

Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

SEBASTIAN ORLOWSKI
ROLAND WESSÄLY

The Effect of Hop Limits on Optimal Cost in Survivable Network Design

The Effect of Hop Limits on Optimal Cost in Survivable Network Design

Sebastian Orlowski* Roland Wessälly†

Abstract

We investigate the impact of hop-limited routing paths on the total cost of a telecommunication network. For different survivability settings (dedicated protection, link and path restoration), the optimal network cost without restrictions on the admissible path set is compared to the results obtained with two strategies to impose hop limits on routing paths.

In a thorough computational study on optimal solutions for nine real-world based problem instances, we show that hop limits should be avoided if the technology allows it and network cost is a major planning issue. In this case, column generation should be employed to deal with all routing paths. If hop-limits are required, these should be defined for each demand individually and as large as possible.

Keywords: survivable network design, hop limits, routing, restoration, branch-and-cut algorithm, mixed-integer programming

Mathematics Subject Classification (2000): 90C57, 90C11, 68M10

1 Introduction

In telecommunication network design, it is well-established to use short paths for routing communication demands. This approach is some cases motivated by the underlying routing protocol, such as OSPF, IS-IS, or BGP in IP-networks, or by technological restrictions, such as degradation of signal quality in WDM networks or transmission delay in ATM networks. In other cases, the purpose of a restriction to short routing paths is to reduce the planning complexity.

Each individual demand consumes the least bandwidth in the network if it is routed on a shortest path. However, a shortest path routing is not necessarily cost minimal as soon as modular link capacities are taken into account. Furthermore, the reduction of the solution space caused by a restricted path set potentially increases network cost. The question is: *how much* additional cost is incurred by a restriction to short paths? It is often assumed that a small, well-chosen set of short routing paths is sufficient to reliably obtain near-optimal solutions. However, no published computational study evaluates the necessary additional investment based on provably optimal solutions for different survivability mechanisms. This is the gap we fill with this paper.

On nine real-world based problem instances, we compare the optimal network cost with different kinds of admissible path sets: without any restrictions, with demand-independent

*Konrad-Zuse-Zentrum für Informationstechnik Berlin, orlowski@zib.de

†Konrad-Zuse-Zentrum für Informationstechnik Berlin, wessaely@zib.de

hop limits, and with demand-dependent hop limits. Each of these path sets is tested with different survivability settings (dedicated protection, link restoration and path restoration), and with different hop limits. The main conclusions drawn from our experiments are the following:

- Refrain from setting hop limits unless there is a technological reason for using short paths only. A restriction to short paths can lead to infeasibility of the planning problem or to very high network cost on practical instances. On the other hand, it is possible to cope with all routing paths computationally with a column generation approach.
- With demand-independent hop limits, paths with at least 6 links are needed for networks with 10–20 nodes to reliably obtain a solution which is at most 10% more expensive than an optimal solution with respect to all routing paths.
- If hop limits are really needed, they should be demand-dependent, e.g., based on the shortest path length between the two end nodes of the demand. However, at least 4 hops should be allowed in addition to the shortest hop count for a particular demand. With smaller hop limits, it is not possible to reliably obtain solutions with at most 10% additional cost compared to an optimal solution with respect to all routing paths. In particular, a shortest path routing can cause up to 161% percent additional cost on practical planning instances.

Many publications on survivable network design present a mathematical model with a path-flow formulation (also called arc-path formulation) to tackle the arising multicommodity flow subproblems. Compared to an edge-flow (also called node-arc) formulation, a path-flow formulation has the main advantage that restrictions on the admissible path set can be more easily modeled. The drawback of such a formulation is a possibly exponential number of path variables.

To cope with the large number of variables and to reduce calculation times, some authors apply column generation techniques [3,9,12,14], while other authors feed a fixed, precalculated set of routing paths into an LP or MIP solver. For the latter approach, several ways of defining the admissible path set have been proposed. In [5,15], a fixed global hop limit is imposed on the routing paths for all demands. In [7], a demand-dependent hop limit is considered, defined by the length of a shortest hop path for a given demand plus some fixed additional number of hops. Eventually, [4] employs a demand-dependent hop limit, which is iteratively raised until a specified number of paths has been found for each demand. We now briefly present the problems investigated in these papers, together with the employed solution approach.

Murakami [9] uses a path-flow formulation to compare the cost of path restoration with stub release and link restoration under a single link failure scenario. The author considers both minimization of continuous spare capacities with respect to a given shortest working path routing, and joint optimization of working and spare capacities. Column generation is applied to generate working and restoration paths only when needed, using a (quadratic) shortest path algorithm.

Dahl and Stoer [3,13] formulate the problem of installing discrete link capacities for the survivability models reservation and diversification with so-called metric inequalities [8]. These inequalities are generated at runtime using a path-flow formulation with routing paths in all operating states, which is solved using column generation.

Poppe and Demeester [12] use a similar model to formulate the problem of installing continuous spare capacities for link and path restoration based on a given shortest working path routing. Column generation is used to identify missing restoration paths.

Wessály [14] determines discrete working and spare capacities using a path-flow formulation for the survivability models reservation, diversification, and path restoration. This path-flow LP is solved by generating working and restoration paths only when needed.

Herzberg and Bye [5] is the only paper known to us in which cost effects of hop limits are investigated. The authors consider the spare capacity assignment problem with respect to a given working path routing. The network is designed for a single link failure scenario with link restoration. A path-flow formulation with integer capacities is presented. Computational results are presented on one small but well connected test instance (11 nodes, 23 links) with integer capacities. The effect of hop limits is tested by enumerating all restoration paths up to a given number of hops, which varies between 3 and 7. On the investigated test instance, the optimal solution values for hop limits 5, 6 and 7 were identical. The authors state having observed similar results on other test instances as well.

Xiong and Mason [15] use a path-flow formulation for path restoration without stub release and for link restoration under a single link failure scenario. A set of working and restoration paths is precalculated, which contains at most 40 paths per demand. A path length restriction of 6 and 10 hops is imposed for the two small and the two larger test instances, respectively.

Iraschko, MacGregor, and Grover [7] compare the cost of link and path restoration (with or without stub release) for single link failures 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. For each demand, the authors enumerate all paths up to a given hop limit, which is the length of a shortest hop path for this demand plus a fixed number of additional hops. This path set is complemented by a small set of link disjoint paths to guarantee a solution. Due to the resulting large path set, the authors report on very long calculation times even on small test instances as soon as working and spare capacities are optimized together (9 hours for link restoration, 2.7 days for path restoration without stub release on an instance with 10 nodes, 22 links, and 45 demands).

Doucette and Grover [4] compare several protection and restoration mechanisms for networks of varying density. The authors use a path-flow formulation with integer capacities and a predetermined path set. All paths up to an iteratively adapted hop limit are enumerated until at least 5, 10, or 20 paths have been found for each demand (the exact number depends on the considered problem).

A column generation approach combined with a branch-and-cut algorithm provides a lower bound and thus a quality guarantee for solutions if the pricing problem (i.e., the problem of identifying missing path variables) is exactly solvable. For instance, if the admissible path set is not restricted at all or by hop limits only, the pricing problem is exactly solvable in polynomial time and this approach can be used to compute provably optimal solutions.

On the other hand, if the pricing problem is not solved exactly (e.g., because it is \mathcal{NP} -hard), the column generation approach no longer provides a valid lower bound. Hence, this approach does not yield a quality guarantee, but can still be used as a good heuristic.

The main advantage of a predefined path set is the fact that “wild” path set restrictions can be incorporated in the model. In practice, however, a restricted path set often consists simply of all paths up to a given number of hops (which may be demand-dependent or not), sometimes complemented by a small set of paths which guarantees a solution. The drawback of a predefined path set is that this approach only leads to heuristic solutions without a lower

bound.

This paper is structured as follows: after a description of our mathematical model and a brief sketch of our algorithmic approach in Section 2, we report on our computational tests in Section 3. Eventually, we conclude with Section 4.

2 Model and Algorithm

We investigate network design problems dealing with an integrated planning of

- a topology,
- modular link capacities,
- a routing during normal operation, and
- a routing in all single link and node failure states.

The mathematical model is derived from the mixed-integer linear programming formulation described in [10], which also covers hardware requirements imposed by network elements and interface cards. These and other extensions (e.g., the possibility to respect existing parts of a network) are implemented in our network planning tool DISCNET [1].

Sections 2 and 2 describe the parts of our model covering topology and link capacity decisions, and routing planning, respectively.

The planning network consists of a set V of nodes and a set E of potential (undirected) links between these nodes. For each link $e \in E$, a set $\mathcal{D}(e)$ of *link designs* (e.g., STM- N capacities or a certain number of WDM wavelengths) is specified, out of which at most one may be chosen. Every link design d installable on a given link e has a capacity C_e^d and a cost value K_e^d assigned to it. The final topology consists of those links for which a link design is chosen; all other links are omitted.

For every link $e \in E$ and every link design $d \in \mathcal{D}(e)$, a binary variable $x_e^d \in \{0, 1\}$ determines whether link design d is installed on link e or not. The condition that at most one link design must be chosen on each link is expressed by

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

For convenience of notation, the auxiliary variable

$$y_e := \sum_{d \in \mathcal{D}(e)} C_e^d x_e^d \quad (2)$$

denotes the capacity of the installed link design on link $e \in E$ since at most one of the link design variables is non-zero. In particular, choosing no link design implies $y_e = 0$.

Our objective is to minimize the sum of all link capacity costs:

$$\min \sum_{e \in E} \sum_{d \in \mathcal{D}(e)} K_e^d x_e^d. \quad (3)$$

In addition to the input for topology and capacity planning, a survivable routing has to be determined. This section presents a path-flow formulation which also integrates survivability

requirements for the concepts *diversification* (as a relaxation of 1+1 protection), *link* and *path restoration*.

Apart from the *normal operating state* (NOS) where the whole network is operational, we consider a set S of *failure states* where at least one link or node fails. In this paper, the set S consists of all single link and node failures.

NOS routing Let \mathcal{D} denote the set of all point-to-point demands. For each demand $uv \in \mathcal{D}$, a demand value d_{uv} must be routed between the end nodes u and v . We assume a bifurcated routing, i.e., several paths may be used for each demand.

Let \mathcal{P}_{uv} be the set of *admissible* paths to route the demand $uv \in \mathcal{D}$ in the normal operating state. These are all loopless paths between u and v satisfying a given hop limit. 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 demand constraints (4) and capacity constraints (5) formulate a multicommodity flow problem with hop limits for the normal operating state:

ROUTING (NOS):

$$\sum_{P \in \mathcal{P}_{uv}} f_{uv}(P) = d_{uv} \quad uv \in \mathcal{D} \quad (4)$$

$$\sum_{uv \in \mathcal{D}} \sum_{\substack{P \in \mathcal{P}_{uv}: \\ e \in P}} f_{uv}(P) \leq y_e \quad e \in E \quad (5)$$

1+1 protection and diversification With 1+1 protection, each demand is doubled at one node and routed on two link- or node-disjoint paths to some target node, which chooses the signal of better quality. In particular, if a single link (or node, respectively) fails, at least one of the two paths remains operational since the two paths are disjoint. We approximate this mechanism by *diversification*. This formulation has been introduced by Dahl and Stoer [3].

For every demand $uv \in \mathcal{D}$, the *diversification parameter* $\delta_{uv} \in (0, 1]$ specifies the fraction of demand uv which is allowed to fail in any considered failure state. This is expressed by the diversification constraints

$$\sum_{\substack{P \in \mathcal{P}: \\ P \text{ fails in } s}} f_{uv}(P) \leq \delta_{uv} d_{uv} \quad uv \in \mathcal{D}, s \in S. \quad (6)$$

and implies that the demand is routed on at least $\lceil \frac{1}{\delta_{uv}} \rceil$ paths. Setting the diversification value δ_{uv} to 0.5 and doubling the demand value, constraints (6) ensure that at least the original demand value d_{uv} survives in any considered failure state. Since every solution for 1+1 protection fulfills these constraints but not vice versa, this formulation is a relaxation of 1+1 protection. It can be used to formulate 1+1 protection exactly by splitting every demand with value d into d demands with value 1, doubling these unit-demands and routing them with diversification value $\delta_{uv} = 0.5$ and integer flow variables. However, since this leads to a very large set of demands and thus to a large integer program, we only use the relaxation here.

Link and path restoration In contrast to 1+1 protection where backup capacity is pre-configured and dedicated to a particular demand, link and path restoration share backup capacity between the demands.

With *link restoration*, each path affected by a failure is patched locally between the two nodes on the path which are adjacent to the failing link or node.

With *path restoration*, every affected path is rerouted between the two end nodes of the corresponding demand. In this paper, we consider only path restoration *without* stub release, i.e., working capacity on operational links of a failing path is *not* released and cannot be used for backup paths.

With each failure state $s \in S$, we associate a set C^s of *failure commodities*. These are point-to-point demands to be satisfied in failure state s ; their exact definition depends on the type of failure and the restoration mechanism.

For instance, a single link failure of link e with link restoration leads to only one failure commodity between the two end nodes of e , whose demand value is the total NOS flow on e . Upon failure of a node v , a failure commodity is created between every pair u, w of nodes adjacent to v , with a demand value equal to the NOS flow on the three-node-path $u \rightarrow v \rightarrow w$. With path restoration, one failure commodity per failure state and failing demand is created, whose demand value is the amount of failing flow for this demand.

In failure state $s \in S$, let \mathcal{P}_c^s be the set of all admissible paths to route failure commodity $c \in C^s$. These are all paths between the end nodes of the failure commodity c which do not contain failing nodes or links and which satisfy given hop limits. For a given failure commodity c , the set \mathcal{IP}_c^s consists of all *interrupted paths* which are restored by failure commodity c in failure state s . The exact definition of this set again depends on the restoration mechanism and the type of failure.

Using non-negative continuous path-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 constraints formulate both the link and path restoration problem, according to the choice of parameters:

ROUTING (failure states):

$$\sum_{c \in C^s} \sum_{\substack{P \in \mathcal{P}_c^s \\ e \in P}} f_c^s(P) + \sum_{uv \in \mathcal{D}} \sum_{\substack{P \in \mathcal{P}_{uv} \\ e \in P}} f_{uv}(P) \leq y_e \quad \begin{array}{l} s \in S, \\ e \in E^s \end{array} \quad (7)$$

$$- \sum_{P \in \mathcal{P}_c^s} f_c^s(P) + \sum_{uv \in \mathcal{D}} \sum_{P \in \mathcal{P}_{uv} \cap \mathcal{IP}_c^s} f_{uv}(P) \leq 0 \quad \begin{array}{l} s \in S, \\ c \in C^s \end{array} \quad (8)$$

The capacity constraints (7) state that in any failure state, the sum of the total NOS flow and the total restoration flow on a given link must not exceed its capacity. The failure demand constraints (8) formulate that all failing flow must be restored.

The problem is solved with our network planning tool DISCNET [1] using Benders decomposition [2]. The central algorithm is a branch-and-cut algorithm based on an LP relaxation containing link design variables but no routing variables. The path flow formulation of the routing constraints is used to test whether given link capacities allow for a feasible routing. Either this test leads to a feasible solution, or generalizations [11, 14] of metric inequalities [8] are generated from the dual solution of the routing LP. These inequalities are added to the LP relaxation in order to cut off the infeasible capacities. To solve the path flow formulation, we use column generation for working and restoration paths. Figure 1 illustrates this decomposition concept.

A more detailed description of this approach, further employed cutting planes, and the methods used to identify missing paths can be found in [10, 11, 14]. The branch-and-bound

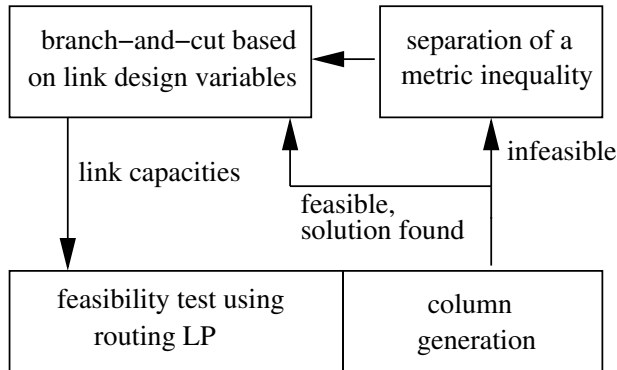


Figure 1: Algorithmic approach

process together with the exact pricing provides us with a lower bound on the optimal network cost and thus to a quality guarantee for obtained solutions. In particular, this lower bound can be used to prove optimality of a solution.

3 Computational Results

In this section, we describe our computational tests to evaluate the effect of two kinds of hop limits on the overall network cost. After a short presentation of our nine test instances stemming from different planning scenarios, we show and discuss our results.

We present two test series, each of them for diversification, link and path restoration. In the first series, a fixed global hop limit between 3 and 7 is imposed. In the second series, demand-dependent hop limits are given by the shortest hop distance for a particular demand, plus some fixed number of hops, which is then varied between 0 and 5. The hop limits are applied to both working and backup paths. For diversification and path restoration, the end-to-end backup paths are hop-limited, whereas for the local survivability mechanism link restoration, the hop limits are applied to the local patching paths.

For each of our test instances, Table 1 shows the number of nodes, potential links, and demands, respectively, together with the average node degree $\bar{d} = 2|E|/|V|$ and the number of available link designs (#ld), which is the same for all links of an instance.

In all computational tests, we have assumed full restoration or protection of all failing flow upon any single link or node failure. In order to be able to compare optimal solutions, we have chosen relatively small networks with 10–20 nodes. Most of the test instances have been provided to us by network operators. They are stemming from SDH-, WDM-, and leased line planning problems and reflect the different cost structures and demand patterns arising in these planning scenarios. In some of the instances, we have added a few links in order to make sure that the network is still connected in any single link or node failure situation.

The longest computation time was about 36 hours. However, most instances could be solved to proven optimality within an hour on a Linux machine with 1 GB of RAM and a 1.7 MHz processor. In fact, the algorithm often found good solutions after a short time, spending most of the time on finding slightly better solutions and proving optimality. All linear programs were solved using CPLEX 9.0 [6].

The column generation approach proved to be well scalable. Whereas for small hop limits

Table 1: Characteristics of the test instances

Name	$ V $	$ E $	$ D $	\bar{d}	#ld
g1	10	25	29	5.0	2
g2	12	19	27	3.2	4
g3	15	22	13	2.9	3
g4	15	24	105	3.2	7
g5	18	26	62	2.8	9
g6	18	32	62	3.6	9
g7	20	32	119	3.2	6
g8	14	21	91	3.0	5
g9	17	28	58	3.3	3

the computation time raised almost linearly, it was nearly constant for larger hop limits. In contrast, enumerating all paths up to a given length leads to an exponential growth in computation time.

In all presented figures, the solution values are scaled such that 100 corresponds to the optimal network cost when *all* routing paths are allowed, regardless of their length. To simplify the following interpretation of the results, we define an additional cost of 10% for a hop-limited solution to be *acceptable*.

3.1 Global hop limit

For each of our nine test instances, Figure 2 shows the relative network cost with diversification when a global hop limit between 3 (leftmost bar) and 7 (rightmost bar) is imposed on all routing paths (note that the relative cost scale starts at 90, not at 0).

Figure 2 reveals that high hop limits are often needed to obtain a feasible solution at all. In five out of the nine instances, there is a demand which needs a path of length at least 6, and in another two instances, hop limit 5 is needed. These long paths are usually due to the failure of a “central” node whose failure implies a long detour for some demands.

Another interesting observation is that whenever a feasible solution is found with paths of length k , hop limit $k + 1$ leads to an acceptable solution with our test instances. This leads to a natural iterative algorithm: start with a small hop limit and raise it until a feasible solution is found. Going one step further, chances are good to obtain an acceptable solution.

In a similar way, Figures 3 and 4 show the results for link and path restoration with a global hop limit. The figures show that in many cases (five out of nine instances with link restoration, four instances with path restoration), paths of length 6 are needed to obtain a feasible solution at all. Other than with diversification, we have not observed any rule of thumb which could give us a hint about the quality of a solution without calculating a lower bound. Even high hop limits can lead to expensive solutions. In several cases, even paths of length 7 are by far not sufficient to find an acceptable solution.

As these results indicate, global hop limits do not seem to be the ultimate method to define a good set of routing paths. The problem is that a global hop limit does not use any information on the network size or of the distance between given nodes. It can be expected that for larger networks, the global hop limit which is needed to obtain a feasible solution

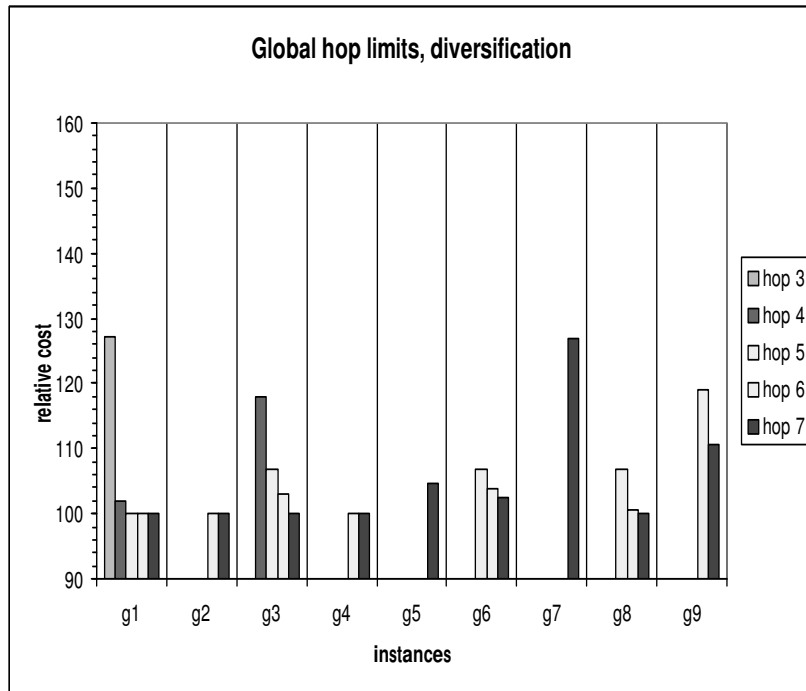


Figure 2: Global hop limit, diversification

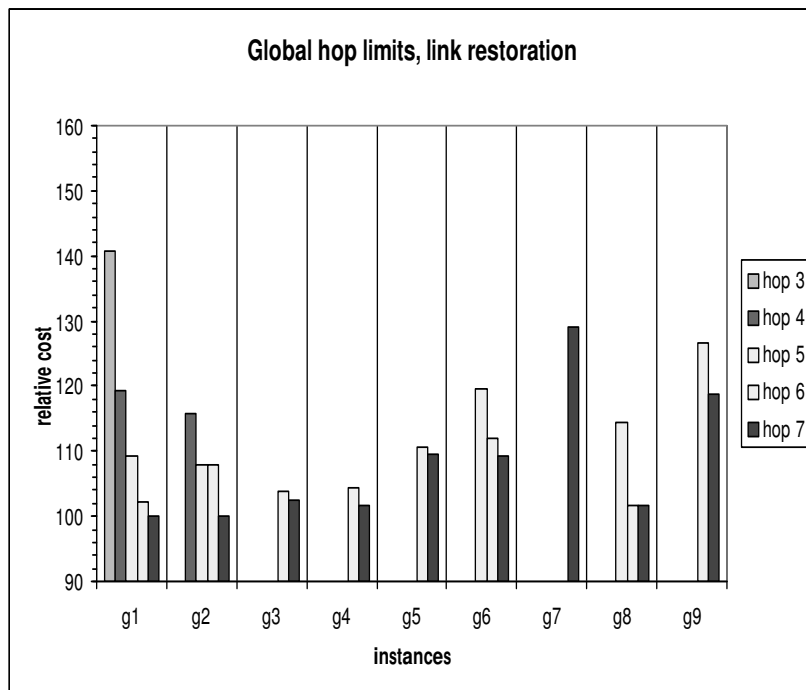


Figure 3: Global hop limit, link restoration

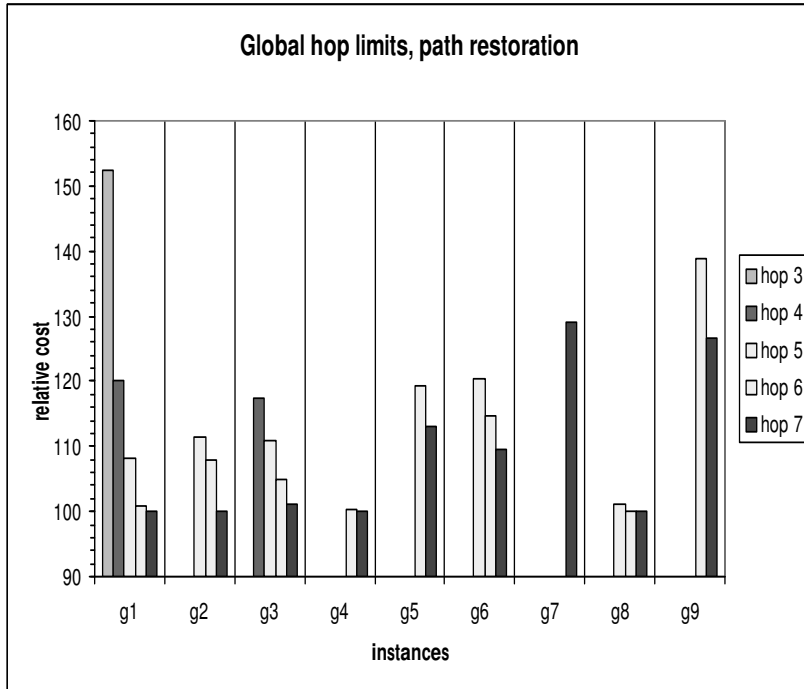


Figure 4: Global hop limit, path restoration

will rise even more, unless the density of the network increases at the same time. Notice that both a larger hop limit and increased density make it difficult if not impossible to enumerate all routing paths.

3.2 Demand-dependent hop limit

Global hop limits do not reflect properties of the network such as its size or density. One possibility to overcome this drawback is to define demand-dependent hop limits based on the shortest hop distance between the end nodes of each demand. For a given value k , the hop limit imposed on a particular demand is k plus the number of links in a shortest hop path between the demand end nodes.

For diversification, the shortest hop distance in the NOS is used for all paths of a demand since it is not known in advance which paths will be used for backup purposes. For restoration, the hop limits for backup paths are defined with respect to the shortest hop distances in the corresponding failure states.

Figure 5 shows the relative cost for each instance with diversification, imposing demand-dependent hop limits. Again, 100 corresponds to the optimal network cost when *all* routing paths are allowed, i.e., the values are comparable to those in Figure 2. The leftmost bar for each instance shows the result for $k = 0$, which implies that each demand is routed on one or more shortest hop paths. The next bars represent the relative cost if $k = 1, \dots, 5$ additional hops are allowed.

Again, we see that a high hop limit ($k \geq 4$) is often needed for feasibility. This is again due to nodes whose failure implies a detour for some demands which is much longer than a

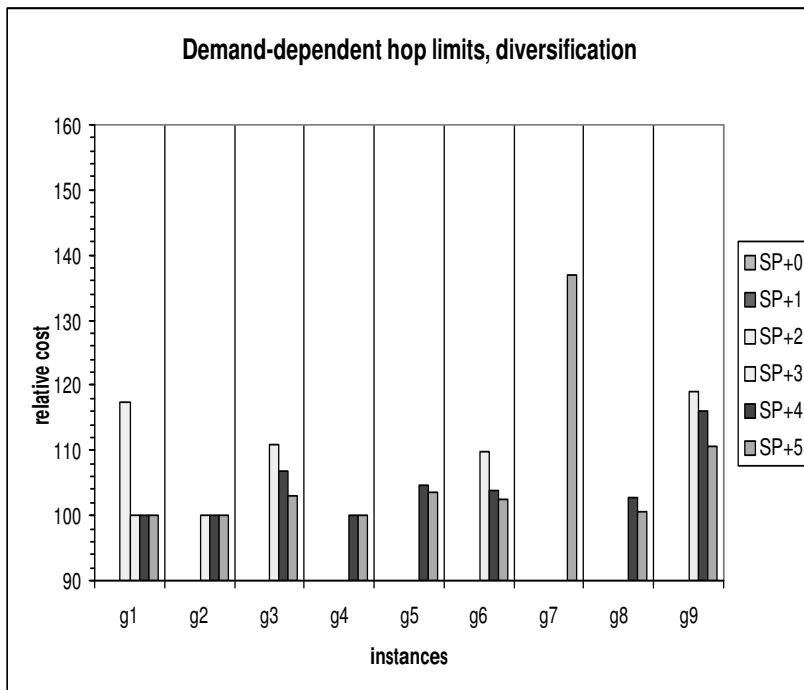


Figure 5: Demand-dependent hop limit, diversification

shortest hop path in the NOS. On the other hand, the smallest feasible hop limit (i.e., the smallest hop limit for which a feasible solution exists) leads to an acceptable solution in five out of nine cases. With the smallest feasible hop limit plus 1, an acceptable solution is found in all cases but one. In summary, the behavior of the network cost with respect to changing hop limits is very similar to the one observed with global hop limits.

Figures 6 and 7 show the results for link and path restoration with demand-dependent hop limits (note the different scale of the cost axis). The values are comparable with those in Figures 3 and 4, respectively.

Contrary to diversification, a restriction to shortest paths always leads to a feasible solution with restoration by definition of the restricted path set, provided that sufficiently large link capacities are available. However, with both link and path restoration, high hop limits are needed to obtain acceptable solutions. In four out of the nine instances, at least $k \geq 4$ is needed, and in two cases, even $k = 5$ still leads to more than 10% additional network cost.

A result that we had expected in its tendency but not to that extent is the cost of a shortest path routing. It is always above 10% additional cost and can be up to 161% more expensive than a solution where all paths are admissible! Both with link and path restoration, a shortest path routing leads to more than 60% additional cost in four out of our nine instances. Notice that we allowed the demands to be routed on more than one shortest path if several of them existed; a restriction to a single shortest path routing would even aggravate the results.

In some of the instances (including g1), not all of the potential links are actually needed in a cost optimal solution with respect to all paths. This partially explains the bad performance of a shortest hop routing (as well a shortest length routing with respect to kilometer length or any other metric link weights) since the direct link between the end nodes of a demand is

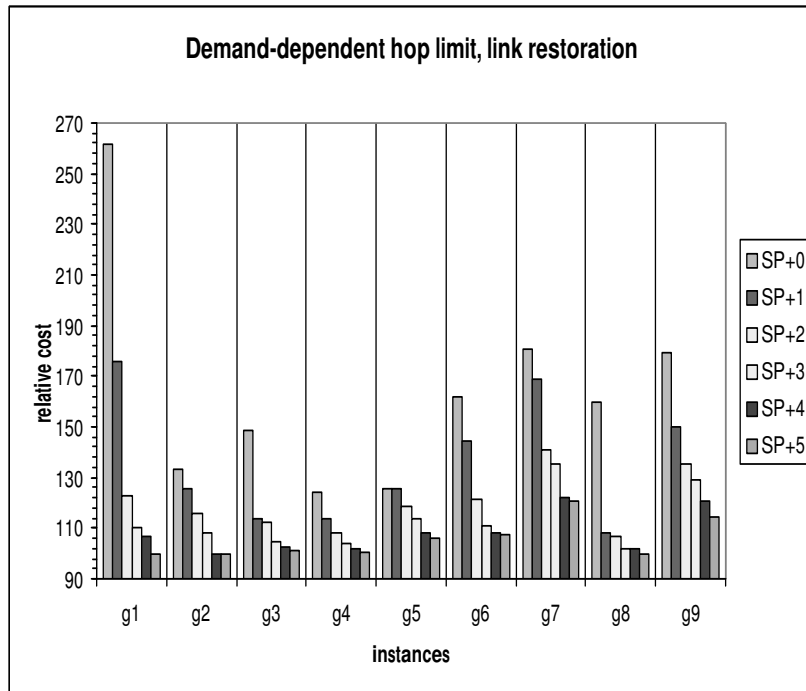


Figure 6: Demand-dependent hop limit, link restoration

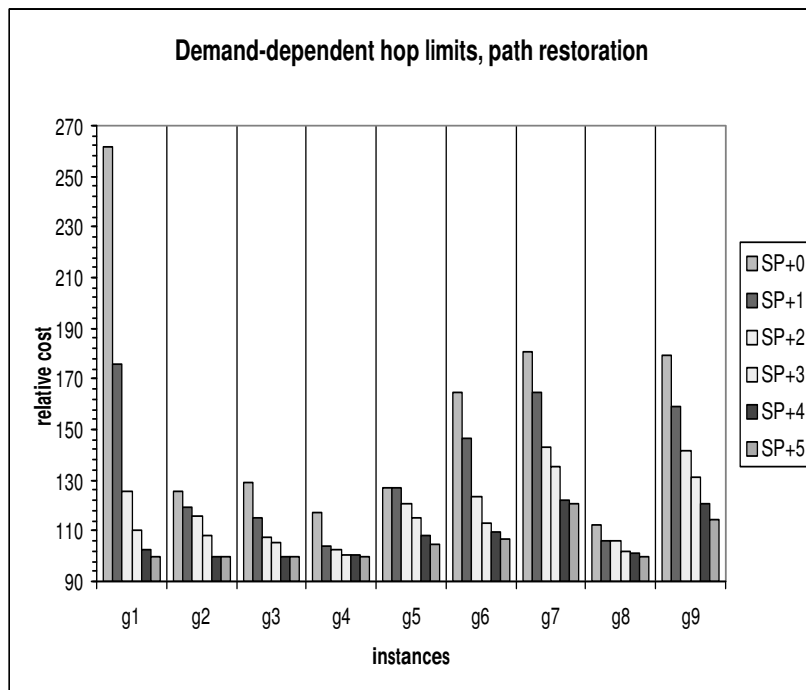


Figure 7: Demand-dependent hop limit, path restoration

the unique shortest path if it exists. This leads to some positive capacity on most links and consequently to high network cost.

4 Conclusion

In this paper, we have studied the impact of varying hop limits on the overall network cost in the planning of a telecommunication network. Based on our mixed-integer linear programming model, we have solved nine problem instances to optimality using a branch-and-cut framework and column generation for working and restoration paths. For each of these instances, we have considered three different classes of admissible path sets and three survivability concepts.

Based on optimal solutions, we have compared the overall network cost without restrictions on the admissible paths to the network cost with hop-limited paths, either with demand-dependent or with global hop limits.

As main conclusion, hop limits should be avoided if the technology allows it, and column generation should be employed for dealing with all routing paths. For many of the instances considered in our tests, a global hop limit of six links or a demand-dependent hop limit of four links in addition to the shortest hop count are needed to obtain an acceptable solution; enumerating all paths up to these hop limits can already exceed available time and memory even for small, sparse networks.

If hop limits are technologically unavoidable or if they are desired for ease of management, the suggested approach depends on the survivability concept: with dedicated protection, it is possible to start with a shortest path routing and to raise the hop limit until a feasible solution is found. Allowing one further link for each path typically leads to a solution which is at most 10% more expensive than an optimal solution without path length restrictions. In contrast to this, restoration requires sufficiently high demand-dependent hop limits (at least four hops in addition to the shortest hop count) to obtain acceptable solutions. In particular, neither global hop limits nor a shortest path routing should be employed.

In any case, the additional network cost caused by hop limits cannot be predicted in advance. On our test instances, neither network size, nor density, the number of demands or the number of installable link designs are reliable indicators.

References

- [1] atesio GmbH, Germany. DISCNET, 2000–2004. www.atesio.de.
- [2] J. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.
- [3] G. Dahl and M. Stoer. A cutting plane algorithm for multicommodity survivable network design problems. *INFORMS Journal of Computing*, 10(1):1–11, 1998.
- [4] J. Doucette and W.D. Grover. Comparison of mesh protection and restoration schemes and the dependency on graph connectivity. In *Proceedings of the Third International Workshop on the Design of Reliable Communication Networks (DRCN 2001)*, Budapest, Hungary, pages 121–128, October 2001.

- [5] M. Herzberg and S. Bye. An optimal spare-capacity assignment model for survivable networks with hop limits. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM'94)*, pages 1601–1606, September 1994.
- [6] ILOG CPLEX Division, 889 Alder Avenue, Suite 200, Incline Village, NV 89451, USA. *ILOG CPLEX 9.0 Reference Manual*, 2002. Information available at <http://www.cplex.com>.
- [7] R. Iraschko, M. MacGregor, and W.D. Grover. Optimal capacity placement for path restoration in stm or atm mesh survivable networks. *IEEE Transactions on Networking*, 6(3):325–336, June 1998.
- [8] 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.
- [9] K. Murakami. *Survivable Network Management for High-Speed ATM Networks*. PhD thesis, Carnegie Mellon University, September 1995.
- [10] S. Orłowski and R. Wessäly. Comparing restoration concepts using optimal network configurations with integrated hardware and routing decisions. In *Proceedings of the Fourth International Workshop on the Design of Reliable Communication Networks (DRCN 2003)*, Banff, Alberta, Canada, pages 15–22, October 2003.
- [11] Sebastian Orłowski. Local and global restoration of node and link failures in telecommunication networks. diploma thesis, Technische Universität Berlin, February 2003.
- [12] F. Poppe and P. Demeester. Economic allocation of spare capacity in mesh-restorable networks: Models and algorithms. In *Proceedings of the 6th International Conference on Telecommunication Systems (ICTSM 1998), Modeling and Analysis*, Nashville, USA, pages 77–86, March 1998.
- [13] M. Stoer and G. Dahl. A polyhedral approach to multicommodity survivable network design. *Numerische Mathematik*, 68:149–167, 1994.
- [14] R. Wessäly. *Dimensioning Survivable Capacitated NETWORKS*. PhD thesis, Technische Universität Berlin, April 2000.
- [15] Y. Xiong and L. Mason. Restoration strategies and spare capacity requirements in self-healing ATM networks. In *Proceedings of the 16th IEEE Infocom 1997*, Kobe, Japan, pages 353–360, 1997.