On the complexity of column generation in survivable network design with path-based survivability mechanisms

Sebastian Orlowski* Michał Pióro◇

*Zuse Institute Berlin
Takustr. 7, 14195 Berlin, Germany

◇Institute of Telecommunications, Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland

◇Department of Electrical and Information Technology, Lund University
Box 118, 221 00 Lund, Sweden

Abstract

This paper deals with path-based linear programming formulations in survivable network design. In a recent survey we have investigated the complexity of the column generation problems for a large variety of protection and restoration mechanisms in a single or multiple link failure scenario, and classified them according to their structure. It turned out that all the considered column generation problems are composed of only few building blocks which determine their complexity. In this paper, we summarize our findings and give an example for each of these building blocks.

Keywords*: network design, routing, column generation, complexity, multiple failures*

1 Introduction

In the literature on network design and traffic engineering, a variety of linear programming (LP) formulations have been developed to incorporate protection and restoration mechanisms against node and link failures into network optimization models. Many of them employ link-path formulations with exponentially many path flow variables representing the total end-to-end demand flow routed on paths in the communication network. To solve such formulations exactly for networks of practical size, a common approach is to use column generation (in our context, path generation). The approach starts with a small set of variables corresponding to an initial set of routing paths, and generates further path flow variables only when necessary to improve the current solution.

For an optimal LP solution corresponding to a given restricted set of path flow variables (LP columns), the *pricing problem* is to identify further routing paths that can improve the LP value, or to discover that no such paths exist. A new path variable can improve the current optimal solution if it has a negative reduced cost, i.e., if it violates its dual constraint. If new columns are found, the LP is solved again with the new variables. This process is repeated until no improving variables are found. To ensure efficiency of the algorithm, it is crucial to know whether the pricing problem can be solved in polynomial time. For some protection or restoration mechanisms, a most violated dual constraint can be found by solving a shortest-path problem with respect to link weights derived from the dual LP solution. For other mechanisms, the pricing problem is known to be $\mathcal{NP}$-hard already for single link failure scenarios; little has been known so far the case of multiple failures.

In a recent technical report [10], we have systematically summarized known results on the complexity of column generation for various path-based network survivability mechanisms and failure scenarios, and investigated the cases with unknown complexity. In particular, we have shown that for almost all the considered mechanisms, the pricing problem becomes $\mathcal{NP}$-hard for multiple link failures. We have distinguished between several ways of end-to-end path protection, single and multiple link failure scenarios, whether the assignment of backup paths to working paths depends on the failure state or not, and whether the capacity of surviving links of a failing path can be released and reused for backup flows (*stub release*) or not. In the case of failure-independent protection, we distinguish whether backup capacity is shared between demands or dedicated to each demand individually. The complexity results are summarized in Table 1.

The primary goal of this conference paper is to illustrate another finding of the survey [10]: the pricing problems for all the considered survivability mechanisms are composed of only few building blocks (called complexity cases in the sequel) that determine their complexity. These are: (a) a classical shortest-path problem, (b) a classical shortest-cycle problem (more precisely, a shortest failure-disjoint pair of paths problem), (c) a shortest path problem where the path length is the sum of given non-negative prices of the failure states in which the path fails, (d) a shortest path problem with link weights depending on the set of the failure states in which the path survives, and (e) a shortest path pair problem with link weights of the backup path depending on the set of failure states in which the primary path fails. In this paper we will discuss examples for four of these classes, and explain what features make the respective pricing problems polynomial or $\mathcal{NP}$-hard.

In the next section, we introduce the notation used in the rest of the paper. In Section 3, we discuss the pricing problem for path diversity protection, which is a shortest path problem in the single-failure case (see case (a) above) but an $\mathcal{NP}$-hard minimum-color shortest-path problem for multiple link failures (case (c)). Section 4 discusses the pricing problems for single-backup path protection with dedicated or shared backup link capacity, which reduce to a classical shortest-cycle problem (b) and to a failure-dependent shortest-cycle problem (e), respectively. The remaining case (d) occurs in the pricing problem for path restoration with stub release, and is known to be $\mathcal{NP}$-hard already for single link failures [9, 8]. Eventually, we summarize the results in Section 5.

2 Notation

The network is described by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For ease of exposition, we assume that the graph does not contain loops nor parallel links. The cost of realizing one unit of demand flow on link $e \in \mathcal{E}$ is denoted by ξ_e, and the capacity of the link (serving as a variable) by y_e. The set $\mathcal{D} \subseteq \mathcal{V}^{|2|}$ represents undirected point-to-point demands, each with demand value $h_d > 0$ to be routed between nodes $u_d \in \mathcal{V}$ and $v_d \in \mathcal{V}$ (in any direction). For notational convenience, at most one demand between each pair of nodes is assumed. All protection/restoration mechanisms discussed in this paper are defined with respect to a given failure scenario, i.e., a family $\mathcal{S} \subseteq 2^{\mathcal{E}}$ of *network states*, each of which corresponds to a subset of failing links. We assume that the size of $\mathcal{S}$ is polynomial in the size of the network, and that $\mathcal{S}$ contains the failure-less state $\emptyset$ in which all links are operational.

Each demand $d \in \mathcal{D}$ has a predefined set $\mathcal{P}_d$ of undirected candidate paths (without loops) between its end-nodes that can be used for realizing the demand flow. The set of all candidate paths is denoted by $\mathcal{P} := \bigcup_{d \in \mathcal{D}} \mathcal{P}_d$. The flow realizing the volume of demand $d \in \mathcal{D}$ on path $p \in \mathcal{P}_d$ will be denoted by variable x_p. The set $\mathcal{P}_d^s$ of candidate paths for demand $d \in \mathcal{D}$ available in state $s \in \mathcal{S}$ is defined as $\mathcal{P}_d^s = \{p \in \mathcal{P}_d \mid p \cap s = \emptyset\} \subseteq \mathcal{P}_d$. Furthermore, $\mathcal{P}_e \subseteq \mathcal{P}$ is the set of all paths containing link $e \in \mathcal{E}$. The notation $\mathcal{S}_p = \{s \in \mathcal{S} \mid p \cap s = \emptyset\}$ and $\bar{\mathcal{S}}_p := \mathcal{S} \setminus \mathcal{S}_p$ refers to the sets of states $s \in \mathcal{S}$ in which path $p \in \mathcal{P}$ is available or unavailable, respectively.

The set of all failure-disjoint primary-backup path pairs for demand $d \in \mathcal{D}$ will be denoted by $\mathcal{R}_d := \{r = (p, q) \mid p \in \mathcal{P}_d, q \in \mathcal{Q}_p\}$, where $\mathcal{Q}_p$ denotes a set of candidate backup paths for path $p \in \mathcal{P}_d$. In $r = (p, q)$, paths p and q never fail simultaneously. For each link $e \in \mathcal{E}$, the set of all path pairs r containing link e (once or twice) will be denoted by $\mathcal{R}_e$. In this case, variable x_r will denote the fraction of demand volume of demand $d \in \mathcal{D}$ realized on path pair $r \in \mathcal{R}_d$ (note that path-flow variable x_p denotes the absolute flow, not the fraction).

Throughout this paper, we assume that all link capacity and flow variables are continuous, nonnegative, and unbounded from above (assumptions naturally satisfied in the LP relaxations of most practical network planning problems). We also assume that the initial set of routing paths (pairs) results in a feasible initial primal LP.

3 Path diversity – PD

Primal problem A conceptually simple way of protecting traffic against failing network components is by flow over-provisioning. The path diversity mechanism may route more flow than the specified demand value h_d in the failure-less state, and ensures that at least a specified fraction h_d^s of the flow survives in each of the considered failure states $s \in \mathcal{S}$ without rerouting any path flow [2, 7, 15, 6]. Of course, the flow must fit into the link capacities. This translates into the following LP formulation:

$$\min_{x,y\geq 0} \{\sum_{e\in\mathcal{E}} \xi_e y_e \mid \sum_{p\in\mathcal{P}_d^s} x_p \geq h_d^s \quad \forall d\in\mathcal{D},\ \forall s\in\mathcal{S}, \quad \sum_{p\in\mathcal{P}_e} x_p \leq y_e \quad \forall e\in\mathcal{E}\} \tag{1}$$

As the objective function minimizes the non-negative continuous y variables, all capacity constraints are tight in the optimum, and the y_e variables could in fact be eliminated from the model. This leads directly to the dual (2).

Dual problem Let λ_d^s denote the dual variables associated with the demand constraints (the first set of constraints) in (1), and define $\Lambda_d := \sum_{s\in\mathcal{S}} \lambda_d^s$ for all demands $d \in \mathcal{D}$. By straightforward LP duality, the dual formulation of LP (1) can be written as

$$\max_{\lambda\geq 0} \{\sum_{d\in\mathcal{D}} \sum_{s\in\mathcal{S}} h_d^s \lambda_d^s \mid \Lambda_d \leq \sum_{e\in p} \xi_e + \sum_{s\in\bar{\mathcal{S}}_p} \lambda_d^s \quad \forall d\in\mathcal{D},\ \forall p\in\mathcal{P}_d\} \tag{2}$$

Given an optimal dual solution λ^* with respect to the current set of candidate routing paths, the goal of the pricing problem for demand $d \in \mathcal{D}$ is to find a new path p for demand d which violates the dual constraint from (2). The following two paragraphs describe the details of this pricing problem for single and multiple link failures. Notice that if the capacities y_e have to satisfy additional integrality restrictions, the values ξ_e in the dual constraints have to be replaced by the dual values of the primal capacity constraints (i.e., with the second set of constraints in (1)). This, however, does not affect the pricing complexity.

Pricing problem for single failures Under a single link failure scenario $\mathcal{S} \subseteq \{\{e\} \mid e \in \mathcal{E}\} \cup \{\emptyset\}$, the pricing problem for PD can be solved in polynomial time, as observed by Wessäly et al. [15]. In this setting, the dual constraint (2) can be written as

$$\sum_{e\in p} (\xi_e + \lambda_d^{\{e\}^*}) \geq \Lambda_d^*. \tag{3}$$

The right-hand side depends only on the demand, and the link weights on the left-hand side are nonnegative. Hence, for each demand $d \in \mathcal{D}$, violation of dual constraint (3) can be tested by finding a shortest path between the end-nodes of d with respect to the demand-dependent link weights $\gamma_d^e := \xi_e + \lambda_d^{\{e\}^*}$ using for example the Dijkstra algorithm, and comparing its length to the value of Λ_d^*. If the computed shortest path p for demand d violates condition (3) then it should be added to the LP relaxation; otherwise, no path for this demand violates its dual constraint for the current set of optimal dual variables.

Pricing problem for multiple failures In a multiple failure state a group of links, a so-called *shared risk link group* (SRLG) [12], fails simultaneously. In this paragraph, we show by reduction from the *minimum-color shortest-path problem* (MC-PATH) that path generation for PD is difficult in general. In the MC-PATH problem, a set $\mathcal{C}$ of colors with weights $w_c > 0$ is given, and every link $e \in \mathcal{E}$ is associated with a subset $\mathcal{C}_e \subseteq \mathcal{C}$ of the colors. The length of a path p in this colored network is defined as the total weight of different colors traversed by p. In contrast to the classical shortest path problem, the weight of a used color is counted only once even if the path contains several links with that color. Given two nodes $u, v \in \mathcal{V}$, the goal of MC-PATH is to find a u-v-path with the minimum length according to this definition. This problem has been shown to be $\mathcal{NP}$-hard for a general color setting, and various inapproximability results are known [17, 1, 3].

The PD pricing problem for demand $d \in \mathcal{D}$, PRICE-PD, consists in finding a path p from u_d to v_d minimizing

$$\langle p\rangle = \sum_{e\in p} \xi_e + \sum_{s\in\bar{\mathcal{S}}_p} \lambda_d^{s*}. \tag{4}$$

The second sum contains the dual values of those network states $s \in \mathcal{S}$ where path p fails. If the path contains several failing links from s, the weight λ_d^{s*} is counted only once, as in MC-PATH. By identifying failure states with colors and assuming all costs ξ_e to be zero, we observe that the pricing problem (4) contains MC-PATH as a special case, which shows that PRICE-PD is $\mathcal{NP}$-hard in general. Also the inapproximability results for MC-PATH can be directly transferred to PRICE-PD. Results by Coudert et al. [1] on MC-PATH imply that PRICE-PD is $\mathcal{NP}$-hard already for the polynomially bounded set of all double link failures. Some special cases, however, are known to be polynomial, e.g., link failures induced by single node failures. Tomaszewski et al. [14] have proved that problem PD (1) is $\mathcal{NP}$-hard itself already for a single demand and a certain multiple link failure scenario containing $|\mathcal{V}|$ failure states. This yields an alternative proof that PRICE-PD is not polynomial for a polynomial number of failure states. Problem PRICE-PD can be exactly formulated and solved (at least for small to medium size instances) as a mixed-binary programming problem. Alternatively, it can be treated as a shortest-path problem with resource constraints [5, 10].

4 Single backup path protection – SB

Single backup path protection (SB) assumes that the entire primary flow is routed on a single primary path and restored on a single backup path, used in all failure situations when the primary paths fails. When the backup capacity of links is split and reserved separately for each particular backup path (SB-D), then we talk about *1+1 protection* or *1:1 dedicated path protection*. When the backup capacity can be shared by backup paths of different demands in different failure states (SB-S), then we deal with *1:1 shared path protection*.

SB with dedicated protection capacity – SB-D In this paragraph we assume that protection capacity is reserved for each particular demand. The primal problem SB-D reads:

$$\min \sum_{e \in \mathcal{E}} \xi_e y_e \tag{5a}$$

$$[\lambda_d \geq 0] \qquad \sum_{r \in \mathcal{R}_d} x_r \geq 1 \qquad d \in \mathcal{D} \tag{5b}$$

$$[\pi_e \geq 0] \qquad \sum_{d \in \mathcal{D}} h_d \sum_{r \in \mathcal{R}_d \cap \mathcal{R}_e} x_r \leq y_e \qquad e \in \mathcal{E} \tag{5c}$$

$$y_e \geq 0, \, x_r \in \{0, 1\} \tag{5d}$$

It is a straightforward exercise to derive the dual problem to the LP relaxation of SB-D, and to show that column generation consists in finding, for each demand $d \in \mathcal{D}$, a minimum cost cycle through its end-nodes with respect to the dual capacity cost π_e. This can be done by solving a min-cost-flow problem with capacities 1 and value 2 [13]. Intuitively, this is the survivable analogon of a simple multi-commodity flow, where the pricing problem searches for minimum-cost paths with respect to the dual capacity cost.

If the total capacity y_e is replaced by working capacity y_e^1 for the primary flows and backup capacity y_e^2 for the backup flow with corresponding dual variables π_e^1, π_e^2, the pricing problem for each demand is to find a disjoint path pair $r = (p, q)$ minimizing $\langle r \rangle = \sum_{e \in p} \pi_e^1 + \sum_{e \in q} \pi_e^2$. This problem is $\mathcal{NP}$-hard already for single link failures [16]. If, however, the capacity of a link is defined exactly by its flow, as in model (5), both dual values can be replaced by the original cost ξ_e of link $e \in \mathcal{E}$, and both the pricing problem and the original problem reduce to finding a cycle $r \in \mathcal{R}_d$ minimizing $\sum_{e \in r} \xi_e$. We note that already for double link failures, the pricing problem of finding a shortest failure-disjoint pair of paths is $\mathcal{NP}$-hard [4].

SB with shared protection capacity – SB-S Now we assume that the pool of protection capacity is shared between the demands and states. This means that in a failure state $s \in \mathcal{S}$, capacity is used by all working paths and by those backup paths whose primary path fails in state s (in contrast to all backup paths for SB-D). This leads to the following relaxation of the primal problem SB-S, which is equivalent to the LP relaxation of failure-independent

problem	stub release, sharing	restoration type	failure type and complexity	
			single	multiple
PD	yes, dedicated	none	polynomial (a) [15]	$\mathcal{NP}$-hard (c) [Sec. 3]
UR	yes, shared	unrestricted	polynomial (a)	polynomial (a)
FD-nSR	no, shared	restricted, FD	polynomial (a) [9]	$\mathcal{NP}$-hard (c) [10]
FD-SR	yes, shared	restricted, FD	$\mathcal{NP}$-hard (d) [9, 8]	$\mathcal{NP}$-hard (d)
FI-nSR	no, shared	restricted, FI	$\mathcal{NP}$-hard (e) [11]	$\mathcal{NP}$-hard (e)
FI-SR	yes, shared	restricted, FI	$\mathcal{NP}$-hard (d) [10]	$\mathcal{NP}$-hard (d)
SB-D	no, dedicated	single-path, FI	polynomial (b) [13]	$\mathcal{NP}$-hard (b) [4]
SB-S	no, shared	single-path, FI	$\mathcal{NP}$-hard (e) [11]	$\mathcal{NP}$-hard (e)

Table 1: Complexity of pricing problems for different survivability mechanisms

shared path protection as studied in [11]:

$$\min \sum_{e \in \mathcal{E}} \xi_e y_e \tag{6a}$$

$$[\lambda_d \geq 0] \qquad \sum_{r \in \mathcal{R}_d} x_r \geq h_d \qquad d \in \mathcal{D} \tag{6b}$$

$$[\pi_e^s \geq 0] \qquad \sum_{d \in \mathcal{D}} h_d \sum_{\substack{r=(p,q) \in \mathcal{R}_d \cap \mathcal{R}_e: \\ s \in \bar{\mathcal{S}}_p}} x_r \leq y_e \qquad e \in \mathcal{E},\ s \in \mathcal{S} \tag{6c}$$

$$x, y \geq 0. \tag{6d}$$

The pricing problem for SB-S consists in finding, for each demand $d \in \mathcal{D}$, a pair of failure-disjoint paths $r = (p, q)$ from u_d to v_d minimizing

$$\langle r \rangle = \sum_{e \in p} \xi_e + \sum_{e \in q} (\sum_{s \in \bar{\mathcal{S}}_p} \pi_e^{s*}). \tag{7}$$

The link metrics for calculating the length of the primary path $p \in \mathcal{P}_d$ are equal to the true unit link costs ξ_e, while the link metrics for the backup path $q \in \mathcal{Q}_p$ depend on the states where the primary path fails. Stidsen et al. [11] showed by reduction to 3-SAT that this problem is $\mathcal{NP}$-hard already for single link failures, and that it can be modeled and solved as a shortest-path problem with $|\mathcal{S}|$ resource constraints.

5 Summary and conclusions

In the previous sections, we have given examples of pricing problems for most of the complexity cases of path generation listed in the introduction. Table 1 summarizes the complexity of the pricing problems considered in this paper and in [10]. In addition to PD and SB-D/SB-S, which have been discussed in this paper, the table also covers the following survivability mechanisms: unrestricted reconfiguration (UR), where all flows can be freely rerouted in case of a failure; failure-dependent restoration (FD), where the backup path for a given working path depends on the network situation; and, finally, failure-independent restoration (FI), where every working path has a backup path that is used in any situation where the primary path fails. The second column states whether non-failing but unused capacity is released in a failure situation (stub release, SR) or reserved for the normal network state (no stub release, nSR), and whether capacity is dedicated to a particular demand or shared between demands. The last two columns show the complexity of the pricing problem for single and multiple link failures, together with the information which of the complexity cases defined in the introduction determines the problem complexity, and the appropriate references. Note that wherever applicable, the $\mathcal{NP}$-hardness for multiple failures follows from the $\mathcal{NP}$-hardness for single failures.

In practical applications, node failures may have to be considered in addition to link failures. We have not discussed this case here due to lack of space. Although single node failures can be seen as a special case of multiple link failures, it makes sense to distinguish these cases. In [10], we have demonstrated that in contrast to general multiple link failures, incorporating single node failures into the set of failure states does not change the complexity of the pricing problems compared to single link failures.

Acknowledgment

The authors wish to express their gratitude to Andreas Bley, David Coudert, Arie Koster, Dritan Nace, Thomas Stidsen, Artur Tomaszewski, Marie-Émilie Voge, Roland Wessäly, and Mateusz Żotkiewicz for valuable discussions and helpful remarks on the presented subject.

This work has been funded by the following institutions: DFG Research Center MATHEON, Polish Ministry of Science and Higher Education (MNiSW, grant no. N517 397334 "Optimization Models for NGI Core Networks"), and Swedish Research Council (Vetenskapsrådet, grant no. 621-2006-5509 "Modeling and Design of Core Internet Networks"). The research has been carried out within the framework of two EC projects: COST Action 293 "Graphs and Algorithms in Communication Networks", and FP7 Network of Excellence "Euro NF".

References

[1] D. Coudert, P. Datta, S. Perennes, H. Rivano, and M-E. Voge. Shared risk resource group: Complexity and approximability issues. *Parallel Processing Letters*, 2006. To appear.

[2] G. Dahl and M. Stoer. A cutting plane algorithm for multicommodity survivable network design problems. *INFORMS Journal on Computing*, 10(1):1–11, 1998.

[3] R. Hassin, J. Monnot, and D. Segev. Approximation algorithms and hardness results for labeled connectivity problems. *Journal of Combinatorial Optimization*, 14(4):437–453, November 2007.

[4] J.Q. Hu. Diverse routing in optical mesh networks. *IEEE Trans. Com.*, 51(3):489–494, 2003.

[5] S. Irnich and G. Desaulniers. Shortest path problems with resource constraints. In G. Desaulniers, J. Desrosier, and M.M. Solomon, editors, *Column Generation*, pages 33–65. Springer, 2005.

[6] A.M.C.A. Koster and A. Zymolka. Demand-wise shared protection and multiple failures. In *Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium*, April 2007.

[7] A.M.C.A. Koster, A. Zymolka, M. Jäger, and R. Hülsermann. Demand-wise shared protection for meshed optical networks. *Journal of Network and Systems Management*, 13(1):35–55, March 2005.

[8] J-F. Maurras and S. Vanier. Network synthesis under survivability constraints. *4OR*, 2(1):53–67, March 2004.

[9] S. Orlowski. Local and global restoration of node and link failures in telecommunication networks. M.Sc. thesis, Technische Universität Berlin, February 2003. `http://www.zib.de/orlowski/`.

[10] S. Orlowski and M. Pióro. On the complexity of column generation in survivable network design with path-based survivability concepts. ZIB report ZR-08-51, November 2008. `http://opus.kobv.de/zib/volltexte/2008/1146/`.

[11] T. Stidsen, B. Petersen, K.B. Rasmussen, S. Spoorendonk, M. Zachariasen, F. Rambach, and M. Kiese. Optimal routing with single backup path protection. In *Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium*, 2007.

[12] J. Strand, A. L. Chiu, and R. Tkach. Issues for routing in the optical layer. *IEEE Communications Magazine*, pages 81–87, 2001.

[13] J. W. Suurballe. Disjoint paths in a network. *Networks*, 4:125–145, 1974.

[14] A. Tomaszewski, M. Pióro, and M. Żotkiewicz. On the complexity of resilient network design. *Networks*, 2009. To appear.

[15] R. Wessäly, S. Orlowski, A. Zymolka, A.M.C.A. Koster, and C. Gruber. Demand-wise shared protection revisited: A new model for survivable network design. In *Proceedings of the 2nd International Network Optimization Conference (INOC 2005), Lisbon, Portugal*, pages 100–105, March 2005.

[16] D. Xu, Y. Chen, Y. Xiong, C. Qiao, and X. He. On the complexity of and algorithms for finding the shortest path with a disjoint counterpart. *IEEE/ACM Transactions on Networking*, 14(1):147–158, 2006.

[17] S. Yuan, S. Varma, and J.P. Jue. Minimum-color path problems for reliability in mesh networks. In *Proceedings of the 24th IEEE Infocom 2005, Miami, USA*, volume 4, pages 2658–2669, March 2005.