

Optimal Routing with Single Backup Path Protection

Thomas Stidsen, *Technical University of Denmark*

Bjørn Petersen, *University of Copenhagen*

Kasper Bonne Rasmussen, *Technical University of Denmark*

Simon Spoorendonk, *University of Copenhagen*

Martin Zachariassen, *University of Copenhagen*

Franz Rambach, *Siemens Networks*

Moritz Kiese, *Technical University of Munich*

Keywords: single backup path protection, Dantzig-Wolfe decomposition, protection capacity minimization

1. Introduction

Today's information society relies increasingly on advanced communication networks. This has led to massive investments in increased communication network capacity. In order to utilize these investments the network operators perform *traffic engineering*, i.e., route communication in order to maximize the utilization of the capital invested in the communication network. Here we will consider traffic engineering of circuit switched networks where protection against single link failures is required.

The standard model of a circuit switched communication network is a directed graph $G = (V, A)$ consisting of a set of nodes V and a set of arcs A . The nodes correspond to telecommunication switches. The telecommunication switches route the communication signals through cables. We will assume that all cables enable bidirectional communication and hence model a single cable using two opposite arcs. We assume that a static communication demand is given which requires unidirectional communication between a origin node o_k and a terminating node d_k of volume DEMAND_k for a set of demands $k \in K$. We furthermore assume that a linear cost term c_a for using capacity on arc a has to be paid. If the optimization furthermore includes limitations on the capacity of each cable, i.e., limits the sum of the communication over the corresponding two arcs, the well known Multi Commodity Flow (MCF) problem [1] or one of its many variants has to be solved in order to perform traffic engineering.

Communication networks are increasingly required to be *reliable*. In this paper we only consider single cable failures [7, 12]. The classic path protection method employed in circuit switched networks is 1+1 protection, where two cable disjoint circuits (and hence arc disjoint circuits) are established and actively used. In case an arc fails on one path, the other path will survive and enable the receiving node to restore communication by just switching to the other incoming signal. This method is simple, there are well defined standards, but the required network capacity is always more than twice the required Non-Failure (NF) network capacity.

In this paper we will consider traffic engineering optimization methods for the Single Backup Path Protection (SBPP) method for circuit switched networks. This protection method is also called Shared Backup Path Protection [7] or Global Backup Path Protection [3]. The SBPP method is a slight variation of 1+1 protection: Instead of actively sending data packets on both paths, one path is designated the primary path — and only when the primary path fails are data packets sent along the other so-called backup path. Since we only guarantee protection against single cable failures, several backup paths may share capacity in the network.

In order to compare various protection methods, we define the following two measures: 1) Restoration Over Build (ROB) network capacity, which is the extra network capacity necessary to ensure protection, i.e. the network capacity subtracted by NF network capacity, assuming shortest path routing. 2) Relative Restoration Over Build (RROB) network capacity, which is the relative extra network capacity necessary to ensure protection, i.e. ROB network capacity divided by NF network capacity.

In order to utilize the path protection methods traffic engineering has to be performed in order to minimize the required RROB network capacity. When working with 1+1 protection this is a well studied problem for

which there exist polynomial-time algorithms [14]. This is *not* the case for the SBPP method. Because of the possibility of sharing the capacity for the backup paths, the best choice of primary path and backup path for each end-to-end demand node pair becomes interdependent. As noted in [7], p. 415: “This leads to a surprisingly difficult problem for exact solution and is currently an open area of research”.

In [12] the full SBPP traffic engineering problem is considered. A column generation approach, similar to the approach considered in this paper, is employed. The same mathematical model for the column generation master problem is formulated, but the sub-problem is *not* formulated. This means that if an optimal solution is required, the full set of disjoint paths has to be pre-generated, and this is only feasible for small networks.

The main focus of this paper is an exact method for solving the SBPP problem. We present a column generation approach, with an NP-hard sub-problem that is solved using an efficient labeling method. We quantify the exact relationship in costs for primary and backup paths and prove that the resulting optimization problem is NP-hard. Our experiments demonstrate that the SBPP method is a very cost-efficient protection method.

2. Dantzig-Wolfe formulation of the SBPP method

Given, as previously defined, a directed graph $G = (V, A)$ with nodes V and arcs A . For each failure situation $s \in S$ we define the set of failed arcs $a \in F_s$, where $F_s \subseteq A$. There is a cost c_a for using one unit of the capacity of an arc a . We further assume to know a static set of demand node pairs for which protected circuits using the SBPP method should be established. A directed connection between an origin node o_k and a terminating node d_k with a size of DEMAND_k should be established for each demand $k \in K$. The optimization objective is to minimize the cost of the required capacity when applying the SBPP method to protect the established circuits. The size of the necessary arc capacity is given by the variable θ_a which is multiplied to the cost per. unit of arc capacity c_a . This further means that for each demand a *pair* of directed failure disjoint paths needs to be found: A primary path p^{pri} and a backup path p^{bac} , both connecting node o_k to node d_k . A SBPP circuit therefore consists of a pair $\pi = (p^{pri}, p^{bac})$ of failure disjoint paths, and the variable λ_π^k gives the amount of communication flow through circuit π for demand $k \in K$. Each circuit belongs to the set of circuits which can satisfy a demand, i.e. $\pi \in P_k$. The arcs which constitute the primary path p^{pri} of circuit π are identified by the incidence matrix PRI_π^a , i.e., $\text{PRI}_\pi^a = 1$ if the primary path in π uses the arc a , otherwise $\text{PRI}_\pi^a = 0$. Correspondingly, the arcs which constitutes the backup path p^{bac} of circuit π are identified by the incidence matrix BAC_π^a . The requirement that primary path and backup path are failure disjoint can be formulated as follows:

$$\left(\sum_{a \in F_s} \text{PRI}_\pi^a \right) \cdot \left(\sum_{a \in F_s} \text{BAC}_\pi^a \right) = 0 \quad \forall k \in K, \pi \in P_k, s \in S$$

To simplify the formulation of the master program we define a new incidence matrix to decide whether it is necessary to use the backup path of circuit π in failure situation s :

$$\text{SWITCH_ON}_\pi^s = \left(1 - \prod_{a \in F_s} (1 - \text{PRI}_\pi^a) \right) \quad \forall k \in K, \pi \in P_k, s \in S$$

Given these definitions we are ready to formulate the LP-relaxed version of the MIP model for the SBPP traffic engineering cost minimization model. Given a set of disjoint path pairs $\pi \in P_k, k \in K$, and the corresponding incidence matrixes PRI_π^a and BAC_π^a , we can optimize the cost of the required cable capacities. If this set of paths is complete, the solution to the SBPP traffic engineering cost minimization model is optimal. The problem is that the number of disjoint path pairs grows exponentially with the network size and hence the complete model can only be solved for small network sizes. Instead, we will use a column generation algorithm such that only a subset of the path pairs are generated. The optimization subproblem to generate new path pairs with negative reduced costs is discussed in the next section.

minimize

$$\sum_{a \in A} c_a \cdot \theta_a \quad (1)$$

subject to

$$\sum_{\pi \in P_k} \lambda_\pi^k = \text{DEMAND}_k \quad \forall k \in K \quad (2)$$

$$\sum_{k \in K} \sum_{\pi \in P_k} \text{PRI}_\pi^a \cdot \lambda_\pi^k + \sum_{k \in K} \sum_{\pi \in P_k} \text{SWITCH_ON}_\pi^s \cdot \text{BAC}_\pi^a \cdot \lambda_\pi^k \leq \theta_a \quad \forall s \in S, a \in A \setminus F_s \quad (3)$$

$$\lambda_\pi^k, \theta_a \in R_+$$

The objective function (1) is the total network capacity cost. The demand constraints (2) ensures that enough capacity is established on the disjoint paths. The capacity constraints (3) ensures that enough capacity is allocated to route the communication on each arc a in each failure situation s which does not disrupt the arc.

3. Quadratic Cost Disjoint Path Problem

For the master problem for SBPP, let $\alpha_k \geq 0$, $k \in K$, be the dual variables associated with the constraints (2), and let $\beta_a^s \geq 0$, $s \in S$, $a \in A \setminus F_s$, be the dual variables associated with constraints (3). Our task is to decide if there exists a pair of primary and backup paths $\pi = (p^{pri}, p^{bac})$ from some origin node o_k to some terminating node d_k with negative reduced cost (for any $k \in K$).

The reduced cost of a pair of paths (p^{pri}, p^{bac}) is computed as follows. The cost of an arc $a \in p^{pri}$ is $\sum_{s \in S} \beta_a^s$, while the cost of an arc $a \in p^{bac}$ is $\sum_{s \in S, F_s \cap p^{pri} \neq \emptyset} \beta_a^s$. Note the asymmetry in the definition of arc costs in primary and secondary paths: For an arc on the primary path the sum taken over *all* failure situations, while for an arc on the backup path the sum is only taken over the failure situations that affect an arc on the *primary* path. The total reduced cost of (p^{pri}, p^{bac}) is now

$$-\alpha_k + \overbrace{\sum_{a \in p^{pri}} \sum_{s \in S} \beta_a^s}^{\text{primary path cost}} + \overbrace{\sum_{a \in p^{bac}} \sum_{s \in S, F_s \cap p^{pri} \neq \emptyset} \beta_a^s}^{\text{backup path cost}}$$

The *Quadratic Cost Disjoint Path Problem (QCDPP)* is to compute a pair of paths $\pi = (p^{pri}, p^{bac})$ with minimum total cost. Since the dual variables β_a^s are non-negative, there clearly exists an optimal solution where both the primary path p^{pri} and the backup path p^{bac} are simple. Hence in the following we require that the paths p^{pri} and p^{bac} are simple and arc disjoint.

We consider two variants of failure situations: In the *single arc* failure variant there is one failure situation for each arc in A . In the *single link* failure variant there is one failure situation for each pair of opposite arcs, i.e., when the corresponding undirected edge is broken.

NP-Completeness

QCDPP is NP-hard for both the single arc and single link failure variants. Here we briefly sketch the proof for the single arc variant. In the single arc variant the set of failure situations S is identical to the set of arcs A . The decision version of QCDPP with single arc failures is formally defined as follows (where the constant term $-\alpha_k$ in the objective function of QCDPP is ignored):

INSTANCE: Directed graph $G = (V, A)$, pairwise (integer and non-negative) costs β_a^f for all ordered pairs of arcs $(a, f) \in A \times A$, origin node $o_k \in V$, terminating node $d_k \in V$ and integer C .

QUESTION: Does there exist a pair of simple arc disjoint paths $\pi = (p^{pri}, p^{bac})$ from o_k to d_k in G such that

$$\sum_{a \in p^{pri}} \sum_{f \in A} \beta_a^f + \sum_{a \in p^{bac}} \sum_{f \in p^{pri}} \beta_a^f \leq C ?$$

We prove that this problem is NP-complete by reduction from 3-SATISFIABILITY (3SAT). It is obvious that the decision version of QCDPP is in NP, since given $\pi = (p^{pri}, p^{bac})$ we can compute the corresponding cost and compare it to C in polynomial time.

Let (U, C) be an instance of 3SAT, where $U = (x_1, x_2, \dots, x_n)$ is a finite set of n variables and $C = (c_1, c_2, \dots, c_m)$ is a set of clauses where $|c_i| = 3$, $i = 1, \dots, m$. We assume without loss of generality that each variable appears in at least one clause.

Based on the 3SAT instance we create an instance of the QCDPP. The constructed graph consists of two chains of parallel arcs connecting o_k and d_k — the so-called top chain and the bottom chain. Two node disjoint paths from o_k to d_k have the property that one of the paths travels through the top chain while the other travels through the bottom chain. The arcs in the top chain are denoted *variable* arcs, while the arcs in the bottom chain are denoted *clause* arcs. For each clause $c_i \in C$ we have 8 parallel arcs, one for each combination of assignments for the three literals. Similarly, we have two variable arcs for each variable x_j , one arc for $x_j = 0$ and one arc for $x_j = 1$. By assigning pairwise costs β_a^f for all ordered pairs of arcs $(a, f) \in A \times A$ appropriately, and setting $C = 0$, we can prove that we have YES-instance for QCDPP if and only if we have a YES-instance for 3SAT. Therefore, the decision version of QCDPP is NP-complete.

Label-setting algorithm for the QCDPP

The QCDPP can be formulated as a Shortest Path Problem with Resource Constraints (SPPRC). The SPPRC can be solved in pseudo-polynomial time when the number of resources are constant. Formulating the QCDPP as a SPPRC leads to a graph with a resource for each failure situation. When regarding the single arc (link) variant this adds up to one resource per arc (link), i.e., the complexity becomes exponential.

The SPPRC is a common subproblem in many graph based problems when using Dantzig-Wolfe decomposition, e.g. the Vehicle Routing Problem with Time Windows [9, 10] and Crew Pairing [4]. The SPPRC can be solved with a label-setting algorithm which is based on dynamic programming [8, 9, 11]. A label-setting algorithm enumerates all paths by extending partial paths in all feasible directions. Dominance is applied so only pareto-optimal paths are extended. The partial paths are denoted labels. For further details we refer to the papers mentioned above.

Next we consider the transformation of the QCDPP into an SPPRC. Consider the graph $G = (V, A)$ for the QCDPP where a least cost primary and a backup path must be found from o_k to d_k . Note that the cost of the backup path depends on the arcs used on the primary path. The nodes are duplicated into V' , the arcs are duplicated and reversed into A' , i.e. $(i, j) \in A \Leftrightarrow (j', i') \in A'$, and a connecting arc from d_k to d'_k is added resulting in the graph $G' = (V \cup V', A \cup A' \cup \{(d_k, d'_k)\})$. Also for each failure situation $s \in S$ make a duplicate of the set F_s , i.e., $(j', i') \in F'_s$ is the set of arcs from A' that are duplicates of the arcs $(i, j) \in F_s$ belonging to A . Now the primary path is sought in the first part of the graph containing the nodes V and arcs A ; then shifting to the other part of the graph when reaching d_k and looking for a reverse backup path from d'_k to o'_k using nodes V' and arcs A' . Given a failure situation $s \in S$ it is enforced that no arcs $(j', i') \in F'_s$ can be used on the backup path if an arc $(i, j) \in F_s$ is used on the primary path. This can be done by adding a binary resource per failure $s \in S$, i.e., for each set F_s a resource is added enforcing that if any arc in F_s is used then no arcs in F'_s can be used.

The cost of an arc $a \in A$ of the primary path is given as $c_a = \sum_{s \in S} \beta_a^s$ while the cost of an arc $a' \in A'$ of the backup path is a function depending on the label L being extended:

$$c(a', L) = \sum_{s \in S: s(L)=1} \beta_a^s$$

where $s(L)$ is the resource indicating if failure situation s is covered or not. The cost of the connecting arc (d_k, d'_k) is $-\alpha_k$. Solving the SPPRC on the graph G' for a demand k given the resources for each failure situation gives a path p with reduced cost:

$$c_{reduced}^k = \sum_{a \in A(p)} \sum_{s \in S} \beta_a^s + \sum_{a' \in A(p)} \sum_{s \in S: s(p)=1} \beta_{a'}^s - \alpha_k$$

This still leaves $|K|$ pricing problems to be solved in order to prove optimality. However, if there exist several demands with the same origin node, an alternative is to solve a pricing problem for each such origin node to all the terminating nodes in a single SPPRC. In such a pricing problem add arcs between (d_k, d'_k) for the relevant terminating nodes with cost $-\alpha_k$. This results in several linking arcs from V to V' , but does not change the complexity of the SPPRC compared to the SPPRC for a single demand.

4. Experimental results

In this section the SBPP method is tested by applying the column generation algorithm, described in Section 2, to 5 real-world networks. Because it is difficult to obtain realistic demand matrices, we choose the very simple demand matrix $D^{kl} = 1$ for $k < l$, i.e. one connection between each pair of nodes.

The size of the test networks and the results when applying the column generation for SBPP protection is given in Table 1. The first column gives name of the network and the three next the number of nodes, links and the average node degree of the network. The NF network capacity is given in the fourth column. The fifth column gives the complete rerouting lower bound for protection [13] and the sixth column the corresponding RROB value for the lower bound. The seventh column gives the absolute network capacity required for the SBPP method, the eighth column the RROB value for the SBPP method and finally the ninth column contains the RROB gap between the lower bound for protection and the SBPP method.

	Network size		Avg.	NF	CR		SBPP		RROB
	nodes	links	Node Degree	Capacity	Abs.	RROB	Abs.	RROB	gap
Cost239 [2]	11	26	4.73	86	97.6	1.13	102.3	1.19	0.06
Europe	13	21	3.23	158	248.0	1.57	260	1.65	0.08
USA [5]	28	45	3.21	1273	1914.2	1.50	1967.8	1.55	0.05
Italy [6]	33	68	4.12	1718	2299.4	1.34	2352.7	1.37	0.03
France [5]	43	71	3.3	3473	5077	1.46	5220.3	1.50	0.04
Avg.						1.40		1.45	0.05

Table 1: Network protection requirements for SBPP protection compared to NF and CR.

We find the results in Table 1 interesting because it shows how efficient the SBPP method is. The RROB gap is always lower than 8% and it seems to improve as the average node degree increases and as the size of the networks grows. A possible explanation could be that when the SBPP method gets a greater freedom to look for capacity sharing it improves compared to the lower bound. These results are preliminary and many more tests are needed to confirm them. Still we find it very interesting that the protection method is so efficient. It hence seems rather un-necessary to search for even more efficient methods through more complex protection methods such as Full Backup Path Protection, which is the theoretically most efficient path protection method, but which is also significantly more complicated.

The described model for single backup path protection is quite abstract and in the future we intend to make it more realistic in two different respects: 1) Modular capacities: The current model allows allocation of any necessary capacity to each link. 2) Bifurcation: The current model allows the demand volume be split in fractional parts.

If integer constraints are put on the paths and the capacities, the model becomes a mixed integer programming

model and hence a branch-and-price algorithm is necessary. We intend to develop such an algorithm in the near future.

References

- [1] Ravindra K. Ahuja, James B. Orlin, and Thomas L. Magnanti. *Network Flows : Theory, Algorithms, and Applications*. Prentice Hall, 1993.
- [2] P. Batchelor, B. Daino, P. Heinzmann, D. R. Hjelm, P. Leuthold, R. Inkret, G. De Marchis, H. A. Jager, F. Matera, M. Joindot, B. Mikac, A. Kuchar, H.-P. Nolting, E. Coquil, J. Spath, F. Tillerot, B. Caenegem, N. Wauters, and C. Weinert. Study on the implementation of optical transparent transport networks in the european environment-results of the research project cost 239. *Photonic Network Communication*, 2(1):15–32, 2000.
- [3] E. Calle, J.L. Marzo, and A. Urrea. Protection performance components in mpls networks. *Computer Communications*, 27(12):1220–1228, 2004.
- [4] Guy Desaulniers, Jacques Desrosiers, Y. Dumas, S. Marc, B. Rioux, Marius M. Solomon, and Francios Soumis. Crew pairing at air france. *European Journal of Operations Research*, 97:245 – 259, 1997.
- [5] J. Doucette, D. He, W.D. Grover, and O. Yang. Algorithmic approaches for efficient enumeration of candidate p -cycles and capacitated p -cycle network design. *Design of Reliable Communication Networks*, pages 212–220, 2003.
- [6] W. Grover, J. Doucette, M. Clouqueur, D. Leung, and D. Stamatelakis. New options and insights for survivable transport networks. *IEEE Communications Magazine*, 40(1):34–41, 2002.
- [7] W. D. Grover. *Mesh-Based Survivable Networks*. Prentice Hall PTR, 2004.
- [8] S. Irnich and G. Desaulniers. Shortest path problems with resource constraints. In G. Desaulniers, Jacques Desrosiers, and M.M. Solomon, editors, *Column Generation*, GERAD 25th Anniversary Series, chapter 2, pages 33–65. Springer, 2005.
- [9] Stefan Irnich and Daniel Villeneuve. The shortest path problem with resource constraints and k -cycle elimination for $k \geq 3$. Forthcoming in: *INFORMS Journal of Computing*, 2006.
- [10] Mads Jepsen, Bjørn Petersen, Simon Spoorendonk, and David Pisinger. Subset row inequalities applied to the vehicle routing problem with time windows. Forthcoming in: *INFORMS Operations Research*, 2006.
- [11] Bjørn Petersen and Simon Spoorendonk. Label-setting algorithms for variants of the shortest path problem with resource constraints. DIKU University of Copenhagen, Denmark, Working paper, 2006.
- [12] Michal Pioro and Depankar Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Elsevier, 2004.
- [13] T. Stidsen and P. Kjærulff. Complete rerouting protection. *Journal of Optical Networking*, 2005. accepted.
- [14] J.W. Suurballe. Disjoint paths in a network. *Networks*, 4(2):125–45, 1974.