

# Integrated Optimization of Hardware Configuration and Capacity Dimensioning in SDH and opaque WDM networks

Alexander Kröller and Roland Wessäly<sup>†‡</sup>

## Abstract

We suggest a new model for the design of telecommunication networks which integrates decisions about the topology, configuration of the switching hardware, link dimensioning, and protected routing of communication demands. Applying the branch-and-cut-algorithm implemented in our network planning and optimization tool DISCNET, we demonstrate that real-world based network planning instances of such an enhanced model can be solved.

**Keywords:** Integer Programming, Network Models, Network Planning, Routing, Resource Allocation, Survivability, Topological Design, Sonet/SDH Networks, Optical Networks.

## 1 Introduction

We propose a mixed-integer linear programming approach to deal with the *integrated* optimization of

- hardware configuration,
- topology and link dimensioning, and
- protected routing of demands

in communication networks. Over the last decade, mathematical models as well as solution methods for the design of a broad range of technologically different communication networks have been developing. Starting from pure topology optimization with connectivity requirements [9,18], many authors considered network planning problems including link dimensioning and routing, see e.g. [5,6,14]. Others extended these models to protect the networks against single node and single link failures, see e.g. [1,7,17,19]. We are aware that these lists of references are non-exhaustive and only cover a few publications where a branch-and-cut approach is used to compute (optimal) solutions. The models proposed are of great variety. There is a number of different ways to deal with the capacity structure on the links, e.g. continuous capacities, multiples of some basic capacities, or a discrete set of capacity choices. The routing can be continuous or integer, it can be on a single path or on multiple paths. The survivability models cover different kinds of failure scenarios, various protection mechanisms, e.g. 1+1 protection and diversification, or restoration mechanisms such as reservation, meta-mesh, and link and path restoration in its different versions.

---

<sup>†</sup>Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustraße 7, D-14195 Berlin, Germany.

<sup>‡</sup>Corresponding author. E-mail: wessaely@zib.de URL: <http://www.zib.de/wessaely>

To our knowledge, there are only a few publications where the total flow through each node is limited either by a constant or a discrete set of capacity bounds, see e.g. [4, 13]. There are, however, no models dealing appropriately with the modular cost and capacity structure of the hardware in nowadays networks. The network elements used in node locations of SDH networks, for instance, are add-drop-multiplexers or cross-connects which have a fixed switching capacity, a fixed number of slots to plug in interface cards of particular types, which have the ports to connect fibers. Given some link capacities, it is therefore not clear whether there exists a compatible hardware configuration, and even if it does, its overall cost might be far from being optimal.

In Section 2, we propose a mathematical optimization model which extends previously suggested models in order to cover hardware requirements. This model is independent of a particular technology. It has been used, for instance, in planning scenarios stemming from SDH networks and opaque WDM networks. Considering just the hardware configuration part of the model, it can also be used for ATM, IP, and MPLS networks. In Section 3, we sketch our branch-and-cut framework, and in Section 4, we show computational results for various structurally different planning problems, all of which are based on real-world problems. The results reveal that the hardware configuration can and should be incorporated into optimization tools for telecommunication networks.

## 2 Problem and Model

In this section, we present our mathematical model for the design of survivable communication networks. Before going into the details of this model in Sections 2.2–2.4, the next section sketches the hardware of two important transport network technologies, namely SDH and WDM networks.

### 2.1 Hardware in SDH and opaque WDM networks

While transmission technology differs between SDH and WDM networks, the node location hardware used is of similar structure for SDH and opaque WDM networks. The network elements are ADMs (add-drop-multiplexers) and DXCs/OXCs (digital/optical cross-connects). These network elements have an internal switching unit which performs the routing of virtual containers (VCs) and optical channels. Due to technical limitations such as internal bus systems, the capacity of this switching unit is typically restricted to some multiple of STM-1 (155 Mbit/s), the basic transmission capacity in these transport networks. ADMs and cross-connects provide slots to plug in certain interface cards, to which fibers can be connected. As special property, ADMs have two so-called aggregates as distinguished slots. These are equipped with predetermined (single port) interface cards and are aimed at establishing ring networks. Interface cards have a specific number of ports to attach fibers operated at specific capacities, i.e., STM-N capacities in SDH and wavelength/bitrate assignments for WDM. By attaching a duplex or a pair of directed simplex fibers, an undirected link with fixed transmission capacity to another node location can be established.

WDM technology allows to operate multiple wavelengths over a single fiber. In the aforementioned opaque networks, node equipment always converts optical signals to electric before processing them. Therefore, the ability to operate multiple wavelengths on one fiber is merely a method to increase capacity. In transparent WDM networks, which are not considered in

this article, an optical channel can be forwarded fully optical, and wavelength assignment has a great impact on hardware configurations.

## 2.2 Hardware Configuration and Link Dimensioning

The potential network is modeled as an undirected *supply graph*  $G = (V, E)$ , where  $V$  is the set of potential node locations and  $E$  is the set of admissible links. To deal with the hardware requirements at the nodes and the capacities to be installed on the links, we introduce *node designs* to represent ADMs, DXCs, OXCs, or more complex configurations using these devices, *modules* as abstraction of interface cards, and *link designs* to deal with capacities to be installed on the links, see Figure 1 for a simplified introductory picture.

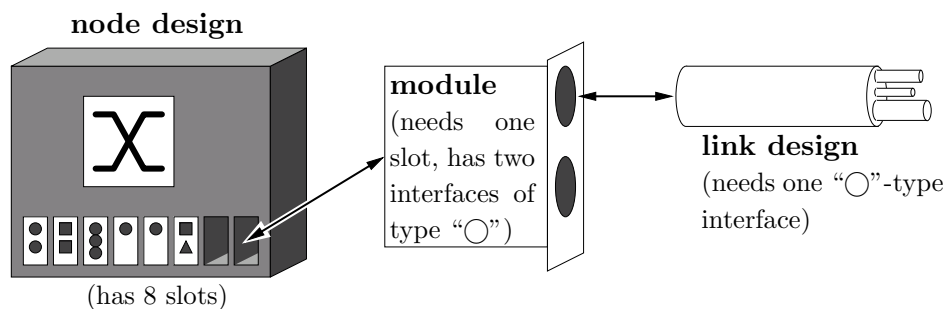


Figure 1: Hardware configuration example.

For each particular node  $v \in V$ , a set  $\mathcal{D}(v)$  of admissible node designs is given from which at most one design can be chosen. Key properties of a node design  $d \in \mathcal{D}(v)$  are its maximum switching capacity  $C^d \in \mathbb{Z}_+$ , the types of supported modules  $\mathcal{M}(v) \in \mathbb{Z}_+$  and the number  $S^d \in \mathbb{Z}_+$  of slots available to install modules into the node design. Figure 1 shows a node design with eight slots, six of which are already occupied. It is possible to install multiple modules at a node design. For each module  $m \in \mathcal{M}(v)$ , however, at most  $M^{d,m} \in \mathbb{Z}_+$  are admissible at node design  $d \in \mathcal{D}(v)$ . Each installed module  $m \in \mathcal{M}(v)$  occupies  $S^m \in \mathbb{Z}_+$  many slots; the sum of the slot requirements of all modules installed at a node design  $d \in \mathcal{D}(v)$  should not exceed  $S^d$ . Each module  $m$  provides  $I^{i,m} \in \mathbb{Z}_+$  interfaces of type  $i \in \mathcal{I}$ , where  $\mathcal{I}$  is the set of all interfaces. These interfaces allow to attach link designs (and therefore links) to nodes. Interfaces appearing in Figure 1 are “□”, “△” and “○”.

For each particular link  $e \in E$ , a set  $\mathcal{L}(e)$  of admissible link designs is given from which at most one design can be chosen. A link design  $\ell \in \mathcal{L}(e)$  is determined by its (payload) capacity  $C^\ell \in \mathbb{Z}_+$  and, for each interface  $i \in \mathcal{I}$ , the number  $I^{i,\ell}$  of interfaces it requires at both end nodes of the link. Each link has a preinstalled link design  $\ell_e$  which is fixed and can not be removed, but can be a dummy providing no capacity at all. Figure 1 shows a link design requiring only one interface of type “○”.

Using decision variables  $x_{d,v} \in \{0, 1\}$  for all  $v \in V$  and all node designs  $d \in \mathcal{D}(v)$ , non-negative integer variables  $x_{m,v} \in \mathbb{Z}_+$  for the number of modules  $m \in \mathcal{M}(v)$  installed at  $v$ , and decision variables  $x_{\ell,e} \in \{0, 1\}$  for all  $e \in E$  and all link designs  $\ell \in \mathcal{L}(e)$ , the problem of selecting a topology, including node and link designs, can be stated as follows:

$$\sum_{d \in \mathcal{D}(v)} x_{d,v} \leq 1 \quad v \in V \quad (1)$$

$$\sum_{\ell \in \mathcal{L}(e)} x_{\ell,e} \leq 1 \quad e \in E \quad (2)$$

$$\sum_{e \in \delta(v)} \left( I^{i,\ell_e} + \sum_{\ell \in \mathcal{L}(e)} I^{i,\ell} x_{\ell,e} \right) - \sum_{m \in \mathcal{M}(v)} I^{i,m} x_{m,v} \leq 0 \quad \begin{array}{l} v \in V, \\ i \in \mathcal{I} \end{array} \quad (3)$$

$$\sum_{e \in \delta(v)} \left( C^{\ell_e} + \sum_{\ell \in \mathcal{L}(e)} C^{\ell} x_{\ell,e} \right) - \sum_{d \in \mathcal{D}(v)} C^d x_{d,v} \leq 0 \quad v \in V \quad (4)$$

$$\sum_{m \in \mathcal{M}(v)} S^m x_{m,v} - \sum_{d \in \mathcal{D}(v)} S^d x_{d,v} \leq 0 \quad v \in V \quad (5)$$

$$x_{m,v} - \sum_{d \in \mathcal{D}(v)} M^{d,m} x_{d,v} \leq 0 \quad \begin{array}{l} v \in V, \\ m \in \mathcal{M}(v) \end{array} \quad (6)$$

Inequalities (1) and (2) state that at most one design must be chosen for each node and each link, respectively; the topology consists of those graph elements where exactly one design is chosen. Inequalities (3) and (4) ensure for each node that enough interfaces of each type are available and that the switching capacity of the selected node design is sufficient to attach the designs of the incident links. Eventually, (5) and (6) ensure for each node that the selected node design provides sufficiently many slots and that the maximum number of admissible modules is not exceeded.

### 2.3 Routing and Protection

In addition to the topology and the hardware configuration, a feasible network design comprises a survivable routing. As communication requirements a set  $\mathcal{D}$  of demands is given, where three parameters are associated with each demand  $uv \in \mathcal{D}$ : the demand value  $d_{uv} \in \mathbb{Z}_+$  which must be routed between the end nodes  $u$  and  $v$  (assuming a bifurcated routing, i.e., several paths may be used for one demand), a path length restriction (also called hop limit)  $\ell_{uv} \in \mathbb{N}$  which specifies the maximum number of links of an admissible path between  $u$  and  $v$ , and the diversification value  $\delta_{uv} \in (0, 1] \subseteq \mathbb{R}_+$  which specifies the fraction of the demand  $d_{uv}$  which can maximally be routed through the graph elements of a *failure state*, which typically is a single node or a single link failure, but can also be a multi-link failure (if failures from another network layer must be protected in this network). The set  $S$  comprises all failure states for which diversification restrictions of single demands must be respected. Notice that setting the diversification values to  $\frac{1}{2}$  for all demands and multiplying all demand values by two, a mechanism similar to 1+1 protection can be implemented.

Let  $\mathcal{P}$  be the set of all simple paths in  $G$  and let  $\mathcal{P}_{uv}$  be the subset of admissible paths to route the demand  $uv \in \mathcal{D}$ , which are all paths between  $u$  and  $v$  satisfying the path length restriction  $\ell_{uv}$ . Furthermore, let a failure state  $s \in S$  be active for a path  $P \in \mathcal{P}$ , if some of its failing nodes or edges are passed by this path. Using non-negative continuous flow variables  $f_{uv}(P) \in \mathbb{R}_+$  for all demands  $uv \in \mathcal{D}$  and all paths  $P \in \mathcal{P}_{uv}$ , the following constraints

formulate the routing requirements as a multi-commodity flow problem with path-length and protection restrictions:

$$C^{\ell_e} + \sum_{\ell \in \mathcal{L}(e)} C^{\ell} x_{\ell,e} - \sum_{uv \in \mathcal{D}} \sum_{\substack{P \in \mathcal{P}_{uv}: \\ e \in P}} f_{uv}(P) \geq 0 \quad e \in E \quad (7)$$

$$\sum_{P \in \mathcal{P}_{uv}} f_{uv}(P) = d_{uv} \quad uv \in \mathcal{D} \quad (8)$$

$$\sum_{P \in \mathcal{P}_{uv}: s \cap P \neq \emptyset} f_{uv}(P) \leq \delta_{uv} d_{uv} \quad uv \in \mathcal{D}, s \in S \quad (9)$$

The capacity constraints (7) restrict the flow over a particular link to the capacity of the selected link design, constraints (8) ensure that all demands are routed with respect to their value. Eventually, with constraints (9) the diversification (protection) requirements are met, since the flow of all paths active in a failure state is appropriately bounded for each demand.

## 2.4 Cost Minimization

We aim at designing cost-minimal survivable networks. For each node  $v \in V$ , the installation of node design  $d \in \mathcal{D}(v)$  incurs a cost of  $K^d$  and equipping this node design with a module  $m \in \mathcal{M}(v)$  incurs a cost of  $K^m$ . Similarly, for each link  $e \in E$ , the installation of link design  $\ell \in \mathcal{L}(e)$  incurs a cost of  $K_e^{\ell}$ . Using this notation, the objective function reads as follows:

$$\min \sum_{v \in V} \left( \sum_{d \in \mathcal{D}(v)} K^d x_{d,v} + \sum_{m \in \mathcal{M}(v)} K^m x_{m,v} \right) + \sum_{e \in E} \left( K_e^{\ell_e} + \sum_{\ell \in \mathcal{L}(e)} K_e^{\ell} x_{\ell,e} \right). \quad (10)$$

Summarizing, the overall planning problem is to simultaneously decide the topology, the hardware configuration, the link capacities and a survivable routing satisfying (1)–(9) such that the network cost (10) are minimal.

## 3 A Branch-and-Cut Algorithm

Our solution approach is based on Benders decomposition [2]. The central procedure is a branch-and-cut algorithm [16] based on a relaxation of the problem described in Section 2 (see [12] for a detailed description). This relaxation includes the hardware configuration constraints (1)–(6) and an arc-flow formulation of the routing problem with node-aggregated commodities, ignoring path-length and survivability restrictions. Instead of using the link design variables, the LP is formulated with incremental link design variables resembling incremental capacities as presented in [7, 19].

To strengthen our formulation at each node of the branch-and-cut tree, we separate inequalities which are violated by the optimal solution of the current relaxation. For this purpose, we apply separation algorithms for band inequalities [7], GUB cover inequalities [20], and generalizations [7, 19] of metric inequalities [10]. The branching process chooses among the incremental link design variables only. Notice that this is equivalent to partitioning the

link designs of some link into two sets: those with smaller and those with larger capacity. The restriction to link design variables ensures that eventually, the respective variables are integer. Node design and module variables, however, might still be fractional at this point.

At each branch-and-bound node, the link capacities of the optimal solution of the current relaxation are passed to a path-based LP comprising inequalities (7)–(9). This LP is solved using column generation techniques and either yields a violated metric inequality or a valid demand routing. In the first case, the inequality is added to the LP relaxation. In the latter case, a cost-minimal hardware configuration supporting the particular routing is determined by an IP solver, using a formulation consisting of inequalities (1)–(6) and the routing’s link utilization as capacity bounds. If such a configuration exists, a complete and feasible solution for the original problem is identified.

## 4 Computational Results

In this section, we report on results of numerical tests on eight real-world based instances stemming from structurally and technologically different planning problems. We distinguish between three types of instances. The first type contains three “plain” problems where a discrete set of capacities is available through link designs, and both modules and node designs have a rather simple structure where at most one type of node design is available at each node. The second type has two problems stemming from opaque WDM networks. The node designs are OXCs with ports provided by OXC-specific modules. Each link design is a predefined wavelength assignment and the number of available channels is its capacity. The last type consists of three SDH planning instances, where node designs are sets of ADMs and DXCs with fixed interconnections, modules are interface cards, and link designs are combinations of STM-N capacities. In contrast to the greenfield planning problems of the first two types, the SDH instances have preinstalled link designs to represent expansion planning problems. The cost structure reflects the modularity of the available hardware. Costs are incurred for network components such as ADMs, DXCs or OXCs, for interface cards, for WDM-systems and for repeaters. In some instances, the cost of the link designs, is solely defined by fiber cost which depend on the length of the link in kilometer and the number of its fiber pairs.

Table 1 lists characteristic properties for each instance. From top to bottom these are the number of nodes and links of the underlying network, the average node degree, the number of links carrying a preinstalled capacity, the number of point-to-point demands, the average number of link designs per link and node designs per node, and finally the total number of available modules. All modules can be installed at every node, but the respective availability of node and link designs is not the same for all nodes and links of a single problem instance.

Table 1 lists our computational results which have been computed on a Linux machine with a 1.0 GHz Athlon processor and 256 MBytes of RAM. Except for the basic operating system, no other software was running on the host during the tests. The total time spent for solving one instance was limited to four hours. For each instance, the results for three different demand scalings (50%, 100%, 200%) are presented: the original demand values of the real-world application, and the values multiplied by  $\frac{1}{2}$  and 2, respectively. The following information is provided for each demand scaling: the *cost* of the best solution computed, where 100 represents the value of the best solution computed for the original demand values within four hours; the *gap* in percent between the best solution and the lower bound; and the *time* spent solving an instance, given as “hours:minutes”.

	Plain			WDM		SDH			
	p1	p2	p3	w1	w2	s1	s2	s3	
Nodes	15	19	36	18	19	18	22	24	
Links	21	34	122	33	148	33	34	55	
Avg. degree	2.8	3.6	6.8	3.7	15.6	3.7	3.1	4.6	
Preinstalled	0	0	0	0	0	13	25	28	
Demands	105	171	630	153	171	153	231	276	
Linkdesigns/Link	4.8	9.1	5.0	12.0	5.0	6.0	6.0	12.0	
Nodedesigns/Node	1.0	1.0	1.0	3.0	3.1	5.0	5.2	8.2	
Modules	7	0	5	3	3	4	4	4	
Cost	×50%	88	58	70	93	85	79	100	90
	×100%	100	100	100	100	100	100	100	100
	×200%	121	159	162	114	128	140	115	159
Gap	×50%	0.0	1.4	31.9	2.8	14.1	0.4	1.2	1.0
	×100%	0.0	12.7	23.9	5.7	18.2	2.4	1.2	1.2
	×200%	0.0	7.5	19.2	9.8	17.2	6.5	1.0	8.6
Time	×50%	0:00	≥4:00	≥4:00	≥4:00	≥4:00	0:03	0:02	0:03
	×100%	0:00	≥4:00	≥4:00	≥4:00	≥4:00	≥4:00	0:01	≥4:00
	×200%	0:12	≥4:00	≥4:00	≥4:00	≥4:00	≥4:00	0:42	≥4:00

Table 1: Test instance overview and computational results

In 8 out of the 24 test instances, our algorithm terminates with an optimal solution. From an operators point of view, however, proven optimality is not the major concern and it is sufficient to compute solutions which are close to optimality. Therefore, it is also interesting to note that in 17 out of the 24 test instances, the optimality gap is less than 10%. In the remaining cases, our algorithm has not been able to close the gap within the time limit of four hours. However, as it can often be observed for branch-and-cut algorithms, a good or even optimal solution can be computed in relatively short time and the remaining time is needed to close the gap.

Due to the three different demand scalings it is unlikely that the results could be obtained because of a misrelation between the available link capacities and the demand values (results are questionable, for instance, if the smallest capacities are always sufficient). Furthermore, a comparison of the results for the different scalings admits another practically interesting observation: solution cost are not scaling linearly with the demand values. The average cost of the 50% and 200% demand scaling is 83 and 137, respectively. This supports our claim that an accurate model of the modular capacity and cost structure of the available hardware in communication networks is a must since this structure has a significant impact on solutions.

## 5 Conclusions

In this article we suggest a mixed-integer linear programming model to integrate constraints imposed by typical hardware limitations. The computational results achieved with our network planning and optimization tool DISCNET (with a branch-and-cut algorithm at its core) demonstrate the applicability of this model for SDH and opaque WDM networks. We can conclude that it is possible to accurately model real-world network planning problems and

to find solutions of sufficiently high quality in reasonably short time. To close the optimality gap faster, however, additional research on the polyhedral structure is still necessary.

## Acknowledgments

This work was supported by the DFG Research Center “Mathematics for key technologies” (FZT86), Berlin.

## References

- [1] D. Alevras, M. Grötschel, R. Wessäly, “Cost-efficient network synthesis from leased lines”. *Annals of Operations Research*, 76, 1–20, 1998.
- [2] J. Benders, “Partitioning procedures for solving mixed-variables programming problems”. *Numerische Mathematik* 4, 238–252, 1962.
- [3] F. Barahona, “Network design using cut inequalities”. *SIAM Journal on Optimization*, 6(3), 823–837, 1996.
- [4] D. Bienstock, I. Saniee, “ATM network design: traffic models and optimization-based heuristics”. *Telecommunication Systems*, 16, 399–421, 2001.
- [5] D. Bienstock, S. Chopra, O. Günlük, C.-Y. Tsai, “Minimum cost capacity installation for multicommodity network flows”. *Mathematical Programming*, 81, 177–199, 1998.
- [6] D. Bienstock, O. Günlük, “Computational experience with a difficult mixed-integer multicommodity flow problem”. *Mathematical Programming*, 68, 213–237, 1995.
- [7] G. Dahl, M. Stoer, “A Polyhedral Approach to Multicommodity Survivable Network Design”. *Numerische Mathematik*, 68(1), 149–167, 1993.
- [8] O. Günlük, “A branch-and-cut algorithm for capacitated network design”. *Mathematical Programming*, 86, 17–39, 1999.
- [9] M. Grötschel, C. Monma, M. Stoer, “Facets for polyhedra arising in the design of communication networks with low-connectivity constraints”. *SIAM Journal on Optimization*, 2(3), 474–504, 1992.
- [10] M. Iri, “On an extension of the maximum-flow minimum-cut theorem to multicommodity flows”. *Journal of the Operations Research Society of Japan*, 13(3), 129–135, 1971.
- [11] DISCNET, “Network planning and configuration engine.” atesio GmbH, [www.atesio.de](http://www.atesio.de), 2000–2003.
- [12] A. Kröller, “Network optimization: Integration of Hardware Configuration and Capacity Dimensioning”. Diploma thesis, TU Berlin, 2003.
- [13] K. Lee, K. Park, S. Park, “Spare channel assignment for DCS mesh-restorable networks”. *Proceedings of the 3rd International Conference on Telecommunication Systems: Modeling and Analysis*, 296–307, 1995.



- [14] T. Magnanti, P. Mirchandani, R. Vachani, “Modeling and solving the two-facility capacitated network loading problem”. *Mathematical Programming*, 60(2), 233–250, 1995.
- [15] M. Minoux, “Optimum synthesis of a network with non-simultaneous MCF requirements”. *Studies on Graphs and Discrete Programming*, P. Hansen (ed.), North Holland Publishing Company, 269–277, 1981.
- [16] M. Padberg, G. Rinaldi, “A branch and cut algorithm for the resolution of large-scale symmetric traveling salesman problems”. *SIAM Review*, 33, 60–100, 1991.
- [17] F. Poppe, P. Demeester, “The design of SDH mesh-restorable networks: Problem formulation and algorithm”. *Proceedings of the 5th International Conference on Telecommunication Systems, Modeling and Design*, 88–97, 1997.
- [18] M. Stoer, “Design of Survivable Networks”, *Lecture Notes in Mathematics*, Springer, Vol. 1531, 1993.
- [19] R. Wessäly, “Dimensioning Survivable Capacitated NETWORKS”. PhD thesis, TU Berlin, 2000.
- [20] L.A. Wolsey, “Valid inequalities for 0/1 knapsacks and MIPs with Generalised Upper Bound constraints”. *Discrete Applied Mathematics*, 29, 251–261, 1990.