Survivable Logical Network Design with an Underlying Physical Layer

S. Orlowski*, R. Wessäly*†
* Zuse Institute Berlin, Germany, orlowski@zib.de
† atesio GmbH, Berlin, Germany, wessaely@atesio.de

***Abstract*— We consider the design of a logical network topology, together with node hardware, link capacities, and a survivable routing of demands. In addition to all single node failures, the routing must also survive multiple logical link failures caused by single failures in the underlying physical network. Furthermore, the number of logical links supported by a physical link is bounded. We propose an integer linear programming model for this design problem, together with a branch-and-cut based solution approach combined with column generation. The model and algorithm are tested on three real-world test instances, and preliminary results are given.**

***Index Terms*— multi-layer networks, survivable network design, multiple failures, integer programming, branch-and-cut**

Support

Supported by the DFG Research Center Matheon „Mathematics for key technologies“ in Berlin.

I. Introduction

When designing the logical part of a multi-layer network, several constraints are imposed by an underlying physical network. For instance, in SDH/SONET network design, the number of logical (also called virtual) links traversing a given physical link is often limited by the number of available fibers or by the capacity of a radio link. Furthermore, single failures in the underlying physical layer may cause multiple failures in the logical network.

In this paper, we propose an integer linear programming (ILP) model for designing a logical topology, node hardware, link capacities, and a routing in the logical network which takes the above constraints caused by an underlying physical network into account. For the solution of the planning task, a branch-and-cut based solution approach combined with column generation is proposed. Preliminary results on three real-world test instances stemming from SDH and WDM planning contexts are given. For the full paper, an $\mathcal{NP}$-completeness proof for the column generation problem is in preparation.

A. Practical background

We consider the design of a logical network, given an underlying physical network which might consist of optical fibers, copper cables, or radio links with corresponding hardware at the nodes. The goal is to design a *logical* topology together with suitable hardware (e.g., routers, multiplexers, or interface cards), link capacities, and a routing which survives all *physical* node and link failures, such that total installation cost in the logical layer is minimized.

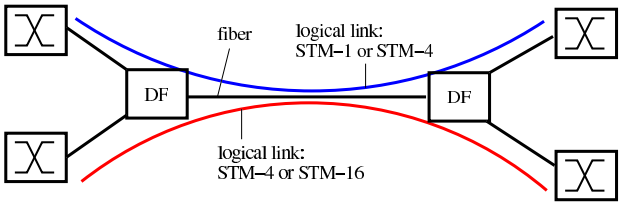

Fig. 1. Two SDH locations with multiplexers and a distribution frame (DF). At most one out of several candidate STM-N links using the same fiber link has to be chosen and equipped with a capacity.

An additional constraint is induced by the underlying physical network: Only a limited number of logical links can traverse a given physical link at the same time. We assume that for each potential logical link, a representation by a path in the physical network is specified, which is used if the logical link is selected. Part of the decisions to take are which of the potential logical links to install and which capacity to assign to them.

There are several practical applications of this planning scenario. For instance, in Figure 1, several SDH multiplexers at the same location are connected to a distribution frame (or patch-panel), which is connected to a similar location by a dark fiber link. The fiber supports one out of several candidate logical STM-N links whose capacity is determined by the interface cards and ports installed at the multiplexers. The question is which of these logical links should be selected and which capacity (e.g., STM-1, STM-4, or STM-16) should be installed on it. Another example is a network design problem where several STM-1 links may either be installed (or rented) or not, and an underlying radio link provides a capacity of 4xSTM-1. The logical links have to be selected such that at most four logical STM-1 links employing this physical link are used and every demand routed in the logical network survives all single physical link and node failures. Similar problems occur in IP/MPLS over WDM network planning, where demands are routed on lightpaths which are in turn embedded into the fiber layer.

In any such planning scenario, the model and algorithm proposed in this paper can be used in a study to evaluate the network cost with a given set of the above constraints originating in the underlying physical network. This can help answering the question whether it would pay to install or to

lease an additional fiber, or to augment the capacity of a radio link, for instance.

B. Literature

A major part of the available literature on survivable network design considers single-layer network design with single link and/or node failures. Only recently, multiple failures induced by an underlying physical layer have been taken into account. The following literature overview is restricted to the latter kind of papers.

Most of these publications concentrate on embedding a logical topology consisting of WDM lightpaths into a physical topology consisting of optical fibers, such that the logical network is not disconnected by any single physical link failure [1]–[6] or by any single physical link or node failure [7]. From a mathematical point of view, the presented models describe a single-layer routing problem: route a set of unit demands on the fiber topology such that (i) capacity constraints on a fiber (maximal number of wavelengths) are satisfied and (ii) any single link or node failure is survived. The objective is usually to minimize the total number of links of all lightpaths.

Such topology considerations can serve as a basis for link protection or link restoration at the logical layer: If the logical topology has been designed as described, a backup lightpath can always be found in the physical layer in any of the considered failure scenarios, provided that either sufficiently high capacities are available or that enough preemptible traffic can be dropped in case of a failure. However, a routing of demands on the lightpaths is not considered in these papers.

In [1], the tabu search heuristic Disjoint Alternate Path (DAP) is described. In [3], [4], an integer linear programming (ILP) formulation with an exponential number of cut inequalities is given for link failures, which is generalized to node failures in [7]. For solving the problem, a tabu search heuristic is proposed. An alternative ILP formulation with a polynomial number of constraints is discussed in [6]. In all these papers, the proposed algorithms are tested on randomly generated logical topologies. With the exception of [1], the underlying physical topologies are assumed to be ring networks.

In [2], [5], theoretical results are provided on the existence of a survivable logical topology based on an underlying physical ring topology. In [3], [4], [6], the number of lightpaths traversing a fiber is bounded by the number of available wavelengths in the presented ILPs, but this constraint is dropped in the computational tests. Out of the papers using a tabu search heuristic for the logical topology design problem, only [7] includes such a constraint. As far as we know, a constraint of the number of logical links traversing a physical one has not been considered yet in any paper which goes beyond topology planning in the design of a logical network.

In [8], the spare capacity assignment problem in the WDM layer is considered. Given working lightpaths for each demand, the goal is to find backup lightpaths for a given set of Shared Risk Link Groups (SRLGs) using link protection, such that total spare capacity cost is minimized. An ILP formulation is proposed for the important special case of single and double link failures. Any number of lightpaths can be routed on a link, with link cost being proportional to the capacity and the length of a link. In the test computations on three networks, the ILP is solved to optimality with respect to a previously enumerated set of 10 admissible backup paths for each failure situation. By varying different parameters, the authors investigate the impact on total network cost of different failure scenarios.

In this paper, we generalize the above approaches. During the design of the logical network, two constraints imposed by the underlying physical layer are taken into account: (i) demands routed in the logical network must survive multiple logical failures caused by single physical link or node failures, and (ii) a constraint is given on the number of logical links traversing a physical one. Furthermore, practical hardware configuration cost and constraints are taken into account.

We present an ILP model for solving the planning problem and propose a branch-and-cut algorithm combined with column generation for solving it. In view of the high complexity of the planning problem, a branch-and-cut algorithm provides a suitable framework which can be enhanced by good primal heuristics (such as tabu search). Preliminary computational results on three real-world instances are given which show the importance of taking constraints from an underlying physical layer into account. For the final paper, an $\mathcal{NP}$-completeness proof for the column generation problem is in preparation.

This article is organized as follows. After presenting our mathematical model in Section II, we propose an integrated solution approach in Section III. After presentation and discussion of some preliminary results in Section IV, we conclude this working paper with Section V.

II. Mathematical Model

In this section, our integer programming model for an integrated planning of

- a logical topology with physical topology constraints,
- logical link capacities,
- a hardware configuration, and
- a survivable routing with respect to physical failures

is presented. The model, based on the one presented in [9], decomposes into a hardware and a routing part, connected by the link capacities. In this paper, the model is extended by an underlying physical graph which gives rise to an additional class of constraints for installing the logical link capacities. In the routing part, the underlying physical graph is used to derive the set of considered failure states to make sure that the routing in the *logical* network survives any single *physical* link or node failure.

A. Virtual Topology, Hardware, and Link Capacities

The hardware model, illustrated by Figure 2, covers the selection of a topology, network elements, and interface cards, as well as the restrictions caused by slot and port constraints.

1) Notation: The network is modeled by an undirected graph $G = (V, E)$ representing the logical network to be designed. Given is a set $\mathcal{P}$ of *port types*, e.g., corresponding to different STM-N interfaces. For every node $v \in V$, a set $\mathcal{D}_v$ of installable *node designs* is given, representing network

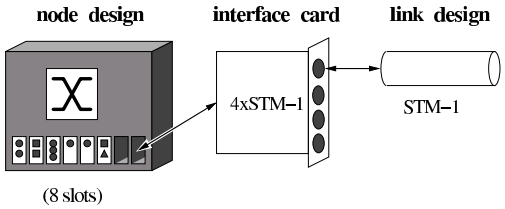

Fig. 2. Hardware configuration model

elements of different types (multiplexers, cross-connects) or vendors, or racks with a different number of shelves, for example. Similarly, for every logical link $e \in E$, a set $\mathcal{D}_e$ of installable *link designs* is given which may, e.g., correspond to different STM-N capacities. The set $\mathcal{I}$ refers to the *interface cards*, each of which provides a certain number of ports to link designs and consumes slots at a network element.

On every node $v \in V$ and on every link $e \in E$, at most one design must be chosen. If a node or link is not equipped with a design, it is not included in the final topology. Every node design $d \in \mathcal{D}_v$ offers $S^d \in \mathbb{Z}_+$ slots. An interface card $i \in \mathcal{I}$ requires $S^i \in \mathbb{Z}_+$ slots and provides $P^i_p \in \mathbb{Z}_+$ ports of type $p \in \mathcal{P}$. Each link design $d \in \mathcal{D}_e$ requires $P^d_p \in \mathbb{Z}_+$ ports of type $p \in \mathcal{P}$ and provides a *routing capacity* of $C^d_e \in \mathbb{Z}_+$ to the routing.

2) Variables: For every node $v \in V$ and every node design $d \in \mathcal{D}_v$, the variable $x^d_v \in \{0,1\}$ indicates whether d is installed at v or not. Similarly, for every link $e \in E$ and every link design $d \in \mathcal{D}_e$, the variable $x^d_e \in \{0,1\}$ indicates whether d is installed at e or not. Finally, for every node $v \in V$ and every interface card $i \in \mathcal{I}$, the variable $x^i_v \in \mathbb{Z}_+$ states how often i is installed at v.

3) Constraints: The following set of linear inequalities models the compatibility requirements of the hardware installed in the network:

$$\sum_{d\in\mathcal{D}_v} x^d_v \le 1 \qquad \forall v \in V \qquad (1)$$

$$\sum_{d\in\mathcal{D}_e} x^d_e \le 1 \qquad \forall e \in E \qquad (2)$$

$$\sum_{e\in\delta(v)} \sum_{d\in\mathcal{D}_e} P^d_p x^d_e - \sum_{i\in\mathcal{I}} P^i_p x^i_v \le 0 \qquad \begin{array}{l}\forall v \in V,\\ \forall p \in \mathcal{P}\end{array} \qquad (3)$$

$$\sum_{i\in\mathcal{I}} S^i x^i_v - \sum_{d\in\mathcal{D}_v} S^d x^d_v \le 0 \qquad \forall v \in V \qquad (4)$$

$$x^i_v \in \mathbb{Z}_+,\ x^d_v,\ x^d_e \in \{0,1\}$$

Inequalities (1) and (2) state that at most one design must be chosen on each node and on each logical link. The port inequalities (3) ensure that for every node and every port type, the installed interface cards provide enough ports for the link designs on incident links. The slot inequalities (4) state that for every node, the number of slots required by the installed interface cards must not exceed the number of slots provided by the installed node designs.

4) Extension by a physical graph: We now extend the model by an underlying physical graph $G_p = (V_p, E_p)$, where $V_p \supseteq V$, and assume that a physical representation is specified for each logical link which is used if the link is selected. Given the restriction that no more than c_{e_p} logical links may traverse physical link $e_p \in E_p$, the decision to be made is which links should be chosen from the potential logical topology and which capacity should be assigned to them.

Let $e_p \in e$ denote the fact that the physical path by which the logical link $e \in E$ would be realized contains the physical link $e_p \in E_p$. The above constraint can be written as

$$\sum_{e\in E: e_p\in e} \sum_{d\in\mathcal{D}_e} x^d_e \le c_{e_p} \qquad e_p \in E_p. \qquad (5)$$

Since at most one link design may be chosen on each logical link, the left-hand side counts exactly the number of logical links routed over e_p.

B. Objective

The objective is to minimize total installation cost for network elements, interface cards, and logical links. With every node design $d \in \mathcal{D}_v$ installable at a node $v \in V$, a cost value $K^d_v \in \mathbb{Z}_+$ is associated. Likewise, each interface card $i \in \mathcal{I}$ incurs a cost $K^i \in \mathbb{Z}_+$, and the cost of link design $d \in \mathcal{D}_e$ on link $e \in E$ is $K^d_e \in \mathbb{Z}_+$. The objective function can be written as

$$\min \sum_{v\in V} \left(\sum_{d\in\mathcal{D}_v} K^d_v x^d_v + \sum_{i\in\mathcal{I}} K^i x^i_v \right) + \sum_{e\in E} \sum_{d\in\mathcal{D}_e} K^d_e x^d_e\,. \qquad (6)$$

C. Survivable Routing

The computations for this paper are based on the formulation from [9], but multiple failures in the logical layer caused by failures in the underlying physical graph are taken into account now. We will now briefly present the routing model. Let

$$y_e := \sum_{d\in\mathcal{D}_e} C^d_e x^d_e$$

denote the total link capacity of logical link $e \in E$ (with respect to some link design vector x).

Given is a set of *failure states* $s \in \mathcal{S}$ which have to be considered in the design of the logical network. Each of them is characterized by its failing components in the logical layer, i.e., $s \subset V \cup E$. The failure situations considered in our computations are those induced by the failure of a single node $v \in V_p$ (in any layer) or of a single physical link $e_p \in E_p$. In particular, several logical links may fail at the same time, caused by an underlying physical failure. Such failure states are also known as Shared Risk Link Groups [8].

Let $\mathcal{D}$ be a set of point-to-point communication demands. For every demand $uv \in \mathcal{D}$, a *demand value* $d_{uv} \in \mathbb{Z}_+$ is specified which has to be routed in the failure-free state. Furthermore, a *diversification parameter* $\delta_{uv} \in (0,1]$ is given for each demand, denoting the maximum fraction of the demand value which is allowed to fail in any considered failure state. By setting this parameter to 0.5 and doubling the demand value, the following formulation provides a good relaxation of 1+1 protection which is computationally tractable using a column generation procedure.

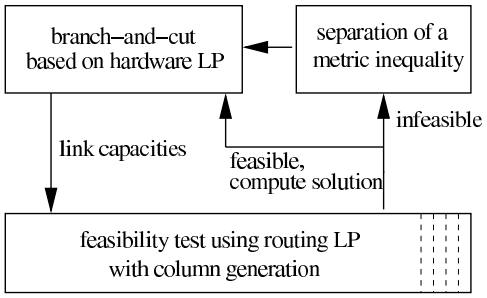

Fig. 3. Algorithmic approach.

A demand $uv \in \mathcal{D}$ may be routed on one or more paths from the set $\mathcal{P}_{uv}$, which comprises all simple uv-paths (i.e., without loops). In failure state $s \in \mathcal{S}$, the subset $\mathcal{P}^s_{uv} \subseteq \mathcal{P}_{uv}$ denotes all surviving uv-paths. Using non-negative integer flow variables $f_{uv}(P) \in \mathbb{Z}_+$ for all demands $uv \in \mathcal{D}$ and all paths $P \in \mathcal{P}_{uv}$, the routing part of the model reads as follows:

$$\sum_{P \in \mathcal{P}_{uv}} f_{uv}(P) \geq d_{uv} \qquad uv \in \mathcal{D} \tag{7}$$

$$\sum_{uv \in \mathcal{D}} \sum_{\substack{P \in \mathcal{P}_{uv}: \\ e \in P}} f_{uv}(P) \leq y_e \qquad e \in E \tag{8}$$

$$\sum_{P \in \mathcal{P}_{uv} \setminus \mathcal{P}^s_{uv}} f_{uv}(P) \leq \delta_{uv} d_{uv} \qquad \begin{array}{l} \forall uv \in \mathcal{D}, \\ \forall s \in \mathcal{S} \end{array} \tag{9}$$

$$f_{uv}(P) \in \mathbb{Z}_+$$

The demand constraints (7) and capacity constraints (8) formulate a multicommodity flow problem. For every failure state, the diversification constraints (9) ensure for each demand that at most the specified fraction of the demand value fails in any failure state. Notice that the extension of this model by further requirements such as hop limits is straightforward by restricting the set of admissible paths appropriately.

III. Branch-and-Cut with Column Generation

Our proposed solution approach is similar to Benders decomposition [10]. The central procedure is a branch-and-cut algorithm [11] based on an LP relaxation which contains only hardware variables and constraints, but no routing information. To strengthen the LP relaxation, cutting planes are separated during the whole branch-and-bound process, such as Gomory mixed-integer rounding cuts [12], GUB cover inequalities [13], and generalizations [14] of metric inequalities [15].

Each time integer an link design vector $\bar{x}$ is identified during the branch-and-bound process, the corresponding (integer) link capacity vector

$$\bar{y}_e = \sum_{d \in \mathcal{D}_e} C^d_e \bar{x}^d_e$$

is tested for feasibility with respect to the (missing) routing constraints. If no feasible routing exists within $\bar{y}$, a violated metric inequality is derived from the dual objective function of the routing formulation, which is then added to the relaxation in order to cut off the infeasible capacities. If, on the other hand, a feasible routing can be found within these capacities, a cheapest hardware configuration over these link capacities is computed using a separate integer program based on the hardware formulation. This process is illustrated in Figure 3.

To test whether a feasible routing exists within given link capacities $\bar{y}_e$, the continuous relaxation of the routing ILP (7)–(9) is solved with $\bar{y}$ as right-hand side in the capacity constraints (8). As the routing LP has an exponential number of path variables, column generation has to be employed in order to decide whether a routing exists or not.

Quality guarantees for a given solution are available if the branch-and-cut process yields a valid lower bound on the optimal solution value. In particular, this lower bound can be used to prove optimality of a solution. It is valid as long as no optimal capacity vector $\bar{y}$ is cut off by a separated inequality.

The integrality of the routing variables does *not* affect the validity of the lower bound: $\bar{y}$ is only cut off by a metric inequality if no *fractional* routing exists, in which case no feasible *integer* routing exists, either. If a fractional routing is found, we try to turn it into a feasible integer solution using rounding and rerouting techniques.

Furthermore, all necessary path variables have to be found either in the computation of an initial path set, or during the column generation process. It has already been shown that finding two physically edge- or node-disjoint paths in the logical layer is $\mathcal{NP}$-hard [7], [16]; we employ two heuristics for finding such paths.

The pricing problem, i.e., the question which further path variables should be added to the LP, is polynomially solvable with the presented formulation as long as only single logical link or node failures are considered [14]. As soon as multiple failures are involved, we strongly suspect the pricing problem to be $\mathcal{NP}$-hard; a proof is in progress and will be included in the final paper. Currently, we are solving the pricing problem for multiple failures approximately. As a possible consequence, a feasible capacity vector might be cut off by an invalid inequality if not enough path variables have been generated to allow for a feasible (fractional) routing. This may lead to an invalid lower bound if an optimal vector is accidently cut off.

From a practical point of view, however, notice that (i) even with approximate pricing, missing path variables are often identified, (ii) an invalid separated inequality need not cut off all optimal solutions, and (iii) this approach still allows for better solutions than just enumerating some fixed set of admissible paths. In fact, we generated such a restricted path set in the beginning and use it as a starting point for the column generation procedure. In regard of these considerations, even if the obtained lower bound might be invalid in some cases, it is probably not far from a valid one.

IV. Preliminary Computational Results

We have implemented the approach presented in the previous section and tested it on three real-world test instances of different size stemming from SDH and WDM planning contexts. The number of nodes, logical links, and demands in the test instances are (10, 17, 10) for the instance p1, (48, 92, 73) for p2, and (44, 129, 861) for p3, respectively.

c_{e_p}	p1	p2	p3
1	14308000	infeasible	-
2	10800000	63474000	-
3	10800000	63248000	-
5	10800000	62966000	7358207
10	10800000	63550000	7346222
∞	10800000	63044000	7119533

TABLE I

BEST SOLUTION VALUES FOR THE THREE TEST INSTANCES

Table IV reports on preliminary results for these test instances. The first column gives the problem name, while the next two columns contain the number of logical paths allowed to traverse any given physical link and the best obtained solution value for each of the problem instances. The computation times range from several seconds to the time limit of 5,5 hours.

With a branch-and-cut based algorithm, three scenarios can happen: (i) optimality of a solution can be proven, (ii) it can be shown that no solution exists, or (iii) a solution is computed but little can be said about its quality; this situation is similar to the case where only a primal heuristic is employed.

In the test runs on the 10-node instance, our lower bound was always equal to the upper bound or no unexplored nodes were left in the branch-and-cut process, which indicates that the obtained solutions are probably optimal. It turns out that with only one logical link being allowed to traverse any physical link, the logical network cost increases by more than 30%, compared to the case where two logical links per physical link are allowed! In this case, it would probably pay off to install a further fiber on selected physical links.

For the large and dense 44-node instance (average node degree 5.8, almost full demand matrix), we could compute solutions for a limit of 5 and 10 logical links per physical link, as well as for the unrestricted case. As expected, the solution values are decreasing as the restriction is relaxed. For $c_{e_p} \in \{1, 2, 3\}$, no solutions could be identified, and it is not known whether the corresponding planning problems are feasible or not. For the final paper, we will try to further improve our bounds so as to either identify a solution or to get indications that no solution exists.

For the 48-node instance, the problem was reported to be infeasible in the case $c_{e_p} = 1$, after adding some cutting planes to the hardware formulation. For larger limits, solutions were found, but the solution values are not monotonically decreasing. However, notice that a solution for a given value of c_{e_p} is also feasible for a larger value, which provides a means to improve some of the solution values. On this instance, the solution values do not differ very much, which means that laying down new fibers would probably not pay off.

V. CONCLUSION

In this paper, the design of a logical network including hardware, link capacities, and a survivable routing has been considered. Several constraints arising from an underlying physical layer have been taken into account in the planning process: Multiple logical failures may be caused by single physical failures, and a physical link supports only a limited number of logical links, e.g., due to fiber restrictions.

We have proposed an integer programming model and a branch-and-cut algorithm with column generation to solve the problem. Despite the high complexity of the problem, we could compute solutions on realistic large and dense networks with up to 44 nodes and 192 links using the described solution approach. For one of the large test instances, we could give strong indications that the planning problem has no solution if only one logical link is allowed to traverse a physical link. We observed that such a limitation can easily lead to infeasibility of the planning problem or to a significant increase network cost. It is therefore of major practical importance that such constraints be taken into account in the planning process.

In the full paper, a proof will be included that the pricing problem in column generation is $\mathcal{NP}$-hard, and further computational studies will be presented and discussed in more detail.

REFERENCES

[1] O. Crochat and J.-Y. L. Boudec, "Design protection for WDM optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 16, September 1998.

[2] H. Lee, H. Choi, and H. Choi, "Restoration in IP over WDM optical networks," in *Proceedings of the 30th International Workshop on Optical Networks, (ICPP'01)*, pp. 263–268, September 2001.

[3] E. Modiano and A. Narula-Tam, "Survivable routing of logical topologies in WDM networks," in *Proceedings of the 20th IEEE Infocom 2001,* Anchorage, USA, pp. 348–357, 2001.

[4] E. Modiano and A. Narula-Tam, "Surivable lightpath routing: A new approach to the design of WDM-based networks," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 800–809, May 2002.

[5] H. Lee, H. Choi, S. Subramaniam, and H. Choi, "Survivable embedding of logical topology in WDM ring networks," in *Proceedings of the Joint Conference on Information Sciences 2002,* Research Triangle Park, North Carolina, USA, pp. 1393–1397, March 2002.

[6] Q. Deng, G. Sasaki, and C.-F. Su, "Survivable IP over WDM: An efficient mathematical programming formulation," in *Proceedings of 40th Annual Allerton Conference on Communication, Control, and Computing 2002,* Illinois, USA, 2002.

[7] A. Todimala and B. Ramamurthy, "Survivable virtual topology routing under shared risk link groups in WDM networks," in *First International Conference on Broadband Networks (BROADNETS'04),* San Jose, CA, USA, pp. 130–139, October 2004.

[8] J. Doucette and W. Grover, "Shared-risk logical span groups in span-restorable optical networks: Analysis and capacity planning model," *Photonic Network Communications*, vol. 9, no. 1, pp. 35–53, 2005.

[9] A. Kröller and R. Wessäly, "Integrated optimization of hardware configuration and capacity dimensioning in SDH and opaque WDM networks," in *Proceedings of the First International Network Optimization Conference (INOC 2003), Paris*, pp. 349–354, October 2003.

[10] J. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik*, vol. 4, no. 1, pp. 238–252, 1962.

[11] M. W. Padberg and G. Rinaldi, "A branch and cut algorithm for the resolution of large–scale symmetric traveling salesman problems," *SIAM Review*, vol. 33, pp. 60–100, 1991.

[12] H. Marchand and L. Wolsey, "Aggregation and mixed integer rounding to solve MIPs," Tech. Rep. 9839, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, 1998.

[13] L. Wolsey, "Valid inequalities for 0/1 knapsacks and MIPs with Generalized Upper Bound constraints," *Discrete Applied Mathematics*, vol. 29, pp. 251–261, 1990.

[14] R. Wessäly, *Dimensioning Survivable Capacitated NETworks.* PhD thesis, Technische Universität Berlin, April 2000.

[15] M. Iri, "On an extension of the maximum-flow minimum-cut theorem to multicommodity flows," *Journal of the Operations Research Society of Japan*, vol. 13, no. 3, pp. 129–135, 1971.

[16] R. Karp, "On the computational complexity of combinatorial problems," *Networks*, vol. 5, pp. 45–68, January 1975.