

# Two deterministic strategies for finding shortest pairs of edge-disjoint paths

Marina Girolimetto<sup>1</sup>, Marcia H. M. Paiva<sup>1</sup>, Rodrigo S. Tessinari<sup>1</sup>,  
Claunir Pavan<sup>2</sup>, Fabio O. Lima<sup>3</sup>

1: LabTel – Laboratório de Telecomunicações, UFES, Vitória, Brazil, 29075-910;  
marina.girolimetto@aluno.ufes.br, marcia.paiva@ufes.br, stange@inf.ufes.br

2: Ciência da Computação, UFFS, Chapecó, Brazil, 89815-899; claunir.pavan@uffs.edu.br

3: Engenharia de Controle e Automação, IFES, Serra, Brazil, 29173-087; fabio.lima@ifes.edu.br

## Resumo

From the observation that there may exist more than one shortest pair of edge-disjoint paths interconnecting a given pair of nodes in a network, and that the Suurballe and Tarjan's algorithm, which is usually used for finding them, is not deterministic, this work proposes two deterministic versions of that algorithm, which can be used to exploit different protection schemes for telecommunication networks. Given a network topology, the algorithms find what we call the more balanced and the less balanced pairs of edge-disjoint paths for each pair of nodes.

## 1 Introdução

Nas redes de telecomunicações, o projeto topológico deve garantir uma rede confiável e tolerante a alguns tipos de falhas, como a ruptura de um enlace ou falta de energia em um nodo [GP16]. Mecanismos de proteção contra falhas são importantes para evitar a suspensão da transmissão de dados na ocorrência de algum problema na rede. Quando há uma falha em um nodo ou em um enlace, diversos pares de nodos origem e destino podem ser afetados. Por isso, as redes devem ser sobreviventes a falhas. Consideramos que uma rede sobrevive a uma falha se houver caminhos alternativos para enviar o tráfego destes pares.

Para que uma rede sobreviva a uma falha em qualquer um de seus nodos ou enlaces, é necessário que

cada par origem e destino da rede esteja interligado por pelo menos dois caminhos que não compartilhem nodos. Por sua vez, se há pelo menos dois caminhos que não compartilham enlaces interligando cada par origem e destino, então a rede sobrevive a uma falha em um enlace. Neste trabalho, os dois caminhos definidos para cada par origem e destino são denominados *caminho de trabalho* e *caminho de proteção* e as redes apresentadas são sobreviventes a falhas de enlace.

Há políticas de roteamento nas quais, para cada par de nodos origem-destino, há um caminho de trabalho responsável por carregar o tráfego enquanto a rede estiver operando em condições normais, e um caminho de proteção responsável por carregar o tráfego em caso de falhas ou manutenção da rede. Existem ainda políticas de roteamento nas quais o tráfego do nodo origem ao nodo destino é enviado simultaneamente pelos caminhos de trabalho e de proteção, e apenas o sinal que chega no nodo destino com melhor qualidade é considerado [MBN99].

As topologias de redes de telecomunicações podem ser modeladas por grafos, onde os vértices representam os nodos, e as arestas representam os enlaces entre os nodos. Um grafo é dito conexo se e somente se existe pelo menos um caminho interligando cada par de nodos origem e destino. O comprimento de um caminho é a soma dos pesos das arestas que o constituem. A quantidade de arestas percorridas define o número de saltos do caminho. Considerando arestas de peso unitário, o comprimento do caminho equivale ao número de saltos. Quando existem pelo menos dois caminhos disjuntos por arestas interligando qualquer par de nodos origem e destino da rede, o grafo é dito *2-aresta-conexo* [Net06]. Aqui, as topologias consideradas são *2-aresta-conexas* com arestas de peso unitário.

Para definir os caminhos de trabalho e de proteção, geralmente é utilizado o algoritmo de roteamento de *Suurballe* e *Tarjan* (*ST*) [ST84], que tem a finalidade de encontrar, para cada par de nodos origem e destino de uma rede, um par de caminhos disjuntos por arestas tais que a soma de seus comprimentos seja mínima. Estes pares de caminhos definem ciclos de comprimento mínimo interligando cada par de nodos. Neste trabalho, consideramos o menor destes caminhos como o caminho de trabalho e o outro como o caminho de proteção.

Para certas topologias, pode haver mais de um ciclo de comprimento mínimo  $C$  interligando um determinado par de nodos, e estes ciclos podem ser desmembrados em caminhos de trabalho e de proteção de comprimentos diferentes. Por exemplo, dois ciclos de comprimento mínimo  $C = 16$  podem corresponder a: *i*) um par *mais balanceado*, com um caminho de trabalho com 8 saltos e um caminho de proteção com 8 saltos ( $C = 8 + 8$ ); ou *ii*) um par *menos balanceado*, com um caminho de trabalho com 3 saltos e um caminho de proteção com 13 saltos ( $C = 3 + 13$ ). Esta situação acontece em uma rede real, como será ilustrado posteriormente, na Seção 4 (Figura 3).

A existência de mais de um ciclo de comprimento mínimo associado a um mesmo par de nodos origem e destino nos permite explorar diferentes estratégias para a escolha dos caminhos de trabalho e de proteção, que podem ser utilizadas na definição de políticas de roteamento apropriadas para redes com diferentes tipos de serviço. Por exemplo, em caso de serviço sensível à latência, o ideal é que os caminhos de trabalho e de proteção tenham o mesmo comprimento ou, pelo menos, que seus comprimentos sejam o mais próximo possível. Neste caso, deve-se adotar a estratégia de buscar o que é denominado neste trabalho como o par de caminhos *mais balanceado*. Essa exigência costuma ocorrer quando não há, na prática, distinção entre caminhos de trabalho e de proteção: ambos os sinais são enviados simultaneamente para determinado destino e apenas o de melhor qualidade é utilizado. Em outros cenários, o ideal é que o caminho de trabalho seja o mais curto possível, pois o sinal é lançado somente nele, e o caminho de proteção é ativado apenas em caso de necessidade. Neste caso, a estratégia mais apropriada seria buscar o que é chamado neste trabalho de par de caminhos *menos balanceado*, de modo que o menor caminho do par seja utilizado como caminho de trabalho e o maior fique como proteção. Ambas as estratégias de proteção são observadas em casos práticos como, por exemplo, em redes ópticas de transporte [MBN99].

Em [GP16] foi observado que o algoritmo *ST* não é determinístico, i.e., havendo mais de um ciclo de comprimento mínimo associado a um mesmo par de nodos

origem e destino, ao ser executado mais de uma vez dada uma mesma entrada, o algoritmo *ST* pode retornar pares diferentes de caminhos de trabalho e de proteção. No exemplo anterior, o algoritmo *ST* ora encontrava o par de caminhos de comprimentos 8 e 8, ora o par de caminhos de comprimentos 3 e 13.

A partir desta observação, e a fim de explorar diferentes estratégias para a escolha dos caminhos de trabalho e proteção, este trabalho propõe duas versões determinísticas do algoritmo *ST* para encontrar, para uma determinada topologia, pares de caminhos de trabalho e de proteção mais balanceados e menos balanceados para cada par de nodos. No melhor do nosso conhecimento, este é o primeiro trabalho a propor versões determinísticas para o algoritmo de *Suurballe* e *Tarjan*. As versões propostas são denominadas: algoritmo de *Suurballe* e *Tarjan* mais balanceado (*Suurballe and Tarjan More Balanced, STMB*) e algoritmo de *Suurballe* e *Tarjan* menos balanceado (*Suurballe and Tarjan Less Balanced, STL*).

## 2 Algoritmo de Suurballe e Tarjan

Seja  $G = (V, E)$  um grafo *2-aresta-conexo* de ordem  $N$ , onde  $V = \{v_1, v_2, \dots, v_N\}$  é o conjunto de vértices e  $E$  é o conjunto de arestas, de pesos unitários. Para cada par origem e destino, o algoritmo *ST* encontra um ciclo de comprimento mínimo,  $C$ , contendo dois caminhos disjuntos por arestas, a serem usados como caminhos de trabalho e de proteção. Isto é possível sempre que o grafo for *2-aresta-conexo*, conforme o Teorema de Menger [Net06]. O algoritmo *ST* segue os seguintes passos:

1. Executa o algoritmo de *Dijkstra*, que encontra o menor caminho  $P_1$  interligando  $v_i$  a  $v_j$ ;
2. Os pesos das arestas de  $P_1$  são alterados. Esta parte é conhecida como “transformação de custos” e ocorre da seguinte forma:

$$c_{ij} = \begin{cases} N & , se (v_i, v_j) \in P_1 \\ c_{ij} + c_i - c_j & , se (v_i, v_j) \notin P_1 \\ & e (v_j, v_i) \notin P_1 \\ 0 & , se (v_j, v_i) \in P_1 \end{cases}$$

onde  $c_i$  é o custo do nodo  $v_i$ , e  $c_{ij}$  é o custo da aresta interligando  $v_i$  a  $v_j$ ;

3. Executa o algoritmo de *Dijkstra* mais uma vez, com os pesos das arestas de  $P_1$  modificados e encontra o segundo menor caminho  $P_2$ ;
4. Por fim, identifica se  $P_1$  e  $P_2$  compartilham alguma aresta. Se sim,  $P_1$  e  $P_2$  são redefinidos como  $P'_1$  e  $P'_2$ , excluindo-se a aresta em comum de ambos os caminhos. A redefinição dos caminhos é apresentada com detalhes no exemplo a seguir.

O par de caminhos resultante do algoritmo *ST*, seja  $P_1$  e  $P_2$  ou  $P'_1$  e  $P'_2$ , é usado para definir os caminhos de trabalho e de proteção. Por exemplo, para o grafo ilustrado na Figura 1, considerando que o nodo origem é o nodo  $v_1$ , o nodo destino é o nodo  $v_8$  e o peso de cada aresta seja 1, o menor caminho  $P_1$  obtido com o algoritmo de *Dijkstra* será:  $v_1 \leftrightarrow v_5 \leftrightarrow v_4 \leftrightarrow v_8$ , com 3 saltos. Em seguida, os pesos de todas as arestas de  $P_1$  são alterados. Assim, é encontrado um segundo menor caminho  $P_2$  com o algoritmo de *Dijkstra*,  $v_1 \leftrightarrow v_2 \leftrightarrow v_3 \leftrightarrow v_4 \leftrightarrow v_5 \leftrightarrow v_6 \leftrightarrow v_7 \leftrightarrow v_8$ .

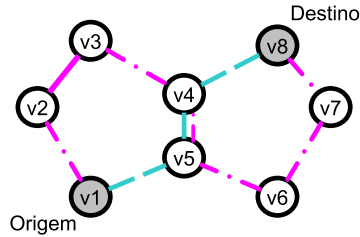


Figura 1: Menor caminho (em linha tracejada azul) e segundo menor caminho (em linha com ponto e traço rosa) do nodo origem  $v_1$  para o nodo destino  $v_8$ , obtidos com o algoritmo de *Dijkstra*.

É identificado se  $P_1$  e  $P_2$  compartilham arestas. Neste exemplo, existe aresta compartilhada,  $v_4 \leftrightarrow v_5$ , portanto  $P_1$  e  $P_2$  são redefinidos como  $P'_1$  e  $P'_2$ , respectivamente. O caminho  $P'_1$  será formado pelas arestas iniciais de  $P_1$ , até a última aresta antes da aresta compartilhada; seguidas pelas arestas finais de  $P_2$ , até a última aresta após a aresta compartilhada. A aresta compartilhada não é adicionada em nenhum dos caminhos  $P'_1$  e  $P'_2$ . O caminho  $P'_2$  segue o mesmo esquema, porém possui as arestas iniciais de  $P_2$  e finais de  $P_1$ . Respectivamente, temos  $P'_1$  dado por  $v_1 \leftrightarrow v_2 \leftrightarrow v_3 \leftrightarrow v_4 \leftrightarrow v_8$ , com 4 saltos, e  $P'_2$  dado por  $v_1 \leftrightarrow v_5 \leftrightarrow v_6 \leftrightarrow v_7 \leftrightarrow v_8$ , também com 4 saltos. Ambos os caminhos são ilustrados na Figura 2.

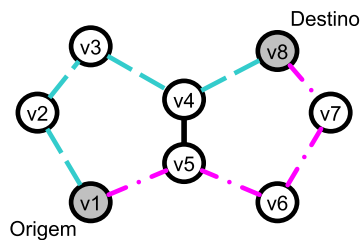


Figura 2: Caminhos de trabalho e de proteção disjuntos obtidos com o algoritmo *ST* do nodo origem  $v_1$  para o nodo destino  $v_8$ .

### 3 Algoritmos Propostos

Como observado na Seção 1, o algoritmo de *Suurballe* e *Tarjan* não é determinístico. Esta seção descreve

duas versões determinísticas deste algoritmo, denominadas: algoritmo de *Suurballe* e *Tarjan* mais balanceado (*Suurballe and Tarjan More Balanced, STMB*) e algoritmo de *Suurballe* e *Tarjan* menos balanceado (*Suurballe and Tarjan Less Balanced, STLB*).

O comprimento do caminho de trabalho no *STMB* será chamado de  $W^+$  (*Work Path More Balanced*) e o comprimento do caminho de proteção será chamado de  $B^+$  (*Backup Path More Balanced*). De modo similar, no *STLB*,  $W^-$  e  $B^-$  representam os comprimentos dos caminhos de trabalho e de proteção.

#### 3.1 Suurballe e Tarjan Mais Balanceado

Para cada par origem e destino, o algoritmo *STMB* encontra um ciclo de comprimento mínimo com pares de caminhos de trabalho e de proteção de comprimentos iguais ou com a mínima diferença possível em número de saltos entre eles. A sequência de passos do algoritmo é apresentada a seguir:

1. Executa o algoritmo *ST* padrão. A saída será, para cada par de nodos origem e destino da topologia, o tamanho do ciclo de comprimento mínimo,  $C$ , interligando o nodo origem ao nodo destino;
2. Para cada par origem e destino, com base no tamanho do ciclo de comprimento mínimo, as variáveis auxiliares  $R$  e  $S$  são definidas. O  $R$  definirá o novo comprimento do  $W^+$  e seu valor será o arredondamento para baixo (piso) do resultado da divisão de  $C$  por 2. O  $S$  definirá o novo comprimento do  $B^+$  e seu valor será o resultado da subtração de  $C$  por  $R$ ;
3. Usando o algoritmo de Yen [Yen70], cria uma lista para armazenar todos os caminhos de comprimento  $R$ , para um dado par origem e destino;
4. Encontra um caminho de comprimento  $S$ , também utilizando o algoritmo de Yen;
5. Compara caminhos de comprimentos  $R$  e  $S$ . Para isso, é tomado um caminho da lista de comprimento  $R$  e o caminho de comprimento  $S$ . É verificado se há alguma aresta compartilhada entre eles. Se não há qualquer aresta compartilhada, os valores de  $R$  e  $S$  são atribuídos a  $W^+$  e  $B^+$ , respectivamente. Se há aresta compartilhada nesses caminhos, testa-se o próximo caminho de comprimento  $R$  da lista. Se nenhum caminho de comprimento  $R$  for disjunto ao de comprimento  $S$ , é verificado se existe outro caminho de comprimento  $S$ . Em caso positivo, retorna ao Item 4. Caso contrário,  $R$  é decrementado e retorna ao Item 3. Este procedimento será realizado até serem encontrados dois caminhos disjuntos por arestas.

Sempre que o algoritmo *ST* for bem sucedido, ou seja, se existirem na topologia dois caminhos disjuntos por arestas para cada par origem-destino, os passos do algoritmo *STMB* também serão.

### 3.2 Suurballe e Tarjan Menos Balanceado

Para cada par origem e destino, o algoritmo *STLB* encontra um ciclo de comprimento mínimo,  $C$ , com pares de caminhos de trabalho e de proteção em que o caminho de trabalho tem o menor comprimento possível em número de saltos.

O algoritmo *STLB* atua de maneira semelhante ao algoritmo *STMB* com duas pequenas diferenças: *i*) para cada par origem e destino, o  $R$  inicia com o comprimento do menor caminho encontrado pelo algoritmo de *Dijkstra*, para tentar obter o menor caminho de trabalho possível; e *ii*) quando não forem encontrados dois caminhos disjuntos por arestas de comprimentos  $R$  e  $S$ , o  $R$  é incrementado. Assim como no *STMB*, estes passos serão bem sucedidos sempre que o algoritmo *ST* padrão obtiver sucesso.

## 4 Resultados

Os resultados dos algoritmos propostos são apresentados para a rede “Arnes”, ilustrada na Figura 3, que tem 17 nodos e 20 arestas.

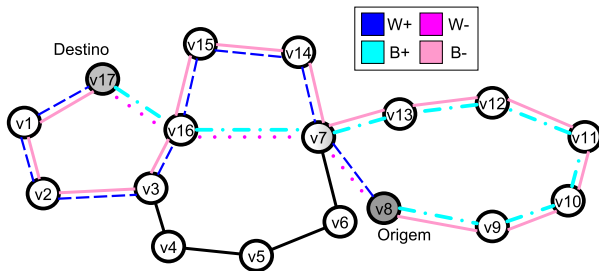


Figura 3: Caminhos de trabalho e de proteção definidos pelos algoritmos *STMB* e *STLB* na rede “Arnes” para o par de nodos origem  $v_8$  e destino  $v_{17}$ .

Dado o par de origem  $v_8$  e destino  $v_{17}$ , para o qual tem-se  $C = 16$  arestas, a Figura 3 apresenta os caminhos de trabalho e de proteção obtidos pelo *STMB* e pelo *STLB*. Tanto o caminho de trabalho como o de proteção obtidos pelo *STMB* tem 8 saltos, ou seja, temos  $W^+ = B^+ = 8$ . No *STLB*, o caminho de trabalho obtido tem 3 saltos, i.e.,  $W^- = 3$ , enquanto o de proteção tem 13 saltos, i.e.,  $B^- = 13$ . Este é um dos pares de nodos origem-destino para os quais ocorre a maior diferença entre o *STMB* e o *STLB*. O  $W^+$  tem 5 saltos a mais que o  $W^-$  e consequentemente o  $B^-$  tem 5 saltos a mais que o  $B^+$ .

As versões *STLB* e *STMB* serão implementadas na linguagem  $C++$ , utilizando o *framework ElasticO++* [TPCG16], e ambos os algoritmos serão tes-

tados num conjunto de 40 redes de telecomunicações reais disponíveis em [RMdRP13].

## 5 Conclusão

Neste trabalho foram propostos dois novos algoritmos que acrescentam previsibilidade ao algoritmo de *Suurballe* e *Tarjan* para obter, para cada par de nodos origem-destino de uma rede, caminhos de trabalho e de proteção disjuntos por arestas tais que a soma de seus comprimentos seja mínima. Estas estratégias consistem em encontrar os pares de caminhos mais e menos balanceados, o que permite explorar diferentes esquemas de proteção para aplicações em diferentes cenários. Como trabalhos futuros, os algoritmos propostos serão testados em um conjunto de 40 topologias de redes reais de telecomunicações.

## Referências

- [GP16] M. Girolimetto and C. Pavan. The average number of hops for working and backup paths in telecommunication networks. *SSN 2016 Spring School on Networks*, 1727, 2016. Session 2: Posters.
- [MBN99] J. Manchester, P. Bonenfant, and C. Newton. The evolution of transport network survivability. *IEEE Communications Magazine*, 37(8):44–51, 1999.
- [Net06] P. O. Boaventura Netto. *Grafos: Teoria, Modelos, Algoritmos*. Edgard Blucher, São Paulo, 4th edition, 2006.
- [RMdRP13] S. K. Routray, R. M. Morais, J. R. F. da Rocha, and A. N. Pinto. Statistical model for link lengths in optical transport networks. *IEEE/OSA J. Opt. Commun. Netw.*, 5(7):762–773, 2013.
- [ST84] J. W. Suurballe and R. E. Tarjan. A quick method for finding shortest pairs of disjoint paths. *Networks*, 14:325–336, 1984.
- [TPCG16] R. S. Tessinari, B. Puype, D. Colle, and A. S. Garcia. *ElasticO++*: An elastic optical network simulation framework for *OMNeT++*. *Optical Switching and Networking*, 22:95–104, 2016.
- [Yen70] Jin Y. Yen. An algorithm for finding shortest routes from all source nodes to a given destination in general networks. *Quarterly of Applied Mathematics*, 27:526–530, 1970.