Optimal Routing with Single Backup Path Protection

Simon Spoorendonk

DIKU, University of Copenhagen, Denmark

24th of November

Based on joint work with: Thomas Stidsen (DTU), Bjørn Petersen (DIKU), Kasper Bonne Rasmussen (DTU), and Martin Zachariasen (DIKU) [4]
1. **Path Protection**
   - Path protection schemes

2. **Protection Pricing**

3. **SBPP**
   - Mathematical Model
   - Example

4. **QCDPP**
   - Mathematical Model
   - NP-hard
   - Branch-and-Bound Algorithms
   - Shortest Path Problem with Resource Constraints

5. **Test Results**

6. **Open Problems**

Simon Spoorendonk

Optimal Routing with Single Backup Path Protection

DIKU, University of Copenhagen, Denmark
Tele Communication Networks

- Minimize capacity usage on network when meeting customer demand
- Provide reliable service, e.g., in case of cable failure

![Diagram of telecommunication network with backup paths highlighted.](image)
Path Protection

We define path protection in the following way:

- Each circuit contains **one primary connection**. This connection functions in the normal situation.
- If the primary connection fails, the signal is routed along **one backup connection**.
- The paths available for the backup connection depends on the path protection scheme.
1+1 Path Protection

- Sends packets on both paths
- More than twice the non-failure capacity required
Full Backup Path Protection

- Theoretical must efficient method
- Each possible cable failure is protected by a possible unique backup path
- Lots of capacity sharing between backup paths

The backup path for this particular edge failure
Single Backup Path Protection (SBPP)

- Only sends packets on backup path if a failure occurs
- Allows sharing capacity of backup paths
Reverse Single Backup Path Protection (Haskin)

- Variant of SBPP where data is routed back to start node
- Increases capacity requirements but decreases notification time

The global backup path

Start node

Active Path

Terminating node

Simon Spoorendonk
DIKU, University of Copenhagen, Denmark
Segment Backup Path Protection

- Segments of primary path cables are protected by same backup path

The backup path for this segment

1. Start node
2. Active Path
3. Notification
4. Terminating node
Local Backup Path Protection (LBPP)

- Reroutes the backup path between the failed end nodes of the primary path

![Diagram of LBPP]

1. Start node
2. Active Path
3. Terminating node

The local backup path
Dynamic Backup Path Protection

- Variant of LBPP where backup paths are routed directly from the start of failed connection to the target node.
Front Dynamic Backup Path Protection

- Another variant of LBPP where backup paths are rerouted from start node to end of failed connection

The front local backup path

Start node → → → Active Path → Terminating node
Given we have a network, what should users pay to use it?

- Obviously the users should pay for the capacity they use.
- We will assume the users pay a price linearly dependent on the use of cable capacity.
- The price for a circuit connection is then simply the sum of the capacity costs.
- This leads to an shortest path routing for establishing unprotected circuit connections.
The Problem

First the easy problem, 1+1 protection:

- How much should connection A and connection B pay for the use of the arc $a_{56}$
The Problem

First the easy problem, 1+1 protection:

- How much should connection A and connection B pay for the use of the arc $a_{56}$
- But what about sharing? How much should connection A and connection B pay for the use of the arc $a_{56}$?
The Problem

First the easy problem, 1+1 protection:

- How much should connection A and connection B pay for the use of the arc $a_{56}$
- But what about sharing? How much should connection A and connection B pay for the use of the arc $a_{56}$?
- And now, how much should connection A and connection B pay for the use of the arc $a_{56}$?
Shared Capacity

In general: How should users pay for shared protection capacity?

- Wayne Grover: ”This leads to a surprisingly difficult problem for exact solution and is currently an open area of research” [1].

- But paying for protection capacity is not a unique telecommunication planning problem, the same issue arise in e.g. electricity production planning ...
We focus on SBPP for the following reasons:

- Simple intuitive and easy to explain because of the close resemblance with 1+1 protection
- Very capacity efficient
- Can be controlled from edge routers
- Seems ideally suited for Multi Path Label Switching [5] implementation used in backbones

There are drawbacks with SBPP:

- Not very fast recovery time, because of the notification time.
- Not the most efficient path protection method
The abstract model

We choose to use a very simple model of the path protection problem:

- We assume an oriented network $G(V, A)$ of nodes $i \in V$ and arcs $a \in A$
- We assume a known static demand matrix $D^k$ of the volumes of protected circuit connections from origin node $o_k$ to terminating node $t_k$
- The (linear) cost for using an arc is $c_a$
- The objective to be minimized is the sum of costs for the capacity use in each arc (link).
Failure Situations

We model the possible failures with a number of failure situations $s \in S$:

- The set $F_s \subseteq A$ enumerates the arcs which fail in situation $s$, i.e. the failed arcs are $f \in F_s$
- We model single link failures by creating the failure sets $F_s$ of the two arcs modeling each link
- In principle we can model any failure situation, but only if the network stays connected ...
Mathematical Model

Single Backup Path Protection

We will formulate the problem directly as a column generation problem where the variables:

- $\lambda^k_p$: Correspond to a path-pair $p \in P_k$ for demand $k$.
  - $PRI^k_{p,a} \in \{0, 1\}$: The incidence matrix for the primary path, i.e. 1 when primary path $k \in P_k$ for demand $k$ use arc $a \in A$
  - $BAC^k_{p,a} \in \{0, 1\}$: The incidence matrix for the backup path, i.e. 1 when primary path $k \in P_k$ for demand $k$ use arc $a \in A$
  - The primary path and the backup path needs to be failure disjoint, i.e. $PRI^k_{p,a} + BAC^k_{p,a'} \leq 1 \ \forall k, p, s: a, a' \in F_s$
- $\theta_a$: Used to record how much capacity is needed
When to backup?

Given the incidence matrixes when should the backup path be used?

- The backup paths should be used when the primary path fails in a failure situation $s$
- $SWITCH\_ON^k_{p,s}$: The incidence matrix for switching on the backup path:

$$SWITCH\_ON^k_{p,s} = (1 - \prod_{a \in F_s} (1 - PRI^k_{p,a})) \quad \forall \ k, p, s$$
The Master problem

Min:

$$\sum_{a \in A} c_a \cdot \theta_a$$

s.t.:

$$\sum_{p \in P_k} \lambda_k^p = d_k \quad \forall \, k$$

$$\sum_k \sum_{p \in P_k} PRI_{p,a}^k \cdot \lambda_k^p + \sum_k \sum_{p \in P_k} SWITCH\_ON_{p,s}^k \cdot BAC_{p,a}^k \cdot \lambda_k^p \leq \theta_a \quad \forall \, a, s : a \notin F_s$$

$$\lambda_k^p, \theta_a \in R_+$$
Exponential paths ...

The number of possible paths grows exponentially with the size of the network (and also if the average node degree increases ...)  

- We want to generate the path possibilities on the fly!
- In this way we (in principle) only need a (tiny) subset of the path-pairs, $|K| + \frac{1}{2} \cdot |A| \cdot |A|$
- We will start with dummy path pairs which are guaranteed to be too expensive to be selected in the final solution ...
- ... we simply select all links in all failure situations.
- How can we select improving path pairs ?
- And more fundamentally: What is the price of SBPP path-pairs ?
The prices

Dual variables are:

- $\alpha_k \geq 0$: The (highest) price we are currently paying for satisfying demand $k$ with a disjoint path pair

- $\beta_s^a \geq 0$: The price the *backup* path has to pay for using arc $a$ in failure situation $s$
We need to calculate the *reduced cost* of the path pairs:

- \( c_{\text{reduced}} = c_{\text{original}} - \sum_i \pi_i a_i \)
- In our case it consists of:
  - \( c_{\text{original}} = 0, \) there are no direct costs!
  - \( c_{\text{demand}} = \alpha_k \) dual variables only participate once
  - \( c_{\text{primary},k,p \in P_k} = \sum_a PRI_{p,a}^k (\sum_s \beta_s^a) \)
  - \( c_{\text{backup},k} = \sum_a \sum_s SWITCH\_ON_{p,s}^k \cdot BAC_{p,a}^k \cdot \beta_s^a \)

- Notice we for a given demand \( k \) only need to consider the cost of the path-pair since the reward \( \alpha_k \) is constant.
A numerical example

A very simple example network, with $\beta$ arrays for each arc (the dash "-" ) correspond to the arc it self...
A numerical example

- A very simple example network, with $\beta$ arrays for each arc (the dash “-”) correspond to the arc it self ...
- There are two possible path pairs:
Example

A numerical example

- A very simple example network, with $\beta$ arrays for each arc (the dash “-”) correspond to the arc itself ...
- There are two possible path pairs:
- $[(AB, BC), (AD, DC)] = 3.5 + 6 + 3.4 + 1.3 = 14.2$
A numerical example

- A very simple example network, with $\beta$ arrays for each arc (the dash “-”) correspond to the arc it self ...
- There are two possible path pairs:
  - $[(AB, BC), (AD, DC)] = 3.5 + 6 + 3.4 + 1.3 = 14.2$
  - $[(AD, DC), (AB, BC)] = 5.4 + 5.4 + 0.5 + 1.3 = 12.6$ (The winner)
Quadratic Cost Disjoint Path Problem (QCDPP)

How hard is the QCDPP?

- The problem is NP-complete

How can we solve the QCDPP? Optimal solution methods:

- Formulate MIP and solve with Branch-and-Bound
- Reformulate as an SPPRC and solve with a Label Setting Algorithm

Heuristic approaches:

- Regard reduced networks
- General MIP heuristics on MIP formulation
- Truncated Label Setting Algorithms (relaxed dominans criteria, truncated state space)
Mathematical Model

The Path-Pair generation problem I

Min:

\[ c_{\text{reduced}}^k = \sum_{a \in A} \left( \sum_{s \in S} \beta^s_a \right) \cdot x_a + \sum_{a \in A} \sum_{s \in S} \beta^s_a \cdot z^s_a - \alpha_k \]

s.t.:
The Path-Pair generation problem II

s.t.:

\[
\sum_{a \in \delta_i} x_a - \sum_{a \in \gamma_i} x_a = \begin{cases} 
1 & i = s \\
-1 & i = t \quad \forall \ i \\
0 & i 
\end{cases}
\]

\[
\sum_{a \in \delta_i} y_a - \sum_{a \in \gamma_i} y_a = \begin{cases} 
1 & i = s \\
-1 & i = t \quad \forall \ i \\
0 & i 
\end{cases}
\]
Mathematical Model

The Path-Pair generation problem II

\[ \begin{align*}
\text{s.t.:} & \\
|F_a| \cdot u_s & \geq \sum_{a \in F_a} x_a \quad \forall s \\
|F_a| \cdot v_s & \geq \sum_{a \in F_a} y_a \quad \forall s \\
u_s + v_s & \leq 1 \quad \forall s \\
z^s_a & \geq u_s + y_a - 1 \quad \forall s, a \\
x_a, y_a, u_s, v_s & \in \{0, 1\}, \quad z_{ij,qr} \in [0, 1]
\end{align*} \]
QCDPP NP-completeness

The decision version of the QCDPP problem states the following question: Does there exist a pair of simple arc disjoint paths $(p^{pri}, p^{bac})$ from $s$ to $t$ in $G$ such that

$$
\sum_{a \in p^{pri}} \sum_{f \in A} \beta_a f \quad + \quad \sum_{a \in p^{bac}} \sum_{f \in p^{pri}} \beta_a f \quad \leq \quad C
$$
QCDPP NP-completeness

To prove NP-completeness (in the strong sense) we have to:

- Prove that we can check a solution in polynomial time (easy)
- Select an existing NP-complete problem to reduce from: 3SAT
- Describe a reduction from the 3SAT [2] problem to the QCDPP problem.
Polynomial checking of QCDPP solutions

Given a solution:

$$(p^{pri}, p^{bac}) = ([o, x_1, x_2, ..., x_n, t], [o, y_1, y_2, ..., y_n, t])$$

- Check that primary path and backup path are failure disjoint, i.e. link disjoint.
- Calculate the summed cost of the primary path and the backup path.
3SAT and QCDPP

Given a pair \((U, C)\) where \(U = (x_1, x_2, ..., u_n)\) is a finite set of \(n\) variables and \(C = (c_1, c_2, ..., c_m)\) is a finite set of \(m\) clauses like: \((x_1 \lor x_2 \lor \overline{x_3})\). The basic question: Is there a setting of the variables \(U\) such that all clauses are satisfied, i.e. \(c_1 \land c_2 \land ... \land c_m = \text{TRUE}\)
Simpler QCDPP

To make the proof simpler we make a few alterations which simplifies the proof:

- Each arc corresponds to a failure situation (easy to change).
- We will construct a multigraph (again, easy to change).
- We only consider 0 or 1 values for the $\beta_a^s$ dual values.
Given a 3SAT problem construct the following figure:

Set the values for $\beta^s_a$ for all pairs of arcs to zero, except:
Set the values for $\beta^s_a$ for all pairs of arcs.

- For the bottom (clause) arcs 000, set $\beta^s_a$ to 1 for the arc $a$ to one randomly chosen variable $s$.
- For the upper (variable) arcs, set $\beta^s_a = 1$ for all pairs of arcs $a$ and $s$ such that a clause containing that variable fails.
Given this setup we claim:

- Given a network corresponding to a 3SAT problem solving the QCDPP problem leads to a 3SAT solution, i.e. cost 0

- Given a 3SAT solution and the network we can (in polynomial time) transform the 3SAT solution to a solution of the QCDPP problem by choosing the corresponding arcs.
What if we simply solve the subproblem using standard MIP solvers?

- Feasible approach for small to medium problems only...
- Possible because problem is quite constrained and we are using sparse networks
Shortest Path Problem with Resource Constraints (SPPRC)

Problem definition:

- A directed graph with edge weights (possible negative)
- Resource constraints, e.g. capacity, time
- Find the least cost path between a pair of nodes.

Very common subproblem in decomposition algorithms
Shortest Path Problem with Resource Constraints

Algorithm

Pseudo-polynomial running time when the number of resources are constant.

Label Setting Algorithm:

- Dynamic programming approach
- Extend partial paths (labels)
- Only maintain pareto-optimal paths through dominance
- Implicit enumeration algorithm
Reformulating the QCDPP into a SPPRC

- Duplicate the graph into a primary and a backup part
- In the backup part all arcs are reversed
- Connect the end nodes of a pair with each other, i.e. this is the link between the two parts of the new graph
- A path from the start node to the copy of the start node corresponds to a primary path and a reverse backup path
Shortest Path Problem with Resource Constraints

Arc costs

- Arc $a \in A$ on primary path: $c_a = \sum_{s \in S} \beta^s_a$
- Arc $a' \in A'$ on backup path depends on label $L$: $c(a', L) = \sum_{s \in S : s(L) = 1} \beta^s_a$
- Arc from end to copy of end node: $c(d_k, d'_k) = -\alpha_k$
- The path cost is:

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

Simon Spoorendonk
DIKU, University of Copenhagen, Denmark

Optimal Routing with Single Backup Path Protection
Single pair disjoint paths

We copy the original network to two different parts:
Single to many disjoint paths

Add several links between all terminating node to their copies for a single start node:
Shortest Path Problem with Resource Constraints

Observations

- All arc costs are positive: $\beta^f_a \geq 0$
- If primary path cost more than $\alpha_k$ no need to extend path
- However, hard to dominate paths before switching to backup path
- Branch-bound-Bound is competitive if dominance criteria is weak
Test Networks

- 5 Networks the SNDlib [6]
- RROB: Relative Restoration Over Build, i.e. the network capacity necessary for protection divided by the network capacity necessary for the un-protected case.

<table>
<thead>
<tr>
<th></th>
<th>Nodes</th>
<th>Edges</th>
<th>Avg. Node Degree</th>
<th>Number of Demands</th>
</tr>
</thead>
<tbody>
<tr>
<td>newyork</td>
<td>16</td>
<td>49</td>
<td>6.12</td>
<td>240</td>
</tr>
<tr>
<td>ta1</td>
<td>24</td>
<td>51 (55)</td>
<td>4.58</td>
<td>396</td>
</tr>
<tr>
<td>france</td>
<td>25</td>
<td>45</td>
<td>3.6</td>
<td>300</td>
</tr>
<tr>
<td>norway</td>
<td>27</td>
<td>51</td>
<td>3.78</td>
<td>702</td>
</tr>
<tr>
<td>cost266</td>
<td>37</td>
<td>57</td>
<td>3.08</td>
<td>1332</td>
</tr>
</tbody>
</table>

Table: The tested networks
SBPP efficiency

Table: Network protection requirements for SBPP protection compared to NF and CR [3]
Very little difference!
But also large architecture dependence
What about the Average node degree?
It is difficult to conclude something based on so few examples
RROB seems to be dropping as the average node degree grows ...
... but there are large variations ...
Efficiency Comments

The results seems to indicate:

- SBPP protection is *very* efficient ...
- ... can at most be improved by less than 5%
## Running time: Brute force version

<table>
<thead>
<tr>
<th>Location</th>
<th>It</th>
<th>Total Sec.</th>
<th>Master Sec.</th>
<th>Sub Sec.</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>newyork</td>
<td>1622</td>
<td>123284.15</td>
<td>3461.59</td>
<td>119818.33</td>
<td>97.19</td>
</tr>
<tr>
<td>ta1</td>
<td>1190</td>
<td>5850.22</td>
<td>840.66</td>
<td>4998.58</td>
<td>85.44</td>
</tr>
<tr>
<td>france</td>
<td>1100</td>
<td>3003.47</td>
<td>722.55</td>
<td>2271.49</td>
<td>75.63</td>
</tr>
<tr>
<td>norway</td>
<td>1593</td>
<td>16612.94</td>
<td>2848.51</td>
<td>13748.79</td>
<td>82.76</td>
</tr>
<tr>
<td>cost266</td>
<td>2079</td>
<td>26075.23</td>
<td>12758.92</td>
<td>13253.95</td>
<td>50.83</td>
</tr>
</tbody>
</table>

**Table:** Running times in seconds
Running Time Comments

What can be said about the running times?

- Solution of the sub-problem (QCDPP) dominates the execution time, as expected ...

- ... but for the largest example, the solution time of the master problem is significant
On-line routing problem

QCDPP routing problem is also important in the On-Line case:

- Given a network, how to allocate SBPP connections one by one?
- If we have the $\beta_s^a$ values (or approximations of it) we can route SBPP connections by solving the QCDPP
Open Problems/Extension for SBPP

- Stub release SBPP, releasing capacity from primary path on unfailing arcs
- Not Bifurcation, i.e. all circuits for a demand is sent along the same path-pair (req. Branch-and-Price)
- Modular Capacities on edges, discount on large demand (req. Branch-and-Price)
- Capacity limitations on edges
- Node failure protection
- Double edge protection, protection against multiple failures
- Full Backup Path Protection, it is hard just to write the model
Grover W. D. 
*Mesh-Based Survivable Networks.*

Michael R. Garey and David S. Johnson.
Freeman, 1979.

T. Stidsen and P. Kjærulff.
Complete rerouting protection.

Thomas Stidsen, Bjørn Petersen, Kasper Bonne Rasmussen, Simon Spoorendonk, and Martin Zachariasen.
Optimal routing with single backup path protection.

G. Swallow and L. Andersson.
Mpls working group.

R. Wessly and M. Piro.
Snd-lib: The data source for all people working on optimizations of telecommunications networks.
http://sndlib.zib.de/home.action.