

Spare Capacity Allocation in Multi-Layer Networks

Yu Liu[†], David Tipper[‡], Korn Vajanapoom[‡]

[†]OPNET Technologies, Inc.

200 Regency Forest Drive, Cary, North Carolina 27511, USA

Email: yliu@opnet.com

[‡]Department of Information Science and Telecommunications
University of Pittsburgh, Pittsburgh, Pennsylvania 15260, USA

Email: tipper@tele.pitt.edu, kov2@pitt.edu

Abstract—In this paper, we consider the problem of provisioning spare capacity in multi-layer backbone networks in order to meet survivability requirements. A matrix based model is presented showing how failure propagation can be mapped across network layers. Two different multi-layer spare capacity allocation optimization problems are formulated utilizing the failure propagation matrix to determine the location and amount of spare capacity in each network layer. In order to scale the models a fast and efficient approximation algorithm based on our early successive survivable routing (SSR) technique is developed. Numerical results for a variety of networks show that near optimal solutions are found by the proposed heuristic algorithm.

I. INTRODUCTION

Survivability in the face of failures has become an essential property of backbone transport networks. Current backbone data networks are converging towards a two layer architecture of IP/MPLS or GMPLS over an optical transport layer. Typically, each layer of a network will need self-healing capabilities to address faults occurring in their own layer. One of the major problems in the survivability of such multi-layer networks is *failure propagation*, which occurs when the failure of a bottom-layer link or node results in the simultaneous failure of multiple top-layer links. If failure propagation is not considered appropriately in multi-layer networks, the critical communication services could be interrupted upon failure. Several approaches have been proposed to design survivable virtual topologies in the top layer [1], [2], including work focused on backup path routing [3], and working paths optimization [4], [5] on two layer networks [6], [7].

In this paper, we provide two models for the spare capacity allocation (SCA) problem in multi-layer networks using fault independent path restoration. The first model captures failure propagation by extending the

matrix-based SCA formulation in [8]. The second model further improves the first one to increase the amount of spare capacity sharing. Modifications to the basic models for failure-dependent path restoration and the stub release cases are provided. Finally numerical results are given showing that the proposed algorithm can find near optimal solutions.

II. THE SPARE CAPACITY ALLOCATION MODEL

In this section, the spare capacity allocation (SCA) problem is formulated following [8], using failure independent (FID) path restoration for an arbitrary failure condition. FID is also called path restoration with disjoint routes, where a backup path is always disjoint from its working path. We assume all traffic flows require a 100% restoration level for any failure. This level of restoration requires that all affected flows be detoured to their backup paths upon any given failure. Provisioning enough spare capacity is the prerequisite condition to such restoration.

We represent an uncapacitated network by a directed graph of N nodes, L links, and R flows. A flow r , $1 \leq r \leq R$ is specified by its origin/destination node pair $(o(r), d(r))$ and traffic demand m_r . Working and backup paths of flow r are represented by two $1 \times L$ binary row vectors $\mathbf{p}_r = \{p_{rl}\}$ and $\mathbf{q}_r = \{q_{rl}\}$ respectively. The l -th element in one of the vectors equals to one if and only if (iff) the corresponding path uses link l . The path link incidence matrices for working and backup paths are the collections of these vectors, forming two $R \times L$ matrices $\mathbf{P} = \{\mathbf{p}_r\}$ and $\mathbf{Q} = \{\mathbf{q}_r\}$ respectively. Let $\mathbf{M} = \text{Diag}(\{m_r\}_{R \times 1})$ denote the diagonal matrix representing the demands of each flow.

The topology is given by the node-link incidence matrix $\mathbf{B} = (b_{nl})_{N \times L}$ where $b_{nl} = 1$ or -1 if and only if node n is the origin or destination node of link l . $\mathbf{D} = (d_{rn})_{R \times N}$ is the flow node incidence matrix

where $d_{rn} = 1$ or -1 iff $o(r) = n$ or $d(r) = n$. We characterize K failure scenarios in a binary matrix $\mathbf{F} = \{\mathbf{f}_k\}_{K \times 1} = \{f_{kl}\}_{K \times L}$. The row vector \mathbf{f}_k in \mathbf{F} is for failure scenario k and its element f_{kl} equals one iff link l fails in scenario k . In this way, each failure scenario includes a set of one or more links that will fail simultaneously in the scenario. For a failed node, all its adjacent links are marked as failed. We also denote a flow failure incidence matrix $\mathbf{U} = \{\mathbf{u}_r\}_{R \times 1} = \{u_{rk}\}_{R \times K}$, where $u_{rk} = 1$ iff flow r will be affected by failure k , and $u_{rk} = 0$ otherwise. A flow tabu-link matrix $\mathbf{T} = \{\mathbf{t}_r\}_{R \times 1} = \{t_{rl}\}_{R \times L}$ has $t_{rl} = 1$ iff the backup path of flow r should not use link l , and $t_{rl} = 0$ otherwise. We can find \mathbf{U} and \mathbf{T} given \mathbf{P} and \mathbf{F} as shown in (7) and (8) respectively. Note that, a binary matrix multiplication operation “ \odot ” is used in these two equations. It modifies the general addition in $1 + 1 = 2$ to Boolean addition in $1 + 1 = 1$ [9]. Using this binary operator, the complicated logical relations among link, path and failure scenarios are simplified into two matrix operations.

We let $\mathbf{G} = \{g_{lk}\}_{L \times K}$ denote the *spare provision matrix* whose elements g_{lk} are the minimum spare capacity required on link l when link k fails. Note that $K = L$ when protecting all single link failures. Given the backup paths \mathbf{Q} , demand bandwidth matrix \mathbf{M} , and working path \mathbf{P} , the spare provision matrix can be determined as in (3). The minimum spare capacity required on each link is denoted by the column vector $\mathbf{s} = \{s_l\}_{L \times 1}$ which is found in (2). The function \max in (2) asserts that an element in \mathbf{s} is equal to the maximum element in the corresponding row of \mathbf{G} . It is equivalent to $\mathbf{s} \geq \mathbf{G}$ in this optimization model, where the operator \geq between a column vector \mathbf{s} and a matrix \mathbf{G} guarantees that any element in \mathbf{s} is always not less than any elements in the corresponding row of \mathbf{G} . Let ϕ_l denote the cost function of spare capacity on link l . $\boldsymbol{\phi} = \{\phi_l\}_{L \times 1}$ is a column vector of these cost functions and $\boldsymbol{\phi}(\mathbf{s})$ gives the cost vector of the spare capacities on all links. The total cost of spare capacity on the network is $\mathbf{e}^T \boldsymbol{\phi}(\mathbf{s})$, where \mathbf{e} is unit column vector of size L . For simplicity, in this paper we assume all cost functions are identity functions.

Given the notation and definitions above the spare capacity allocation (SCA) problem can be formulated as follows.

$$\min_{\mathbf{Q}, \mathbf{s}} \quad S = \mathbf{e}^T \mathbf{s} \quad (1)$$

$$\text{s.t.} \quad \mathbf{s} = \max \mathbf{G} \quad (2)$$

$$\mathbf{G} = \mathbf{Q}^T \mathbf{M} \mathbf{U} \quad (3)$$

$$\mathbf{T} + \mathbf{Q} \leq 1 \quad (4)$$

$$\mathbf{Q} \mathbf{B}^T = \mathbf{D} \quad (5)$$

$$\mathbf{Q} : \text{binary} \quad (6)$$

$$\mathbf{U} = \mathbf{P} \odot \mathbf{F}^T \quad (7)$$

$$\mathbf{T} = \mathbf{U} \odot \mathbf{F} \quad (8)$$

The SCA problem has the objective to minimize the total spare capacity in (1) with the constraints (2)–(8). The decision variables are the backup path selection (i.e., backup path matrix) \mathbf{Q} and spare capacity allocation (i.e., the vector \mathbf{s}). Constraints (2) and (3) associates these variables, i.e., the spare capacity allocation \mathbf{s} is derived from the backup paths in \mathbf{Q} . Constraint (4) guarantees that backup paths will not use any link which might fail simultaneously with their working paths. Flow conservation constraint (5) guarantees that backup paths given in \mathbf{Q} are feasible paths of flows in a directed network. Note, the incidence matrices \mathbf{U} and \mathbf{T} are precomputed to indicate the failure cases that will influence the working paths, and the links that should be avoided in the backup paths respectively. A more detailed explanation of the model is in [8, eq.(7)-(14)].

The SCA model as formulated above is a mixed integer programming problem which is NP-complete. Hence, solving the problem for large networks is infeasible using standard integer programming solution methods. In [8] we proposed the successive survivable routing (SSR) heuristic algorithm to solve the SCA problem. The SSR algorithm finds near optimal solutions by routing backup paths iteratively. Each backup path computation uses the shortest path algorithm. The link routing metric is the *incremental spare capacity*. It is computed from the most recent spare provision matrix that is further based on the previous routed backup path. After all flows find their backup paths, SSR continues to update existing backup paths whenever a new one could use less spare capacity. This process keeps reducing total spare capacity until it converges, (i.e., no more backup path updates). Different random ordering of the flows for routing backup paths are used to provide various solutions and avoid local minima. The best ordering is the final near optimal solution [8].

III. SCA MULTI-LAYER MODELS

Here we consider a two layer network and extend the SCA model. In the top layer network, the notation of the previous section is reused, and the same notation with the superscript “ b ” is used for the bottom layer. A top-layer link is carried by a bottom-layer path. The overlay

information is defined by the interlayer link incidence matrix $\mathbf{H} = \{h_{ij}\}_{L \times L^b}$, where $1 \leq i \leq L$, $1 \leq j \leq L^b$. Element h_{ij} equals to one *iff* the top-layer link i passes the bottom-layer link j . Given the top-layer spare capacity allocation vector \mathbf{s} , its equivalent bottom-layer spare capacity vector \mathbf{s}^b is below:

$$\mathbf{s}^b = \mathbf{H}^T \mathbf{s}. \quad (9)$$

The interlayer link incidence matrix \mathbf{H} should guarantee that a failure of any single bottom-layer link will not partition the top layer topology. This property is called *immunity from failure propagation*. A math programming model to find such a mapping \mathbf{H} is provided in [2]. Its matrix format is called the survivable topology layout problem in [10, Section 6.2]. The overlay information \mathbf{H} is the input to the following SCA models.

First we consider the SCA problem for the failure-independent (FID) path restoration case. In FID each flow has a single backup path disjoint from any failures that affect its working path. Based on the layers where the restoration scheme exists, the SCA models could be classified as follows.

A. Restoration at the bottom layer

If the restoration only happens at the bottom layer, each top-layer *link* has a bottom-layer backup path preplanned besides its working path. The top-layer traffic flow is not aware of the bottom-layer restoration. This scheme is simple and provides fast restoration. However, it might not protect all top-layer failures such as the failure of a top-layer router or its interfaces. The SCA model for the overlay network here is simplified to one at the bottom layer only. The only modification beyond [8] is that the bottom-layer working paths are derived from the layout information \mathbf{H} in (10).

$$\mathbf{P}^b = \mathbf{H} \quad (10)$$

B. Restoration at the top layer

When the restoration is at the top layer, equations (1)-(8) in the previous section are extended based on the two alternate used of the overlay information \mathbf{H} . We denote the two models proposed as [A] and [B] superscripts added to key SCA model components that are changed in the multi-layer model.

Model A: *Use of overlay information for failure propagation:* To protect against failure propagation of any single bottom layer link failures, the overlay information \mathbf{H} is used to derive the failure scenario matrix \mathbf{F} for the top layer SCA model in (11). The flow failure

incidence matrix \mathbf{U} and the spare provision matrix $\mathbf{G}^{[A]}$ are modified in (12) and (13) to replace (7) and (3).

$$\mathbf{F} = \mathbf{H}_f^T = \mathbf{F}^b \odot \mathbf{H}^T \quad (11)$$

$$\mathbf{U} = \mathbf{P} \odot \mathbf{F}^T = \mathbf{P} \odot \mathbf{H} \odot \mathbf{F}^{bT} \quad (12)$$

$$\mathbf{G}^{[A]} = \mathbf{Q}^T \mathbf{M} \mathbf{U} = \mathbf{Q}^T \mathbf{M} (\mathbf{P} \odot \mathbf{H} \odot \mathbf{F}^{bT}) \quad (13)$$

In addition, the objective function to minimize the total spare capacity (1) is replaced by (14), where $\phi^T = \mathbf{e}^T \mathbf{H}^T$ is used to compute the actual spare capacity on the bottom layer reserved by the top-layer links.

$$\min_{\mathbf{Q}} S^{[A]} = \mathbf{e}^T \mathbf{s}^b = \mathbf{e}^T \mathbf{H}^T \mathbf{s} = \phi^T \max \mathbf{G}^{[A]} \quad (14)$$

Model B: *Use the overlay information for both failure propagation and cross-layer spare capacity reservation:* In addition to using the overlay information for failure propagation, the second model computes spare capacity sharing with a finer granularity. It keeps track of spare capacity sharing at the bottom layer, instead of at the top. Since every top link passes one or more bottom links, spare capacity sharing on these bottom links could provide the finer granularity than on the top links. This could further minimize the total spare capacity. The disadvantage of this model is the requirement of an approach for the top layer to record and reserve spare capacity on the bottom-layer links. This might require complicated cross-layer protocols in the control plane.

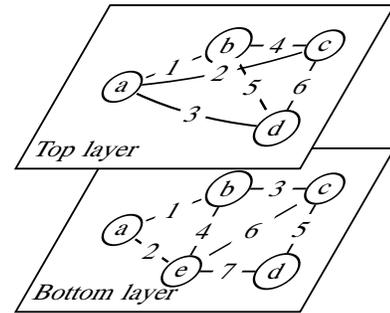


Fig. 1. Network 0: 5-node overlay network

An example is given to illustrate the advantage of the finer granularity. In Fig. 1, two working flows $a-b$ and $c-d$ at the top-layer pass the bottom-layer links 1 and 5 respectively. Their backup paths might be $a-c-b$ and $c-b-a-d$. They pass the top-layer links 2 ($a-c$) and 3 ($a-d$) respectively. The bottom-layer paths of these two links have overlapped on the bottom-layer link 2 ($a-e$) whose spare capacity could be shared by the above backup paths. In this example, the top-layer not only uses the

overlay information \mathbf{H} to avoid the failure propagation, but also to reserve the spare capacity shared at the bottom layer in order to achieve lower redundancy.

This spare capacity sharing scheme is equivalent to: (i) converting all working and backup paths at the top layer into ones at the bottom by multiplexing them with \mathbf{H} , as modeled in (9); (ii) then minimizing the objective function in (15).

$$\min_{\mathbf{Q}} S^{[B]} = \mathbf{e}^T \mathbf{s}^b = \mathbf{e}^T \max(\mathbf{H}^T \mathbf{G}^{[A]}) \quad (15)$$

Since $\max(\mathbf{H}^T \mathbf{G}^{[A]}) \leq \mathbf{H}^T \max(\mathbf{G}^{[A]})$, the total spare capacity using Model B will be equal to or smaller than that in Model A.

$$S^{[B]} \leq S^{[A]} \quad (16)$$

This model also requires modifications in the successive survivable routing (SSR) algorithm in [8]. Since the spare provision matrix $\mathbf{G}^{[A]}$ is converted to its bottom layer equivalent in (15), the incremental spare capacity vector \mathbf{v}_r used at the top layer in [8, Eq. (17)] has to be derived from its equivalent \mathbf{v}_r^b at the bottom layer in (17) below.

$$\mathbf{v}_r = \mathbf{H} \mathbf{v}_r^b, \quad (17)$$

where $\mathbf{v}_r^b = \mathbf{s}^{b*}(\mathbf{e} - \mathbf{t}_r^b) - \mathbf{s}^{b-r}$. The definitions of \mathbf{s}^{b*} and \mathbf{s}^{b-r} are the bottom layer equivalents of \mathbf{s}^* and \mathbf{s}^{-r} whose definitions are in step 3 of the SSR section in [8, Section V].

C. Restoration at both Layers

When both layers have path restoration schemes implemented and share their spare capacity, they might achieve better redundancy. This concept is called the *common pool survivability* in [11]. Assuming both layers are immune from single bottom-layer link failure, the spare capacity sharing between both layers could be done if their spare provision matrices are exchanged. The top-layer spare provision matrix $\mathbf{G}^{[A]}$ will be transformed and merged with the spare provision matrix \mathbf{G}^b at the bottom layer in (18).

$$\mathbf{G}^{[C]} = \mathbf{G}^b + \mathbf{H}^T \mathbf{G}^{[A]} \quad (18)$$

The objective function for the SCA problem is updated to (19).

$$\begin{aligned} \min_{\mathbf{Q}, \mathbf{Q}^b} S^{[C]} &= \mathbf{e}^T \mathbf{s}^b = \mathbf{e}^T \max \mathbf{G}^{[C]} \\ &= \mathbf{e}^T \max(\mathbf{G}^b + \mathbf{H}^T \mathbf{G}^{[A]}) \\ &\leq \mathbf{e}^T \max \mathbf{G}^b + \mathbf{e}^T \max(\mathbf{H}^T \mathbf{G}^{[A]}) \\ &= S^b + S^{[B]} \end{aligned} \quad (19)$$

In the SSR algorithm, both layers perform their single layer SSR algorithms but their link metrics are different. The top layer has the knowledge of \mathbf{H} and uses it to convert its bottom-layer link metric \mathbf{v}_r^b back to \mathbf{v}_r using (17). In this way, both layers share a common spare capacity provision matrix $\mathbf{G}^{[C]}$. They cooperate to further improve spare capacity sharing.

D. failure-dependent path restoration at the top layer

All of the models above assume failure-independent (FID) path restoration. The SCA problem for the failure-dependent (FD) path restoration is given in [12]. Its extension for the multi-layer networks is below.

We use Model A where the overlay information \mathbf{H} is used for the failure propagation. As shown in (11), any arbitrary bottom-layer failure has been captured in failure matrix \mathbf{F} . Similar to (14) in [12], in order to compute backup paths for individual failures, equation (13) that finds the spare provision matrix $\mathbf{G}^{[A]}$ should be replaced by $\mathbf{G}^{[D]}$ in (20).

$$\mathbf{G}_k^{[D]} = \mathbf{Q}^{kT} \mathbf{M} \mathbf{U}_k, \quad 1 \leq k \leq K, \quad (20)$$

The k -th column vector $\mathbf{G}_k^{[D]} = \{g_{lk}\}_{L \times 1}$ of $\mathbf{G}^{[D]}$ is determined by the k -th column vector $\mathbf{U}_k = \{u_{rk}\}_{R \times 1}$ of the failure matrix \mathbf{U} in (12), the demand matrix \mathbf{M} , and the backup path matrix \mathbf{Q}^k .

IV. NUMERICAL RESULTS

The SCA models and the SSR algorithm were studied on eight different multi-layer networks. The top and bottom layer topologies are provided in Fig. 1–8. Two cases of top layer topologies were studied: full mesh and partial mesh. In the full mesh case all top layer nodes were interconnected in a full mesh. In the partial mesh case the top layer has a sparser interconnection and the topology is given in Fig. 1–8. The number of links and nodes and each layer are provided in Table I. We first consider the failure-independent (FID) path restoration case for both Models A and B.

Two algorithms, the branch and bound (BB) and the successive survivable routing (SSR), are used to find the solutions for Model A and B in Table I. The total spare capacity S from both models are listed in the results and the total working capacity W is given for comparison purposes. Since SSR finds the near optimal solutions, a range of solutions from 64 random cases are listed. BB results are obtained from the commercial software CPLEX that could find the optimal solution for small networks.

TABLE I
RESULTS OF TOTAL SPARE CAPACITY ALLOCATIONS AND THE CPU TIME IN SECOND

Net	$ N^b $	$ L^b $	Top Layer Topology	$ N $	$ L $	$ R $	W	$S^{[A]}$		$S^{[B]}$		Time of $S^{[A]}$		Time of $S^{[B]}$	
								BB	SSR	BB	SSR	BB	SSR	BB	SSR
Net 0	5	7	Full mesh	4	6	6	9	11	11-13	11	11	≈ 0	≈ 0	≈ 0	≈ 0
Net 1	10	22	Full mesh	6	15	15	27	15	16-22	15	16-22	0.07	1.67	0.08	2.36
			Partial mesh	6	9	15	34	32	32-35	31	32-33	0.03	1.63	0.04	2.06
Net 2	12	25	Full mesh	7	21	21	39	28	30-37	28	29-35	4.09	1.78	0.54	3.63
			Partial mesh	7	12	21	54	45	46-50	45	46-50	0.06	1.73	0.07	2.79
Net 3	13	23	Full mesh	8	28	28	63	39	40-53	37	40-49	29.37	2.01	1.09	5.7
			Partial mesh	8	14	28	72	49	49-57	48	49-57	0.1	1.8	0.17	3.46
Net 4	17	31	Full mesh	10	45	45	106	48	56-68	48	54-68	88.04	3.03	11.98	20.55
			Partial mesh	10	16	45	136	97	98-103	97	98-104	0.34	2.01	0.29	7.12
Net 5	18	27	Full mesh	10	45	45	134	121	129-141	118	124-131	56.53	2.91	64.49	13.67
			Partial mesh	10	18	45	157	114	115-125	114	115-123	0.55	2.02	30.36	6.78
Net 6	23	33	Full mesh	10	45	45	157	121	132-147	117	125-135	1.145	3.01	290	23.52
			Partial mesh	10	22	45	184	162	164-172	160	161-170	1.12	2.16	0.71	9.99
Net 7	26	30	Full mesh	8	28	28	116	103	103-130	102	105-111	1.07	2.01	3.54	7.14
			Partial mesh	8	13	28	126	100	100-108	100	100-111	0.34	1.77	1.37	4.64

Note: Branch and Bound (BB) results are from AMPL/CPLEX v9.0 on a Sun Fire V240 Server with 1GHz CPU and 2GByte memory. SSR results are from a PC with Intel Pentium M 1.3GHz CPU.

All demands were unit bandwidth requests at the top layer. The demand matrices are given as $M = \mathbf{I}_{|R| \times |R|}$, where $|R|$ is the number of flows at the top layer, $|R| = |N| * (|N| - 1)/2$ for full meshed demands, and $|N|$ is the number of top layer nodes. The working paths are given from the shortest-hop paths. The total work capacity at the bottom layer can be derived by $W = e^T P H e$. Assume all flows and links are symmetric so the problem can be modeled in an undirected graph. The bottom layer failure matrix $F^b = \mathbf{I}_{|L^b| \times |L^b|}$, where $|L^b|$ is the number of the bottom-layer links.

From the results in Table I, one can see that the SSR algorithm closely approximates the optimal BB solution. In network 0, both BB and SSR find the optimal solutions for both models. In the other 7 networks, the SSR results using Model B have a slightly smaller range for the solution than those found from Model A. This indicates the spare capacity can be reduced in Model B, but the reduction is very small – less than 5% when using Model B solved by either BB or SSR.

The CPU time for BB to find the optimal solution increases from significantly when the network size increases. On the contrary, SSR can find all 64 solutions in less than 30 seconds. Because BB does not scale well, SSR could be a good alternative for large networks.

Next, we use Model A and SSR to compare path restoration schemes, i.e. FID, FD, and FD with Stub Release, to protect either single link or single node failures. Model B is not used in this comparison because of its requirement of a complicated cross-layer capacity

reservation protocol and its little gain in the total spare capacity in the previous results.

The SCA results on Network 5 for the failure-dependent (FD) path restoration are shown in Table II. The FD path restoration finds a little bit lower values for the total spare capacity. This is reasonable since multiple backup paths are allowed for each flow at the top layer. This increases the chance of sharing spare capacity, hence reduces redundancy. Using the FD path restoration scheme with the stub release function (FD-StubR), the total spare capacity can be further reduced. The stub release feature allows backup path to reuse the working capacity released by the working paths that are interrupted by the failure. The model of stub release is discussed in [10, Ch. 6].

Furthermore, the SCA problem for protecting any single node failures at the bottom layer could be derived by combining Model A and the Node Failure extension in [8]. The results are compared for the above three path restoration methods, FID, FD and FDStubR, in Table II. FDStubR could provide the lowest redundancy while FD is slightly higher than FDStubR but much lower than the FID results.

It is important to be aware that the lower spare capacity values in the node failure does not indicate better efficiency. The lower value could come from the dropped demands that cannot be recovered at the failure of their end nodes. On the contrary, the link failure can guarantee a 100% demand to be recovered on 2-connected networks.

TABLE II
COMPARISON OF $S^{[A]}$ FOR PATH RESTORATIONS IN FIG. 6

Restoration	Link failure	Node failure
FID	115–125	106–115
FD	104–118	97–109
FDSStubR	103–115	94–107

V. CONCLUSION

In this paper, several variations of the SCA problem for multi-layer networks were formulated as integer programming problems. Numerical results were given showing that the successive survivable routing algorithm can be used to efficiently find near optimal spare capacity solutions.

REFERENCES

- [1] O. Crochat, J.-Y. Le Boudec, and O. Gerstel, "Protection interoperability for WDM optical networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 3, pp. 384–395, June 2000.
- [2] E. Modiano and A. Narula-Tam, "Survivable routing of logical topologies in WDM networks," in *Proceeding of IEEE INFOCOM*, Apr. 2001.
- [3] S. De Maesschalck, D. Colle, and et al, "Intelligent optical networking for multilayer survivability," *IEEE Communications Magazine*, pp. 42–49, Jan. 2002.
- [4] M. Kodialam and T.V. Lakshman, "Integrated dynamic IP and wavelength routing in IP over WDM networks," in *Proceeding of IEEE INFOCOM*, Apr. 2001.
- [5] Wayne Grover, *Mesh-based Survivable Networks: Options and Strategies for Optical, MPLS, and ATM Networking*, Prentice Hall PTR, 2003.
- [6] J. Vasseur, M. Pickavet, and P. Demeester, *Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP and MPLS*, Morgan Kaufmann MKP, 2004.
- [7] M. Pioro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*, Morgan Kaufmann MKP, 2004.
- [8] Y. Liu, D. Tipper, and P. Siripongwutikorn, "Approximating optimal spare capacity allocation by successive survivable routing," *IEEE/ACM Transactions on Networking*, vol. 13, no. 1, pp. 198–211, Feb. 2005.
- [9] Bernard Kolman, Robert C Busby, and Sharon Ross, *Discrete Mathematical Structures*, Prentice Hall, 1996.
- [10] Yu Liu, *Spare capacity allocation method, analysis and algorithm*, Ph.D. dissertation, School of Information Sciences, University of Pittsburgh, 2001, <http://www.sis.pitt.edu/yliu/dissertation/>.
- [11] P. Demeester and M. Gryseels, "Resilience in multilayer networks," *IEEE Communications Magazine*, vol. 37, no. 8, pp. 70–76, 8 1999.
- [12] Y. Liu and D. Tipper, "Spare capacity allocation for non-linear cost and failure-dependent path restoration," Third International Workshop on Design of Reliable Communication Networks (DRCN), Budapest, Hungary, October 7–10 2001.

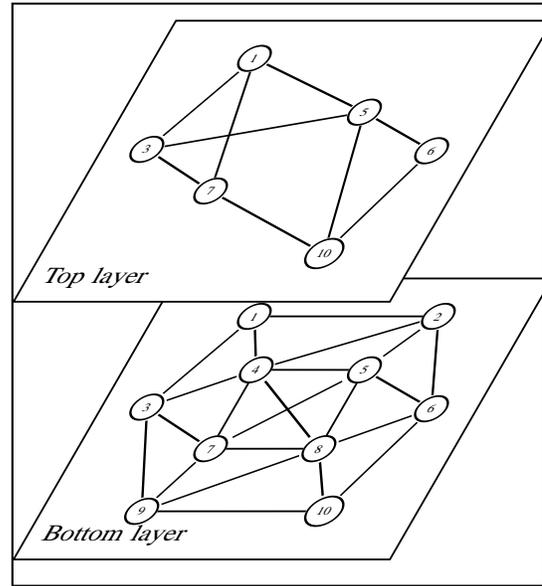


Fig. 2. Net 1 ($N = 6, L = 9, N^b = 10, L^b = 22$)

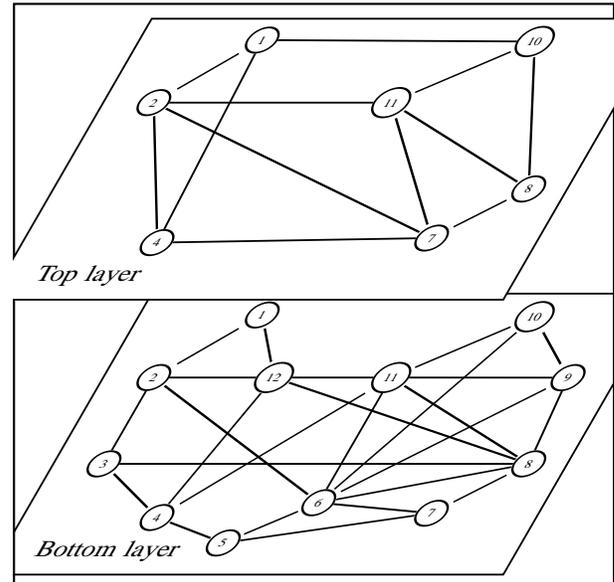


Fig. 3. Net 2 ($N = 7, L = 12, N^b = 12, L^b = 25$)

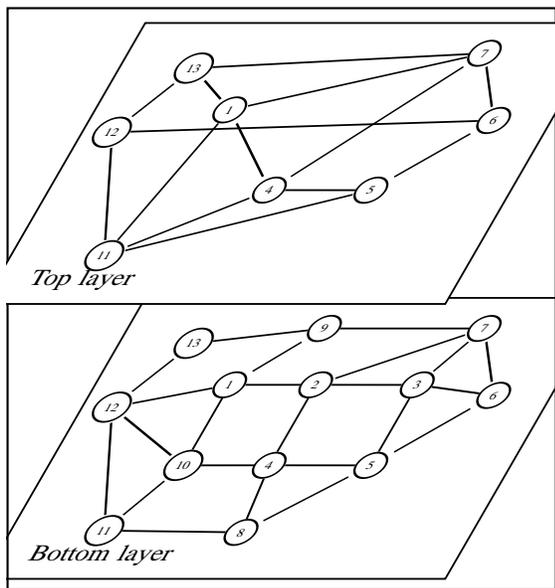


Fig. 4. Net 3 ($N = 8, L = 14, N^b = 13, L^b = 23$)

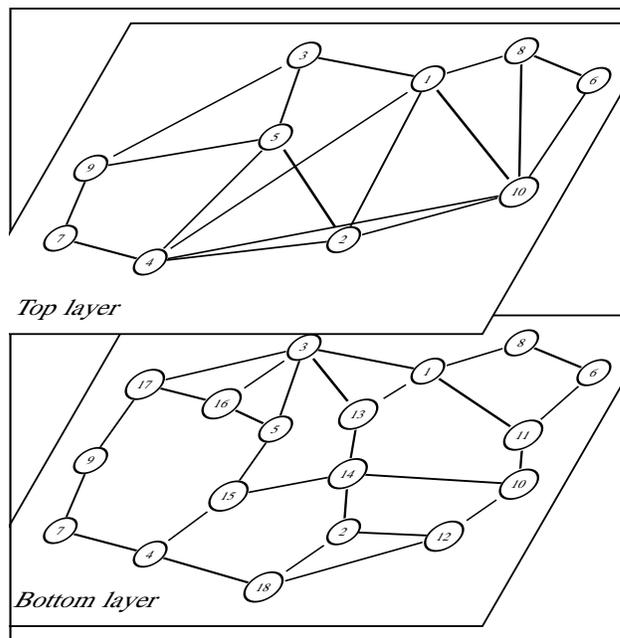


Fig. 6. Net 5 ($N = 10, L = 18, N^b = 18, L^b = 27$)

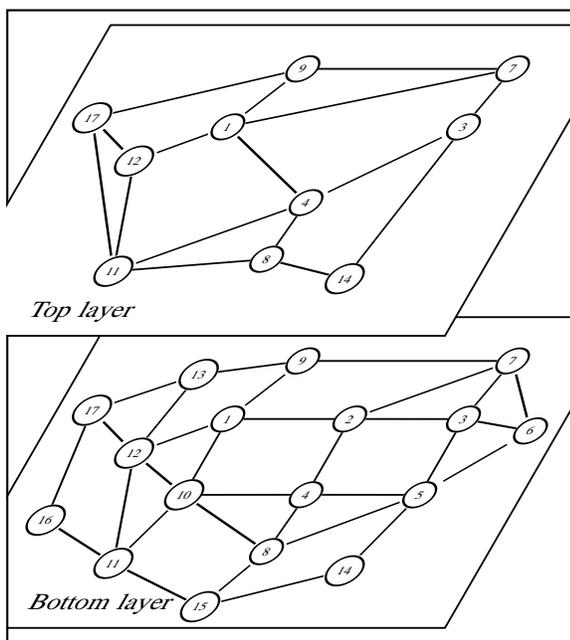


Fig. 5. Net 4 ($N = 10, L = 16, N^b = 17, L^b = 31$)

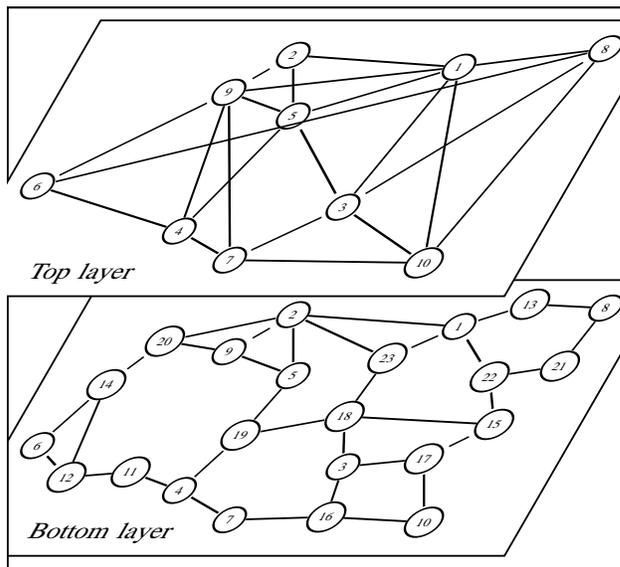


Fig. 7. Net 6 ($N = 10, L = 22, N^b = 23, L^b = 33$)

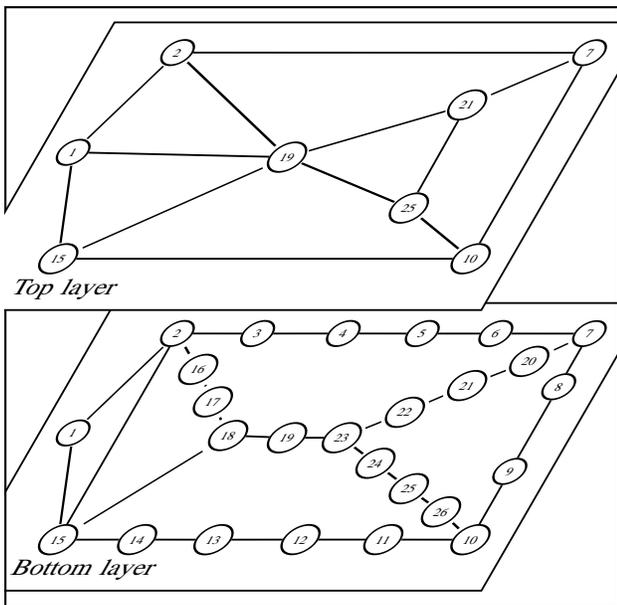


Fig. 8. Net 7 ($N = 8, L = 13, N^b = 26, L^b = 30$)