).

$$s_i = f_i + 2h_i^{pi} - \sum_{j \in J} \max\{0, v_j - c_{ij}\} \quad \forall i \in I, \tag{25}$$

$$S_{il} = d_{il} + u_{il} - \sum_{k \in K} \max\{0, h_l^k - h_i^k\} - \sum_{k \in K} \max\{0, h_i^k - h_l^k\} \quad \forall (i, l) \in E. \tag{26}$$

The above slack variables s_i and S_{il} will be hereafter referred to as a node slack and an edge slack respectively. The dual feasibility is then met when all node and edge slack variables become nonnegative.

3.2 The Decomposition of the Condensed Dual

Our immediate goal is now to increase the $\overline{Z}_2$ of the condensed dual CDP as fast as possible. The goal may then be achieved by considering the following two subproblems: one is the multiobjective problem MOP for maximizing h_i^i under a given set of u_{il}' s and the other, referred to as SDP, for maximizing $\Sigma_{j \in J} v_j$ for a given set of u_{il}' s and h_i^i' s. A formal statement of the two subproblems is given as follows:

(*MOP*)

$$\max \; H = (h_1^1, h_2^2, \ldots, h_m^m)$$

$$\text{s.t.} \; \sum_{k \in K} \max\{0, h_l^k - h_i^k\} + \sum_{k \in K} \max\{0, h_i^k - h_l^k\} - u_{il} \le d_{il}, \quad \forall (i, l) \in E. \tag{27}$$

(*SDP*)

$$\max \; \overline{Z}_2 = \sum_{j \in J} v_j - \sum_{(i,l) \in E} u_{il}$$

$$\text{s.t.} \; \sum_{j \in J} \max\{0, v_j - c_{ij}\} - 2h_i^i \le f_i, \quad \forall i \in I. \tag{28}$$

Note that, once given a set of $\{u_{il}\}$, the above two subproblems have the same structure as that of the single onnection problem in paper. That is, once given a solution of MOP, $\{h_i^k\}$ and $\{u_{il}\}$, SDP is none other than the dual of the UFLP for which Erlenkotter' s dual ascent process can be directly applied[7]. On the other hand, MOP is a multiobjective version of the dual of Wong' s Steiner tree problem[17].

3.3 Dual Ascent Procedure

As we have seen in the described subsection, the special structure of CDP suggests an iterative dual ascent method whose efficiency depends on how effectively the updating of the dual variables, v_j' s, u_{il}' s, and h_i^k' s, is done at each iteration. For that, we note that the dual objective function consists of two kinds of terms, one for v_j' s and the other for u_{il}' s, although these two variables are interrelated with the other dual variable h_i^k' s. This observation renders the two types of dual ascent procedures: one based on the increase of v_j and the other that of u_{il}. We shall refer to them as Procedure v-ascent and Procedure u-ascent respectively. With all dual variables being initialized at zero as in the conventional practice, two such dual ascent procedures are skeletonized:

Procedure v-ascent

Procedure v-ascent is a procedure that increases the dual objective value by manipulating $\{v_j\}$ and $\{h_i^k\}$ for a given $\{u_{il}\}$. To increase rapidly the dual objective value, we utilize the special structure of CDP, which is described as the dual structure of UFLP embedded in STP. Therefore the procedure is mostly taken from the Erlenkotter s method for UFLP and the Wong s method for STP.

The outline of the Procedure v-ascent is now presented: (a) With the initial values of $\{h_i^k\}$=0 and $\{u_{il}\}$=0, apply Erlenkotter' s dual ascent process for the corresponding SDP. (b) On finding that an node slack s_i becomes zero during the step-by-step increase of $\{v_j\}$, the process of increasing h_i^{pi} to find the next solution of MOP is activated. For this, we use the Wong s method for STP with slightly modifications. This makes the node slack s_i larger instead of consuming arc slack S_{il}. (c) Appl· Erlenkotter' s process of incrementally increasing v_j' s for the SDP with the solution of the MOP at hand. (d) Constinue this alternate application between Erlenkotter' s and the Wong' s process until $\{v_j\}$ is not increased any further, i.e., up to the point at which an efficient solution of MOP is attained.

Note that the MOP has two differences from the dual of STP that Wong has dealt with. First, the goal of MOP is to find an efficient solution $(h_1^1, h_2^2, \ldots, h_k^k)$ which maximizes $\Sigma_{j=1}^n v_j$ (the objective of CDP), while the one for Wong' s STP is to maximize $\Sigma_{k=1}^m h_k^k$. Secondly, the MOP has considered an undirected problem while the STP dealt by Wong has only considered directed problem. Still, the Wong' s ascent strategy is adopted with few modifications, acknowledging that effective redistribution over v_j' s of incremental increase in the node slacks by Erlenkotter' s method somehow guarantees that the larger the total amount of increase in node slacks is, the larger the contribution to $\Sigma_{j=1}^n v_j$ is. One refinement on the Wong' s method to note is that an incremental node slack increase is made on the node at the epoch when its node slack is dropped to zero during Erlenkotter' s ascent process.

As usual in dual ascent procedure, we assume that for each j in J, all c_{ij} are ordered in nondecreasing order $c_j^{i'}$, $i'=1,2,\ldots,m$, and set $c_j^{m+1}=+\infty$. For the remainder of this paper, the notation and the terminology by Erlenkotter and Wong will be used unless otherwise specified. For details not listed here, refer to [7,17]. As notations concerning the increase of h_k^k, $C(k)$ denotes the set of nodes which are connected to the node k through zero-slack arcs. $E(k)$ is the associated cutset separating $C(k)$ and the remaining nodes in I, and E^* is the set of zero-slack arcs. This procedure can be described formally as follows:

Step 0. (Initialize)

(a) For each $j \in J$, all c_{ij} are ordered in nondecreasing order as $c_j^{i'}$, $i'=1,2,\ldots,m$, and set $c_j^{m+1}=+\infty$.

(b) Start with a feasible dual solution $\{v_j, h_i^k\}$ such that $v_j \geq c_j^1$ for each $j \in J$ and $h_i^k=0$ $\forall i \in I$, and $k \in K$. For each $j \in J$, define $i'(j)=\min\{i' : v_j \leq c_j^{i'}\}$. If $v_j=c_j^{i'(j)}$, increase $i'(j)$ by 1.

(c) Calculate the slacks s_i and S_{il} for constraints (25) and (26) respectively.

(d) Set $C(k)=\{k\}$ $\forall k \in K$, $E^*=\emptyset$, $F=J$, $V=0$, and $\delta=0$.

Step 1. (*Evaluate the possible amount of v_j inrease*)

(a) Select a user node $j \in F$.

(b) Define $s_{k'(j)}=\min\{s_i | i \in I, v_j - c_{ij} \geq 0\}$

(c) Set $\triangle_j=s_{k'(j)}$ and if $\triangle_j < c_j^{i'(j)} - v_j$, go to Step 4.

Step 2. (*Increase v_j*)

(a) If $\triangle_j > c_j^{i'(j)} - v_j$, set $\triangle_j=c_j^{i'(j)} - v_j$ and $\delta=1$, and increase $i'(j)$ by 1

(b) Decrease s_i by $\triangle_j$ for each $i \in I$ with $v_j - c_{ij} \geq 0$. Update $v_j=v_j + \triangle_j$ and $V=V+1$.

Step 3. (*Terminate*)

(a) Set $F=F - \{j\}$ and if $F \neq \emptyset$, go to Step 1.

(b) If $\delta=1$, reset $F=J$ and $\delta=0$, and go to Step 1; otherwise stop.

Step 4. (*Increase $\{h_i^k\}$*)

(a) Set $k=k'(j)$ and $\alpha=0$ and if $\{1\} \in C(k)$, go to Step 2.

(b) Calculate $S_{i^*l^*}=\min\{S_{il} | (i, l) \in E(k), l \in C(k), i \in I \setminus C(k)\}$ and set $\alpha=S_{i^*l^*}$.

(c) Adjust $h_l^k=h_l^k + \alpha$ for each $\{l\} \in C(k)$, $S_{il} = S_{il} - \alpha$ for each $(i, l) \in E(k)$, and $s_k=s_k + 2\alpha e$.

(d) Update $E^*=E^* \cup \{(i^*, l^*)\}$ and $C(k)=C(k) \cup \{i^*\}$

(e) If $\alpha=0$, repeat Step 4; otherwise go to Step 1-b.

Recall that, as we have noted ahead, the final value of h_k^k represents the shortest path distance from node 1 to node k using the arc lengths q_{il}^k. It would be instructive to see how the values of node and arc slacks change. As our procedure progresses, arc slack S_{il} monotonically nonincreases from its initial value of d_{il}. On the other hand, node slack s_i nonincreases from its initial value of f_i, but once it becomes zero, it immediately jumps upward to α as in Step 4-c, and then nonincreases again by the activation of Erlenkotter s ascent process. Another point to note is that using zero slack arcs in E^*, two paths are found from node 1 to node k for which the condition in Step 4-a is met, which will be utilized to construct a primal feasible solution.

Procedure *u*-ascent

The dual objective value $\overline{Z_2}$ of the CDP can be further increased from the current value by the increase of node slacks s_i, whose nodes block the increase of v_j owing to the lack of its slack. The Procedure u-ascent is a procedure that adjusts the dual variables $\{u_{il}, h_i^k, s_i^k\}$ to increase the dual objective value Z_2. To increase the dual objective value by repeatedly updating $\{u_{il}\}$, we note that the dual objective value may be improved only when u_{il}' s are increased on those edges with zero slacks, i.e., S_{il}=0. However, because of the penalty term $\Sigma_{(i,l)\in E}\, u_{il}$ imposed on the Z_2, whether or not the resulting $\overline{Z_2}$ which is accompanied by the increase of u_{il} can be still increased monotonically should be carefully investigated prior to the increase of u_{il}. The ascent operation to increase the u_{il}' s must be performed only whenever the resulting $\overline{Z_2}$ preserves the monotonically increasing property.

Thus we consider the following simple but effective scheme for adjusting $\{u_{il}\}$: Beginning a user node $j\in J$, identify the backbone nodes $i\in I$ to block the increase of v_j owing to zero slack (s_i=0). If the possible increase of v_j is higher than the penalty cost associated with the increase of u_{il}, increase u_{il} focusing on the increase of associated s_i' s only. With this increased u_{il}, apply the labelingascent method of Procedure v-ascent to further increase the associated v_j. This ascent operation is successively applied from one user node to another. This series of ascent operations is continued insofar as the resulting dual objective is not decreased. This procedure can be described formally as follows:

Step 0. (*Initialize*)

Reset $C(k) = \{k\}\ \forall k\in K$, $F = J$, $F = \emptyset$, $\overline{E} = 0$, and $\delta = 0$.

Step 1. (*Node selection*)

(a) Select a user node $j\in J$.

(b) Define $\overline{F} = \{i \mid i\in I,\ s_i = 0, \text{ and } v_j - c_{ij}\geq 0\}$.

(c) If $|\overline{F}|\geq 2$, then go to Step 6.

(d) Define k be the remaining node i of F.

Step 2. (*Edge Selection*)

(a) If $\{1\} \in C(k)$, then go to Step 6.

(b) Define $E_k = \{(i, l) \mid (i, l) \subset E(k)$, such that $l \in C(k)$, $i \in I \setminus C(k)$ and $S_{il} = 0\}$.

(c) If $E \langle 2$, then go to Step 3.

(d) Update $C(k) = C(k) \cup \{i \mid (i, l) \in \bar{E}_k\}$ and go to Step 2-a.

Step 3. (*Evaluate the possible increase* $\{u_{il}\}$)

(a) Define (i^*, l^*) be the remaining edge (i, l) of $\bar{E}_k$.

(b) Calculate $\beta = \min\{S_{il} \mid (i, l) \subset E(k) \setminus \{(i^*, l^*)\}, l^* \in C(k), i^* \in I \setminus\ C(k)\}$. and $\triangle_j = c_j^{i'(j)} - v_j$.

(c) Define $(i', l') = \{(i, l) \mid (i, l) \subset E(k), S_{il} = \beta\}$.

(d) Set $\alpha = \min\{\beta, \triangle_j/2\}$.

Step 4. (*increase* $\{u_{il}\}$)

(a) Update $u_{i^*l^*} = u_{i^*l^*} + \alpha$.

(b) Adjust $h_l^k = h_l^k + \alpha$ for each $\{l\} \subset C(k)$ and $S_{il} = S_{il} - \alpha$ for each $(i, l) \in E(k) \setminus \{(i^*, l^*)\}$.

(c) Update $E^* = E^* \cup \{(i^*, l^*), : (i', l')\}$ $C(k) = C(k) \cup \{i^*, i'\}$.

(d) Update $s_k = s_k + 2\alpha$ and $V = V - \alpha$.

Step 5. (*Increase* $\{v_j\}$)

(a) Define $s_{k'(j)} = \min\{s_i \mid i \in I, v_j - c_{ij} \geq 0\}$.

(b) If $s_{k'(j)} \rangle \triangle_j$, then set $s_{k'(j)} = \triangle_j$ increase $i'(j)$ by 1.

(c) Decrease s_i by $s_{k'(j)}$ for each $i \in I$ with $v_j - c_{ij} \geq 0$.

(d) Reset $\delta = 1$, and update $v_j = v_j + s_{k'(j)}$ and $V = V + s_{k'(j)}$.

Step 6. (*Termination*)

(a) Update $F = F - \{j\}$.

(b) If $F \neq 0$, then go to Step 1.

(c) If $\delta = 1$, then reset $F = J$ and go to Step 1; otherwise stop.

As we have seen from the constraints (27) and (28), the node slack s_i can be increased two times as much as the increase of u_{il}' s. Instead the increase of u_{il}' s directly decreases the dual objective value $\overline{Z_2}$ by the same amount as that of u_{il}' s because of the penalty term $\Sigma_{(i,l) \subset E} u_{il}$ on the $\overline{Z_2}$. Then the increased dual objective value for each increase of u_{il} is $v_j - \Sigma_{(i,l) \in E} u_{il}$. Therefore u-ascent operation must be performed to the edge such that the term $v_j - \Sigma_{(i,l) \in E} u_{il}$ becomes positive. Note that the term $v_j - \Sigma_{(i,l) \subset E} u_{il}$ becomes positive only when the following two conditions must be satisfied: (1) the number of nodes with zero slack concerning to each increase of v_j cannot be greater than 2. (2) The number of edges with zero slack among the edges in the cutset $E(k)$ associated with the node k which is intended to increase its slack cannot be greater than 2. This condition is utilized to select an edge to increase u_{il} in the Procedure u-ascent. Step 1 and 2 of the Procedure u correspond to the condition (1) and (2) respectively.

Note that the Procedure u-ascent ensure that two zero-slack edges are added to the E^*, which is the set of zero-slack edges. Hence we can find two types of paths from node 1 and node k using zero-slack edges in E^*, which will be utilized to construct a initial feasible design.

3.4 Dual-Based Heuristic

Besides providing a lower bound $\overline{Z}_2$ on the optimal value of the P, the dual solution can also generate an initial primal feasible solution. Presented below are the complementary slackness conditions for P and CDP.

$$y_i\,[f_i + 2h_i^i - \sum_{j\in J} \max\{0, v_j - c_{ij}\}] = 0, \quad \forall i \in I, \tag{29}$$

$$(y_i - x_{ij})\max\{0, v_j - c_{ij}\} = 0, \quad \forall i \in I, j \in J, \tag{30}$$

$$y_{il}\,[d_{il} + u_{il} - \sum_{k\in K} \max\{0, h_l^k - h_i^k\} - \sum_{k\in K} \max\{0, h_i^k - h_l^k\}] = 0, \quad \forall (i, l) \in E, \tag{31}$$

$$(y_{il} - z_{il}^{pk})\max\{0, h_l^k - h_i^k\} = 0 \text{ and}$$

$$(y_{il} - z_{li}^{pk})\max\{0, h_i^k - h_l^k\} = 0, \quad \forall (i, l) \in E, k \in K, p \in P \tag{32}$$

$$u_{ij}(1 - y_{il}) = 0, \quad \forall (i, l) \in E. \tag{33}$$

We shall construct an initial primal feasible solution which satisfies the complementary slackness conditions as much as possible. We will use this solution as a strarting point for local improvement heuristic.

For a feasible dual solution $\{v_j^+, h_i^i, u_{il}\}$ at hand, let I^+ denote the set of facility sites such that the following two conditions are met: (a) $\Sigma_{j\in J}\max\{0, v_j^+ - c_{ij}\} = f_i + 2h_i^i$ for each $i\in I^+$ where $\{h_i^i\}$ is the present efficient solution of MOP, and (b) for each j, $v_j^+ \geq c_{ij}$ for some $i\in I^+$. For each j, define c_j^+ and $i^+(j)\in I^+$ as follows:

$$c_j^+ = c_{i^+(j),j} = \min_{i\in I^+} c_{ij}, j \in J.$$

If we let

$$y_i^+ = \begin{cases} 1, i \in I^+ \\ 0, \text{ otherwise} \end{cases} \quad \forall i \in I \tag{34}$$

$$x_{ij}^+ = \begin{cases} 1, i = i^+(j) \\ 0, \text{ otherwise} \end{cases} \quad \forall j \in J. \tag{35}$$

The $\{y_i, x_{ij}\}$ part of the primal solution then satisfies the complementary slackness condition (29) with possible violation of the condition (30).

We now focus on the construction of the remaining primal variables $\{y_{jl}^+, z_{il}^k\}$, for which the conditions (31), (32) and (33) are related. Recall that the root node is connected to every backbone nodes opened along at least two paths containing only zero slack edges in E^* at the termination of the dual process. This property will be utilized for constructing an initial primal feasible solution. Let A^* be a set of directed arcs which is obtained by splitting each edge $(i, l) \in E^*$ into two directed arcs (i, l) and (l, i). From the arc set A^*, obtain the subgraph $G=(N^+, A^+)$. N^+ is the set of nodes in I which is connected from node 1, such that there is a directed path consisting of arcs in A^* from node 1. G is the graph induced by N^+, i. e., $A^+=\{(i, l) \in A^* \mid i \in N^+ \text{ and } l \in N^+\}$. Note that $I^+ \subset N^+$, since otherwise our dual ascent procedure would not terminate. Those candidate sites not in I^+ but belonging to N^+, which will be denoted by $\bar{I}^+$, may be interpreted as a transhipment node for connection purpose only, since $y_{jl}^+ = 0$ for these nodes in the above construction. Another point to note is that some arcs in A^* which seem redundant are not included in A^+.

From the graph G, we first extract the minimum spanning tree rooted at node 1, denoted by $\overline{T}$. Removing from $\overline{T}$ all the leaf nodes to $\bar{I}^+$ along with their arcs, we obtain a smaller rooted directed tree, denoted by T'. Using the same method, then, another minimum spanning tree at rooted at node 1 is extracted from the graph $G' = (N^+, A^+ \setminus T')$, denoted by T''. Note that two-connectivity constraints are satisfied by T' and T'' since the primary and secondary path from node 1 to every backbone nodes opened can be found along with the arcs on T' and T'' respectively. Based on T' and T'', the remaining portion of primal variables are defined as follows:

$$y_{il}^+ = \begin{cases} 1, & (i, l) \text{ or } (l, i) \in T' \cup T'' \\ 0, & \text{otherwise} \end{cases} \quad \forall (i, l) \in E, \tag{36}$$

$$z_{il}^{1k+} = \begin{cases} 1, & (i, l) \in P'(k) \\ 0, & \text{otherwise} \end{cases} \quad \forall k \in K, (i, l) \in E, \tag{37}$$

$$z_{il}^{2k+} = \begin{cases} 1, & (i, l) \in P''(k) \\ 0, & \text{otherwise} \end{cases} \quad \forall k \in K, (i, l) \in E, \tag{38}$$

where $P'(k)$ and $P''(k)$ denote the primal and the secondary paths from node 1 to k on T' and T'' respectively. The primal variables $\{y_{jl}^+, z_{jl}^{pk}\}$ constructed as above always satisfy the complementary slackness conditions (31) and (33). But violations of the other condition (32) occur when the paths $P'(k) \in T'$ and $P''(k) \in T''$ are different from the shortest path of each path type from node 1 to k. By this construction we now have an initial upper-bound for our original problem P.

4. Computational Experiments

Our solution procedutre was coded in FORTRAN IV on an IBM/PC compatible 486 system. We simply use randomly generated problems to test our solution procedure because it is very hard to obtain a large volume of real data. Problems were generated by same procedure used to the Chapter 3 with a few modification. The big differences between these network generators are that the procedure used in this in this paper generates an undirected and a two-connected network, while that of paper generates only a directed network. Note that, to generate the two-connected network, we constructed two arbitrary spanning tree covering all the selected backbone nodes by using different edges, and then overlapped them.

Test runs with a number of problems, renging in size=(# of nodes, arcs, and demand points), from (20, 70, 50) to (50, 310, 200), were carried out. In all we solved a total of 31 problems which were classified into three different sets. The first set of problems listed in Table 1 denotes the test problems of different size for a given cost parameter with FC/LC=2 and AC/LC=0.8. The other sets of problems denote the test problems according to the variation of FC/LC, and AC/LC and are listed in Table 2 and Table 3 respectively.

To evaluate the effectiveness of our lower and upper bounding heuristics, we focus on two algorithmic performance measures: One is algorithmic effectiveness, measured by the % Gap to the initial dual and primal solution, defined as the difference between upper and lower bounds as a % of the lower bound. The other is computational time, measured by CPU times.

Table 1 summarizes the result of the first problem set . Our computer implementation as shown in Table 1 shows that % Gap is highly data-dependant and tends to increase with the network size, but decreases consistently according to the increase of the AC/LC ratio., % Gap can be further decreased by using the improvement heuristic such as the Add-Drop, Problem Reduction, and Interchange rule.

5. Discussion

This paper has dealt with the problem for designing two-level hierarchical networks with survivability constraints. Since the TTHN design problem is an extension of the THNTS design problem, there exist some common features between them. After investigating differences between the two problems in the mathematical point of view, we have suggested a solution method from that of the THNTS design problem. A modified dual ascent heuristic has been proposed.

The solution procedure has been tested on a variety of randomly generated sample problems. The computational results showed that our solution procedure is effective in solving the TTHN design problem. However, there still remains elaboration for tightening the bound by using survivability constraints in TTHN design model and the primal improbement heuristic procedure.

Table 1: Summary of the Computational Experiments

Problem No.	Problem Size[a] $\|I\|\times\|E\|$	$\|N\|$	Lower Bound	Upper Bound[b]	% Gap[c]	CPU Time (sec)
1	20× 70	50	11408	12790	10.8	0.050
		100	14199	14950	5.0	0.110
		200	26955	27332	1.3	0.170
2	25×100	50	10990	12576	12.6	0.160
		100	14571	18126	19.6	0.220
		200	27656	29337	5.7	0.330
3	30×130	50	11174	13257	15.7	0.270
		100	15453	20050	22.9	0.490
		200	25963	27602	5.9	0.550
4	35×150	50	13699	14746	7.1	0.380
		100	17021	22146	23.1	0.660
		200	29355	31443	6.6	0.440
5	40×200	50	13402	15088	11.1	0.270
		100	17039	22060	22.7	0.550
		200	30230	32199	6.1	0.660
6	45×260	50	13059	15978	18.2	1.540
		100	17069	24778	31.1	1.210
		200	33005	39708	16.8	1.810
7	50×310	50	13146	19556	32.7	0.820
		100	16633	27421	39.3	2.040
		200	32185	41213	21.9	1.930

[a] Problem size is given by $|I|\times|E|$ and $|N|$, where $|I|$, $|E|$ and $|N|$ denote the number of candidate nodes, arcs, and demand nodes respectively.

[b] Upper bound denotes the value of initial primal feasible solution.

[c] $\% \text{ Gap} = \dfrac{\text{Upper Bound - Lower Bound}}{\text{Lower Bound}} \times 100$

Table 2: Effect of FC/LC Ratio*

Problem No.	FC/LC Ratio	Lower Bound	Upper Bound	% Gap	CPU Time (sec)
1	0.6	13436	18189	26.13	3.520
2	0.8	12665	15682	19.24	1.150

3	1	13058	20042	34.85	2.080
4	2	16174	22393	25.62	1.310
5	4	18848	23348	19.28	0.820

* The size of test problem : ($|I|, |E|, |N|$) = (50, 310, 100);
AC/LC = 0.5; Scale factor = 10.

Table 3: Effect of AC/LC Ratio*

Problem No.	AC/LC Ratio	Lower Bound	Upper Bound	% Gap	CPU Time (sec)
1	0.2	8775	14892	41.08	1.870
2	0.8	23197	35251	34.20	2.470
3	1	26497	32623	18.78	2.250
4	1.5	35883	42141	14.85	2.640
5	2	45031	47014	4.22	1.700

* The size of test problem : ($|I|, |E|, |N|$) = (50, 310, 100);
FC/LC = 2; Scale factor = 10.

References

[1]. A. Balakrishnan and T.L. Magnanti, and R. T. Wong, "A Dual-Ascent Procedure for Large-Scale Uncapacitated Network Design," *Opns. Res.* 37 (1989), 716-740.

[2] A. Balakrishnan, T.L. Magnanti, and P. Mirchandani, "The Multi-Level Network Design Problem", Working Paper, OR Research Center, Massachusetts Institute of Technology, Cambridge, 1991.

[3] A. Balakrishnan, T.L. Magnanti, and P. Mirchandani, "A Dual-Based Algorithm for Multi-Level Network Design Problem", Working Paper, OR Research Center, Massachusetts Institute of Technology, Cambridge, 1991.

[4] D. Bienstock, E. F. Brickell, and C. L. Monma, "On the Structure of Minimum-Weight *k*-Connected Spanning Networks", *SIAM J. Disc. Math.* 3 (1990), 320-329.

[5] R.R. Boorstyn and H. Frank, "Large-scale Network Topological Optimization", *IEEE Trans. Commun.* COM-25 (1977), 29-46.

[6] S.H. Chung, Y.S. Myung and D.W. Tcha, "Simultaneous Design of a Network with a Two-Level Hierarchical Structure", To appear in *Europ. J. of Opns. Res.*

[7] D. Erlenkotter, "A Dual-based Procedure for Uncapacitated Facility Location", *Opns. Res.* 26 (1978), 992-1009.

[8] M. X. Goemans and D. J. Bertimas, "Survivable Networks, Linear Programming Relaxations and the Parsimonious Property", MIT OR Research Center Work-

ing Paper, (1990).

[9] M. Gro"tschel and C. L. Monma and M. Stoer, "Facets for Polyhedra Arising in the Design of Communication Networks with Low-Connectivity Constraints", Working Paper, No. 187, Augsburg University, West Germany (1989).

[10] M. Gro"tschel and C. L. Monma and M. Stoer, "Computational Results with a Cutting Plane Algorithm for Designing Communication Networks with Low-Connectivity Constraints", Working Paper, No. 188, Augsburg University, West Germany (1989).

[11] M. Gro"tschel and C. L. Monma and M. Stoer, "Polyhedral Approaches to Network Survivability", Working Paper, No. 189, Augsburg University, West Germany (1990).

[12] Jae-gyun Kim and Dong-wan Tcha, "Optimal Design of a Two-Level Hierarchical Network with Tree-Star Configuration", *Computers and Industrial Engineering* Vol. 22, No. 3, 273-281, 1992.

[13] C. L. Monma, B. S. Munson, and W. R. Pulleyblank, "Minimum-Weight Two-Connected Spanning Networks", *Math. Prog.* 46 (1990), 153-171.

[14] C. L. Monma and D.F. Shallcross, "Methods for Designing Communications Networks with Certain Two-Connected Survivability Constraints", *Opns. Res.* 37 (1989), 531-541.

[15] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & Sons, 1988.

[16] A. Segev, "The Node-weighted Steiner Tree Problem", *Networks* 17, 1-17, 1987.

[17] R.T. Wong, "A Dual Ascent Approach for Steiner Tree Problems on a Directed Graph", *Math. Prog.* 28, 271-287, 1984.