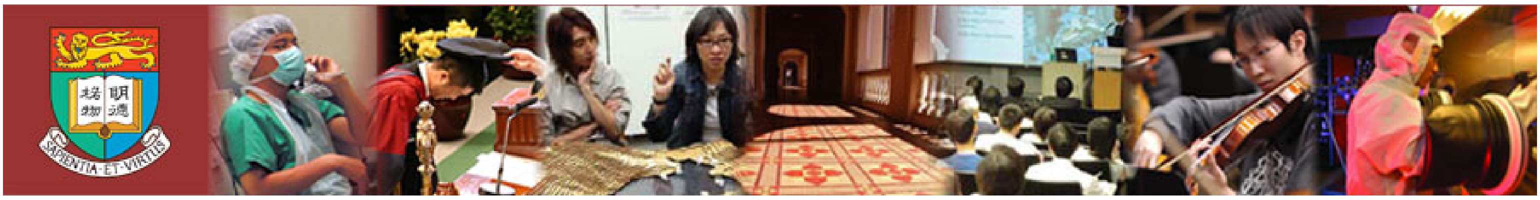


ILP-Based Design of Survivable Broadcast Trees

Minjing Mao, Bin Wu and Kwan L. Yeung

Department of Electrical and Electronic Engineering

The University of Hong Kong, Pokfulam, Hong Kong

Email :{mjmao, binwu, kyeung}@eee.hku.hk

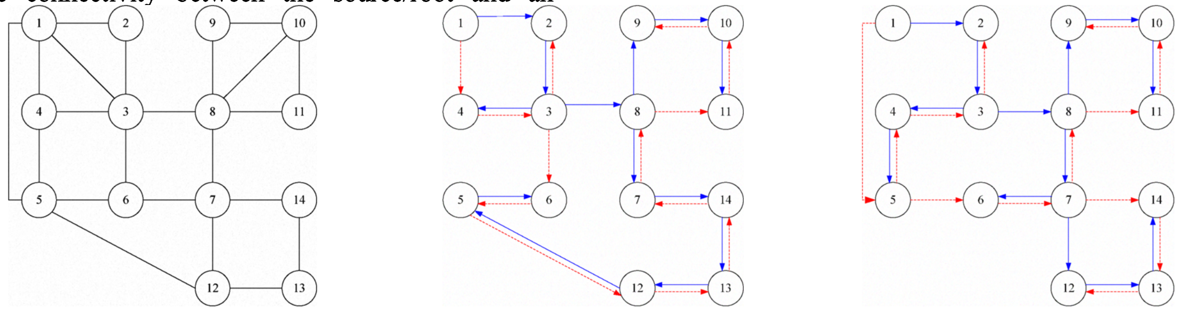
***Abstract*—We study survivable broadcast in high-speed networks against a single link/node failure. We follow the classic approach of blue/red tree [1] to construct a pair of spanning trees (i.e. a *blue tree* and a *red tree*) such that the connectivity between the root and an arbitrary node is ensured (via at least one tree) in the presence of a single link/node failure. To ensure that the blue and red trees are constructed in a cost-efficient way, heuristic algorithms have been designed to minimize the cost involved in tree construction. In this paper, we tackle the same problem but resorting to Integer Linear Programming (ILP) for optimal solutions. Two efficient ILPs are formulated, one for protecting against single link failure (MinCost-E) and the other for node failure (MinCost-V). Numerical results show that our ILPs can generate optimal solutions in relatively short amount of time. As compared with the existing heuristic algorithms, we observe a significant improvement in performance.**

Keywords- Integer Linear Programming (ILP); survivable broadcast trees; blue/red tree.

I. INTRODUCTION

Due to the high-speed nature of our backbone networks nowadays, even a very short service downtime can result in enormous data loss. Therefore, network survivability is of paramount importance. To this end, various protection and restoration schemes have been extensively studied in the literature, e.g. [1]–[15]. In this paper, we focus on designing protection schemes for broadcast applications. In general, broadcast applications are more sensitive to failures than unicast because more downstream nodes will be affected by a single failure in the network. For fast broadcast protection [1]–[5], the blue/red tree approach is usually adopted, in which the connectivity between the source/root and an arbitrary node is always ensured (via either blue or red or both trees) in the presence of a single link/node failure. An example is shown in Fig. 1. Source node 1 is broadcasting to all other nodes in the network. A pair of blue and red trees for protecting against any single link failure (in Fig. 1(a)) are constructed in Fig. 1(b). If node 1 is broadcasting along the blue tree and link (5, 12) fails, (downstream) nodes 5 and 6 are disconnected from the blue tree. Since they are still connected by the red tree, the receiving can be quickly resumed when node 3, the first upstream node of nodes 5 & 6 in the red tree and also a node in the uninterrupted blue tree, starts to forward a copy of data it received from the blue tree onto the red tree branch/link (3, 6).

It is shown in [1] that for an arbitrary node-redundant (i.e. 2-vertex connected) or edge-redundant (i.e. 2-edge connected) network, there always exists such a pair of blue tree T^B and red tree T^R to protect against any single link or node failure. For a given broadcast session, the blue/red tree solution (i.e. the pair of blue/red trees found) is not unique, e.g. Figs. 1 (b) & (c) are two possible solutions for the same broadcast session rooted at node 1. Each solution is associated with a different network cost, which is defined in [2] as the number of *distinct* links traversed by the two trees. That means if a link is traversed by both trees, e.g. link (5, 12), it is only counted once. This is because at any time on a specific link, only one tree (either T^B or T^R) can be active, and thus the transceivers at the two ends of the link as well as the link bandwidth can be shared by both trees. From Figs. 1(b) & (c), we can see that the network costs for the two solutions are 16 and 17, respectively.



(a) Network topology. (b) Our blue/red tree solution. (c) Blue/red tree solution from [2].

Fig. 1. Blue/red tree solutions for the example network in [2], where edges not on the trees are removed in (b) & (c).

978-1-4244-5174-6/09/$26.00 ©2009 IEEE

In order to minimize the network cost, the tree construction algorithm should maximize the number of common links traversed by the two trees. To this end, some heuristic tree construction algorithms are proposed in [3], and are subsequently refined in [2]. In this paper, we also target at finding the minimum cost blue/red tree pair. But unlike [2, 3], we adopt the Integer Linear Programming (ILP) for optimal solutions. Efforts are made in formulating efficient ILPs to speed up the optimization process. Although the ILP running time is still generally longer than heuristic algorithms, we show that the reduction in network cost is significant.

The rest of this paper is organized as follows. Section II formulates two ILPs for network cost minimization, one for link failure (MinCost-E) and one for node failure (MinCost-V). Numerical results are presented in Section III. We conclude the paper in Section IV. Throughout the paper, "link" and "edge" as well as "vertex" and "node" will be used interchangeably.

II. Minimizing Total Network Cost

The network cost is minimized if the number of common links traversed by the blue/red tree pair is maximized. It should be noted that a common link cannot be traversed by both trees in the *same* direction. Otherwise, a failure at this edge/link will disconnect all the downstream nodes from the root due to the disruption of both trees.

A. MinCost-E: surviving from any single link failure

We first formulate an ILP that minimizes the total network cost involved in finding a pair of blue/red trees to survive from any single link failure.

1) Notations:

We use (i,j) to denote an undirected edge connecting nodes *i* and *j*, and use $\langle i,j\rangle$ to denote a directed edge from node *i* to *j*. The notations used in MinCost-E are described as follows.

r_i : Binary variable. It equals to 1 if node *i* is the root of T^B and T^R, and 0 otherwise.

S : The given common root of T^B and T^R.

$e_b(i,j)$: Binary variable. It equals to 1 if edge $\langle i,j\rangle$ is on T^B, and 0 otherwise.

$e_r(i,j)$: Binary variable. It equals to 1 if edge $\langle i,j\rangle$ is on T^R, and 0 otherwise.

v_b^i : Auxiliary voltage value assigned to node *i* on T^B.

v_r^i : Auxiliary voltage value assigned to node *i* on T^R.

α : A small constant that serves as the minimum step of voltage increase along each tree; $1/\|V\| \geq \alpha > 0$.

$c_e(i,j)$: Binary variable. It equals to 1 if edge (i,j) is a common edge on both T^B and T^R, and 0 otherwise.

$pe_b(i,j,d)$: Binary variable. It equals to 1 if edge $\langle i,j\rangle$ is on the path from root to destination node *d* on T^B, and 0 otherwise.

$pe_r(i,j,d)$: Binary variable. It equals to 1 if edge $\langle i,j\rangle$ is on the path from root to destination node *d* on T^R, and 0 otherwise.

V : The set of all the nodes in the network.

E : The set of all the directed edges in the network.

2) ILP Formulation:

Given a network $G(V,E)$, the total cost of the two trees can be minimized by solving the ILP below (denoted as MinCost-E).

Objective:

$$Maximize \sum_{(i,j)\in E} c_e(i,j). \tag{1}$$

Subject to:

$$\sum_{i\in V} r_i = 1. \tag{2}$$

$$r_S = 1, \qquad S \in V. \tag{3}$$

$$e_b(i,j) + e_b(j,i) \leq 1, \qquad \forall\langle i,j\rangle \in E. \tag{4}$$

$$e_r(i,j) + e_r(j,i) \leq 1, \qquad \forall\langle i,j\rangle \in E. \tag{5}$$

$$\sum_{\langle i,j\rangle\in E} e_b(i,j) = 1 - r_j, \qquad \forall j \in V. \tag{6}$$

$$\sum_{\langle i,j\rangle\in E} e_r(i,j) = 1 - r_j, \qquad \forall j \in V. \tag{7}$$

$$v_b^j - v_b^i \geq \alpha e_b(i,j) - (1 - e_b(i,j)), \quad \forall\langle i,j\rangle \in E. \tag{8}$$

$$v_r^j - v_r^i \geq \alpha e_r(i,j) - (1 - e_r(i,j)), \quad \forall\langle i,j\rangle \in E. \tag{9}$$

$$e_b(i,j) + e_r(i,j) + e_b(j,i) + e_r(j,i) - 2c_e(i,j) \geq 0, \qquad \forall\langle i,j\rangle \in E. \tag{10}$$

$$e_b(i,j) + e_r(i,j) + e_b(j,i) + e_r(j,i) - c_e(i,j) \leq 1, \qquad \forall\langle i,j\rangle \in E. \tag{11}$$

$$e_b(i,j) + e_r(i,j) \leq 1, \qquad \forall\langle i,j\rangle \in E. \tag{12}$$

$$pe_b(i,j,j) = e_b(i,j), \qquad \forall\langle i,j\rangle \in E: j \neq S. \tag{13}$$

$$pe_r(i,j,j) = e_r(i,j), \qquad \forall\langle i,j\rangle \in E: j \neq S. \tag{14}$$

$$pe_b(i,S,d) = 0, \qquad \forall\langle i,S\rangle \in E, \; \forall d \in V: d \neq S. \tag{15}$$

$$pe_r(i,S,d) = 0, \qquad \forall\langle i,S\rangle \in E, \; \forall d \in V: d \neq S. \tag{16}$$

$$pe_b(i,j,k) + e_b(k,d) - 1 \leq pe_b(i,j,d), \qquad \forall\langle i,j\rangle, \forall\langle k,d\rangle \in E: j \neq S, k \neq S, d \neq S. \tag{17}$$

$$pe_r(i,j,k) + e_r(k,d) - 1 \leq pe_r(i,j,d), \qquad \forall\langle i,j\rangle, \forall\langle k,d\rangle \in E: j \neq S, k \neq S, d \neq S. \tag{18}$$

$$pe_b(i,j,d) + pe_r(i,j,d) + pe_b(j,i,d) + pe_r(j,i,d) \leq 1, \qquad \forall\langle i,j\rangle \in E, \forall d \in V: i \neq S, j \neq S, d \neq S. \tag{19}$$

3) Explanation:

Objective (1) aims at maximizing the number of common edges trarversed by the two trees. Constraints (2)-(9) ensure that both T^B and T^R are spanning trees with a common root.

Specifically, (2) and (3) require the two trees share a common root, (4) and (5) define directed edges on the trees, (6) and (7) require each non-root node to have a single ingress edge, and (8) and (9) prevent looping on T^B and T^R by assigning each child node a higher auxiliary voltage value than its parent.

Constraints (10) and (11) define the number of common edges trarvesred by the two trees $c_e(i,j)$. Constraint (12) ensures that any edge is *not* traversed by T^B and T^R in the same direction. Constraints (13)-(19) formulate the properties of the blue/red trees against single link failure. In particular, (13) and (14) indicate that if an edge is on a tree, then the path on the tree between the root and its ending node must pass through this edge. (15) and (16) indicate that the root cannot have any ingress edge on the trees. (17) and (18) say that, if there is a path on a tree connecting the root S and a node k via an intermediate edge $\langle i,j\rangle$, then we can find a path via the same edge $\langle i,j\rangle$ connecting S and a neighbor d of k if $\langle k,d\rangle$ is on the tree. Constraint (19) guarantees that the paths from the root to each destination node on T^B and T^R pass through different directed edges.

B. *MinCost-V: surviving from any single node failure*

1) Notations:

We now extend the MinCost-E ILP to provide node failure protection. We call it MinCost-V. For node protection, two variables $pv_b(i,d)$ and $pv_r(i,d)$ are defined as follows to replace $pe_b(i,j,d)$ and $pe_r(i,j,d)$ in MinCost-E, while all other variables are the same.

$pv_b(i,d)$: Binary variable. It equals to 1 if the intermediate node i is on the path from S to d on T^B, and 0 otherwise.

$pv_r(i,d)$: Binary variable. It equals to 1 if the intermediate node i is on the path from S to d on T^R, and 0 otherwise.

2) ILP Formulation:

MinCost-V reuses (1)-(12) from MinCost-E, but (13)-(19) are replaced by constraints (20)-(32) below.

$$pv_b(S,d)=1, \qquad \forall d\in V: d\neq S. \qquad (20)$$

$$pv_r(S,d)=1, \qquad \forall d\in V: d\neq S. \qquad (21)$$

$$pv_b(i,S)=0, \qquad \forall i\in V: i\neq S. \qquad (22)$$

$$pv_r(i,S)=0, \qquad \forall i\in V: i\neq S. \qquad (23)$$

$$pv_b(i,i)=1, \qquad \forall i\in V. \qquad (24)$$

$$pv_r(i,i)=1, \qquad \forall i\in V. \qquad (25)$$

$$pv_b(i,d)+pv_b(d,i)\leq 1, \quad \forall i,\forall d\in V: i\neq d. \qquad (26)$$

$$pv_r(i,d)+pv_r(d,i)\leq 1, \quad \forall i,\forall d\in V: i\neq d. \qquad (27)$$

$$pv_b(i,j)\geq e_b(i,j), \qquad \forall\langle i,j\rangle\in E. \qquad (28)$$

$$pv_r(i,j)\geq e_r(i,j),, \qquad \forall\langle i,j\rangle\in E. \qquad (29)$$

$$pv_b(k,i)+e_b(i,j)-1\leq pv_b(k,j), \quad \forall k\in V,\ \forall\langle i,j\rangle\in E: i\neq S, j\neq S. \qquad (30)$$

$$pv_r(k,i)+e_r(i,j)-1\leq pv_r(k,j), \quad \forall k\in V,\ \forall\langle i,j\rangle\in E: i\neq S, j\neq S. \qquad (31)$$

$$pv_b(i,d)+pv_r(d,i)\leq 1, \quad \forall i,\forall d\in V: i\neq S, d\neq S \qquad (32)$$

3) Explanation:

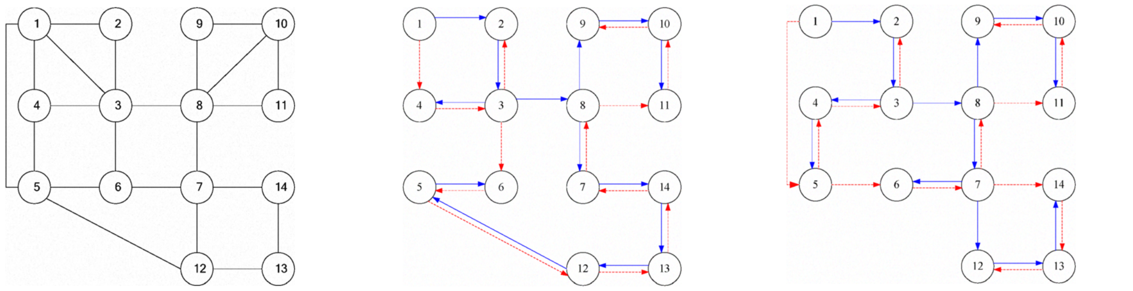
Since (1)-(12) are the same as those in MinCost-E, we only explain new constraints (20)-(32) below. Among them, (20)-(23) say that we can always find a path on T^B and T^R to connect the root to any other node but the root cannot be a destination. Constraints (24) and (25) stipulate that the root can be connected to any destination node by a path on the tree via the destination node itself. By (26) and (27), if node i is an intermediate node on the path from the root to another node d on T^B and T^R, then d cannot be an intermediate node on the path from the root to i on the same tree. Constraints (28) and (29) specify that if a directed edge $\langle i,j\rangle$ is on a tree (T^R or T^B), then there must be a path on the tree connecting the root to j via node i. Constraints (30) and (31) say that, if there is a path on a tree connecting the root S and a node i via an intermediate node k, then we can find a path on the tree via k connecting S and a neighbor j of i if edge $\langle i,j\rangle$ is on the tree. Finally, (32) ensures that the paths from the root to each destination node on T^B and T^R pass through different nodes.

III. Numerical Results And Discussion

Our ILPs are solved by CPLEX 11.0 on a Core2 2.66 GHz Windows PC with 2G bytes of memory. For comparison, the two heuristic algorithms proposed in [2] are implemented. They are F-GMFBT-E (for link protection) and F-GMFBT-V (for node protection).

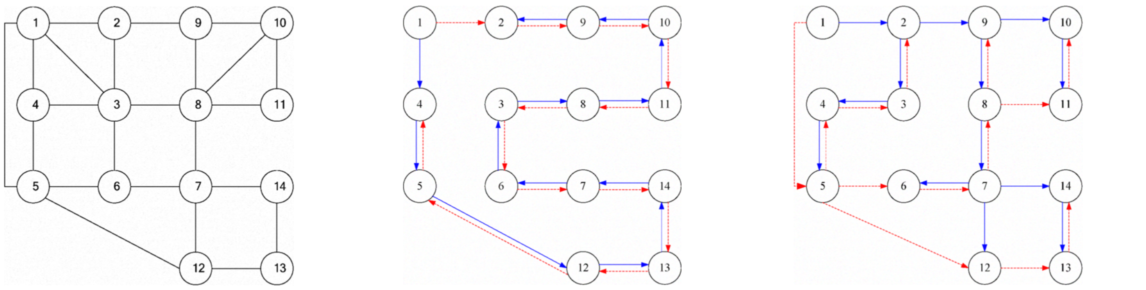
We first consider four benchmark networks. They are EN1 in Fig. 2(a) and EN2 in Fig. 3(a), as taken from [2], the pan European COST 239 in Fig. 4(a) (11 nodes and 26 links), and the high-speed Italian network in Fig. 5(a) (21 nodes and 36 links). Without loss of generality, we always take node 1 as the root. The blue/red tree solutions found using different algorithms are plotted in Figs. 2 to 5, where edges on T^B are blue/solid lines and edges on T^R are red/dashed lines, and edges not on the trees are removed for simplicity.

Tables I-III provide a concise summary, where the quality of our ILP-based solutions can be easily compared with the existing heuristic algorithms F-GMFBT-E and F-GMFBT-V. We can see that for larger network sizes and more complicated network topologies, our ILPs give more significant performance gain over the heuristics. From Table III, we can see that the running time of our ILPs is actually quite short.



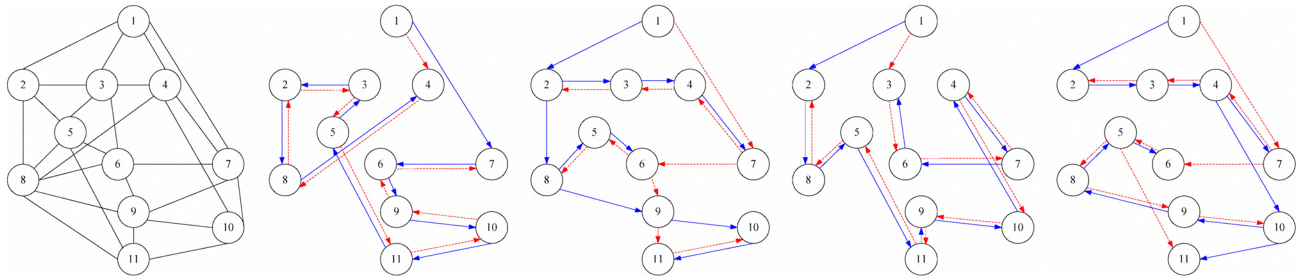
(a) EN1 in [2] (size: 14N22L). (b) Solution obtained from MinCost-E. (c) Solution obtained from F-GMFBT-E.

Fig. 2. Case study for network EN1. (Only link protection is considered).



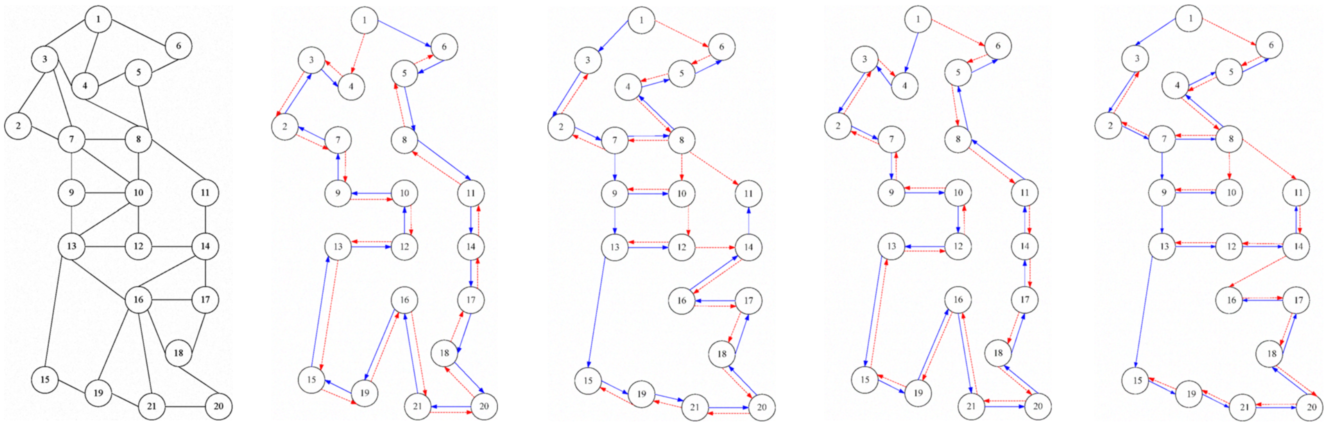
(a) EN2 in [2] (size: 14N23L). (b) Solution obtained from MinCost-V. (c) Solution obtained from F-GMFBT-V.

Fig. 3. Case study for network EN2. (Only node protection is considered).



(a) Pan European COST 239 network (size: 11N26L). (b) Solution obtained from MinCost-E. (c) Solution obtained from F-GMFBT-E. (d) Solution obtained from MinCost-V. (e) Solution obtained from F-GMFBT-V.

Fig. 4. Case study for the pan European COST 239 network. (Both link and node protections are considered).



(a) High-speed Italian network (size: 21N36L). (b) Solution obtained from MinCost-E. (c) Solution obtained from F-GMFBT-E. (d) Solution obtained from MinCost-V. (e) Solution obtained from F-GMFBT-V.

Fig. 5. Case Study for the high-speed Italian network. (Both link and node protections are considered).

TABLE I

Total Cost

Scheme	**Total cost on example networks (edges)**			
	EN1	EN2	pan European COST 239 network	high-speed Italian network
MinCost-E	16	-	11	21
F-GMFBT-E	17	-	14	25
MinCost-V	-	14	11	21
F-GMFBT -V	-	18	13	24

TABLE II

Percentage Improvement Over The Existing Heuristics

Scheme	**Percentage improved on example networks (%)**			
	EN1	EN2	pan European COST 239 network	high-speed Italian network
MinCost-E vs F-GMFBT-E	5.88	-	15.38	8.70
MinCost-V vs F-GMFBT-V	-	22.22	15.38	12.50

TABLE III

Solution Time

Scheme	**Solution time on example networks (sec.)**			
	EN1	EN2	pan European COST 239 network	high-speed Italian network
MinCost-E	1.47	-	101.11	313.45
MinCost-V	-	1.22	38.09	20.02

Next, we apply the heuristic algorithms in [2] and our ILPs on randomly generated networks. The network topologies have three different sizes: 10N20L (i.e. 10 nodes 20 links), 15N30L and 20N40L. For each network size, we generate 20 2-edge connected networks for link protection and 20 2-vertex connected networks for node protection.

Table IV gives the average network cost of the rblue/red tree solution found using different tree construction algorithms, F-GMFBT-E, F-GMFBT-V, MinCost-E and MinCost-V. We can see that our ILPs, MinCost-E and MinCost-V, cut down the solution cost ranging from 11-15%. On the other hand, Table V lists the average solution time of our ILPs, which ranges from 3.16 to 1358.78 seconds.

TABLE IV

Average Total Cost

Scheme	**Average total cost for three network sizes (edges)**		
	10N20L	15N30L	20N40L
MinCost-E	10	15.05	20.25
F-GMFBT-E	11.55	17.4	24.05
MinCost-V	10	15	20.05
F-GMFBT -V	11.25	17	22.65

TABLE V

Average Solution Time

Scheme	**Average solution time for three network sizes (sec.)**		
	10N20L	15N30L	20N40L
MinCost-E	13.80	397.05	1358.78
MinCost-V	3.16	60.16	397.40

IV. Conclusion

We formulated two efficient ILPs, MinCost-E and MinCost-V, to construct a pair of survivable broadcast trees with the same root. Upon any single link or node failure, the root is still connected to any other node in the network by at least one tree. MinCost-E and MinCost-V minimize the total cost as required. For medium-size networks, our ILPs can generate optimal solutions in a reasonable amount of running time, with much higher solution quality than the existing heuristic counterparts.

References

[1] M. Médard, S. G. Finn, R. A. Barry and R. G. Gallager, "Redundant Trees for Preplanned Recovery in Arbitrary Vertex-Redundant or Edge-Redundant Graphs," *IEEE/ACM Transactions on Networking*, vol. 7, no. 5, pp. 641-652, Oct. 1999.

[2] W. Zhang, G. Xue, J. Tang and Krishnaiyan Thulasiraman, "Faster Algorithms for Construction of Recovery Trees Enhancing QoP and QoS," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 642-655, Jun. 2008.

[3] G. Xue, L. Chen and K. Thulasiraman, "Quality-of-Service and Quality-of-Protection Issues in Preplanned Recovery Schemes Using Redundant Trees," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 8, pp. 1332-1345, Oct. 2003.

[4] S. G. Finn, M. Médard and R. A. Barry, "A Novel Approach to Automatic Protection Switching Using Trees," *IEEE ICC '97*, vol. 1, pp. 272-276, Jun. 1997.

[5] A. Itai and M. Rodeh, "The Multi-Tree Approach to Reliability in Distributed Networks," *25th Annual Symposium on Foundations of Computer Science*, pp. 137-147, Oct. 1984.

[6] G. Ellinas and T. E. Stern, "Automatic Protection Switching for Link Failures in Optical Networks with Bi-directional Links," *IEEE GLOBECOM '96*, vol. 1, pp. 152-156, Nov. 1996.

[7] D. Habibi, V. Q. Phung and H. N. Nguyen, "Minimizing the Number of Constraints for Shared Backup Path Protection (SBPP) in Shared Risk Link Group (SRLG) Optical Mesh Networks," *IEEE ICON '07*, pp. 246-251, Nov. 2007.

[8] B. Wu, K. L. Yeung and P.-H. Ho, "ILP Formulations for *p*-Cycle Design without Candidate Cycle Enumeration," minor revision in *IEEE/ACM Transactions on Networking* (in Oct. 2008). Available at http://www.eee.hku.hk/ research/doc/tr/TR2008001_IFDCC.pdf.

[9] T. Y. Chow, F. Chudak and A. M. Ffrench, "Fast Optical Layer Mesh Protection Using Pre-Cross-Connected Trails," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 539-548, Jun. 2004.

[10] B. Wu, P.-H. Ho, K. L. Yeung, J. Tapolcai and H. T. Mouftah, "CFP: Cooperative Fast Protection," *IEEE INFOCOM '09* (Miniconference), April 2009.

[11] S. Ramamurthy and B. Mukherjee, "Survivable WDM Mesh Networks, Part I-Protection," *IEEE INFOCOM '99*, vol. 2, pp. 744-751, March 1999.

[12] S. Ramamurthy and B. Mukherjee, "Survivable WDM Mesh Networks, Part II-Restoration," *IEEE ICC '99*, vol. 3, pp. 2023-2030, June 1999.

[13] V. Anand and C. Qiao, "Dynamic Establishment of Protection Paths in WDM Networks, Part I," *IEEE ICCCN '00*, pp. 198-204, Oct. 2000.

[14] A. Jukan, A. Monitzer, G. de Marchis and R. Sabella (ed.), "Restoration Methods for Multi-Service Optical Networks," *Optical Networks: Design and Modelling*, Kluwer Academic Publisher, 1998, pp. 3-12.

[15] B. Mukherjee, "WDM Optical Communication Networks: Progress and Challenges," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1810-1824, Oct. 2000.