Comparing Restoration Concepts using Optimal Network Configurations with Integrated Hardware and Routing Decisions

Sebastian Orlowski and Roland Wessäly
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Email: {orlowski,wessaely}@zib.de

***Abstract*— We investigate the impact of link and path restoration on the cost of telecommunication networks. The surprising result is the following: the cost of an optimal network configuration is almost independent of the restoration concept if (i) the installation of network elements (ADMs, DXCs, or routers) and interface cards, (ii) link capacities, and (iii) working and restoration routings are simultaneously optimized.**

We present a mixed-integer programming model which integrates all these decisions. Using a branch-and-cut algorithm (with column generation to deal with all potential routing paths), we solve structurally different real-world problem instances and show that the cost of optimal solutions is almost independent of the used restoration concept.

In addition, we optimize spare capacities for given shortest working paths which are predetermined with respect to different link metrics. In comparison to simultaneous optimization of working and restoration routings, it turns out that this approach does not allow to obtain predictably good results.

***Index Terms*— survivable network design, link and path restoration, branch-and-cut algorithm, hardware configuration, routing**

I. Introduction

To secure a network against node and link failures, operators must pick from a variety of protection and restoration concepts. Each of these has its assets and drawbacks with respect to cost, ease of implementation, maintenance effort and recovery time. A good planning decision has to be founded on a thorough analysis of the trade-offs between these competing features.

Based on optimal solutions with respect to a mathematical model that integrates decisions about the topology, hardware (network elements, interface cards), modular link capacities, as well as working and failure routings, we compare the influence of different restoration concepts on the total network cost. We consider a generalization of link restoration (also covering node failures) and path restoration with and without stub release.

Using link restoration, information local to a failing component is sufficient to restore affected traffic, which makes it easier to implement than path restoration where traffic is restored between the end nodes of each affected demand. These two restoration concepts are illustrated in more detail in Section II-B.

This work was supported by the DFG Research Center "Mathematics for key technologies" (FZT86), Berlin, `www.fzt86.de`.

In our computational study on several real-world problem instances stemming from structurally different planning scenarios of SDH-, WDM-, and leased line networks, we reveal that the restoration concept has almost no influence on the total cost of an optimal network configuration. This conclusion can be drawn for single link failures, for single node failures and for the combination of both. In other words, network costs can be a minor criterion in the choice of the restoration concept.

Several authors already compared the influence of the restoration concept on network cost and concluded with results different from ours. Two categories of comparisons can be distinguished: those which solely optimize spare capacity based on predetermined working paths (usually some shortest path between the demand end nodes) and those which jointly optimize the working and failure routings. The following literature review is restricted to cost comparisons between link and path restoration and does *not* cover the numerous studies for a particular restoration concept.

Kennington, Nair, Spiride [1], Poppe, Demeester [2] and Doucette, Grover [3] compare the spare capacity requirements for given working paths. In [1], modular link capacities and single link failures are considered. The authors suggest a branch-and-cut-approach based on an arc-flow formulation for the failure routings. They report that for the eight problem instances which could be solved to optimality (all of which are artificial networks), link restoration requires on average 12% more spare capacity than path restoration without stub release.

In [2], the problem of installing continuous spare capacities is formulated with so-called metric inequalities. These inequalities are generated at run-time using a path-flow formulation of the restoration problem that is solved with a column generation procedure. The presented formulation provides an abstraction of the particular restoration concept and includes node failures. The authors report on considerably different spare capacity requirements for link and path restoration, and demonstrate on three real-world networks (with two demand patterns each) that node failures are less expensive than link failures.

A comparison of various protection and restoration mechanisms is given in [3]. While link restoration is considered with and without given working paths, the path restoration version (with stub release) relies on a predetermined shortest path routing. The authors use a path-flow formulation with

arbitrary integer capacities and a predetermined set of paths. In their comparison based on 18 networks of different density, all originating from the same master topology, they found link restoration to be at least 10% more expensive than path restoration with stub release (for given working paths), independent of network density.

Contrary to the previously discussed papers, Murakami [4], Xiong, Mason [5], Caenegem, Wauters, Demeester [6], and Iraschko, MacGregor, Grover [7], perform a joint optimization of the working and the failure routing.

In [4], a path-flow formulation with continuous link capacities is used. A column generation procedure for missing routing paths is suggested as solution method. Based on computational tests with four realistic and four artificial networks and generated demand requirements, the author reports that with predefined shortest working paths, optimal solutions are 5–25% more expensive than those where the working and failure paths are jointly optimized. The additional cost for link restoration compared to path restoration (with stub release) varies widely but is beyond 10% of the path restoration cost in most cases. With a predetermined shortest working path routing, the advantages of path restoration are even higher. The cost differences between path restoration and link restoration as well as between joint and non-joint working and spare capacity optimization are reported to be higher for the artificial networks than for the real-world networks.

A similar model is presented in [5], but instead of using column generation, a set of hop-limited path variables is precalculated. The authors compare path restoration without stub release and link restoration under a single link failure scenario on one artificial and two realistic networks. With joint working and spare capacity optimization, they obtain almost the same network cost for link and path restoration in all three test instances; with spare capacity optimization based on a predetermined shortest working path routing, they found the difference to be higher (about 5–10%). Both results coincide with our observations.

In [6], path restoration without stub release and link restoration of single link failures are compared for given shortest working paths. In contrast to [4] and [5], the admissible link capacities as well as the flow values are integers. The authors suggest a simulated annealing algorithm to compute low cost solutions. In order to avoid unacceptably long computation times, the set of eligible restoration routes is restricted to the 10 shortest paths for each demand. On one network with two different demand patterns (uniform and estimated) this heuristic yields results where link restoration is 20-25% more expensive than path restoration.

In [7], the cost of link and path restoration (with or without stub release) for single link failures is compared both with a predefined shortest path routing and with joint working and spare capacity optimization. Capacities and flow variables in the path-flow model are allowed to take any integer value. The authors present computational results on five (real-world based and artificial) test instances using a predetermined path set. With spare capacity optimization based on a predefined routing, they obtain about 5–25% higher cost for link restoration than for path restoration with stub release, whereas with joint optimization, the differences are generally found to be lower.

As already mentioned, most of these articles report on substantial cost differences between the restoration concepts, contrary to our results. There are several reasonable explanations for this discrepancy. To our knowledge, there is no comparison of restoration concepts yet which is based on an accurate model for the modular cost and capacity structure of nowadays hardware and, at the same time, a joint optimization of working and failure routings. Furthermore, despite the more complex mathematical model, we are still able to make our comparison based on optimal solutions since the branch-and-cut algorithm in conjunction with a column generation procedure (both implemented in our network planning tool DISCNET [8]) provides accurate lower bounds for the minimal network costs.

An additional computational study presented in this article reveals a slightly stronger influence of the restoration concept on the network cost for the case of predefined working paths. But also in this case, the cost difference between link and path restoration rarely exceeds 10%.

However, the way *how* working paths are predefined has a significant impact on the solution quality. In particular, upon calculation of some shortest working paths, the results heavily depend on the used link weights and on the implementation of the shortest path algorithm. In the worst case, a feasible restoration problem can be made infeasible by a wrong choice of predefined working paths.

Since it is widespread in the literature to use some predefined shortest path routing, we compare five different strategies (including shortest hop and shortest length routing) to obtain some rule of thumb for finding "good" working paths. Our results reveal that joint optimization of working and failure paths is a must: the additional cost of using a predefined shortest path routing ranges between 0 and 164% and is thus more or less unpredictable. Furthermore, none of the tested link weights clearly outperforms all others.

This paper is organized as follows. In Section II, we present a unifying mathematical model which covers the configuration of hardware and link capacities and which abstracts from both the restoration concepts and the considered failure scenarios. Parameterizing this model appropriately, all mentioned restoration concepts (and others like reservation [9] and meta-mesh [10]) as well as arbitrary network component failures (link-, node-, or multi-failures) can be formulated. In Section III, we sketch our branch-and-cut algorithm and briefly describe a column generation procedure to implicitly deal with *all* potential routing paths. In Section IV, we present our quantitative results in more detail and eventually, we conclude with Section V.

II. Problem and Mathematical Model

We investigate network design problems dealing with an integrated planning of

- a topology,

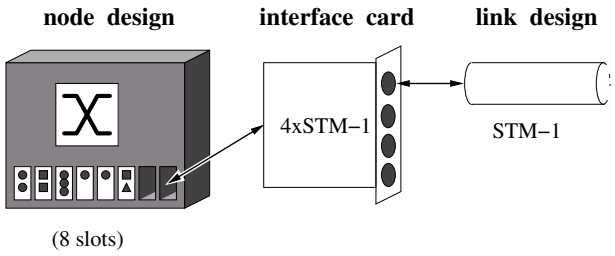

Fig. 1. Hardware configuration example.

- a hardware configuration,
- modular link capacities,
- a routing during normal operation, and
- routings for all single link and node failure situations.

The survivable routing must respect one of the restoration concepts under investigation. Our model consists of two parts, connected by the link capacities: one comprising hardware constraints (see Section II-A) and the other formulating routing and restoration restrictions (see Section II-B). The objective is to minimize the total cost of installing hardware and link capacities (see Section II-C).

A. Topology, Hardware Configuration and Link Capacities

Given are all potential node locations and point-to-point links which may be included in the final topology. For SDH networks, these links are typically restricted to existing fiber cables, while for leased line networks all point-to-point links may be admissible. This potential network is modeled by an undirected *supply graph* $G = (V, E)$, where V is the set of node locations and E is the set of links.

For each node location $v \in V$, a list of potential *node designs* $\mathcal{D}(v)$ and a list of potential *interface cards* $\mathcal{M}(v)$ can be specified. Each node design $d \in \mathcal{D}(v)$ is characterized by its maximum switching capacity C_v^d and the number of slots S^d for interface cards. At most M_m^d interface cards of type $m \in \mathcal{M}(v)$ can be installed at node design d. Each interface card m provides I_i^m interfaces of type $i \in \mathcal{I}$ (the set of all interfaces) and requires S^m many slots if used in some node design.

Similarly, for each admissible link $e \in E$, a list of potential *link designs* $\mathcal{D}(e)$ can be specified. Each link design $d \in \mathcal{D}(e)$ consumes a capacity C_e^d as well as I_i^d interfaces of type $i \in \mathcal{I}$ at each end node of its link.

Example: Figure 1 shows an example of a link design with a capacity of STM-1 which is attached to an interface card providing 4 STM-1 interfaces. This card is plugged into one out of 8 slots of the node design.

Although we have extended the following model to cope with an existing hardware infrastructure in our network design tool DISCNET [8], such enhancements are omitted in this paper for the sake of simplicity.

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

HARDWARE:

$$\sum_{d \in \mathcal{D}(v)} x_v^d \leq 1 \quad v \in V \tag{1}$$

$$\sum_{d \in \mathcal{D}(e)} x_e^d \leq 1 \quad e \in E \tag{2}$$

$$\sum_{e \in \delta(v)} \sum_{d \in \mathcal{D}(e)} I_i^d x_e^d - \sum_{m \in \mathcal{M}(v)} I_i^m x_v^m \leq 0 \quad \begin{matrix} v \in V, \\ i \in \mathcal{I} \end{matrix} \tag{3}$$

$$\sum_{e \in \delta(v)} \sum_{d \in \mathcal{D}(e)} C_e^d x_e^d - \sum_{d \in \mathcal{D}(v)} C_v^d x_v^d \leq 0 \quad v \in V \tag{4}$$

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

$$x_v^m - \sum_{d \in \mathcal{D}(v)} M_m^d x_v^d \leq 0 \quad \begin{matrix} v \in V, \\ m \in \mathcal{M}(v) \end{matrix} \tag{6}$$

Inequalities (1) and (2) state that at most one design must be chosen for each node and each link; the final topology consists of those graph elements where exactly one design is chosen, all other nodes and links are not included in the topology. A given full or partial physical topology can be respected by forcing equality in the respective inequalities (1) and (2). 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 interface cards is not exceeded.

B. Routing and Restoration

In addition to topology and hardware decisions, a feasible survivable network design comprises a routing for every considered operating state. The set S of operating states is composed of the *normal operating state* (NOS) and a subset S^* of all single node and single link failures, the so-called *failure states*. We denote by $s = 0$ the normal operating state, by $s = v$ the failure of node $v \in V$, and by $s = e$ the failure of link $e \in E$. Notice that the failure of a node $v \in V$ implicitly induces the failure of all its incident links $e \in \delta(v)$. The set $V^s \subseteq V$ consists of all non-failing nodes in operating state $s \in S$. Similarly, $E^s \subseteq E$ contains all usable links in s, where a link is considered usable if neither the link itself nor one of its end nodes fail.

For the normal operating state, a set $\mathcal{D}$ of communication demands is given. With each demand $uv \in \mathcal{D}$, two parameters are associated: the demand value d_{uv} 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), and a path length restriction (also called hop limit) ℓ_{uv} which specifies the maximum number of links of an admissible path between u and v during normal operation.

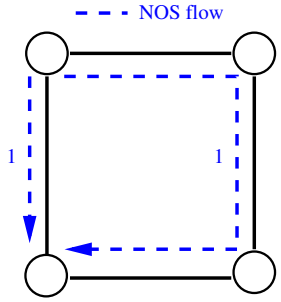

Fig. 2. Example NOS network.

Figure 2 shows the normal operating state routing subsequently used to illustrate the different restoration concepts. In this example, one demand of value 2 is routed on two paths.

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 $uv \in \mathcal{D}$ in the normal operating state. These are all paths between u and v satisfying the path length restriction ℓ_{uv}. 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 capacity constraints (7) and demand constraints (8) formulate a multi-commodity flow problem with path length restrictions for the normal operating state:

ROUTING (NOS):

$$\sum_{d \in \mathcal{D}(e)} C_e^d x_e^d - \sum_{uv \in \mathcal{D}} \sum_{\substack{P \in \mathcal{P}_{uv} \\ e \in P}} f_{uv}(P) \quad \geq 0 \qquad e \in E \tag{7}$$

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

We distinguish between three different restoration concepts to protect the network against the failure of single network components.

a) Link/Local restoration (LR):

Each failing path is locally restored. In case of a failing link, each affected path is restored between the end nodes of the link. In case of a node failure, each failing path which does not start or terminate in the failing node is restored between the two nodes on the path which are adjacent to the failing node. An example for both cases is provided in Figure 3. Notice that this notion is slightly more general than the common link restoration concept since node failures are covered as well.

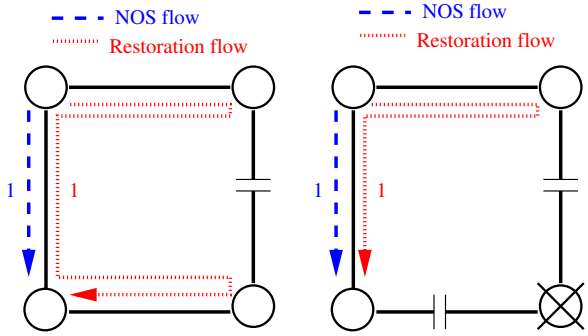

Fig. 3. Link restoration, link and node failure.

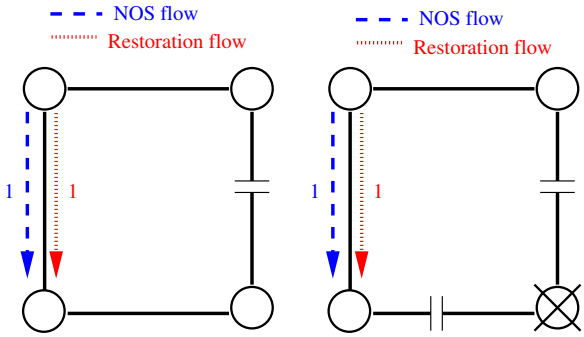

Fig. 4. Path restoration, link and node failure.

b) Path restoration without stub release (PR-sr):

Each failing path is globally restored between its end nodes. Link capacities reserved for normal operation are not released and cannot be used for restoration flow. An example for path restoration of a link and a node failure is shown in Figure 4.

c) Path restoration with stub release (PR+sr):

With stub release, capacity is released on the non-failing parts (stubs) of failing working paths, and may be reused for restoration flow.

Now, we present a generic mathematical formulation which is able to cope with all three restoration concepts, if parameterized appropriately. Although we concentrate on failures of single network components and on the three presented restoration concepts in this paper, the model also allows to formulate multiple link and node failures or other restoration concepts like *reservation* (see [9], [11], [12]), where the NOS routing is completely discarded and re-planned in case of a failure, and mixtures of link and path restoration like *meta-mesh* [10].

For each failure state $s \in S^*$, we introduce a set C^s of *failure commodities*. These are point-to-point demands to be satisfied in failure state s. Let $\mathcal{P}_c^s$ be the set of all paths between the end nodes of failure commodity $c \in C^s$. Furthermore, let $\mathcal{IP}_c^s$ be the set of NOS paths which are to be restored by failure commodity $c \in C^s$; these are the so-called *interrupted paths*. See Table I for a precise definition of failure commodities and interrupted paths for some failure state s. The stub release parameter $\gamma^s(P) \in \{0, 1\}$ indicates

TABLE I

RESTORATION DEPENDENT SETTINGS

Model	C^s	$\mathcal{IP}_c^s,\ c \in C^s$
LR ($s = e$)	$\{uw : e = uw\}$	$\{P \in \mathcal{P} : e \in P\}$
LR ($s = v$)	$\{uw : uv, vw \in \delta(v)\}$	$\{P \in \mathcal{P} : u, v, w \in P\}$
PR-sr/PR+sr	$\{uv : uv \in \mathcal{D},\ u, v \in V^s\}$	$\{P \in \mathcal{P}_{uv} : s \in P\}$

for every path $P \in \mathcal{P}$ and every failure state $s \in S^*$ whether NOS capacity has to be reserved on the non-failing links of P in s. Hence, $\gamma^s(P)$ is always 1 for LR and PR-sr, while for PR+sr, $\gamma^s(P) = 1$ if and only if P survives in failure state s.

Using non-negative continuous flow variables $f_c^s(P) \in \mathbb{R}_+$ for all failure states $s \in S^*$, all failure commodities $c \in C^s$ and all admissible restoration paths $P \in \mathcal{P}_c^s$, the following capacity constraints (9) and restorability constraints (10) formulate a restoration problem for all failure states $s \in S^*$:

ROUTING (failure states):

$$\sum_{d \in \mathcal{D}(e)} C_e^d x_e^d - \sum_{c \in C^s} \sum_{\substack{P \in \mathcal{P}_c^s: \\ e \in P}} f_c^s(P) - \sum_{uv \in \mathcal{D}} \sum_{\substack{P \in \mathcal{P}_{uv}^0 \\ e \in P}} \gamma^s(P) f_{uv}(P) \geq 0 \quad \begin{matrix} s \in S^*, \\ e \in E^s \end{matrix} \tag{9}$$

$$\sum_{P \in \mathcal{P}_c^s} f_c^s(P) - \sum_{uv \in \mathcal{D}} \sum_{P \in \mathcal{IP}_c^s} f_{uv}(P) \geq 0 \quad \begin{matrix} s \in S^*, \\ c \in C^s \end{matrix} \tag{10}$$

Note that the model can easily be extended to partial restoration of the failing flow by changing the coefficient of the NOS flow variables in (10).

C. 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_v^d and equipping this node design with an interface card $m \in \mathcal{M}(v)$ incurs a cost of K_v^m. Similarly, for each link $e \in E$, the installation of link design $d \in \mathcal{D}(e)$ incurs a cost of K_e^d. Putting this together, the objective function is

$$\min \sum_{v \in V} \left(\sum_{d \in \mathcal{D}(v)} K_v^d x_v^d + \sum_{m \in \mathcal{M}(v)} K_v^m x_v^m \right) + \sum_{e \in E} \sum_{d \in \mathcal{D}(e)} K_e^d x_e^d \, .$$

Our branch-and-cut algorithm is able to compute optimal solutions with respect to the objective of Section II-C and constraints (1)–(10), simultaneously deciding the topology, the hardware configuration, the link capacities and a routing for the NOS and all failure states.

III. A Branch-and-Cut Algorithm

Our solution approach is similar to Benders decomposition [13]. The central procedure is a branch-and-cut algorithm (see [14] for a detailed description) based on a relaxation of the problem described in Section II, which consists of the hardware configuration constraints (1)–(6).

To strengthen our formulation at each node of the branch-and-cut tree, we try to separate valid inequalities violated by the optimal solution of the current relaxation. As cutting planes, we use band inequalities [9], GUB cover inequalities [15], and generalizations [12], [16] of metric inequalities [17].

Each time an integer hardware configuration is identified which is feasible for the relaxation, it is tested for feasibility with respect to the (missing) routing constraints. If the hardware configuration allows a feasible routing, one is generated; otherwise, a violated metric inequality is added to the relaxation in order to cut off the infeasible hardware configuration.

The subroutine used to test feasibility of integer hardware configurations and to separate metric inequalities is based on a linear program consisting of constraints (7)–(10) with fixed link capacities, see [9], [12], [16]. Since this formulation has an exponential number of path variables, we apply a column generation procedure, starting with a small subset of all possible variables. The initial paths are typically short paths with respect to some link metric (hop count, length, etc.). Further path variables (columns) are generated whenever necessary by computing shortest (hop-limited) paths with respect to link weights derived from an optimal dual solution of the linear program.

For path restoration without stub release and link restoration, the column generation problem can be solved exactly, that is, path variables necessary to prove feasibility of some link capacities can be identified in polynomial time. For path restoration with stub release, this problem is $\mathcal{NP}$-hard [16]. When solving the column generation problem approximately, feasible capacity vectors may accidently be rejected. In a thorough computational study, however, the best solution value obtained has always been the same for various initializations of the initial path set, heuristics to set up the link weights, and algorithms to compute short paths. Consequently, even for path restoration with stub release there is a high probability that no optimal capacity vector has been rejected and a globally optimal solution has been identified.

IV. Computational Results

In this section, we report on the results of numerical tests on 10 real-world based instances stemming from SDH-, WDM-, and leased line planning problems. After a short presentation of the test instances in Section IV-A, we provide a comparison of optimal network costs using link or path restoration in Section IV-B. Finally, Section IV-C shows our results using predetermined shortest working paths with respect to different link weights.

A. General

For each test instance, Table II shows the number of potential nodes and links, the number of demands, together with the average node degree $\bar{d} = 2|E|/|V|$, as well as the number of available link designs (#ld) and node designs (#nd). The number of designs is the same for all links and nodes of an instance, respectively. The last column in the table shows the hop limit ℓ, which is the same for all demands of an instance.

The instances reflect the variety of capacity and cost structures of nowadays technologies. For optical networks, for example, the modularities of WDMs and OXCs are considered; for SDH-networks, different interface cards (1xSTM1, 4xSTM1, 1xSTM4, or 1xSTM16) and DXCs are taken into account. Eventually, the cost structures for leased line networks exhibit economies of scale with respect to capacity and length of a link. Two particularities of the test instances are worth mentioning. First, instance e6 is the same as e5 with six

TABLE II

CHARACTERISTICS OF THE TEST INSTANCES

Name	$\|V\|$	$\|E\|$	$\|D\|$	$\bar{d}$	#ld	#nd	ℓ
e1	10	25	29	5.0	2	1	4
e2	12	18	27	3.0	4	1	∞
e3	15	21	13	2.8	3	1	∞
e4	15	22	105	2.9	7	1	∞
e5	18	21	62	2.3	9	1	5
e6	18	27	62	3.0	9	1	∞
e7	20	28	119	2.8	6	2	6
e8	14	21	91	3.0	5	1	∞
e9	12	18	27	3.0	9	6	∞
e10	12	18	27	3.0	5	2	∞

additional links but without hop limits, and second, instances e9 and e10 have the same underlying planning topology and demand pattern as e2, but significantly different cost and capacity structures.

All numerical tests are performed on undirected networks, assuming full restoration of all single node failures, all single link failures, or both. To avoid side effects from a disproportion between the demand values and the capacities of the available node and link designs, three test series are performed, where all demands are scaled by 0.5, 1.0 or 2.0, respectively.

The computation times range between a few seconds and several hours on a Linux machine with 1 Gigabyte of RAM and a 1.7 Mhz processor; typically, the problems can be solved to optimality within an hour.

B. Comparison of Optimal Network Costs

Table III shows the optimal network cost for all demand scaling factors and for the three considered restoration concepts: link restoration (LR), path restoration without stub release (PR-sr) and path restoration with stub release (PR+sr).

For each demand scaling factor, three columns are presented: for all single node failures, all single link failures, and both. All solution values are scaled such that 100 corresponds to the optimal network cost without any survivability requirements and demand scaling factor 1.0. Unless stated otherwise, this scaling and rounding implies that two identical entries in the table might actually differ up to 1%.

Table III allows several interesting observations:

1) The restoration concept has only minor influence on the network cost: in 68 out of 90 cases (75%), *the minimal network cost is the same for all restoration concepts* (even before scaling and rounding). Furthermore, in only six cases, the most cost-intensive restoration concept is above 5% more expensive than the cheapest one, and in only one out of 90 cases (e6 with scaling factor 1.0 and all single failures), the difference exceeds 10%.
2) Securing a network only against single *link* failures is often as expensive as securing it against *all* single failures, while single *node* failures are cheapest in most cases. In fact, single link failure solutions often turned out to be feasible for all single failures as well. Notice that the cost relation between these failure scenarios may change significantly with different demand values (see e2, e4 and e8, for instance).
3) Given that 100 corresponds to the optimal network cost without any survivability requirements and demand scaling factor 1.0, the additional cost to achieve full restoration of single link failures ranges between 25% (e6) and 132% (e4). Thus, it is difficult to estimate the spare capacity cost relative to the working capacity cost in advance. With slightly different percentages, the same result applies to all failure scenarios.
4) A comparison of those instances with the same underlying planning topology and demand patterns but different capacity and cost structure (e2, e9 and e10) reveals that the only common property is the invariance of network cost with respect to the restoration concept. On the contrary, the cheapest failure scenario, the increase in network cost when scaling the demands, and the cost

TABLE III

OPTIMAL COST USING DIFFERENT RESTORATION CONCEPTS

Name	rest.	factor 0.5			factor 1.0			factor 2.0		
	type	node	link	both	node	link	both	node	link	both
e1	LR	125	128	128	136	138	138	175	186	186
	PR-sr	125	128	128	136	136	136	175	180	180
	PR+sr	125	128	128	136	136	136	172	177	177
e2	LR	100	112	112	183	140	183	232	233	232
	PR-sr	100	112	112	183	140	183	232	233	232
	PR+sr	100	112	112	183	140	183	232	233	232
e3	LR	112	112	112	122	148	148	156	199	199
	PR-sr	112	112	112	122	139	139	156	217	217
	PR+sr	112	112	112	122	139	139	156	199	199
e4	LR	109	98	112	231	232	236	510	508	512
	PR-sr	109	96	111	231	232	236	510	508	512
	PR+sr	107	96	111	231	232	235	510	508	512
e5	LR	119	119	119	145	145	145	252	280	280
	PR-sr	119	119	119	145	145	145	252	280	280
	PR+sr	119	119	119	145	145	145	251	280	280
e6	LR	103	103	103	124	128	145	187	207	208
	PR-sr	103	103	103	124	128	129	187	207	208
	PR+sr	103	103	103	124	125	125	183	207	208
e7	LR	130	146	150	132	157	157	133	187	187
	PR-sr	130	146	150	132	157	157	133	187	187
	PR+sr	130	146	150	132	157	157	133	187	187
e8	LR	100	101	101	159	149	165	300	301	301
	PR-sr	100	101	101	154	147	163	300	301	301
	PR+sr	100	101	101	154	147	163	300	301	301
e9	LR	100	100	100	159	142	174	248	216	300
	PR-sr	100	100	100	159	142	174	248	216	296
	PR+sr	100	100	100	159	142	174	248	216	300
e10	LR	100	103	103	127	137	141	162	170	180
	PR-sr	100	103	103	127	137	141	162	170	180
	PR+sr	100	103	103	127	137	141	162	170	180

incurred for full restoration vary significantly among these instances.

5) In 4 instances (e2, e8, e9 and e10), the costs for node failure restoration with demand factor 0.5 appear to be the same as for factor 1.0 without restoration. However, in 3 out of these 4 cases, the optimal costs *before* scaling and rounding are *not* identical but within the 1% range.

The second observation is in accordance with the results of [2] who also observed node failures to be cheaper than link failures in most cases. On the contrary, our first observation is in contrast to many previously published comparisons, as discussed in Section I. We see two main reasons for this discrepancy:

1) As described in Section II, our mathematical model covers hardware aspects with discrete capacity and cost structures as well as joint optimization of working and failure routings and topology planning.
2) We compare *optimal* values computed by a branch-and-cut algorithm, combined with column generation to deal with *all* admissible routing paths, instead of using only heuristics or fixing a small set of eligible routing paths in advance. Although we do not know of any thorough computational study on the influence of hop limits on the cost of optimal solutions, we observed in practical planning instances that it may be significant.

Some of the comparisons referenced in the introduction explicitly note that part of the input is artificial. Even though we are aware of the difficulties to judge from the description of the problem instances used in any computational study how realistic these are, we remark that another reason for the discrepancy might be that all of our instances are based on real-world planning instances. One might also suspect as a reason that in our instances, node cost is largely dominating total network cost. However, this is not the case. Node costs occur in only 3 out of 10 instances, and for these 3 instances (e7, e9, and e10), the ratios of total node costs to total link costs are 10, 5, and 130 percent, respectively.

C. Shortest path NOS Routing

An analysis of the routings of the optimal solutions presented in Table III revealed that most demands are routed on only one path, few demands on two paths, and almost none on more than two paths. These paths were often shortest paths with respect to the length in kilometers in the final topology of an optimal network but not necessarily in the original planning topology.

This observation raises the following question: does there exist some link metric such that a shortest working path routing in the *planning* topology and a spare capacity optimization based on this routing leads to reliably good solutions compared to joint optimization of working and spare capacity?

An affirmative answer would have several nice implications. First, such a link metric would give the planner a rule of thumb how to define the working routing. Second, the computation times for the spare capacity assignment are substantially smaller than with joint optimization since the routing problem decomposes into one small subproblem for each considered failure state. Therefore, much larger problem instances could be solved close to optimality.

TABLE IV
COST WITH PREDEFINED SHORTEST PATH NOS ROUTING

Name	rest.	$c_e := l(e)$			$c_e := 1$			$c_e := (1 + l(e))^2$		
	type	node	link	both	node	link	both	node	link	both
e1	LR	360	360	360	296	296	296	189	184	189
	PR-sr	360	360	360	296	296	296	187	169	187
	PR+sr	360	360	360	296	296	296	187	169	187
e2	LR	206	152	206	195	161	206	199	156	199
	PR-sr	195	152	206	195	161	206	195	156	199
	PR+sr	195	152	206	195	161	206	195	152	195
e3	LR	129	167	167	152	184	184	145	163	177
	PR-sr	129	148	148	152	184	184	145	161	162
	PR+sr	129	147	147	152	184	170	145	161	162
e4	LR	249	261	266	240	264	250	274	285	285
	PR-sr	248	243	254	240	244	246	265	258	268
	PR+sr	241	243	253	233	239	239	254	254	265
e5	LR	145	159	159	145	151	151	150	170	170
	PR-sr	145	145	145	145	145	145	145	145	145
	PR+sr	145	145	145	145	145	145	145	145	145
e6	LR	173	174	174	172	172	172	156	166	166
	PR-sr	173	173	173	172	172	172	156	156	156
	PR+sr	173	173	173	172	172	172	156	156	156
e7	LR	165	165	178	162	169	169	176	176	188
	PR-sr	164	165	177	162	169	169	176	176	188
	PR+sr	164	165	177	162	169	169	175	175	188
e8	LR	178	162	178	178	162	178	178	162	178
	PR-sr	178	162	178	178	162	178	178	162	178
	PR+sr	178	162	178	178	162	178	178	162	178
e9	LR	159	142	174	172	162	186	160	156	188
	PR-sr	159	142	174	172	162	186	160	142	174
	PR+sr	159	142	174	172	162	186	160	142	174
e10	LR	129	140	142	139	142	145	128	142	144
	PR-sr	129	140	142	139	142	145	128	140	142
	PR+sr	129	140	142	139	142	145	128	140	142

Trying to find such a link metric, we first compare link weights c_e proposed by other authors. We set c_e to either the length in kilometers $l(e)$ of a link (shortest length path) or to uniform lengths 1 (shortest hop path).

All values have been calculated with unscaled demands (factor 1.0) and full failure path pricing, that is, the obtained spare capacity assignment is optimal with respect to the given NOS routing. The values in Table IV are scaled by the same factors as in Table III, such that 100 is the minimal network cost without survivability requirements.

As can be seen from Table IV, the quality of the solutions obtained by fixing a shortest length or shortest hop routing is often poor compared to the optimal values from Table III. The additional cost from fixing a shortest working path routing (defined as *(shortest-path-result – joint) / joint*) ranges from 0% (for e5, e9) up to 164% (for e1), with an average of 24%

(without e1: 11%). Otherwise stated, although it is possible to obtain a globally optimal configuration with a predefined shortest hop or shortest length routing, the results are far from being predictable.

Instance e1 illustrates that this approach can yield almost arbitrarily bad results: with both shortest hop and shortest length routing, overall network cost is far more than twice the optimal value using joint optimization! In this network, only about half of the available links are actually needed in a cost optimal solution. However, a shortest hop or shortest length routing (as any other shortest path routing with respect to link weights defining a metric on the nodes) always chooses the direct link between two demand nodes if available, leading to some positive capacity on every link. Similar observations hold for e6.

Even though with our test instances, the cost of link and path restoration is still the same in 50 out of 90 cases with a predefined shortest path routing (and nearly the same in another 12 cases), it is interesting to note that the cost difference tends to be higher than with joint working and spare capacity optimization. This confirms the corresponding results of [5] and [7]; at the same time, it could explain why other authors who optimized spare capacity with respect to a given NOS routing found link restoration to be usually more expensive than path restoration.

Given that the solution quality with a shortest hop or shortest length routing is quite unpredictable, we now compare with other (non-metric) link weights. Let $avg := 1/|E| \sum_{e \in E} l(e)$ be the average length of a link, and let $M := 5 \max_{e \in E} l(e)$. With these definitions, we test the link weights

1) $c_e := (1 + l(e))^2$,
2) $c_e := (1 + l(e))^3$,
3) $c_e := M + l(e)$ if $l(e) > avg$, $c_e := l(e)$ else.

The first two choices are just meant to make the link weights non-metric while still reflecting the length in kilometers of a link. The choice $M + l(e)$ intends to make long links even longer, in order to give the shortest path algorithm an incentive to make a detour instead of using a long link.

In our computational studies, the link weights $(1 + l(e))^2$, $l(e)$, and 1 perform best on average. Hence, the third column of Table IV shows the results for $c_e := (1 + l(e))^2$; the values for the other link weights are omitted here. Even though none of the three presented link weights clearly dominates the others, the weights $l(e)$ performs quite well in 6 out of our 10 instances. Given the short computation times with a fixed NOS routing, it is advisable to try different link weights and to take the best one. However, none of the tested link weights can be used to obtain predictably good solutions.

V. Conclusion

We have presented a mixed-integer programming model for the network restoration problem which integrates topology, hardware, capacity and routing decisions for all operating states simultaneously. Based on a branch-and-cut algorithm together with a column generation procedure, we have given a comparison of optimal network cost for 10 real-world problem instances using link or path restoration (with or without stub release), for different failure scenarios.

Our main observation is that the optimal network cost is almost independent of the restoration concept. From a practical point of view, this implies that the technological decision for a particular concept should be dominated by other criteria than cost.

In addition, we optimized spare capacity based on a given shortest working path routing with respect to different link weights. However, none of the tested link weights led to reliably good results compared to joint working and spare capacity optimization.

References

[1] J.L. Kennington, V.S.S. Nair and G. Spiride: "Optimal spare capacity assignment for path restorable mesh networks: cuts, decomposition, and an empirical analysis," Technical Report 98-CSE-11, Dept. of Comp. Sci. and Eng., Southern Methodist University, Dallas, 1998.

[2] F. Poppe and P. Demeester: "Economic allocation of spare capacity in mesh-restorable networks: models and algorithms," *6th Int. Conf. on Telecommunication Systems*, Modeling and Analysis, Nashville, pp. 77-86, March 1998.

[3] J. Doucette and W.D. Grover: "Comparison of mesh protection and restoration schemes and the dependency on graph connectivity." *3rd International Workshop on Design of Reliable Communication Networks (DRCN 2001)*, pp. 121-128, October 2001.

[4] K. Murakami: "Survivable network management for high-speed ATM networks," PhD thesis, Carnegie-Mellon University, 1995.

[5] Y. Xiong, L. Mason: "Restoration strategies and spare capacity requirements in self-healing ATM networks," *IEEE Infocom '97*, pp. 353-360, 1997.

[6] B. Van Caenegem, N. Wauters and P. Demeester: "Spare capacity assignment for different restoration strategies in mesh survivable networks." *IEEE International Conference on Communications 1997, pp. 288-292*

[7] R. Iraschko, M. MacGregor, W.D. Grover: "Optimal capacity placement for path restoration in STM or ATM mesh survivable networks," *IEEE/ACM Transactions on Networking* vol. 6, no. 3, pp. 325-336, 1998.

[8] DISCNET: "Network planning and configuration engine." atesio GmbH, `www.atesio.de`, 2000-2003.

[9] G. Dahl and M. Stoer: "A polyhedral approach to multicommodity survivable network design," *Numerische Mathematik* 68(1) pp. 149-167, ZIB-Report SC-93-09, `www.zib.de`, 1993.

[10] J. Doucette and W.D. Grover: "Increasing the efficiency of span-restorable mesh networks on low-connectivity graphs," *3rd International Workshop on Design of Reliable Communication Networks (DRCN 2001)*, Budapest, pp. 99-106, 2001.

[11] M. Minoux: "Optimum synthesis of a network with non-simultaneous MCF requirements," *Studies on Graphs and Discrete Programming*, P. Hansen (ed.), pp. 269-277. North Holland Publishing Company 1981.

[12] R. Wessäly: "DImensioning Survivable Capacitated NETworks," PhD thesis, TU Berlin, 2000.

[13] J. Benders: "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik* 4(1): pp. 238-252, 1962.

[14] M. Padberg, G. Rinaldi: "A branch and cut algorithm for the resolution of large-scale symmetric traveling salesman problems," *SIAM Review* no. 33, pp. 60-100, 1991.

[15] L.A. Wolsey: "Valid inequalities for 0/1 knapsacks and MIPs with Generalised Upper Bound constraints," *Discrete Applied Mathematics* no. 29, pp. 251-261, 1990.

[16] S. Orlowski: "Local and global restoration of node and link failures in telecommunication networks," Diploma thesis, TU Berlin, 2003.

[17] 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.