Assignment of ADM Rings and DCS Mesh in Telecommunication Network

*Chae Y. Lee and **Seok J. Koh

* Department of Industrial Engineering, Korea Advanced Institute of Science and Technology, 373-1 Kusung Dong, Taejon 305-701, Korea

E-mail : cylee@heuristic.kaist.ac.kr

** Protocol Engineering Center, ETRI, 161 Kajung Dong, Taejon, 305-350, Korea

Abstracts

This paper discusses a design of hybrid ring-mesh network in survivable communication network. Given a set of traffic demands, the problem is to assign each traffic demand to rings and mesh such that the cost of ADM and DCS equipments required is minimized. This assignment problem can be considered together with the fiber routing of nodes on rings and mesh. As a solution procedure, tabu search is developed with a recency based short term and a frequency based long term memory structure. In computational experiments, the proposed tabu search is compared with the solutions obtained by the branch and bound procedure of CPLEX. We see that the tabu search provides nearly optimal solution within sufficiently short time periods for all test problems with a gap of approximately 1- 4% from the lower bound.

Key Words : Communication Network Design, ADM Ring, DCS Mesh, Tabu Search

Introduction

In the conventional approach for the survivable network design, a network is divided into several clusters, each of which includes one or two hub switching offices[1,2]. To provide diverse protection from a network failure, each cluster employs add-drop multiplexer (ADM) based self-healing rings (SHR) [3,4]. On the other hand, digital cross-connect system (DCS) based mesh architecture is employed between clusters[5]. This clustering approach reduces the computational complexity of large fiber network design. However, it may not induce a globally optimal cost due to individual design of each cluster. In addition, a ring structure may not be cost-effective to expand as demands increase through time, since a lot of ADM equipment might need to be deployed to cover the increased traffic.

In this paper, a design of integrated ADM ring and DCS mesh network is considered in communication networks. Figure 1 illustrates three alternative network architectures: ring, mesh and ring/mesh. In Figure 1(c), broadband DCS (B-DCS) equipment includes the self-healing ring functionality of ADM. In the study by Doverspike *et. al.*[6], it was proved that an integrated architecture of ring and mesh in Figure 1(c) is more economical than ring in Figure 1(a) and mesh in Figure 1(b).

Figure 1. Survivable network architecture

In spite of the cost effectiveness of the integrated ring-mesh architecture, related design issues have not been studied enough. In fact, such a design is very difficult to effectively solve. As a promising solution, however, we consider the following decomposition approach:

- First, we assign each traffic demand to a ring or mesh such that the cost of required ADM and B-DCS electronic equipment is minimized. According to the assignment of traffic demands, it is determined that ADM or DCS has to be located at each node in the network. In this stage, the costs of fiber material and installation are not considered because the fiber routes for the ring and mesh networks are not determined. This problem can be viewed as grouping all demands into the ring or mesh structure.
- Given the assignment of nodes and demands to the ring or mesh, the next step is to determine specific routes for the ring or mesh network. In this stage, the costs of fiber material, placement and repeaters must be considered together with fiber splicing cost. In the ring network design, the ring routing problem is similar to the traditional traveling salesperson problem[2]. In case of mesh network, the dimensioning of working and spare capacity must be addressed together with routing for mesh traffic in the DCS network[5]. The working capacity is employed to cover mesh traffic in normal operation, while the spare capacity is prepared to restore the service in the event of a network failure.

In a real network design, the above two procedures need to be jointly addressed. Moreover, it is required to interact one procedure with another. For example, the ring routing and spare capacity dimensioning may give a significant impact on the selection of a candidate ring or mesh performed in the first procedure. In particular, mutual adjustment or integration between those two procedures may be required.

In this paper, we examine the problem of assigning ADM rings and DCS mesh, which corresponds to the first procedure in the overall network design. We note that the second procedure related to the routing on rings and mesh has been studied enough[2-4].

The remainder of this paper is organized as follows. First, we present an integer program to find the optimal assignment of ADM rings and B-DCS mesh in the ring-mesh architecture. As an efficient solution procedure, tabu search heuristic is proposed with short-term and long-term memory structures. To show the effectiveness of the proposed tabu search, computational experiments are performed and discussed. Finally, we conclude this paper.

Problem Formulation

The problem of finding an assignment of ADM and DCS on ring-mesh network can be summarized as follows:

- As inputs, a set of central offices (COs) and traffic demand between each pair of COs are given;
- Outputs are an assignment of each traffic demand to ADM ring or DCS mesh such that the total cost of ADM and DCS equipment required is minimized.

The constraints are:

- Each demand is assigned to a ring or mesh. The demand assigned to a ring will be routed within the ADM ring, and the demand assigned to mesh within DCS network;
- For each ring or mesh, the sum of all traffic does not exceed its capacity;
- If a demand between COs *i* and *j* is assigned to a ring, then ADMs are installed on that ring at COs *i* and *j*, or DCSs can take the self-healing functionality of ADM at those COs. On the other hand, if a demand is assigned to mesh, then DCSs are installed on the mesh at those COs.

Given the assignment of demands and nodes to rings and mesh, the fiber routing of nodes on rings and mesh will be determined. The resulting route may consist of one or more hops in the network. Thus the traffic from the origin may undergo several transit nodes until it gets to the destination node. This route selection depends on the fiber/cable cost and the link facility pre-established in the network.

An instance of the problem can be represented as a graph $G(V,E)$ where the $n = |V|$ nodes are the telecommunication offices in the network and the $m = |E|$ edges denote pairs of nodes between which there is a traffic demand $d(e)$ for $e \in$ E. Figure 2 illustrates a problem instance with $n = 10$ and $m = 20$ as a graph. In the figure, the number on edge e represents an amount of traffic demand $d(e)$ in DS3 units or OC-1 line rates .

Figure 2. An example problem

To satisfy the traffic demand, we employ a set of candidate ADM rings $R = \{1, 2, ..., |R|\}$, where $|R|$ represents total number of rings, and DCS mesh denoted by z. Let C_r denote the capacity of each ring $r \in R$ and C_z the capacity of the mesh z. We also define the cost of an ADM for the ring r by a_r, and the cost of a DCS for the mesh z by b_z.

To formulate the problem, let us define the following variables:

$y_{er} = 1$ if traffic demand e is assigned to ring r, and $y_{er} = 0$ otherwise,

$y_{ez} = 1$ if traffic demand e is assigned to mesh z, and $y_{ez} = 0$ otherwise, and

$x_{ir} = 1$ if an ADM is installed at node i for ring r, and $x_{ir} = 0$ otherwise,

$x_{iz} = 1$ if a DCS is installed at node i for mesh z, and $x_{iz} = 0$ otherwise.

Then we obtain an integer programming formulation:

Minimize $$\sum_{r \in R} a_r \sum_{i \in V} x_{ir} + b_z \sum_{i \in V} x_{iz} \quad (1)$$

Subject to $$\sum_{r \in R} y_{er} + y_{ez} = 1 \quad \forall\ e \in E \quad (2)$$

$$\sum_{e \in E} d(e) y_{er} \le C_r \quad \forall\ r \in R \quad (3)$$

$$\sum_{e \in E} d(e) y_{ez} \le C_z \quad (4)$$

$$y_{er} \le x_{ir} + x_{iz},\ y_{er} \le x_{jr} + x_{jz} \quad \forall e = (i, j) \in E,\ r \in R \quad (5)$$

$$y_{ez} \le x_{iz},\ y_{ez} \le x_{jz} \quad \forall e = (i, j) \in E \quad (6)$$

$$y_{er},\ y_{ez},\ x_{ir},\ x_{iz} \in \{0,1\} \quad \forall e = E,\ r \in R,\ i \in V. \quad (7)$$

The objective function (1) minimizes the cost of ADM and DCS equipments required for the assignment of rings and mesh. In fiber network design, the network cost consists of electronic equipment cost and fiber cost including fiber materials and placement cost. However, fiber cost is usually considered in physical fiber routing of nodes on the ring and mesh. Thus in the assignment of ring and mesh, only the electronic equipment cost is employed as the network cost.

Although there exist many various price structures on SONET equipment costs[7], we employ the following cost structure[6]. In the ring architecture, the cost of one OC-3 ADM is approximately \$10,000, while OC-12 and OC-48 ADM are \$18,000 and \$50,000, respectively. The mesh architecture deploys B-DCS equipment that has a cost of \$80,000 with the restoration software cost

of $2,000. The cost of a DCS acting as a ring node may slightly differ from the cost of a DCS used exclusively in the mesh. When a DCS performs the ring functionality, additional cost is required because a ring emulation "intelligence" has to be built in. In the example[6], the DS3 termination cost of an exclusive DCS is $370 for one DS3 port, while the cost of an intelligent DCS with a ring emulation is $970. Thus additional $600 is required for the intelligent DCS. In that case, the model proposed in this paper can be generalized by adding the sum of additional costs for ring traffics which are routed through DCS. However, those additional costs are not considered in this paper since termination costs are relatively low compared to the equipment cost.

The demand constraints in (2) ensure that each traffic demand is assigned to a ring or mesh. In capacity constraints (3), the sum of traffic assigned to ring r cannot exceed the capacity of the ADM ring C_r. This capacity restriction is based on the unidirectional ring, while in bidirectional ring, the capacity of the ring is determined by the maximum link traffic carried within the ring. Thus the load balancing problem must be considered to determine the ring capacity. For readers who are interested in the applications of bidirectional rings, we suggest to refer to the work by Lee and Chang[8]. In constraint (4), C_z represents an amount of total traffic demands that can be cross-connected through DCS mesh. B-DCS networks usually provide a switching capability of 960 equivalent DS3 ports[6]. Thus we employ C_z as 960 DS3 units.

In constraint (5), if a demand between nodes i and j is assigned to a ring r, then ADM or DCS must be installed at nodes i and j. In constraint (6), if the demand between two nodes is assigned to the mesh z, then DCS must be installed at each of two nodes. All variables are binary integers by (7).

In real-world network design, a network designer needs to limit the number of nodes included in a ring. This is a constraint that often arises in practice. In fact, the maximum number of nodes that can be supported by a SONET ring is reported to be sixteen. Given the maximum number of nodes on ring r as M_r, the following constraint can be added.

$$\sum_{i \in V} x_{ir} \leq M_r \qquad (8)$$

The solution to the integer program, $Y = \{y_{er}, y_{ez}\}$ and $X = \{x_{ir}, x_{iz}\}$, can easily be converted into a graphical representation of the network design. Firstly, we classify each demand $e = (i,j)$ into a ring $r \in R$ or mesh z by referring to the solution Y. If $y_{er} = 1$, then two nodes i and j are located on the ring r. Secondly, based on the solution X, we determine if ADM or DCS is located at each node. If $x_{ir} = 1$, then ADM for ring r is located at node i. In case that a node i is assigned to two rings r and s at once, two ADMs may be installed at the node, or two ADMs can be replaced by one DCS, which depends on the cost structure of ADM and DCS equipment.

Figure 3 illustrates an example solution to the problem shown in Figure 2. The network consists of two OC-48 ADM rings and DCS mesh. In the figure, the ordering of nodes on each ring is arbitrarily given. Ring 1 consists or nodes 9, 10, 8 and 7, and Ring 2 consists of nodes 2, 1, 3, 4 and 5. On the other hand, the mesh network is composed of nodes 4, 5, 6, 7, and 8. Note that the DCS nodes 4, 5, 7, and 8 also perform the ADM functionality for two rings. In DCS mesh network,

traffic demands (4,7) and (5,8) are routed via node 6. For example, the demand (4,7) is routed by the links (4,6) and (6,7).

Figure 3. A possible solution to the problem

Unfortunately, the proposed integer programming could not be effectively solved by the conventional branch and bound technique using the CPLEX 6.5 package[9]. In a network with 40 nodes the branch and bound procedure failed to get optimal solutions even after 24 hours of running time. Thus a tabu search heuristic is developed as a promising solution procedure for real-world size problems.

Application of Tabu Search

In this section a heuristic procedure based on tabu search is presented. Tabu search is a search heuristic introduced by Glover[10] that has enjoyed a large amount of success in solving difficult real world combinatorial problems. At each step, the neighborhood of the current solution is explored and the best one is selected as the new solution. This procedure is called a "move". However, as opposed to other local search techniques, the procedure does not stop even when no improvement is obtained. The best solution in the neighborhood is selected, even if it is worse than the current solution. This strategy allows the search to escape from local optima and to explore a larger fraction of the search space.

To prevent cycling in the search process, recently selected solutions are forbidden. For each move leading to a new solution, the inverse move is labeled "tabu" and forbidden to move back. This is called "short-term memory". Another characteristic of tabu search is "long-term memory". After a solution region is searched by the short-term memory, the long-term memory drives the search into a new area that has not been explored until then.

In this section, we present a tabu search procedure and its implementation for the ring-mesh assignment problem. In particular, the tabu search proposed in this paper employs short term and long-term memory structure to explore much better solutions.

Initial Heuristic

Before applying the tabu search, we need to construct an initial solution. In this paper we suggest a heuristic based on the residual ring capacity to construct a feasible ring-mesh assignment. At each iteration, a demand is assigned to the ring with the largest residual capacity.

- Residual Capacity Based Assignment (RCBA)

Step 1. For each ring $r \in R$, assign the largest demand $d(e)$ to the ring if $d(e) \leq C_r$.

Set the residual capacity C_r* as C_r - $d(e)$, and y_{er} = 1 and x_{ir} = x_{jr}= 1 for demand e = (i,j).

Step 2. Do the following steps until no further demand can be assigned to a ring;

2-1. Let r be the ring with the largest residual capacity C_r*.

2-2. Select the largest demand $d(e)$ such that i or j has already been assigned to the ring r for the demand e = (i,j).

2-3. If $d(e) \leq C_r$*, then y_{er} = 1, x_{ir} = x_{jr} = 1 and C_r* = C_r* - $d(e)$.

Otherwise, y_{ez} = 1, and x_{iz} = x_{jz} = 1 and C_z* = C_z* - $d(e)$.

Step 3. For each demand e = (i,j) that has not been assigned in Step 2, assign the demand to DCS-mesh, and set y_{ez} = 1, and x_{iz} = x_{jz} = 1.

Step 4. Let X = {x_{ir}, x_{iz}} be the integer solution obtained from Steps 1, 2 and 3.

For each node $i \in V$, do

If x_{iz} = 1, then set x_{ir} = 0 for all $r \in R$.

If x_{iz} = 0 and $\Sigma_{r \in R}\, a_r x_{ir} > b_z$, then set x_{iz} = 1 and x_{ir} = 0 for all $r \in R$.

Step 5. Stop the algorithm.

Let us consider the example in Figure 2. It is assumed that two OC-48 ADM rings are employed as candidate rings. In Step 1, the demand pairs of (9,10) and (1,3) are assigned to ring 1 and 2, respectively. The residual capacity is set to be 38 for those two rings. In Step 2-1, the ring 1 is considered. In Step 2-2, the demand pair (7,9) which is connected to the demand pair (9,10) is selected. In Step 2-3, the demand (9,10) is assigned to the ring 1. Thus its residual ring capacity is updated as 29. All demands are assigned by Step 2 and 3. Step 4 improves the network cost by adjusting the configuration obtained by the previous steps. If both a DCS and an ADM are to be installed at a node, then the ADM is removed because the B-DCS replaces add-drop functionality. If total cost of ADMs at a node is greater than the cost of a B-DCS, then all ADMs are replaced with a B-DCS. In Step 4 of the RCBA algorithm, the decision variables Y = {y_{er}} is not changed because each B-DCS equipment does not generate new mesh traffic and it just replaces the ADMs.

Design of Moves

Starting from an initial ring-mesh architecture set by RCBA, the solution is improved by applying tabu search moves. In this section we define a move as a transformation of x_{ir} or x_{iz} to 1 - x_{ir} or 1 - x_{iz} for each node $i \in V$.

- Mesh_Ring(i)

In case of x_{iz} = 1, this move improves the network cost by dropping DCS and adding ADM. This is done by setting x_{iz} = 0 and x_{ir} = 1 for a ring r such that this change preserves a feasible routing for the capacity constraint. The concerning demand e = (i,j) is assigned from mesh z to ring r. The other node j is also considered as a candidate move together.

- Ring_Mesh(i)

  In case of x_{iz} = 0, this move tries to add a DCS and drop the current ADM. This is done by setting x_{ir} = 1 and x_{ir} = 0 for ring r. Note that this transformation always preserves the feasibility of the capacity constraints. The concerning demand e = (i,j) is assigned from ring r to mesh z. The other node j is also considered as a candidate move together.

- Ring_Swap(i)

  This move tries to drop the current ring r and add another ring s. This is done by setting x_{ir} = 0 and x_{is} = 1 for the ring s such that its adding preserves the feasibility. In this case, the concerning demand e = (i,j) is assigned to the ring r to ring s. Thus the other node j is also considered as a candidate move together.

At each move of tabu search, the minimum cost among the candidate moves for all nodes is selected as a move. A candidate move that is tabu or that induces infeasibility is discarded. Thus the feasibility check is required to determine whether a given solution X = {x_{ir}, x_{iz}} satisfies the capacity constraint. Based on a solution X each demand e = (i,j) can be classified as one of the following three cases.

A) Both i and j are in the mesh z.

B) Both are in the same ring r.

C) One is in the mesh z and the other is in the ring r.

In each case, $d(e)$ is allocated to a ring or a mesh as follows. In case (A), the demand can be routed through the mesh structure by using DCS switching mechanism. Thus $d(e)$ is assigned to mesh z, and set C_z* as C_z*- $d(e)$. In cases (B) and (C), $d(e)$ is assigned to ring r and set C_r* as C_r*- $d(e)$. If the capacity of each ring and mesh is satisfied for each move, the move induces a feasible traffic routing. Each candidate move that induces an infeasible routing is discarded.

Recency based Short-term Memory

To prevent the search from being trapped in a local optimum, an effective tabu restriction needs to be designed. Tabu search manages the short-term memory to explore better solutions by imposing restrictions on the composition of new solutions generated. For each type of move, we impose restrictions so that a move cannot be "reversed". In particular, if a mesh z is currently dropped from the node i, we forbid this z to move back to i for several iterations. Such a restriction prevents the

search from revisiting a local minimum in short term and greatly diminishes the chance of cycling in the long term.

How long a given restriction operates depends on a parameter called *tabusize* or *tabu tenure*, which identifies the number of iterations that a particular tabu restriction remains in force. The *tabusize* can be either fixed or variable.

In this paper, short term memory is implemented using a "recency" based memory structure. To illustrate this, let *iter* denote the current iteration number. Also denote $tabu(i,r)$ by an iteration value governing the duration that forbids a reversal of the move of transforming x_{ir}. Initially, $tabu(i,r) = 0$ for all i and r, and $iter = 1$. When tabu search restriction is imposed, we update the recency memory as $tabu(i,r) = iter + tabusize$. Thus the restriction that prevents x_{ir} from being transformed to $1 - x_{ir}$ is enforced when $tabu(i,r) > iter$.

Two types of *tabusize* are employed to impose the tabu restriction: fixed and variable. Let n be the number of nodes in the network. In the fixed case, *tabusize* is set to $n/4$ or $n/2$, while the variable *tabusize* is generated according to the uniform distribution ranged from $n/4$ to $n/2$. The efficiency for two types of *tabusize* is compared in the experimental test. In short term memory, tabu search employs a stopping criterion N_{max}, defined as the number of consecutive moves performed without cost improvement. Thus we stop tabu search if a new better solution is not found during N_{max} moves.

Frequency based Long-term Memory

The long term memory we employ makes use of a "frequency" based memory structure to achieve a diversification effect, encouraging the search to explore regions less frequently visited. More specifically, we use this memory to discourage moves that occurred frequently during the search. A transition measure is used to record the number of times that each move from x_{ir} to $1 - x_{ir}$ occurs. Let $frequency(i,r)$ be the number of times that x_{ir} is changed from $1 - x_{ir}$ in the search. Then the frequency can easily updated as follows;

$$frequency(i,r) = frequency(i,r) + 1, \text{ if } i \text{ and } r \text{ are selected as a move.}$$

When long term memory is invoked, each $tabu(i,r)$ is set to the $frequency(i,r)$ for all i and r. Also, the parameter *iter* is updated as a mean integer value of $frequency(i,r)$ for all i and r. Thus approximately a half of candidate moves will be in tabu status during the following search. Note that the frequency information is used to create a tabu restriction. This long term memory contributes significantly to the quality of solutions obtained by our approach, as the subsequent computational results disclose.

Based on discussions so far, the tabu search procedure is described as follows.

- Tabu Search Procedure

Initialization Phase

Step 1-1. Get an initial feasible solution (X,Y) by RCBA;

Step 1-2. Best_cost = cost(X,Y);

Step 1-3. *tabu*(i,r) = *frequency*(i,r) = 0 for all i, r and z;

Step 1-4. *iter* = *nmax* = 0;

Tabu Search Short-Term Memory

while *nmax* <= *Nmax* do

Step 2-1. *iter* = *iter* + 1, *nmax* = *nmax* + 1;

Step 2-2. For each node i = 1 to n do

If x_{iz} = 1 and *tabu*(i,z) ≤ *iter*, then (i,r,z) = Mesh_Ring(i);

Else if x_{iz} = 0 and *tabu*(i,z) ≤ *iter*, then (i,r,z) = Ring_Mesh(i);

Else if x_{iz} = 0 and *tabu*(i,z) > *iter*, then (i,r,z) = Ring_Swap(i);

(In Mesh_Ring and Ring_Swap moves, it is required that *tabu*(i,r) ≤ *iter* for the newly selected ring r.)

Step 2-3. Let (X*, Y*) be the best solution and (i,r,z) be parameters obtained from Step 2-2.

i) (X,Y) = (X*,Y*);

ii) *tabu*(i,r) = *iter* + *tabusize* and *frequency*(i,r) = *frequency*(i,r) + 1;

Step 2-4. If cost(X,Y) < Best_cost, then Best_cost = cost(X,Y), *nmax* = 0;

Tabu Search Long-Term Memory

If a new best solution is not found, stop the search. Otherwise, do the following steps.

Step 3-1. *tabu*(i,r) = *frequency*(i,r) for all i, r and z;

Step 3-2. *iter* = mean of *frequency*(i,r) and *nmax* = 0;

Step 3-3. Go to Tabu Search Short-Term Memory;

Computational Results

To test the performance of the proposed tabu search algorithm, 20 problem sets are generated with 10, 20, 30, 40 and 50 nodes, different ring types, and different number of demand pairs of approximately 2n and 3n.. Table 1 specifies the problem sets to be tested. For each problem, the DS3 demand for each demand pair is randomly generated, which is ranged uniformly between 1 and 10 integer units. The table also shows the number of candidate OC-48 and OC-12 rings required to cover all demands for each problem.

Table 1. Test problems

Before solving the ring-mesh assignment problem we tested the performance of two parameters:

tabusize and stopping criterion *Nmax*. The *tabusize* represents the number of iterations during which a listed node cannot be included into the related ring or mesh. The stopping rule *Nmax* represents the number of consecutive iterations allowed for the search to continue without cost improvement. The experimental results are illustrated in Table 2 and 3. The RCBA algorithm was employed as an initial heuristic for all experiments.

A suitable *tabusize* usually depends on the problem structure. In particular, a fixed or variable *tabusize* including its scale should be determined through various experimental simulations. In Table 2, fixed *tabusize* $n/4$, $n/2$ and a uniformly distributed *tabusize* ranged from $n/4$ to $n/2$ are compared with the initial solution for each problem. From the table, we conclude that tabusize $n/4$ is appropriate for the problem structure considered in this paper. This implies that a large *tabusize* is too restrictive to generate good solutions for the problem. Table 3 shows the performance of tabu search with different value of *Nmax*. Based on the experiments by various *Nmax*, we see that the stopping criteria *Nmax* with $2n$ is appropriate for all the problems.

Table 2. Comparison of tabu search with different tabusize

Table 3. Comparison of tabu search with different Nmax

In Table 4, for the problems with 10, 20, and 30 nodes, the computational results of tabu search are shown with a recency based short term memory, a frequency based long term memory and optimal solution. In the table, optimal solutions are obtained by solving each integer problem by running the CPLEX solver[9]. Each value in the parenthesis represents the CPU time in units of seconds (s), minutes (m), and hours (h). The tabu search using both short term and long term memory (LTS) outperforms the solution by using only short term memory (STS). In the table, tabu search gives optimal solutions within one minute for all problems with 10 and 20 nodes. Nearly optimal solutions are obtained for some problems with 30 nodes. We note that the computational running time for the tabu search increases linearly as the number of nodes gets larger, but the computation time by CPLEX increases exponentially.

Table 4. Results of tabu search in problems with 10, 20 and 30 nodes

Table 5 shows the computational results of tabu search in problems with 40 and 50 nodes. For the problems, we fail to get optimal solutions until the branch and bound nodes of more than 2 millions have been investigated. In particular, the convergence speed of CPLEX becomes very slow after 24 hours. Thus, we take a lower bound for each problem, which is given by the CPLEX after running time of 24 hours. The CPLEX solver produces feasible solutions with lower bounds for a problem, and terminates the branch and bound procedure in the normal case when the feasible solution and the lower bound are the same. The feasible solution is denoted by CPLEX heuristic in the table, and compared with the result of tabu search (LTS). From the table, we see that tabu search gives better solutions than the CPLEX heuristic within 2 minutes with a gap of approximately 10 – 25 %. In the table, it is also shown that the proposed tabu search provides near optimal solutions, the gap of

which is approximately 1 - 4% from the lower bounds or optimal solutions.

Table 5. Results of tabu search in problems with 40 and 50 nodes

In Table 6 the cost of pure mesh network is compared with that of hybrid ring-mesh network. In the previous study[6], it was proved that the mesh network is more cost-effective than the ring network. Thus the cost of the ring network is not presented. In the table, the cost of mesh network is obtained by locating a B-DCS equipment at all nodes. From the table, it seems that the hybrid network can provide the cost savings of approximately 10% - 30%, compared to the mesh network.

Table 6. Comparison of mesh and ring/mesh network cost

Conclusions

In this paper, a problem of assigning traffic demands to the ring and mesh network has been addressed. Each traffic demand is assigned to either ring or mesh such that the cost of ADM and DCS equipment required is minimized. In real-world network design, this assignment problem can be used together with the fiber routing of the ring and mesh nodes.

A tabu search heuristic is developed to solve the problem. Starting from an initial heuristic by a residual capacity based algorithm, three types of moves such as ring_mesh, mesh_ring and ring_swap are employed to improve the network cost. To discourage the search from being trapped in a local optimum, a short term memory structure is implemented using the recency based memory. In the long term memory, frequency information is used to encourage the search to explore regions less frequently visited and to get better solutions.

To test the performance of the proposed tabu search, computational experiments are examined for randomly generated problems. Computational results show that the proposed tabu search provides nearly optimal solutions for all problems with a gap of approximately 1% - 4% from the lower bounds within two minutes. It is also shown that the hybrid network provides the cost savings of approximately 10% - 30%, compared to the mesh network.

References

1. T.-H. Wu (1992). Fiber Network Service Survivability. *Artech House*.

2. O. J. Wasem, T.-H. Wu and R. H. Cardwell (1994) Survivable SONET Networks - Design Methodology. *IEEE Journal on Selected Areas in Communications* **12**: 205-212.

3. C. Y. Lee and S. J. Koh (1997) A Design of the Minimum Cost Ring-Chain Network with Dual-Homing Survivability: A Tabu Search Approach. *Computers and Operations Research* **24**: 883 - 897.

4. A. Sutter, F. Vanderbeck and L. Wolsey (1998) Optimal Placement of Add/Drop Multiplexers: Heuristic and Exact Algorithm. *Operations Research* **46**: 719 - 729.

5. S. Hasegawa, Y. Okanoue, T. Egawa and H. Sakauchi (1994) Control Algorithm of SONET Integrated Self-Healing Networks. *IEEE Journal on Selected Areas in Communications* **12**: 110 - 119.

6. R. D. Doverspike, J. A. Morgan and W. Leland (1994) Network Design Sensitivity Studies for Use of Digital Cross-connect Systems in Survivable Network Architectures. *IEEE Journal on Selected Areas in Communications* **12**. 69 - 78.

7. Federal Technology Service (1999) Wire and Cable Service, http://www.fts.gsa.gov.

8. C. Y. Lee and S. G. Chang (1997) Balancing Loads on SONET Rings with integer Demand Splitting. *Computers and Operations Research* **24**: 221-229.

9. Integer Programming Solver (1999) CPLEX version 6.5. *CPLEX Optimization. Inc*.

10. F. Glover (1989) Tabu Search - Part I. *ORSA Journal of Computing* **3**: 190 – 206.

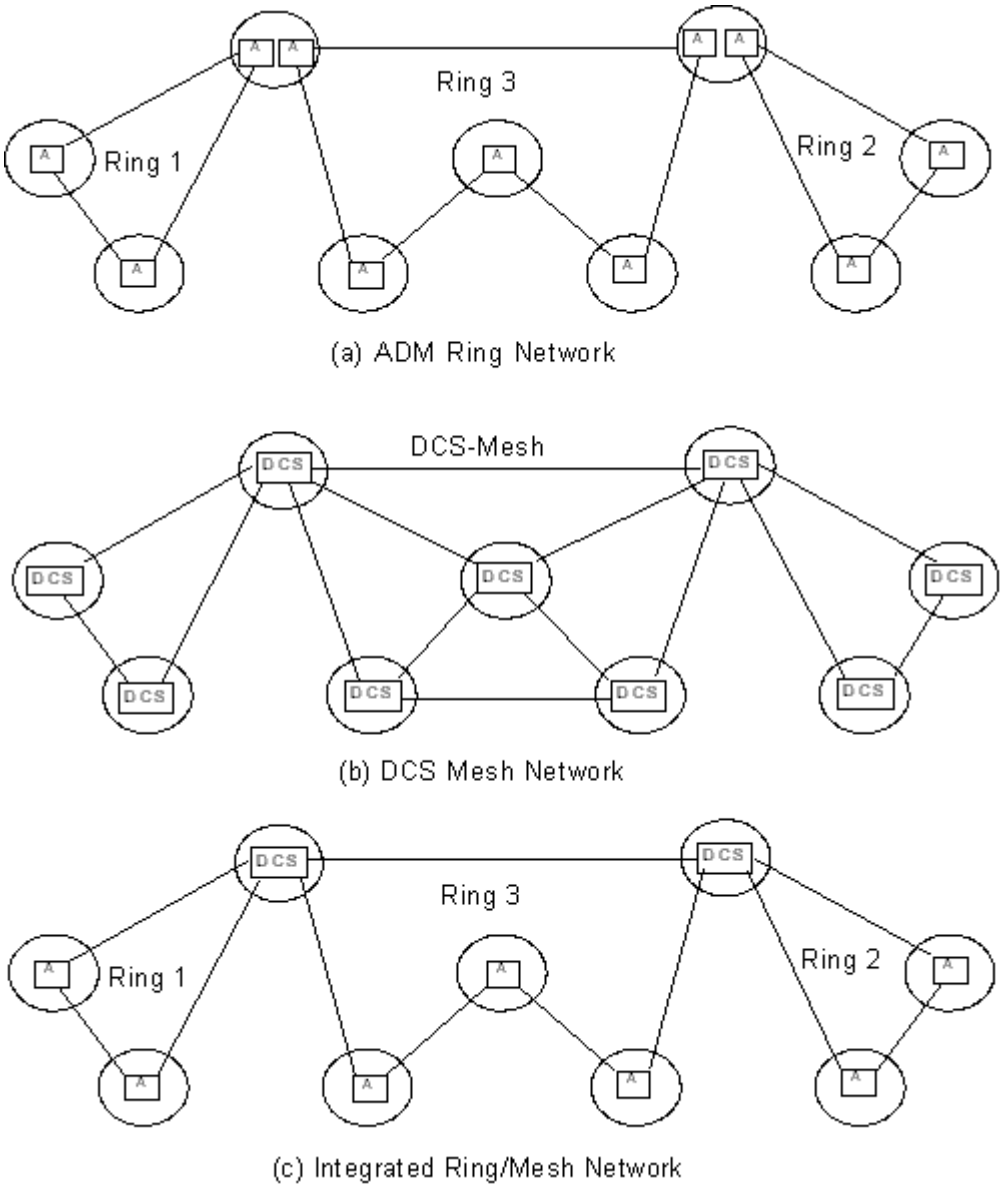

Figure 1. Survivable network architecture

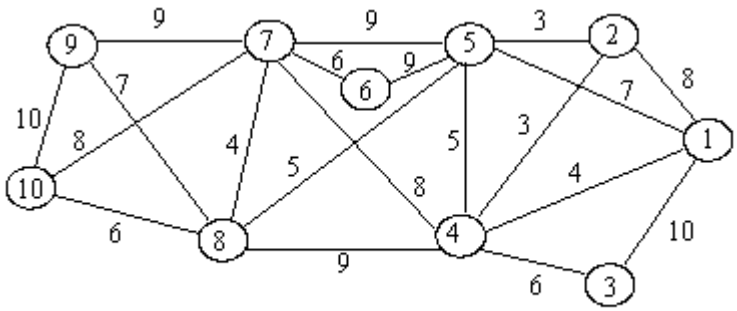

Figure 2. An example problem

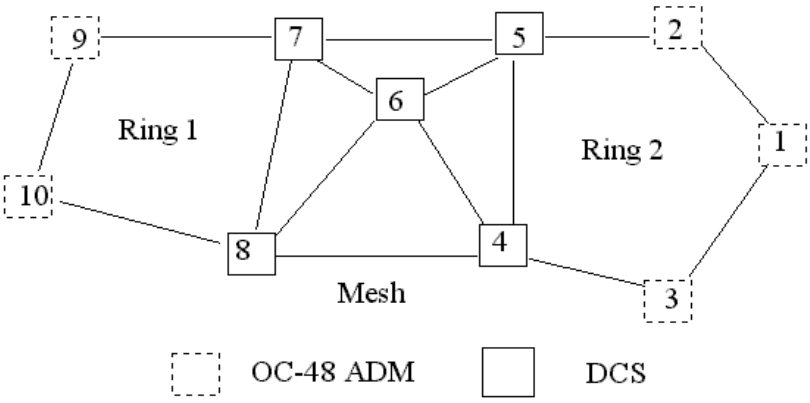

Figure 3. A possible solution to the problem

Table 1. Test problems

Number of nodes	Problem set	Number of demand pairs	Total demands	Number of rings	
				OC-48	OC-12
10	Set 1	20	120	3	0
	Set 2	20	120	2	2
	Set 3	30	177	4	0
	Set 4	30	177	3	3
20	Set 1	40	227	5	0
	Set 2	40	227	4	3
	Set 3	60	336	7	0
	Set 4	60	336	6	4
30	Set 1	60	336	7	0
	Set 2	60	336	6	4
	Set 3	90	496	11	0
	Set 4	90	496	10	2
40	Set 1	80	456	10	0
	Set 2	80	456	9	2
	Set 3	120	650	14	0
	Set 4	120	650	13	3
50	Set 1	100	554	12	0
	Set 2	100	554	11	3
	Set 3	150	800	17	0
	Set 4	150	800	16	3

Table 2. Comparison of tabu search with different *tabusize*

Number of Nodes	Problem set	Initial solution	Tabusize		
			n/4	Uniform	*n*/2
10	Set 1	740	650	654	654
	Set 2	752	656	664	672
	Set 3	800	740	744	746
	Set 4	800	743	743	748
20	Set 1	1480	1330	1342	1342
	Set 2	1493	1336	1336	1344
	Set 3	1540	1360	1362	1368
	Set 4	1555	1374	1378	1390
30	Set 1	2250	1980	1992	1996
	Set 2	2264	1986	1994	1994
	Set 3	2370	2130	2145	2152
	Set 4	2384	2135	2135	2140
40	Set 1	2960	2720	2720	2732
	Set 2	2964	2728	2728	2736
	Set 3	3080	2810	2814	2818
	Set 4	3095	2818	2818	2832
50	Set 1	3650	3250	3280	3280
	Set 2	3662	3258	3262	3266
	Set 3	3880	3370	3380	3396
	Set 4	3893	3382	3392	3392

Table 3. Comparison of tabu search with different *Nmax*

Number of Nodes	Problem set	Initial solution	Nmax		
			n	2*n*	3*n*
10	Set 1	740	654	650	650
	Set 2	752	668	656	656
	Set 3	800	748	740	740
	Set 4	800	753	743	743
20	Set 1	1480	1360	1330	1330
	Set 2	1493	1344	1336	1336
	Set 3	1540	1368	1360	1360
	Set 4	1555	1386	1374	1374
30	Set 1	2250	1996	1980	1980
	Set 2	2264	1998	1986	1986
	Set 3	2370	2145	2130	2130
	Set 4	2384	2148	2135	2135
40	Set 1	2960	2732	2720	2720
	Set 2	2964	2746	2728	2728
	Set 3	3080	2822	2810	2810
	Set 4	3080	2832	2818	2818
50	Set 1	3650	3250	3250	3250
	Set 2	3662	3258	3258	3258
	Set 3	3880	3380	3370	3370
	Set 4	3893	3392	3382	3382

Table 4. Results of tabu search in problems with 10, 20 and 30 nodes

Number of nodes	Problem set	Short-term TS	Long-term TS	Optimal Solution	100(TS-Opt) /Opt
10	Set 1	650 (1.12s)	620 (1.56s)	620 (12s)	0
	Set 2	656 (1.14s)	624 (1.82s)	624 (15s)	0
	Set 3	740 (1.14s)	710 (2.14s)	710 (2m 15s)	0
	Set 4	743 (1.87s)	713 (2.16s)	713 (2m 22s)	0
20	Set 1	1330 (2.24s)	1270 (3.23s)	1270 (42m)	0
	Set 2	1336 (2.67s)	1284 (3.56s)	1284 (46m)	0
	Set 3	1360 (3.87s)	1330 (4.23s)	1330 (1h 25m)	0
	Set 4	1374 (4.34s)	1345 (5.15s)	1345 (1h 36m)	0
30	Set 1	1980 (5.24s)	1950 (6.74s)	1950 (11h 26m)	0
	Set 2	1986 (5.87s)	1954 (7.12s)	1950 (11h 35m)	0
	Set 3	2130 (15.34s)	2104 (18.65s)	2096 (15h 28m)	0.38 %
	Set 4	2135 (16.54s)	2115 (20.78s)	2106 (18h 32m)	0.43 %

Each value in the parenthesis represents the CPU time.

Table 5. Results of tabu search in problems with 40 and 50 nodes

Number of nodes	Problem set	Short-term TS	Long-term TS	CPLEX Heuristic	100(CH-TS)/TS	Lower Bound	100(TS-LB)/LB
40	Set 1	2720 (18.24s)	2670 (22.45)	2943 (24h)	10.22 %	2638 (24h)	1.21 %
	Set 2	2728 (22.49s)	2681 (34.34)	2956 (24h)	10.25 %	2646 (24h)	1.32 %
	Set 3	2810 (26.32s)	2780 (37.54)	3092 (24h)	11.22 %	2731 (24h)	1.79 %
	Set 4	2818 (31.34s)	2794 (41.23)	3113 (24h)	11.41 %	2744 (24h)	1.82 %
50	Set 1	3250 (37.56s)	3220 (47.48)	3875 (24h)	20.34 %	3122 (24h)	3.14 %
	Set 2	3258 (39.34s)	3225 (48.54)	3894 (24h)	20.74 %	3123 (24h)	3.27 %
	Set 3	3370 (1m 15s)	3340 (1m 34s)	4054 (24h)	21.37 %	3227 (24h)	3.50 %
	Set 4	3382 (1m 24s)	3360 (1m 38s)	4183 (24h)	24.49 %	3238 (24h)	3.77 %

Table 6. Comparison of mesh and ring/mesh network cost

Number of nodes	Problem Set	Mesh (M)	Ring/Mesh (RM)	Cost saving (%) 100(M-RM)/RM
Set 1	1	800	620	29.03 %
	3	800	710	12.67 %
Set 2	1	1600	1270	25.98 %
	3	1600	1330	20.31 %
Set 3	1	2400	1950	23.07 %
	3	2400	2096	14.50 %
Set 4	1	3200	2670	19.85 %
	3	3200	2780	15.11 %
Set 5	1	4000	3220	24.22 %
	3	4000	3340	19.76 %