Escola de Engenharia de São Carlos – Universidade de São Paulo

Proteção compartilhada e restauração de tráfego em redes ópticas de segmentação espectral flexível

Natália Santa Bárbara Capelari

Orientador: Prof. Dr. Amílcar Careli César
2016

Proteção compartilhada e restauração de tráfego em redes ópticas de segmentação espectral flexível

Natália Santa Bárbara Capelari

Orientador: Prof. Dr. Amílcar Careli César

Dissertação apresentada à Escola de Engenharia de São Carlos
da Universidade de São Paulo para obtenção do título de
Mestre em Ciências, Programa de Engenharia Elétrica.
Área de concentração: Telecomunicações.

USP – São Carlos
2016

Trata-se da versão corrigida da dissertação. A versão original se encontra disponível na EESC/USP que aloja o Programa de Pós-Graduação de Engenharia Elétrica.

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Capelari, Natália Santa Bárbara
C238p Proteção compartilhada e restauração de tráfego em redes ópticas de segmentação espectral flexível / Natália Santa Bárbara Capelari; orientador Amilcar Careli César. São Carlos, 2016.

Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia Elétrica e Área de Concentração em Telecomunicações -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2016.

1. redes ópticas de segmentação espectral flexível. 2. redes ópticas elásticas. 3. SLICE. 4. MATLAB. 5. proteção compartilhada. 6. algoritmo de roteamento e alocação de espectro. 7. smart-fit. 8. busca lógica. I. Título.

FOLHA DE JULGAMENTO

Candidata: Engenheira **NATÁLIA SANTA BÁRBARA CAPELARI.**

Título da dissertação: "Proteção compartilhada e restauração de tráfego em redes ópticas de segmentação espectral flexível".

Data da defesa: 05/05/2016

Comissão Julgadora:	**Resultado:**
Prof. Titular **Amílcar Careli César (Orientador)** (Escola de Engenharia de São Carlos/EESC)	APROVADA
Profa. Dra. **Mônica de Lacerda Rocha** (Escola de Engenharia de São Carlos/EESC)	Aprovada
Prof. Dr. **Luiz Henrique Bonani do Nascimento** (Universidade Federal do ABC/UFABC)	APROVADA

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica:
Prof. Associado **Luis Fernando Costa Alberto**

Presidente da Comissão de Pós-Graduação:
Prof. Associado **Paulo César Lima Segantine**

AGRADECIMENTOS

Agradeço aos meus pais, João e Mara, por terem me apoiado em todas as minhas decisões durante esses anos e ao Marcelo, por todo amor e carinho que recebi. Sem vocês, meus sonhos não teriam se tornado realidade.

Ao meu orientador e professor Dr. Amílcar Careli César, pelo exemplo de profissional e por ter me dado a oportunidade de ampliar meus conhecimentos.

Ao CNPq, pelo apoio ao projeto CNPq Universal nº 482191/2013-9, que permitiu a obtenção de novos computadores para o laboratório.

À Escola de Engenharia de São Carlos, por toda a infraestrutura oferecida.

Aos meus amigos, colegas e demais docentes com quem convivi em minha jornada.

RESUMO

CAPELARI, N. S. B., "Proteção compartilhada e restauração de tráfego em redes ópticas de segmentação espectral flexível", Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2016.

A busca de sistemas de comunicação eficientes e econômicos é cada vez maior, principalmente com o aumento da demanda por tráfego e banda de transmissão. Conhecidas pela utilização eficiente de largura de banda, as redes ópticas de segmentação espectral flexível têm sido intensamente estudadas e são consideradas boas candidatas para a próxima geração de redes. Tais redes flexíveis utilizam formatos de modulação multinível e multiplexação multiportadora.

Sistemas com alta capacidade de transmissão devem ser extremamente confiáveis, pois a falha em um dispositivo ou enlace da rede acarreta enorme perda de dados. Por isso, as técnicas de proteção e restauração do tráfego devem ser eficientes, de modo a garantir a sobrevivência da rede.

Neste trabalho, foi desenvolvido um algoritmo capaz de lidar com cada requisição de conexão, encontrar um caminho óptico para transmissão e reservar outro caminho para recuperação do tráfego em caso de falha na rede. A seleção dos caminhos primário e de proteção utiliza a técnica *smart-fit*, que escolhe a solução com menor custo final, definido pela distância da rota percorrida somada ao custo do índice inicial da faixa espectral alocada, buscando equilibrar a escolha entre a posição no espectro e rota escolhida. Além disso, são definidos custos diferentes para *slots* de frequência livres e compartilhados, a fim de incentivar o compartilhamento espectral em caminhos de proteção.

Dentre as técnicas de alocação espectral, a varredura com janela espectral varre o espectro em todos os enlaces da rota, *slot* por *slot*, em busca de uma faixa livre com tamanho suficiente para atender uma demanda. Neste trabalho, foi desenvolvida uma técnica chamada busca lógica, que lida simultaneamente com todos os enlaces da rota ao realizar operações lógicas com seus espectros. Em seguida, é realizada uma convolução entre a janela espectral, com tamanho da demanda, e o espectro resultante. A partir desse resultado, é possível saber as posições no espectro onde a demanda poderá ser alocada. Tal técnica, como será demonstrado, é mais veloz que a varredura com janela espectral.

Para garantir a eficácia e confiabilidade do algoritmo, utilizando o *software* MATLAB, avaliou-se a probabilidade de bloqueio e probabilidade de bloqueio de banda, a fragmentação espectral média na rede, o grau de compartilhamento, a taxa de utilização espectral e a taxa de capacidade reservada. O algoritmo desenvolvido foi capaz de organizar o tráfego na rede de modo ordenado e pouco fragmentado, além de utilizar de maneira eficiente os recursos destinados à proteção.

Palavras chave: ***redes ópticas de segmentação espectral flexível, redes ópticas elásticas, SLICE, proteção compartilhada, algoritmo de roteamento e alocação de espectro, smart-fit, busca lógica, software MATLAB.***

ABSTRACT

CAPELARI, N. S. B., "Shared protection and traffic restoration in elastic optical networks", (Master's degree project) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2016.

The search for efficient and economical communication systems is increasing, especially with the increased demand for traffic and bandwidth transmission. The elastic optical networks, well known for their efficient using of bandwidth, have been thoroughly studied and are considered good candidates for the next generation networks. Such flexible networks use multilevel modulation formats and multicarrier multiplexing.

Systems with high transmission capacity should be extremely reliable, as a failure in a device or link brings huge data loss. Therefore, traffic protection and restoration techniques should be efficient, to ensure network survivability.

In this work, we propose an algorithm capable of managing each connection request, finding an optical path for transmission and a protection path to recover traffic in case of network failure. The selection of the primary and protection path uses the smart-fit technique, which chooses the solution with the lower final cost, defined by the route's distance plus the index's cost of the first slot of the allocated spectrum, in order to balance the choice between the position in the spectrum and the chosen route. In addition, different costs for free or shared frequency slots are defined, so as to encourage spectral sharing in protection paths.

Among the spectrum allocation techniques, scanning with spectrum window scans the spectrum in all links of the route, slot by slot, searching for a free band large enough to meet the demand. In this work, we developed a new technique, called logical search, that simultaneously deals with all the links in the route, by performing logical operations with them. Then, a convolution is performed between a spectrum window, with the same size of the demand, and the resulting spectrum. From this result, it is possible to know all the positions in the spectrum where the demand may fit. This technique, as will be shown, is faster than the scan with spectrum window.

To guarantee the efficiency and reliability of the algorithm, using the MATLAB software, we evaluated the blocking probability and bandwidth blocking probability, the average spectral fragmentation in the network, the shareability ratio, the spectral utilization ratio and the spare capacity ratio. The developed algorithm was able to organize the traffic in an orderly and less fragmented way, and use the protection resources efficiently.

Keywords: elastic optical networks, SLICE, shared protection, routing and spectrum allocation algorithm, smart-fit, logical search, MATLAB software.

SUMÁRIO

Resumo i
Abstract iii
Sumário v
Lista de figuras ix
Lista de tabelas xiii
Lista de abreviaturas xv
1 Introdução ..... - 1 -
2 A rede óptica de segmentação espectral flexível ..... - 7 -
2.1 Vantagens e desvantagens da rede SLICE ..... - 8 -
2.2 Multiplexação óptica OFDM ..... - 8 -
2.3 Arquitetura de rede e nós ..... - 10 -
3 Proteção e restauração de redes ..... - 13 -
3.1 Técnicas de proteção ..... - 14 -
3.1.1 Proteção de enlace ou caminho óptico ..... - 14 -
3.1.2 Proteção dedicada (DPP) ..... - 14 -
3.1.3 Proteção compartilhada (SPP) ..... - 15 -
3.2 Técnicas de restauração ..... - 15 -
3.2.1 Restauração reativa ou proativa ..... - 16 -
3.2.2 Restauração de enlace, caminho óptico ou subcaminho óptico ..... - 16 -
3.2.3 Restauração dedicada, compartilhada ou multiplexação de primário/restauração - 16 -
3.2.4 Restauração dependente do tipo de falha ..... - 17 -
4 Roteamento e alocação de espectro (RSA) ..... - 19 -
4.1 Algoritmos heurísticos estáticos e dinâmicos ..... - 20 -
4.2 Roteamento ..... - 20 -
4.2.1 *Shortest path* (SP) ..... - 20 -
4.2.2 *K-shortest path* (K-SP) ..... - 21 -
4.2.3 *Adaptive routing* (AR) ..... - 21 -
4.3 Alocação de espectro ..... - 21 -
4.3.1 *Random-fit* (RF) ..... - 21 -
4.3.2 *First-fit* (FF) e *last-fit* (LF) ..... - 21 -

4.3.3 *Most-used* (MU) e *least-used* (LU) - 22 -
4.3.4 *Lowest starting slot* (LSS) e *highest starting slot* (HSS) - 22 -
4.4 Algoritmo: *spectrum-scan routing* (SSR) - 22 -
4.5 Algoritmo: roteamento, nível de modulação e alocação de espectro (RMLSA) - 23 -
5 Trabalhos relacionados - 25 -
5.1 *Shared-path protection in* OFDM-*based optical networks with elastic bandwidth allocation* (SPP-OFDM) [21] - 27 -
5.2 *Shared backup path protection* (SBPP) [22, 23] - 27 -
5.3 *Shared-path protected, routing, modulation level and spectrum assignment* (SPP-RMLSA) [24] - 29 -
5.4 *Minimum free spectrum block* (MFSB) [25] - 30 -
5.5 Restauração por compressão de banda (BSR) [26, 27] - 30 -
5.6 *Iterative two step approach* (ITSA) [28] - 31 -
5.7 *Single-path provisioning multi-path recovery* (SPPMPR) [29] - 32 -
5.8 *Dynamic load balancing shared-path protection* (DLBSPP) [30, 31] - 33 -
5.9 Algoritmo genético com proteção dedicada de caminho (RSA/DPP/SC) [32] - 34 -
5.10 *Survivable multi-path routing and spectrum allocation* (SM-RSA) [33, 34] - 34 -
5.11 Conclusões do capítulo - 35 -
6 Definição do problema - 37 -
6.1 A topologia da rede - 37 -
6.2 Geração de tráfego - 37 -
6.3 Ocorrência de falhas - 38 -
6.4 Conexões tratadas (*output*) - 38 -
6.5 Restrições - 39 -
6.6 Avaliação do algoritmo - 39 -
7 Algoritmo heurístico de roteamento e alocação de espectro "*smart-fit*" com proteção compartilhada de caminho (RSA_SF_SPP) - 43 -
7.1 Chegada de uma nova conexão - 43 -
7.1.1 Caminho primário - 43 -
7.1.2 Caminho de proteção - 45 -
7.1.3 Método de busca por faixa espectral disponível: Busca Lógica - 47 -
7.2 Falha em um enlace - 52 -
7.3 Fim de uma conexão - 53 -

7.4 Complexidade computacional ..... - 54 -

8 Outros algoritmos utilizados para comparação ..... - 57 -

8.1 Algoritmo heurístico de roteamento e alocação de espectro *first-fit* com proteção compartilhada de caminho (RSA_FF_SPP) ..... - 57 -

8.2 Algoritmo heurístico de roteamento e alocação de espectro *first-fit* e *last-fit* modificado com proteção compartilhada de caminho (RSA_FFMLF_SPP) ..... - 57 -

9 Resultados e discussões ..... - 59 -

9.1 Comparação entre diferentes algoritmos ..... - 60 -

9.1.1 Probabilidade de bloqueio e probabilidade de bloqueio de banda ..... - 60 -

9.1.2 Grau de compartilhamento ..... - 65 -

9.1.3 Fragmentação espectral ..... - 67 -

9.1.4 Taxa de utilização espectral ..... - 68 -

9.1.5 Taxa de capacidade reservada ..... - 70 -

9.1.6 Porcentagem de restauração ..... - 72 -

9.2 Definição dos valores de custo dos *slots* de frequência ..... - 72 -

9.3 Técnicas de alocação espectral SW e busca-lógica com convolução ..... - 80 -

9.4 Simulações para redes com transceptores não configuráveis ..... - 80 -

9.4.1 Probabilidade de bloqueio e probabilidade de bloqueio de banda ..... - 80 -

9.4.2 Grau de compartilhamento ..... - 86 -

9.4.3 Fragmentação espectral ..... - 86 -

9.4.4 Taxa de utilização espectral ..... - 86 -

9.4.5 Taxa de capacidade reservada ..... - 86 -

9.4.6 Porcentagem de restauração ..... - 91 -

10 Conclusões ..... - 93 -

Referências ..... - 97 -

Apêndice – pseudocódigos dos algoritmos ..... - 99 -

LISTA DE FIGURAS

Figura 2-1: Alocação de banda em rede WDM com grade fixa (a) e rede SLICE com grade flexível (b) (adaptada de[5]). ..... - 7 -

Figura 2-2: Utilização do espectro em um sistema WDM (a) e sistema OFDM (b) (adaptada de [12]). ..... - 9 -

Figura 2-3: Sinal OFDM com 4 subportadoras no domínio da frequência (a) e no domínio do tempo (b) (adaptada de[5]). ..... - 10 -

Figura 2-4: Diagrama de blocos de uma rede SLICE (adaptada de[4]). ..... - 11 -

Figura 2-5: Conceito de WSS de largura de banda variável, utilizado no chaveamento do tráfego em rede com grade flexível (adaptada de[6]). ..... - 12 -

Figura 2-6: Suporte para demandas do tipo subcomprimento de onda usando transceptores fixos (a) e transceptores divisíveis (b) (adaptada de[12]). ..... - 12 -

Figura 3-1: Exemplos de caminhos disjuntos e nãos disjuntos. ..... - 13 -

Figura 3-2: Proteção de caminho (a) ou de enlace (b) (adaptada de[14]). ..... - 14 -

Figura 3-3: Mecanismos de restauração de caminho (a), enlace (b) e subcaminho (c) (adaptada de[14]). ..... - 16 -

Figura 4-1: Esquemas de especificação dos recursos espectrais. Grade com frequência padronizada ITU-T (a) e com slots de frequência de 12,5 GHz (b) (adaptada de[6]). ..... - 19 -

Figura 4-2: Diferença entre varredura do espectro dos algoritmos SSR e DSSR (adaptada de[19]). ..... - 23 -

Figura 7-1: Fluxograma dos passos realizados durante a busca pelo caminho primário. ..... - 44 -

Figura 7-2: Fluxograma dos passos realizados durante a busca pelo caminho de proteção. ..... - 45 -

Figura 7-3: Exemplo de um enlace da rede representado pela matriz de auxílio Cmat. ..... - 49 -

Figura 7-4: Enlace representado pela matriz Cmat após ocupação de um novo caminho primário. - 49 -

Figura 7-5: Enlace representado pela matriz Cmat após a ocupação de um novo caminho de proteção. ..... - 49 -

Figura 7-6: Preparação dos enlaces de uma rota para a operação lógica. ..... - 51 -

Figura 7-7: Operação lógica para caminho primário (Equação 7.5) ou caminho de proteção (Equação 7.6). ..... - 51 -

Figura 7-8: Convolução e seleção dos índices iniciais dos slots de frequência onde a conexão pode ser alocada para caminho primário ou caminho de proteção. ..... - 52 -

Figura 7-9: Fluxograma dos passos realizados para recuperar as conexões afetadas por uma falha. ..... - 53 -

Figura 9-1: Rede NSFNET, 14 nós e 21 enlaces bidimensionais (adaptada de[22]). ..... - 59 -

Figura 9-2: Rede COST239, 11 nós e 26 enlaces bidimensionais (adaptada de[22]).....- 59 -
Figura 9-3: Probabilidade de bloqueio para rede NSFNET.....- 61 -
Figura 9-4: Probabilidade de bloqueio de banda para rede NSFNET. .....- 61 -
Figura 9-5: Motivo de bloqueio da conexão para rede NSFNET. .....- 62 -
Figura 9-6: Probabilidade de bloqueio para rede COST239. .....- 63 -
Figura 9-7: Probabilidade de bloqueio de banda para rede COST239. .....- 63 -
Figura 9-8: Motivo de bloqueio da conexão para rede COST239. .....- 64 -
Figura 9-9: Grau de compartilhamento para rede NSFNET. .....- 66 -
Figura 9-10: Grau de compartilhamento para rede COST239. .....- 66 -
Figura 9-11: Fragmentação espectral para rede NSFNET. .....- 67 -
Figura 9-12: Fragmentação espectral para rede COST239. .....- 68 -
Figura 9-13: Taxa de utilização espectral para rede NSFNET. .....- 69 -
Figura 9-14: Taxa de utilização espectral para rede COST239. .....- 69 -
Figura 9-15: Taxa de capacidade reservada para rede NSFNET. .....- 71 -
Figura 9-16: Taxa de capacidade reservada para rede COST239. .....- 71 -
Figura 9-17: Probabilidade de bloqueio para diferentes valores de constantes para rede NSFNET. - 73 -
Figura 9-18: Probabilidade de bloqueio de banda para diferentes valores de constantes para rede NSFNET. .....- 73 -
Figura 9-19: Fragmentação espectral para diferentes valores de constantes para rede NSFNET. ... - 74 -
Figura 9-20: Taxa de capacidade reservada para diferentes valores de constantes para rede NSFNET. .....- 74 -
Figura 9-21: Grau de compartilhamento para diferentes valores de constantes para rede NSFNET. .....- 75 -
Figura 9-22: Taxa de utilização espectral para diferentes valores de constantes para rede NSFNET. .....- 75 -
Figura 9-23: Probabilidade de bloqueio para diferentes valores de constantes para rede COST239. .....- 77 -
Figura 9-24: Probabilidade de bloqueio de banda para diferentes valores de constantes para rede COST239. .....- 77 -
Figura 9-25: Fragmentação espectral para diferentes valores de constantes para rede COST239. .. - 78 -
Figura 9-26: Taxa de capacidade reservada para diferentes valores de constantes para rede COST239. .....- 78 -

Figura 9-27: Grau de compartilhamento para diferentes valores de constantes para rede COST239. .....- 79 -
Figura 9-28: Taxa de utilização espectral para diferentes valores de constantes para rede COST239. .....- 79 -
Figura 9-29: Probabilidade de bloqueio para rede NSFNET com transceptores não configuráveis. - 82 -
Figura 9-30: Probabilidade de bloqueio de banda para rede NSFNET com transceptores não configuráveis. .....- 82 -
Figura 9-31: Motivo de bloqueio de conexões para rede NSFNET com transceptores não configuráveis. .....- 83 -
Figura 9-32: Probabilidade de bloqueio para rede COST239 com transceptores não configuráveis. .....- 84 -
Figura 9-33: Probabilidade de bloqueio de banda para rede COST239 com transceptores não configuráveis. .....- 84 -
Figura 9-34: Motivo de bloqueio de conexões para rede COST239 com transceptores não configuráveis. .....- 85 -
Figura 9-35: Grau de compartilhamento para rede NSFNET com transceptores não configuráveis. .....- 87 -
Figura 9-36: Grau de compartilhamento para rede COST239 com transceptores não configuráveis .....- 87 -
Figura 9-37: Fragmentação espectral para rede NSFNET com transceptores não configuráveis..... - 88 -
Figura 9-38: Fragmentação espectral para rede COST239 com transceptores não configuráveis.... - 88 -
Figura 9-39: Taxa de utilização espectral para rede NSFNET com transceptores não configuráveis. .....- 89 -
Figura 9-40: Taxa de utilização espectral para rede COST239 com transceptores não configuráveis. .....- 89 -
Figura 9-41: Taxa de capacidade reservada para rede NSFNET com transceptores não configuráveis. .....- 90 -
Figura 9-42: Taxa de capacidade reservada para rede COST239 com transceptores não configuráveis. .....- 90 -

LISTA DE TABELAS

Tabela 5-1: Comparação entre artigos de sobrevivência de redes estudados. .....- 25 -
Tabela 5-2: Comparação entre características dos algoritmos estudados. .....- 26 -
Tabela 9-1: Valores das constantes utilizadas nas simulações. .....- 60 -
Tabela 9-2: Comparação do tempo de simulação dos métodos de alocação espectral SW e busca lógica. .....- 80 -

LISTA DE ABREVIATURAS

AR	*Adaptive routing*
BER	*Best-effort recovery*
BRKGA	*Biased random-key genetic algorithm*
BSR	*Bandwidth squeezed restoration*
BVT	*Bandwidth variable transceivers*
BV-WSS	*Bandwidth variable wavelength-selective switch*
COST239	*Ultra-high capacity optical transmission network (Pan-European Network)*
CSSB	*Common shared spectrum block algorithm*
DA	*Distance adaptive*
DFT	*Discrete Fourier transform*
DLB	*Dynamic load balancing*
DLBSPP	*Dynamic load balancing shared-path protection algorithm*
DPP	*Dedicated-path protection*
DSSR	*Discrete spectrum-scan routing*
EUE	*Eficiência de utilização espectral*
FBGR	*Full bandwidth guaranteed recovery*
FF	*First-fit*
FFT	*Fast Fourier transform*
FS	*Frequency slots*
G	*Guard band*
GA	*Genetic algorithm*
HSS	*Highest starting slot*
IDFT	*Inverse discrete Fourier transform*
IFFT	*Inverse fast Fourier transform*
ILP	*Integer linear programing*
ISI	*Inter-symbol interference*
ITSA	*Iterative two step approach*
ITU-T	*International Telecommunications Union – Telecommunication Standardization Sector*
JFP	*Joint failure probability*
K-SP	*K-shortest path*

LF	*Last-fit*
LPF	*Longest path first*
LSS	*Lowest starting slot*
LU	*Least-used*
MATLAB	*Software Matrix Laboratory*
MFSB	*Minimum free spectrum block algorithm*
MILP	*Mixed integer linear programming*
MLF	*Modified last-fit*
MPP	*Multi-path provisioning*
M-PSK	*Multilevel-phase shift keying*
M-QAM	*Multilevel-quadrature amplitude modulation*
MSF	*Most subcarriers first*
MSSB	*Maximum shared spectrum block algorithm*
MU	*Most-used*
NSFNET	*National Science Foundation Network*
OFDM	*Orthogonal frequency division multiplexing*
PAPR	*Peak-to-average power ratio*
PB	*Probabilidade de bloqueio*
PBB	*Probabilidade de bloqueio de banda*
PBGR	*Partial bandwidth guaranteed recovery*
PUF	*Pior utilização de fibra*
RF	*Random-fit*
RMLSA	*Routing, modulation level and spectrum allocation algorithm*
RSA	*Routing and spectrum allocation*
RSA/DPP/SC	*Algoritmo genético com proteção dedicada de caminho (routing and spectrum allocation with dedicated path protection and same channel allocation)*
RSA_FF_SPP	*Algoritmo heurístico de roteamento e alocação de espectro first-fit com proteção compartilhada de caminho (routing and spectrum allocation, with first-fit, and shared path protection algorithm)*
RSA_FFMLF_SPP	*Algoritmo heurístico de roteamento e alocação de espectro first-fit e last-fit modificado com proteção compartilhada de caminho (routing and spectrum allocation, with first-fit modified last-fit, and shared path protection algorithm)*

RSA_SF_SPP	*Algoritmo heurístico de roteamento e alocação de espectro "smart-fit" com proteção compartilhada de caminho (routing and spectrum allocation, with smart-fit, and shared path protection algorithm)*
RWA	*Routing and wavelength assignment*
SBPP	*Shared backup path protection algorithm*
SCC	*Spectrum-continuity constraint*
SCR	*Spare capacity ratio*
SF	*Smart-fit*
SLICE	*Spectrum-sliced elastic optical path*
SM-RSA	*Survivable multipath routing and spectrum allocation algorithm*
SNR	*Signal-to-noise ratio*
SP	*Shortest path*
SPP	*Shared-path protection*
SPPMPR	*Single-path provisioning multi-path recovery*
SPP-OFDM	*Shared-path protection in OFDM-based optical networks algorithm*
SPP-RMLSA	*Shared-path protection, routing, modulation level and spectrum assignment algorithm*
SSR	*Spectrum-scan routing*
SS-RSA	*Survivable singlepath routing and spectrum allocation algorithm*
SUR	*Spectral utilization ratio*
SW	*Spectrum window*
TAR	*Traffic-aware restoration*
WDM	*Wavelength division multiplexing*
WSS	*Wavelength-selective switch*
WXC	*Wavelength cross-connect*

1 INTRODUÇÃO

Nos últimos anos, a demanda por tráfego em redes de internet e sistemas de comunicação tem crescido rapidamente. Estudos indicam que a demanda por recursos continuará a crescer exponencialmente, com estimativas de que, em 2019, o tráfego de internet seja mais de 60 vezes todo o tráfego mundial de 2005 [1]. Tais estudos também destacam o aumento da demanda de vídeos nas redes de comunicações, que em 2014 chegou a 64% de todo o tráfego global e estima-se que chegue a 80% em 2019 [1]. Em resposta às exigências do mercado consumidor, os sistemas de comunicações ópticas têm evoluído e buscado novas tecnologias a fim de se tornarem mais eficientes, flexíveis, robustos e com maior capacidade de transmissão. Estudos também apontam que as redes ópticas ganharão ainda mais importância nas futuras gerações de transmissão de dados, da ordem de Tbps [2, 3]. A fibra óptica é hoje o principal meio de transporte utilizado em sistemas de telecomunicação por possuir inúmeras vantagens, como grande capacidade de transmissão (alguns Tbps), boa relação custo-benefício e flexibilidade em escala [4]. O termo em inglês *scalability* é uma característica importante em todo sistema ou processo, pois indica sua capacidade de crescimento modular, de maneira uniforme, mediante a adição de dispositivos e recursos. Um sistema óptico é dito "modulável" se for capaz de manter sua eficiência quando utilizado em situações de maior escala.

Para se tornarem compatíveis com o aumento de tráfego, é necessário aumentar a capacidade de crescimento dos sistemas ópticos a partir do desenvolvimento de novos dispositivos e tecnologias de multiplexação e modulação de sinais. Técnicas de multiplexação têm sido estudadas com o objetivo de aumentar a eficiência dos sistemas de transmissão.

Um dos principais problemas das redes atuais é sua subdivisão em canais de largura fixa, que leva ao desperdício de banda de transmissão. Costuma-se usar o termo granularidade (do inglês, *granularity*) indicando a extensão à qual um sistema é dividido em partes menores.

As redes ópticas de multiplexação por comprimento de onda (WDM, *wavelength division multiplexing*) atuais, roteadas por comprimento de onda, requerem a alocação de banda com capacidade de um comprimento de onda (tipicamente da ordem de Gbps) para a transmissão de sinal em um caminho óptico. Esse recurso é alocado mesmo que o sinal a ser transmitido tenha banda menor (por exemplo, da ordem de Mbps).

Chamamos subcomprimento de onda (*sub-wavelength*) o sinal com demanda menor que a de um comprimento de onda. O tráfego resultante da agregação de outros sinais menores é chamado supercomprimento de onda (*super-wavelength*) [5]. A agregação de conexões combina diferentes portas ou conexões de um comutador (*switch*) ou roteador, com destino a um mesmo nó, respeitando a capacidade de transmissão do dispositivo e do enlace.

A técnica de multiplexação por divisão de frequência ortogonal (OFDM, *orthogonal frequency division multiplexing*) divide o espectro de transmissão em porções (*slots*) por meio de subportadoras

de acordo com sua frequência central. Com a técnica OFDM podem-se transmitir múltiplos canais ortogonais entre si (não havendo interferência entre símbolos). A sobreposição parcial dos espectros das subportadoras nos sistemas OFDM resulta em utilização mais eficiente da banda de transmissão [6].

Também com o intuito de aumentar a eficiência espectral e a banda de transmissão, formatos de modulação óptica multinível têm sido adotados em sistemas ópticos OFDM. Essas técnicas de modulação codificam vários bits de dados por símbolo, reduzindo a taxa de símbolos em relação à taxa de dados e diminuindo ainda mais a largura espectral do sinal. Os dois principais esquemas de modulação multinível são o M-PSK (*multilevel-phase shift keying*) e M-QAM (*multilevel-quadrature amplitude modulation*) [7-9]. Na modulação M-PSK, o dado é codificado com a modulação da fase de um sinal de referência (portadora) e na modulação M-QAM a codificação é feita modulando a amplitude e a fase do sinal de referência. A escolha correta do tipo de modulação de um sinal, de acordo com a distância de transmissão e as condições físicas do caminho óptico, pode melhorar ainda mais a eficiência na utilização de recursos e a qualidade do sinal recebido [10]. A modulação multinível permite alcançar maior eficiência espectral, porém o sinal passa a ter menor tolerância a ruído [6]. Existem vários estudos a respeito da chamada tecnologia de modulação adaptativa, em que a decisão a respeito do tipo de modulação a ser utilizado em um dado sinal é tomada de acordo com a distância de transmissão, as condições do canal de transmissão e relação sinal-ruído (SNR, *signal-to-noise ratio*) do canal [2, 10, 11]. A modulação adaptativa pode ser realizada via *software*, alterando as configurações dos equipamentos responsáveis pela geração do sinal modulado, sem a necessidade de alterar os equipamentos da rede [2].

A capacidade de alocar recursos com demanda menor ou maior que a de um comprimento de onda (sub e supercomprimento de onda, respectivamente) de maneira eficiente é uma das principais vantagens da rede óptica de segmentação espectral flexível, chamada SLICE (*spectrum-sliced elastic optical path*) [5, 10].

Outra importante característica de uma rede é a robustez a falhas. A falha de um dispositivo ou enlace acarreta a perda de muitos dados, principalmente em redes com altas taxas de transmissão. O sistema de sobrevivência de uma rede pode ser definido pelo planejamento prévio de recursos reservados, ou pode ser dinâmico, em que algoritmos de restauração são ativados apenas no momento em que ocorre a falha de um dos dispositivos da rede.

Esquemas de proteção de rede reservam previamente recursos (fibras, segmentos específicos de espectro ou mesmo alguns dispositivos) a serem utilizados na transmissão em caso de falhas. A restauração da rede, por outro lado, envolve localizar tais recursos sobressalentes para reacomodar o tráfego no momento em que uma falha ocorre. Em geral, as técnicas de restauração são mais eficientes por economizar gastos e manutenção de recursos, enquanto a proteção previamente planejada garante maior velocidade de recuperação da rede. É importante ter em mente que o sistema de restauração da rede deve ser muito bem planejado e seus algoritmos devem ser eficazes, pois, no caso de uma falha, as

técnicas de restauração poderão gastar recursos em excesso e se tornar menos eficientes em comparação com um sistema com proteção previamente planejada.

Durante a configuração de uma rede, o algoritmo de gerenciamento seleciona caminhos fixos por onde o tráfego escoa, chamados caminhos primários. Quando a falha em um dispositivo, nó ou enlace é detectada, o tráfego é redirecionado para outro caminho, dito caminho de proteção (*backup*), até o nó de destino. O caminho de proteção pode ou não estar previamente configurado, dependendo do algoritmo utilizado.

Em uma rede SLICE, o problema de roteamento e alocação de espectro (RSA, *routing and spectrum allocation*) é diferente do tradicional problema de roteamento e alocação de comprimento de onda (RWA, *routing and wavelength assignment*) das redes WDM. Algoritmos de RWA alocam comprimentos de onda e utilizam restrições e variáveis que dependem do número de comprimentos de onda. Algoritmos de RSA, por outro lado, utilizam faixas espectrais para transmissão e restrições que garantam a continuidade espectral.

O problema de RSA pode ser solucionado em uma ou duas etapas. Na aproximação em uma etapa, o roteamento do tráfego e a alocação de espectro para transmissão são feitos em uma única fase do algoritmo. Na aproximação em duas etapas, o problema é dividido em duas partes. O algoritmo soluciona separadamente cada uma das etapas, encontrando primeiramente os caminhos candidatos para roteamento e na segunda etapa é feita a alocação de espectro para o caminho escolhido anteriormente.

Algoritmos heurísticos são muito utilizados na solução do problema de RSA, podendo utilizar aproximações em uma ou duas etapas. Heurística é uma técnica desenvolvida com o objetivo de solucionar um problema de maneira rápida quando os métodos clássicos se tornam muito demorados, ou encontrar uma solução aproximada quando os métodos clássicos falham em encontrar uma solução exata. Problemas de otimização com complexidade polinomial e não determinísticos (como o problema de RSA em redes de grande escala) são resolvidos por meio de algoritmos heurísticos quando exigem soluções viáveis e rápidas, ainda que próximas da ótima. O algoritmo heurístico não tem uma razão de qualidade comprovada matematicamente ou prova formal de convergência, diferentemente dos algoritmos ILP (*integer linear programing*), que buscam uma solução otimizada matematicamente. Os algoritmos heurísticos devem ser exaustivamente testados e seus resultados comparados com outras soluções já conhecidas para verificar sua exatidão.

O intuito deste trabalho de mestrado é estudar algoritmos e técnicas de proteção e restauração em redes SLICE, com a utilização do *software* MATLAB. Foi desenvolvido um algoritmo para roteamento e alocação espectral para tráfego dinâmico e proteção de caminho compartilhada que utiliza uma técnica chamada *smart-fit*. Nesse algoritmo, cada conexão entre dois usuários em uma rede possui um caminho primário de transmissão de dados e um caminho de proteção disjunto em enlace do primário. A técnica de proteção utilizada garante que, na ocorrência de uma falha durante a transmissão, esta possa ser redirecionada para o caminho de proteção rapidamente. Para o caminho primário, utilizou-se roteamento de menor caminho (K-SP, *k-shortest path*) e o algoritmo pode ser configurado para escolher a faixa do

espectro a ser alocado de menor índice (LSS, *lowest starting slot*) ou de maior índice (HSS, *highest starting slot*). Para o caminho de proteção, entre os menores caminhos (K-SP), escolhe-se a faixa espectral que possua menor custo, calculado de acordo com o índice inicial e se os *slots* são livres ou compartilhados. O intuito desse método é incentivar o compartilhamento espectral entre caminhos de proteção, reduzindo a quantidade de recursos reservados da rede.

O método tradicional de alocação espectral, chamado varredura com janela espectral, utiliza uma janela espectral de tamanho igual ao da demanda, que varre todo o espectro, *slot* por *slot*, até encontrar uma faixa do espectro livre onde a demanda pode ser alocada. Tal varredura é realizada em todos os enlaces da rota. Neste trabalho, foi desenvolvida uma nova técnica de busca por *slots* disponíveis, chamada busca lógica. Nessa técnica, realizam-se operações lógicas entre todos os espectros da rota e, em seguida, é feita uma convolução entre a janela espectral e o espectro resultante. Com o resultado da convolução, sabe-se todas as posições em todos os enlaces envolvidos onde a demanda poderá ser alocada. Essa técnica lida com todos os enlaces da rota de uma única vez e é mais veloz que o método tradicional, como será demonstrado.

A avaliação do desempenho do algoritmo desenvolvido foi realizada ao compará-lo com outros dois algoritmos, um que utiliza K-SP e o método *first-fit* (FF) tanto para o caminho primário quanto para o de proteção, e outro algoritmo que utiliza K-SP e FF para o caminho primário e K-SP e uma técnica *last-fit* modificada (MLF) para o caminho de proteção. Diversos resultados foram avaliados, como a probabilidade de bloqueio e probabilidade de bloqueio de banda, a fragmentação espectral média da rede, o grau de compartilhamento, que relaciona a quantidade de *slots* de proteção requeridos pelas conexões e a quantidade de *slots* de proteção reservados na rede, a taxa de utilização espectral, que fornece o total de *slots* de frequência ocupados na rede e a taxa de capacidade reservada, que relaciona o número de *slots* utilizados por caminhos primários e o número de *slots* reservados para proteção. Além disso, em caso de falha em um dos enlaces da rede, foi verificada também a porcentagem de restauração do algoritmo, a fim de verificar se este foi capaz de proteger e recuperar corretamente todas as conexões afetadas.

Os capítulos seguintes estão organizados da seguinte maneira:

No Capítulo 2, apresentaremos a rede óptica de segmentação espectral flexível SLICE e suas características, como os dispositivos que a compõem e a técnica de multiplexação OFDM.

No Capítulo 3, introduziremos os conceitos de proteção e restauração de redes e descreveremos os métodos mais importantes.

No Capítulo 4, descreveremos o problema de RSA, os métodos mais utilizados de roteamento do tráfego e alocação de espectro, com foco nas técnicas mais importantes encontradas na literatura.

No Capítulo 5, serão analisados e comparados alguns dos principais algoritmos de proteção e restauração de redes ópticas elásticas encontrados na literatura. Esses estudos serviram como base para o desenvolvimento do algoritmo deste trabalho de mestrado.

No Capítulo 6, definiremos o problema a ser tratado neste trabalho, suas características e os métodos de avaliação do algoritmo desenvolvido.

No Capítulo 7, apresentaremos o algoritmo desenvolvido. Como são tratadas as conexões, as técnicas de RSA e de proteção utilizadas, assim como as ações do algoritmo ao detectar uma falha em um enlace. Nesse algoritmo, foi desenvolvido um novo método de tomada de decisão para RSA que foi chamado de *smart-fit*. Também será descrito uma nova técnica de alocação espectral desenvolvida, denominada busca lógica.

No Capítulo 8, descreveremos as características e técnicas dos algoritmos utilizados para avaliação do algoritmo desenvolvido.

No Capítulo 9, serão avaliados os desempenhos dos algoritmos estudados, por meio de simulações utilizando o software MATLAB.

Por fim, no Capítulo 10, apresentaremos as conclusões dos estudos realizados neste trabalho.

2 A REDE ÓPTICA DE SEGMENTAÇÃO ESPECTRAL FLEXÍVEL

A rede óptica de segmentação espectral flexível SLICE foi proposta por Jinno e colaboradores [5, 10] com o objetivo de melhorar a eficiência e capacidade de transporte de dados, permitindo alocar recursos espectrais com demanda menor que um comprimento de onda.

Para prover o transporte de dados de 100 Gb/s e além, é necessária a introdução do conceito de granularidade flexível no domínio da frequência óptica. Diferentemente da granularidade fixa de redes WDM, em que o tráfego com banda menor que um comprimento de onda ocupa todo o recurso da rede, na rede SLICE a alocação é feita por segmentação espectral, de modo que a banda tenha tamanho estritamente necessário para a transmissão daquela demanda. Nessa rede, o recurso necessário para a demanda é fatiado (*sliced off*) do total disponível e alocado para aquele caminho óptico.

A capacidade de expansão e contração de banda de acordo com a demanda por tráfego entre os usuários da rede é chamada elasticidade, ou flexibilidade, da banda. Tal flexibilidade permite a acomodação eficiente de diferentes taxas de transmissão. A comparação entre a alocação de banda em uma rede óptica convencional, com grade fixa, e em uma rede SLICE pode ser vista na Figura 2-1 (adaptada de [5]).

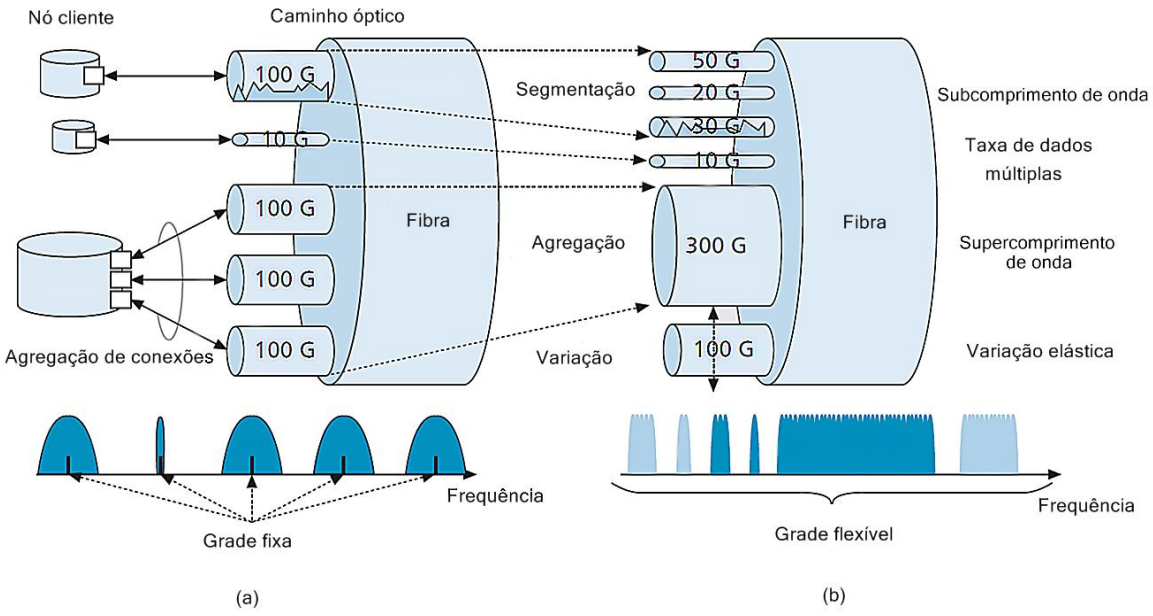

Figura 2-1: Alocação de banda em rede WDM com grade fixa (a) e rede SLICE com grade flexível (b) (adaptada de[5]).

2.1 VANTAGENS E DESVANTAGENS DA REDE SLICE

A rede SLICE possui alta modularidade, alta eficiência espectral e granularidade flexível.

A característica de elasticidade da banda permite e requer muitas inovações nas técnicas de transmissão conhecidas e nos equipamentos existentes. Em seu artigo, Jinno e seus colaboradores [5] dão três exemplos dos benefícios da elasticidade do caminho óptico na rede SLICE:

- *Compartilhamento de banda elástica entre clientes em função do tempo: a banda óptica alocada pode ser compartilhada entre clientes de maneira diferente conforme a utilização e necessidade de banda de cada usuário ao longo do dia. Desse modo, a mesma conexão pode ser utilizada por usuários diferentes em diferentes períodos, de acordo com a demanda de cada um;*
- *Redução do consumo de energia: o número de dispositivos, canais, ou mesmo de subportadoras OFDM pode ser diminuído ou desligado em períodos de baixa utilização da rede;*
- *Restauração da rede via compressão de banda (BSR, de* bandwidth squeezed restoration*): para permitir conectividade mínima em caso de falhas de comunicação, o caminho óptico elástico na rota falha pode ser diminuído (comprimido), aumentando a capacidade de sobrevivência da rede. Desse modo, a transmissão continua possível mesmo que a banda disponível no caminho de desvio não seja suficiente.*

Inovações em *hardware* e *software* são requeridas para o desenvolvimento e a utilização de redes ópticas elásticas no mercado [12]. Para que a rede SLICE possa ter seus benefícios completamente aproveitados é necessário também o desenvolvimento de novos algoritmos de roteamento e alocação de espectro e novos esquemas de controle, proteção e restauração da rede.

A característica de elasticidade da rede SLICE causa o problema de fragmentação do espectro. Conforme a rede é utilizada, o espectro de transmissão é fatiado e alterado continuamente, tornando-se fragmentado. Outro problema semelhante está associado à capacidade de expansão dos supercanais, que utiliza o espectro livre entre caminhos ópticos adjacentes, porém limita seu tamanho máximo em algum momento. Para solucionar os dois problemas relacionados à utilização do espectro, é necessário que este seja periodicamente desfragmentado e que caminhos ópticos sejam realocados de modo a permitir o crescimento dos caminhos ópticos existentes e liberar espectro para futuras conexões.

2.2 MULTIPLEXAÇÃO ÓPTICA OFDM

A tecnologia WDM permite que uma única fibra óptica possa transmitir muitos canais simultaneamente, aumentando drasticamente a taxa de transmissão agregada [4]. Essa tecnologia divide

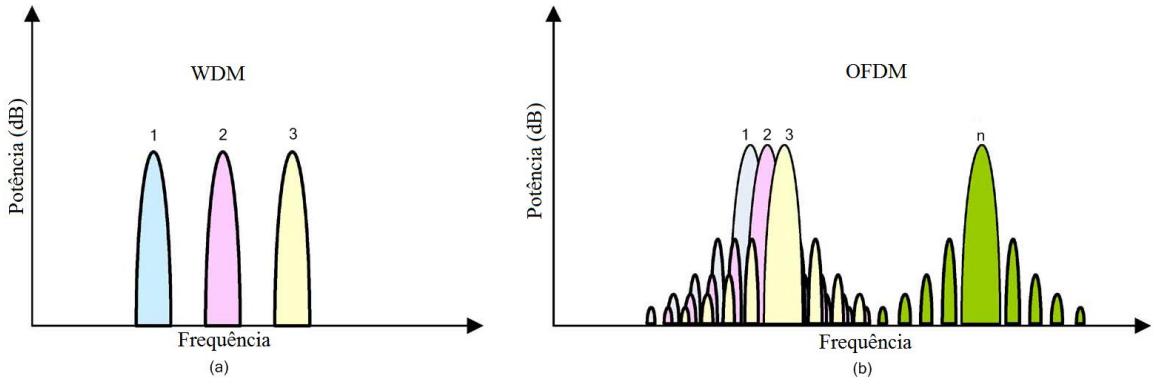

Figura 2-2: Utilização do espectro em um sistema WDM (a) e sistema OFDM (b) (adaptada de [12]).

a banda de transmissão em vários canais com largura de um comprimento de onda. Embora essa tecnologia permita aumentar a capacidade de transmissão de um sistema e seja responsável por avanços importantes nas redes ópticas, ela apresenta desvantagens. A principal delas é a granularidade fixa, que corresponde à de um comprimento de onda. O tráfego, mesmo com demanda menor, utiliza todo o recurso de banda de um comprimento de onda para ser transmitido [5]. Assim, a rede apresenta baixa eficiência em relação à ocupação de banda.

Buscando superar o problema de eficiência espectral da tecnologia WDM e da interferência entre símbolos (ISI, de *inter-symbol interference*) causada pela dispersão do canal de transmissão, outra conhecida tecnologia de multiplexação, que só recentemente foi considerada uma candidata promissora para as novas gerações de redes de transmissão ópticas de banda larga, é a multiplexação por divisão de frequência ortogonal, OFDM. Semelhante à tecnologia WDM, a OFDM divide a banda de transmissão em múltiplas porções de banda menor, chamadas *slots*, e geradas por subportadoras identificadas por suas frequências centrais. No sistema OFDM as subportadoras são ortogonais entre si, permitindo que seus espectros se sobreponham sem que ocorra interferência entre símbolos. Na Figura 2-2 (adaptada de [12]), pode-se ver que a sobreposição dos espectros das subportadoras nos sistemas OFDM proporciona uma utilização mais eficiente da banda de transmissão, diferentemente do sistema WDM, onde os sinais devem estar espaçados entre si para evitar a ISI.

Na Figura 2-3(a) (adaptada de [6]), vê-se que quando uma subportadora é amostrada em sua frequência central, obtém-se seu valor de pico sem a interferência das outras subportadoras, pois, nessa frequência, suas amplitudes são nulas. No domínio do tempo Figura 2-3(b) (adaptada de [6]), o sinal OFDM é uma síntese de várias subportadoras, formando um símbolo OFDM. A condição de ortogonalidade é satisfeita quando as frequências centrais das subportadoras distam n/T_s uma da outra, onde n é um número inteiro e T_s é o período do símbolo [6].

Outra vantagem importante da tecnologia OFDM é que a modulação e demodulação do sinal são implementadas através da transformada discreta de Fourier inversa (IDFT) e transformada discreta de

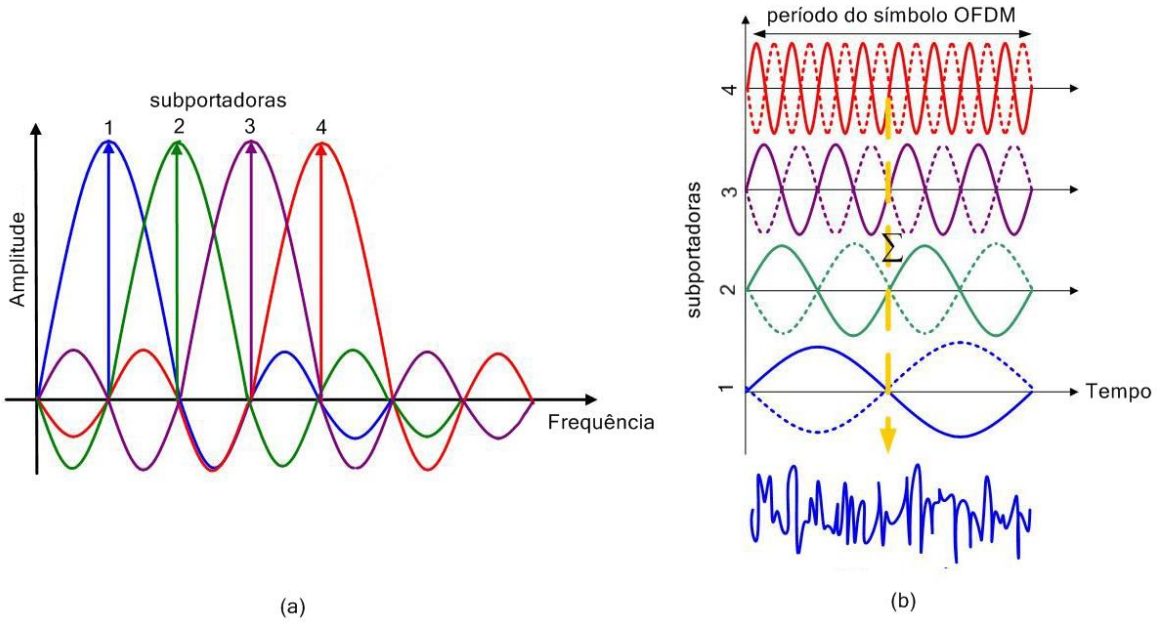

Figura 2-3: Sinal OFDM com 4 subportadoras no domínio da frequência (a) e no domínio do tempo (b) (adaptada de[6]).

Fourier (DFT), respectivamente. A complexidade computacional pode ser reduzida utilizando a transformada rápida de Fourier inversa (IFFT) e a transformada rápida de Fourier (FFT) [6].

Sistemas OFDM possuem inúmeras vantagens, como:

- *Robustez intrínseca à interferência entre símbolos: devido à ortogonalidade das subportadoras;*
- *Alta modularidade: a taxa de transmissão pode crescer aumentando o número de subportadoras (respeitando a condição de ortogonalidade) sem a necessidade de substituição de equipamentos ou dispositivos distintos e/ou complexos no projeto do sistema;*
- *Alta eficiência espectral devido à possibilidade de sobreposição das subportadoras.*

Entretanto, a principal desvantagem da tecnologia OFDM é a sensibilidade a ruídos de fase e frequência, que interferem na condição de ortogonalidade. Outra desvantagem é a alta PAPR (*peak-to-average power ratio*), que relaciona o nível de potência de pico e o nível de potência médio temporal do sinal. Isso distorce o sinal durante a transmissão ou recepção caso estes equipamentos não tenham uma ampla faixa dinâmica [13].

2.3 ARQUITETURA DE REDE E NÓS

Uma rede SLICE é formada por dois equipamentos essenciais: transceptor de banda variável, chamados BVTs (*bandwidth variable transceivers*), e WXCs (*wavelength cross-connects*). Quando um

cliente da rede deseja transmitir um sinal, o BVT é responsável pela geração do sinal óptico a ser transmitido, com banda espectral suficiente. Ele também minimiza a separação entre caminhos ópticos adjacentes, buscando garantir a alta eficiência na utilização dos recursos espectrais. Em seguida, todos os WXCs na rota definida alocam uma banda espectral (com comprimento definido pelo transceptor) criando um caminho óptico para o sinal. Um diagrama simplificado da rede SLICE pode ser visto na Figura 2-4 (adaptada de [5]).

A utilização de comutadores WSS (*wavelength-selective switch*) de largura de banda variável como parte integrante do WXC, além do uso de multiplexação OFDM, aumentam ainda mais a eficiência espectral da rede quando comparada com a rede WDM tradicional [5]. Os mais recentes BV-WSS (*bandwidth variable wavelength-selective switch*) permitem chavear fatias de espectro de diferentes tamanhos [12]. Desse modo, tráfegos com diferentes larguras de banda podem ser chaveados para destinos diferentes, pois o WSS flexível é capaz de alterar a granularidade do chaveamento, como pode ser visto na Figura 2-5 (adaptada de [6]).

O BVT divisível (*sliceable*) é um dos novos conceitos que a rede SLICE utiliza para emprego de sua flexibilidade. Esse transceptor pode ser dividido em transceptores virtuais que servem caminhos ópticos elásticos separados. Tal propriedade evita o "desperdício" ao se utilizar um BVT de alta taxa de transmissão em uma conexão de alguns poucos Gb/s, como acontece com BVTs fixos. Na Figura 2-6 (adaptada de [12]) pode-se ver a diferença de uma rede que transmite três demandas menores que 400 Gb/s (subcomprimento de onda) com BVTs fixos e BVT divisível. Note que, no exemplo, é necessário apenas um BVT divisível, que se divide em três BVTs virtuais para lidar com as três demandas. No caso da utilização de BVTs fixos, são necessários três transceptores diferentes e nenhum deles tem sua capacidade de transmissão totalmente utilizada. Novamente, temos que a flexibilidade da rede e dos equipamentos resulta em uso mais eficiente dos recursos da rede.

Além da capacidade de criar transceptores virtuais para servir diferentes caminhos ópticos elásticos, a nova geração de BVTs permitirá usar diferentes formatos de modulação em cada um deles [12].

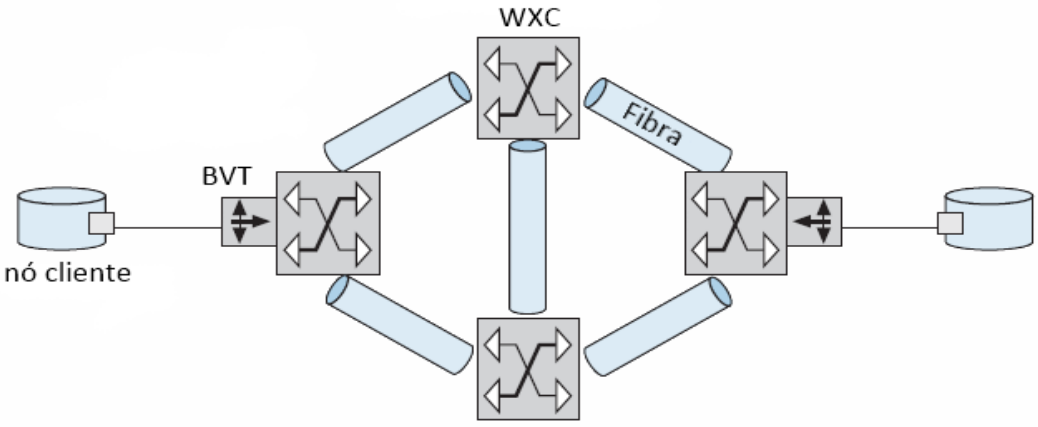

Figura 2-4: Diagrama de blocos de uma rede SLICE (adaptada de[5]).

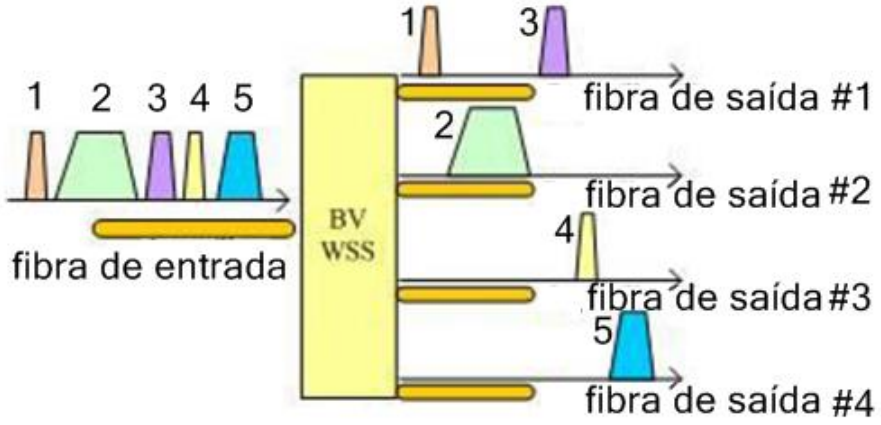

Figura 2-5: Conceito de WSS de largura de banda variável, utilizado no chaveamento do tráfego em rede com grade flexível (adaptada de[6]).

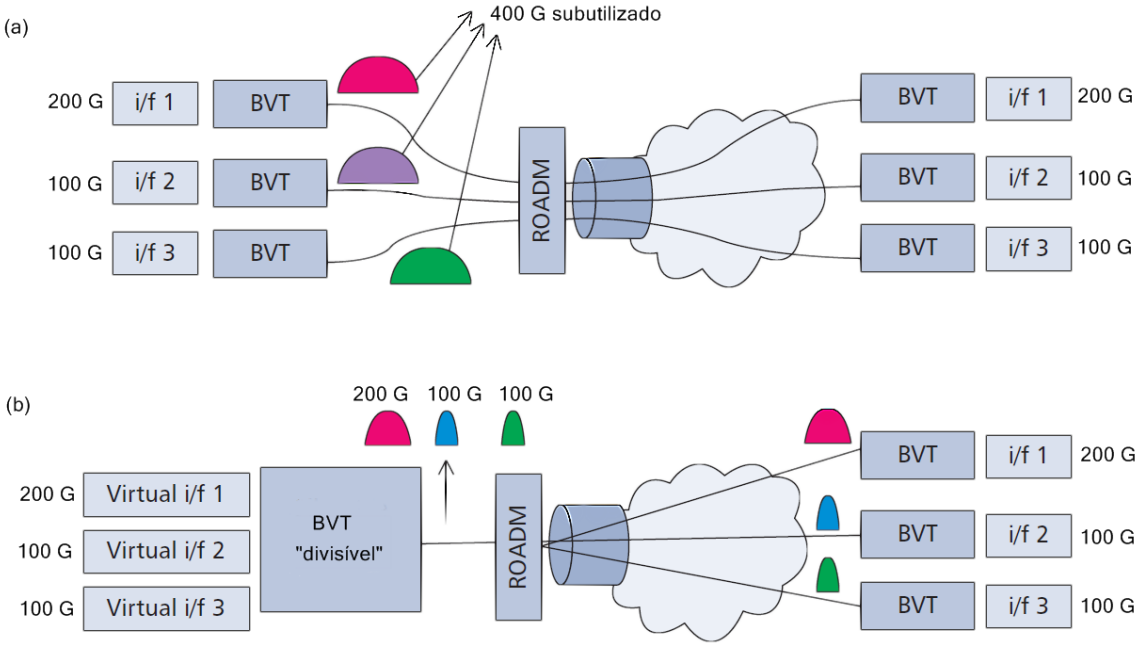

Figura 2-6: Suporte para demandas do tipo subcomprimento de onda usando transceptores fixos (a) e transceptores divisíveis (b) (adaptada de[12]).

3 PROTEÇÃO E RESTAURAÇÃO DE REDES

Chamamos de enlace a conexão entre dois nós. A falha de um enlace ou dispositivo em um dos nós da rede pode resultar na perda de grande volume de dados. Com o aumento das taxas de transmissão, tais falhas tornam-se cada vez mais desastrosas e esquemas eficientes que garantam a sobrevivência do tráfego são essenciais para garantir a qualidade do serviço oferecido.

Durante a configuração de uma rede, o algoritmo de gerenciamento seleciona caminhos fixos por onde os nós da rede se conectam, chamados caminhos primários. O caminho de proteção é aquele para o qual o tráfego é redirecionado quando ocorre uma falha na rede e pode ou não estar previamente configurado.

O sistema de gestão de falhas pode ser baseado na recuperação de um enlace, um caminho (*path*) ou um subcaminho (*sub-path*). Um caminho é um conjunto de enlaces e nós desde o nó fonte até o nó destino. Um subcaminho é um trecho do caminho total, envolvendo um enlace ou um subconjunto de enlaces.

É importante levar em consideração a quantidade de falhas simultâneas que podem ocorrer em uma rede. Embora a probabilidade de ocorrência de duas falhas simultâneas seja muito menor que a probabilidade de apenas uma falha, muito artigos buscam proteger suas redes em caso de mais de uma falha. A ocorrência dessas falhas é geralmente aleatória e todos os enlaces têm a mesma probabilidade de falha. Mas é possível realizar estudos na rede em questão, de modo a determinar enlaces mais suscetíveis a falhas e criar métodos de proteção e restauração específicos para aqueles enlaces.

Caminhos disjuntos são geralmente utilizados como caminho de proteção e caminho principal para garantir a sobrevivência de uma conexão [14]. Há dois tipos de caminhos disjuntos: aqueles que não possuem nós nem enlaces em comum (chamados disjuntos em nó) e aqueles que não possuem enlaces em comum mas podem possuir nós em comum (chamados disjuntos em enlace). Não são considerados os nós fonte e destino. Na Figura 3-1, há exemplos de caminhos disjuntos e não disjuntos.

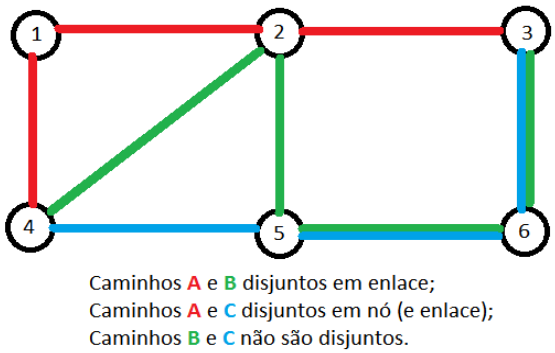

Figura 3-1: Exemplos de caminhos disjuntos e nãos disjuntos.

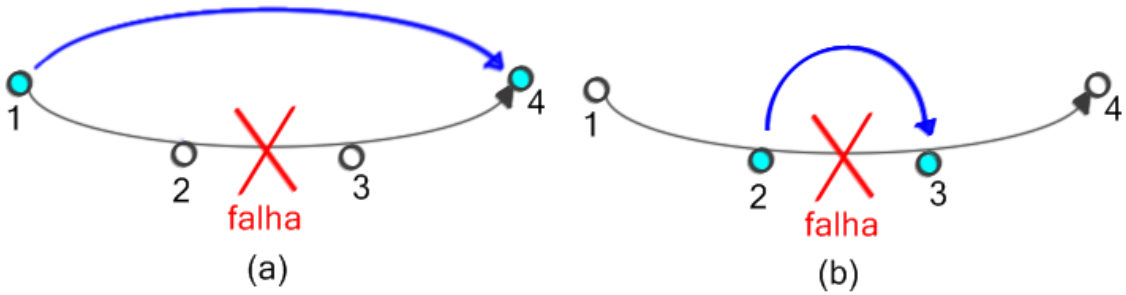

Figura 3-2: Proteção de caminho (a) ou de enlace (b) (adaptada de[14]).

3.1 TÉCNICAS DE PROTEÇÃO

É possível reservar previamente recursos da rede que serão utilizados no caso de falhas. Esses recursos podem ser fibras, dispositivos, ou mesmo alguns segmentos específicos do espectro. Quando é detectada uma falha do sistema, como o rompimento de uma fibra óptica, os recursos reservados são acionados e passam a transmitir os dados. A reserva prévia de recursos é chamada proteção do sistema.

3.1.1 Proteção de enlace ou caminho óptico

A proteção de conexões em uma rede pode ser feita enlace por enlace (chamada proteção de enlace) ou do nó inicial ao final para aquela conexão (chamada proteção de caminho). Ambas podem ser dedicadas ou compartilhadas.

Na proteção de caminho, Figura 3-2(a) (adaptada de [14]), é reservado um caminho de proteção, disjunto (em enlace e/ou nó) do caminho primário. Ao ocorrer uma falha que afete aquela conexão, os nós fonte e destino daquela conexão são informados e a transmissão é realocada para o caminho de proteção.

Na proteção de enlace, Figura 3-2(b), são reservados enlaces e espectros de proteção separadamente para cada enlace do caminho. Ao ocorrer uma falha, apenas os nós diretamente antes e depois do enlace falho precisam redirecionar suas transmissões para enlace de proteção. Esse método é mais veloz que a proteção de caminho em relação ao tempo gasto para redirecionamento do tráfego.

3.1.2 Proteção dedicada (DPP)

A proteção de caminho é classificada como dedicada (DPP, de *dedicated-path protection*), quando cada conexão possui seu próprio recurso para proteção.

No modelo DPP 1+1, cada caminho primário possui o seu próprio caminho de proteção, disjunto do primeiro. O tráfego é enviado simultaneamente pelos dois caminhos e o nó de destino os recebe, compara e escolhe o de melhor qualidade. Caso um enlace de um dos caminhos falhe, o sinal chegará ao seu destino através do outro caminho, sem a necessidade de redirecionamento da conexão. Esse

método garante que os dados serão recebidos em situações em que apenas uma falha ocorre na rede. Em caso de falhas simultâneas, há a possibilidade de os dois caminhos serem afetados, resultando em bloqueio da conexão e em perda dos dados.

De forma semelhante ao método anterior, há o modelo DPP 1:1, no qual cada caminho primário possui um caminho de proteção, mas que fica ocioso ou é utilizado apenas para transmissões de tráfego de baixa prioridade. Quando uma falha no enlace principal é detectada, o caminho de proteção é ativado e passa a transmitir. Esse método é mais lento que o anterior, pois necessita que o tráfego seja redirecionado em caso de falha. Entretanto, consome menos recursos (dispositivos transmitindo e recebendo sinal) e energia da rede.

3.1.3 **Proteção compartilhada (SPP)**

Na proteção de caminho compartilhada (SPP, de *shared-path protection*), os recursos destinados à proteção são compartilhados entre diferentes conexões.

No modelo SPP M:N, *M* conexões primárias compartilham *N* caminhos de proteção, buscando diminuir a quantidade de recursos da rede destinados a proteção. Quando uma falha na rede afeta uma conexão, esta é redirecionada e passa a ser transmitida pelo caminho de proteção. Esse método, entretanto, não protege contra falhas que afetam simultaneamente muitas conexões, pois os caminhos de proteção rapidamente ficarão lotados. Para contornar esse problema, pode-se permitir que apenas conexões cujos caminhos primários são disjuntos compartilhem o mesmo caminho de proteção. Esse método é especialmente eficaz na ocorrência de uma única falha, pois apenas um dos caminhos primários que utiliza aquele caminho de proteção será afetado e precisará ser redirecionado, além de ser o equilíbrio entre eficiência de proteção e utilização de recursos.

3.2 **TÉCNICAS DE RESTAURAÇÃO**

Diferentemente do método de proteção, a restauração do sistema não utiliza recursos previamente reservados. Quando uma falha ocorre, algoritmos buscam por caminhos ópticos e outros recursos disponíveis para reacomodar o tráfego afetado. Embora a proteção do sistema garanta maior velocidade de recuperação da rede, ela é mais dispendiosa, pois envolve gastos com a manutenção dos recursos sobressalentes. A restauração do sistema é mais econômica. A sobrevivência de redes de grande escala exige técnicas mais dinâmicas e com menor gasto em recursos sobressalentes. Para isso, utilizam-se algoritmos de recuperação. Esses algoritmos são responsáveis pela organização da rede, criando caminhos primários, gerenciando o tráfego e, ao detectar uma falha, são responsáveis pelo desvio do tráfego por outros caminhos. Existem várias técnicas utilizadas na restauração de redes e muitos algoritmos diferentes. A sobrevivência de uma rede deve ser planejada com cautela, pois a decisão errada de um algoritmo pode levar a gastos desnecessários de recursos (energia, banda e tempo), diminuindo a qualidade da comunicação.

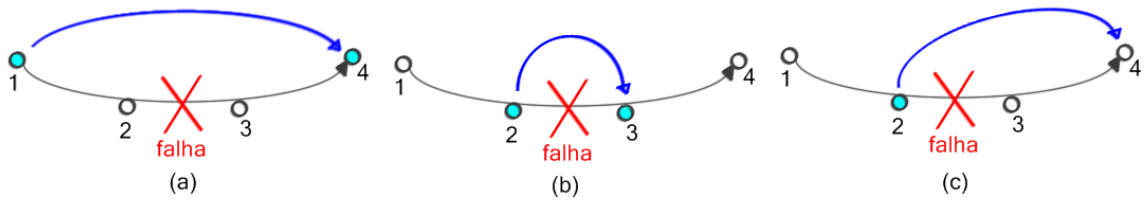

Figura 3-3: Mecanismos de restauração de caminho (a), enlace (b) e subcaminho (c) (adaptada de[14]).

3.2.1 **Restauração reativa ou proativa**

Existem muitas outras características associadas aos sistemas de restauração que definem seu comportamento. Por exemplo, o sistema de restauração pode ser considerado reativo ou proativo. Um sistema reativo é aquele em que os caminhos de proteção são calculados apenas quando ocorre uma falha. Os sistemas proativos já possuem suas rotas alternativas previamente calculadas e definidas quando a rede é configurada. No caso de uma falha, basta alterar a transmissão do caminho principal para o caminho de proteção. Geralmente, sistemas proativos tendem a ser mais eficazes do que os reativos, pois são mais velozes e a chance de sucesso na recuperação da transmissão é maior, já que o algoritmo configura a rede de modo a tolerar tais desvios.

3.2.2 **Restauração de enlace, caminho óptico ou subcaminho óptico**

Diferentemente das técnicas de proteção, a restauração da rede não garante a recuperação da comunicação, pois o algoritmo pode não encontrar um caminho reserva disponível.

O sistema de restauração dinâmico pode ser baseado na recuperação de um enlace, um caminho (*path*) ou um subcaminho (*sub-path*). Na restauração de um caminho, Figura 3-3(a) (adaptada de [14]), os nós fonte e de destino de cada conexão são responsáveis pela descoberta de um novo caminho. Na restauração de um enlace falho, Figura 3-3(b), o algoritmo descobre dinamicamente uma nova rota para todos os sinais que trafegam por aquele enlace e em seguida os nós anterior e posterior ao enlace defeituoso reconfiguram seus comutadores para a nova rota. Caso o algoritmo não encontre uma nova rota que possa transportar o tráfego, a conexão é interrompida.

Quando é necessária a restauração de um subcaminho, Figura 3-3(c), o nó diretamente anterior à localização da falha (dito *upstream*) é responsável pela busca de uma nova rota até o nó destino para cada uma das conexões interrompidas. Como nos casos anteriores, as conexões são bloqueadas se não houver recursos suficientes.

3.2.3 **Restauração dedicada, compartilhada ou multiplexação de primário/restauração**

Assim como nos métodos de proteção, a restauração pode ser dedicada ou compartilhada. Outro método também conhecido é a multiplexação de caminho primário/restauração. Uma faixa de espectro

em um caminho primário *P1*, pode ser configurada como também restauração para outro caminho *P2*. Enquanto o caminho *P1* está sendo utilizado como primário, ele está inacessível como restauração do caminho *P2*, e poderá ser utilizada apenas quando for liberada por *P1*. Esse método apresenta desvantagens, pois se ocorrer uma falha no caminho primário *P2* enquanto *P1* estiver transmitindo, não haverá caminho de proteção disponível para transmitir os dados de *P2* e a conexão será bloqueada.

3.2.4 **Restauração dependente do tipo de falha**

Os métodos de restauração podem também ser classificados em relação à dependência de um certo tipo de falha. Caminhos destinados a restauração totalmente disjuntos do caminho primário são considerados independentes de uma falha em um nó ou enlace em particular, podendo ser utilizados a qualquer instante. Temos também a configuração dependente: são caminhos para restauração disponíveis apenas para utilização em caso de falha em um enlace ou nó em particular.

4 ROTEAMENTO E ALOCAÇÃO DE ESPECTRO (RSA)

O problema de RSA em uma rede elástica é solucionado em função das fatias do espectro de frequência. Para acomodar de maneira flexível diferentes volumes de tráfego na fibra óptica, Jinno e colaboradores propuseram dividir os recursos espectrais nos chamados *slots* de frequência, com largura correspondente à largura do espectro de uma subportadora OFDM [10]. A proposta divide o espectro em *slots* de 12,5 GHz e usa a mesma numeração padrão definida pelo *International Telecommunications Union – Telecommunication Standardization Sector* (ITU-T).

Algoritmos de alocação de espectro lidam com o número de *slots* necessários para transmissão de uma demanda e a posição desses *slots* no espectro. O esquema de divisão dos recursos espectrais pode ser visto na Figura 4-1 (adaptada de [6]).

A restrição de continuidade espectral (SCC, *spectrum-continuity constraint*) [15, 16], pode ser dividida em duas partes:

- *Regra de continuidade espectral: garante que o segmento espectral alocado seja contíguo e o mesmo em todos os enlaces por todo o caminho óptico;*
- *Regra de não sobreposição espectral: garante que dois sinais transportados no mesmo enlace sejam alocados em segmentos espectrais diferentes, de maneira que os sinais não se sobreponham e não interfiram um no outro.*

Essas duas regras são essenciais em todos os algoritmos de roteamento e alocação de espectro.

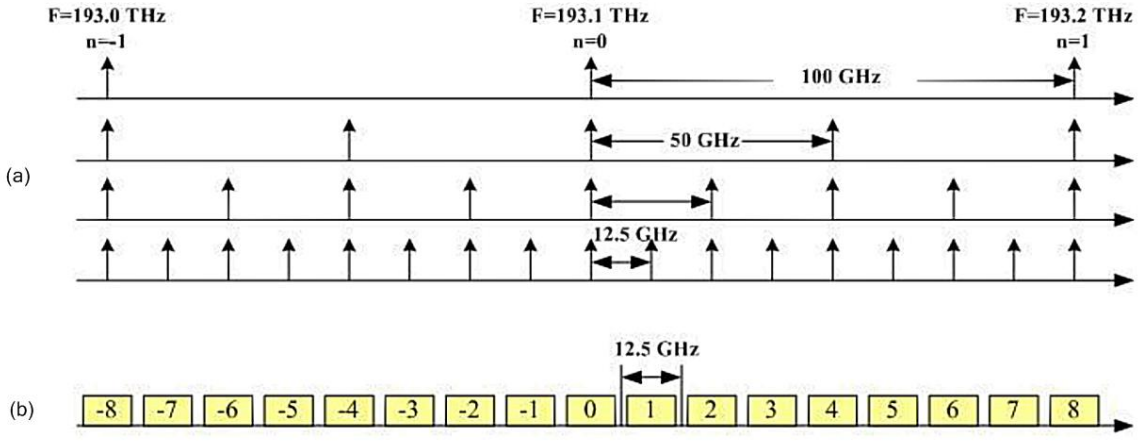

Figura 4-1: Esquemas de especificação dos recursos espectrais. Grade com frequência padronizada ITU-T (a) e com slots de frequência de 12,5 GHz (b) (adaptada de[6]).

4.1 ALGORITMOS HEURÍSTICOS ESTÁTICOS E DINÂMICOS

Em redes de grande escala, o problema de RSA se torna altamente complexo e algoritmos convencionais ILP levam muito tempo para convergir e encontrar uma solução ótima, ou podem não convergir. Nesse caso, opta-se pelo uso de algoritmos heurísticos, capazes de fornecer uma solução viável, ainda que não otimizada, em pouco tempo de processamento.

Os algoritmos de RSA podem ser divididos de acordo com o momento em que são feitos o roteamento e a alocação de espectro. São chamados estáticos aqueles em que as rotas e o espectro que serão utilizados por cada transmissão são pré-determinados e constantes, decididos durante o estágio de planejamento da rede, quando ela ainda está *offline*. Esses algoritmos estáticos, ILP, são capazes de fornecer uma solução ótima para o problema de RSA [15]. O principal objetivo desses algoritmos é minimizar a utilização espectral. Para ter menor complexidade, tais algoritmos podem utilizar a aproximação em duas etapas (R+SA), porém perdem a capacidade de alcançar uma solução otimizada, ainda que encontrem soluções quase-ótimas. Embora mais simples e eficazes, os algoritmos ILP não são bons candidatos para redes de comunicações de grande escala, pois levam muito tempo de processamento para encontrar soluções ótimas [15].

Algoritmos dinâmicos são aqueles em que o RSA é solucionado durante o funcionamento da rede, ou seja, quando está *online*. Cada requisição de conexão é roteada e uma porção do espectro é alocado para transmissão de acordo com as condições em tempo real da rede. Estes algoritmos também podem utilizar aproximações em uma ou duas etapas.

4.2 ROTEAMENTO

O algoritmo de roteamento do tráfego analisa os caminhos disponíveis entre os nós fonte e destino e escolhe um ou mais caminhos, de acordo com condições definidas pelo programador, como caminho a ser utilizado para a conexão entre esses dois nós. Algumas das principais técnicas conhecidas de roteamento são: *shortest path* (SP), *k-shortest path* (K-SP) e *adaptive routing* (AR) [14].

4.2.1 *Shortest path* (SP)

Na técnica SP, o algoritmo analisa todos os caminhos do nó inicial até o nó destino, calculando seu comprimento total. O algoritmo, então, seleciona o caminho mais curto para ser utilizado na comunicação entre os dois nós e descarta todos os demais. Esse algoritmo utiliza o algoritmo para cálculo do caminho de custo mínimo de Dijkstra [14], que calcula o menor caminho entre vértices de um grafo.

O algoritmo utiliza as informações da rede (nós, enlaces e requisições de conexões) e retorna uma lista da rota de menor comprimento para cada requisição.

4.2.2 *K-shortest path* (K-SP)

Semelhante ao método anterior, o K-SP, encontra os *k* primeiros caminhos mais curtos entre o nó fonte e o nó destino. Esse método é mais utilizado do que o anterior, pois diminui a probabilidade de bloqueio de uma requisição de conexão por oferecer mais de um caminho possível.

O algoritmo de Suurballe (*k=2*) fornece duas rotas disjuntas para transmissão, de menor comprimento possível.

O algoritmo de Yen [17] calcula *k* caminhos menores sem *loop* em uma rede, ou seja, sem passar duas vezes por um mesmo nó ou enlace. Ele é muito utilizado por ser considerado extremamente eficiente, pois sua complexidade computacional aumenta linearmente com o valor de *k*.

4.2.3 *Adaptive routing* (AR)

O método AR [14] é o mais complexo entre os três. Nele, o algoritmo escolhe o caminho primário entre o nó fonte e o nó destino de acordo com as condições atuais da rede. Para estar ciente do estado da rede em tempo real, o algoritmo deve receber informações, como quantidade de tráfego, falhas, congestionamentos, ruído e qualidade do sinal. A partir das informações disponíveis o algoritmo pode selecionar o caminho que melhor se encaixa nas condições desejadas. O principal objetivo desse método é diminuir a probabilidade de bloqueio da conexão e aumentar a eficiência da rede de maneira geral, seja aumentando a eficiência espectral, a taxa de transmissão, reduzindo o consumo de energia ou mesmo diminuindo a fragmentação do espectro.

4.3 ALOCAÇÃO DE ESPECTRO

A alocação de espectro pode ser feita de diversas maneiras. O algoritmo procura por uma faixa do espectro que seja contínua em todos os enlaces por todo o caminho a ser percorrido e que tenha o tamanho necessário para o sinal, ou seja, o algoritmo procura por uma sequência de *slots* de frequência (FS, de *frequency slots*) disponível no caminho de transmissão que respeite a SCC.

Vários métodos têm sido estudados, todos com vantagens e desvantagens dependendo da arquitetura da rede ou de outras condições. Alguns dos mais conhecidos são descritos a seguir.

4.3.1 *Random-fit* (RF)

No método de alocação aleatória, o algoritmo analisa todas as sequências de *slots* disponíveis que satisfaçam o requisito de tamanho do sinal a ser transmitido e a SCC de todas as rotas selecionadas e escolhe uma delas aleatoriamente. Geralmente, utiliza-se probabilidade uniforme [18].

4.3.2 *First-fit* (FF) e *last-fit* (LF)

No método FF, o algoritmo de alocação de espectro percorre todo o espectro disponível dos caminhos selecionados na fase de roteamento, buscando por uma sequência de *slots* disponíveis. A faixa

de espectro selecionada pelo algoritmo para transmissão será a primeira encontrada que satisfaça o requerimento de tamanho do sinal a ser transmitido e a SCC. Ao encontrar tal faixa, a busca é finalizada. Além de ser veloz, o objetivo desse método é concentrar os *slots* em uso na porção de menor índice do espectro, aumentando a probabilidade de que rotas mais longas (com maior número de enlaces) encontrem *slots* disponíveis na porção de maior índice [18]. Caso o algoritmo de alocação espectral busque slots disponíveis em mais de uma rota candidata, a busca é interrompida assim que a primeira faixa espectral é encontrada.

Oposto ao método FF, o método LF, analisa várias faixas de espectro disponíveis para transmissão que satisfaçam a SCC e seleciona a última a ser encontrada. Caso o algoritmo procure por slots em várias rotas candidatas, a busca será realizada em todas elas e a faixa selecionada será a última encontrada, independentemente de sua posição no espectro em relação às demais faixas disponíveis encontradas.

4.3.3 ***Most-used*** **(MU) e** ***least-used*** **(LU)**

O método MU procura alocar inicialmente as faixas de espectro mais utilizadas. Contrário a esse método, temos o LU, que procura balancear a utilização espectral por toda a rede. Estudos mostram que o desempenho do método MU supera os métodos FF e LU em relação à utilização espectral [14].

4.3.4 ***Lowest starting slot*** **(LSS) e** ***highest starting slot*** **(HSS)**

No método LSS, o algoritmo busca todos os *slots* consecutivos disponíveis para transmissão que satisfaçam a restrição de continuidade espectral e seleciona o segmento que iniciar no *slot* de menor índice. A vantagem desse método é sua capacidade de preencher espaços vazios sequencialmente, diminuindo a fragmentação espectral e utilizando melhor o espectro [6]. O método HSS, preenche o espectro de forma decrescente, começando pelos *slots* de maior índice. Em comparação aos métodos FF e LF, descritos anteriormente, caso a busca por slots seja realizada em várias rotas candidatas, os métodos LSS e HSS procuram em todas elas e selecionam a faixa espectral de menor ou maior índice, respectivamente, em relação a todas as faixas disponíveis encontradas.

4.4 **ALGORITMO:** ***SPECTRUM-SCAN ROUTING*** **(SSR)**

O algoritmo SSR proposto em [19] e sua extensão *discrete*-SSR (DSSR) [20] possuem complexidade computacional reduzida e são extremamente eficientes em encontrar rotas otimizadas para transmissão. São algoritmos com aproximação em duas etapas, mas que procuram inicialmente pela disponibilidade de espectro para transmissão e depois escolhem o caminho mais curto dentre as opções selecionadas utilizando o algoritmo clássico de Dijkstra. Nesses algoritmos, utiliza-se uma janela de espectro (SW, de *spectrum window*) que tem o tamanho necessário requisitado pela transmissão a ser realizada. Essa janela varre toda a banda de espectro do canal, *slot* por *slot* no modelo SSR, até encontrar

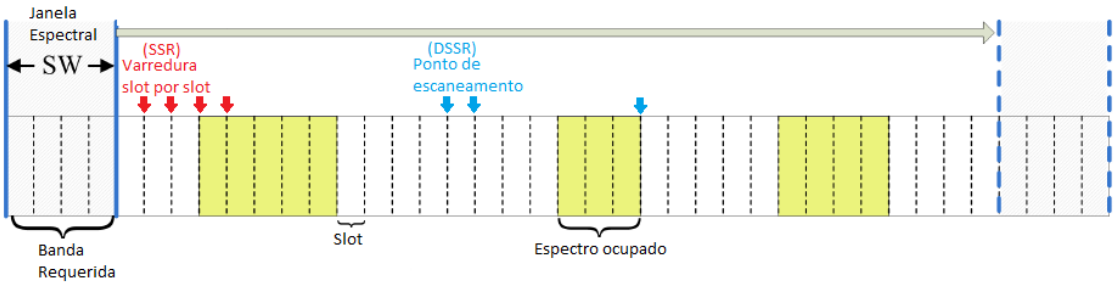

Figura 4-2: Diferença entre varredura do espectro dos algoritmos SSR e DSSR (adaptada de[19]).

um faixa de espectro suficiente para transmissão (do mesmo tamanho, ou maior que a SW) e então utiliza o algoritmo de Dijkstra para selecionar o menor caminho com os enlaces candidatos.

No modelo DSSR, ao invés de varrer a banda *slot* por *slot*, o método busca entre os chamados pontos de escaneamento (*scanning points*), que marcam o fim de um segmento espectral já alocado. Quando o algoritmo faz a varredura do espectro a procura de um segmento livre para transmissão e localiza um segmento já ocupado ele salta para o fim desse segmento (próximo ponto de escaneamento) e continua sua busca, sem precisar varrer todos os *slots* do segmento alocado. Esse método torna a varredura mais rápida e pode ser utilizado em redes flexíveis sem grade espectral. Cada espectro alocado na rede deve ser informado e atualizado na base de dados do algoritmo, para que este possa corretamente localizar regiões utilizadas e livres. Na Figura 4-2 (adaptada de [19]) pode-se comparar os dois métodos de escaneamento do espectro. A complexidade desses dois algoritmos é polinomial. Em [16, 20], o desempenho do algoritmo DSSR é indicado como superior à de outros algoritmos em relação à probabilidade de bloqueio e utilização de recursos.

4.5 ALGORITMO: ROTEAMENTO, NÍVEL DE MODULAÇÃO E ALOCAÇÃO DE ESPECTRO (RMLSA)

A solução do problema de RSA por meio de algoritmos do tipo *distance adaptive* (DA) busca melhorar a utilização espectral, pois considera a possibilidade de seleção do formato de modulação e largura do espectro de acordo com a distância de transmissão. Isso só é possível com a utilização de transceptores nos nós da rede que oferecem a possibilidade de modulação adaptativa e variação da taxa de bits ou largura de banda de transmissão. Desse modo, a utilização espectral é melhorada ao adaptar formatos de modulações em relação à distância de transmissão, utilizando modulação de baixo nível (espectro mais longo) para longas distâncias e de alto nível (com espectro mais curto) para curtas distâncias [6].

O primeiro algoritmo heurístico do tipo *distance adaptive* foi proposto em [10]. Além de calcular os caminhos primários de transmissão utilizando o método K-SP e a alocação de espectro via LSS, o

algoritmo utiliza funções que relacionam a distância de transmissão com o formato de modulação a ser utilizado.

Christodoulopoulos e colaboradores propuseram em [11] um algoritmo que serve sequencialmente a uma lista conhecida de demandas de conexão em uma rede *offline* que realiza o roteamento, define o nível de modulação do sinal e então aloca o espectro para transmissão. O algoritmo, chamado RMLSA (de *routing, modulation level and spectrum allocation*), possui duas versões ILP (de uma e duas etapas) para redes pequenas, e duas versões heurísticas (com a sem ordenação das demandas) para redes de grande porte. O objetivo é diminuir a utilização dos recursos de espectro da rede. No artigo, os autores também avaliam o impacto, em relação à utilização espectral, que a ordem de processamento das demandas tem na solução encontrada pelo algoritmo.

O algoritmo heurístico possui duas versões: com e sem ordenação das demandas. Na primeira, o algoritmo serve as demandas uma a uma, na ordem em que elas aparecem. O algoritmo com ordenação das demandas é dividido, por sua vez, em dois tipos:

- *Maior número de subportadoras primeiro (MSF, de* most subcarriers first*): as demandas são ordenadas de acordo com o número de subportadoras. O algoritmo serve as demandas com mais subportadoras primeiro;*
- *Caminho mais longo primeiro (LPF, de* longest path first*): o algoritmo serve as demandas com maior número de enlaces no caminho mais curto encontrado na fase de roteamento.*

Nos resultados obtidos, o algoritmo heurístico com ordenação de demandas LPF obteve melhor desempenho do que o MSF e ambos se saíram melhores do que o algoritmo sem ordenação de demanda. Como demandas a serem transmitidas em caminhos longos utilizam mais espectro (devido ao formato de modulação necessário), o desempenho da rede pode ser aprimorado ao servir primeiro essas demandas, com a rede ainda vazia. Em relação à variação da modulação do sinal, os algoritmos que permitiam diferentes tipos de modulações em função da distância a ser percorrida pelo tráfego apresentaram melhor utilização dos recursos espectrais da rede.

A seguir, serão apresentados alguns dos principais algoritmos de RSA com proteção e restauração de redes estudados. Nesses algoritmos, as técnicas de RSA são utilizadas tanto para seleção do caminho primário quanto para o de proteção.

5 TRABALHOS RELACIONADOS

Foram estudados trabalhos anteriores sobre sobrevivência em redes ópticas elásticas. Na maioria dos trabalhos estudados, as técnicas de proteção são aplicadas sob cenário estático, ou seja, o tráfego da rede é previamente conhecido.

Em relação ao tipo de recuperação oferecida, os artigos podem ser divididos entre aqueles que oferecem recuperação total e recuperação parcial. Na recuperação parcial o algoritmo busca restaurar apenas parte do tráfego afetado, pois a reserva de recursos destinados à proteção tem apenas uma fração do total de *slots* de frequência requeridos para a transmissão daqueles dados. Por outro lado, na recuperação total, o número de *slots* de frequência reservados para proteção é igual ao número requerido pela conexão, logo todo o tráfego afetado pode ser recuperado.

Os artigos estudados serão melhor detalhados a seguir. A Tabela 5-1 mostra uma comparação entre as principais características desses artigos em relação ao tipo algoritmo (ILP, MILP, heurístico ou meta-heurístico), tráfego (estático ou dinâmico), tipo de proteção e recuperação (proteção dedicada DPP ou compartilhada SPP, restauração R e se é total F ou parcial P), seleção da modulação do sinal (DA), número de falhas (única ou múltiplas falhas) e o ano de publicação.

Tabela 5-1: Comparação entre artigos de sobrevivência de redes estudados.

Artigo	Algoritmo	Tráfego	P & R	DA	Falhas	Ano
[21]	Heurístico	Dinâmico	F-SPP	Não	Única	2012
[22]	MILP	Estático	F-DPP, F-SPP	Não	Única	2012
[23]	ILP	Estático	F-DPP, F-SPP, P-DPP, P-SPP	Não	Única	2014
[24]	Heurístico	Dinâmico	F-SPP	Sim	Única	2013
[25]	Heurístico	Dinâmico	F-SPP	Não	Única	2013
[26, 27]	Heurístico	Estático	P-R	Não	Única	2009, 2011
[28]	Heurístico	Estático	F-SPP	Não	Única	2012
[29]	MILP e Meta-heurístico	Estático	P-DPP+R, P-SPP+R, P-R	Não	Única	2012
[30, 31]	Heurístico	Dinâmico	F-SPP + R	Não	Múltiplas	2011, 2012
[32]	Meta-heurístico	Estático	F-DPP	Não	Única	2013
[33, 34]	ILP e Heurístico	Dinâmico e Estático	F-DPP e P-DPP	Não	Única	2013, 2014

A Tabela 5-2, compara alguns dados mais específicos dos algoritmos, quando disponíveis nos artigos. São comparados o tipo de roteamento e alocação espectral utilizados nos caminhos primário e de proteção, o valor (em número de *slots*) da banda de guarda *G*, o tamanho em GHz de cada *slot* de frequência, o número de *slots* por enlace na rede (*Nsub*), o número de requisições de conexões simuladas (*Nconex*) e a quantidade de *slots* de frequência requeridos pelas conexões (*Rsub*). O formato [*A*,*B*] indica que a demanda é gerada aleatoriamente com distribuição uniforme entre *A* e *B*.

Tabela 5-2: Comparação entre características dos algoritmos estudados.

Artigo	RSA - primário	RSA - proteção ou recuperação	G	FS (GHz)	Nsub	Nconex	Rsub
[21]	K-SP + FF	K-SP + FF	1	-	200	10000	[1,20] e [1,50]
[22]	SP + FF	K-SP + FF	-	-	-	-	[1,5]
[23]	SP + FF	K-SP + FF	-	-	-	-	[1,3], [1,4] e [1,5]
[24]	K-SP + FF	K-SP + MLF	1	12,5	320 (4000 GHz)	250000	[10,400] Gb/s. Depende da modulação selecionada
[25]	K-SP + FF	K-SP + MFSB	1 e 2	-	150	-	[2,5]
[26, 27]	K-SP + LSS	K-SP + LSS (BSR)	-	25	-	-	[1,4]
[28]	K-SP + FF	SP + FF	-	12,5	352 (4400 GHz)	20	2, 4, 6, ou 12.
[29]	-	-	-	12,5 e 25	128 e 64 (1600 GHz)	-	10, 40, 100 e 400 Gb/s (com modulação QPSK)
[30, 31]	Dijkstra modificado (DLB) + FF ou RF	Dijkstra modificado (DLB) + FF ou RF	-	25	100 (25000 GHz)	-	[4,8]
[32]	K-SP + LSS + GA	K-SP + LSS + GA	-	-	1500	-	[2,8] e [2,16]
[33, 34]	MPP	MPP	-	-	-	-	-

Abreviações: BSR: *bandwidth squeezed restoration*, DLB: *dynamic load balancing*, FF: *first-fit*, GA: *genetic algorithm*, K-SP: *k-shortest path*, LSS: *lowest starting slot*, MFSB: *minimum free spectrum block*, MLF: *modified last-fit*, MPP: *multipath provisioning*, RF: *random-fit*, SP: *shortest path*.

5.1 *SHARED-PATH PROTECTION IN* OFDM-*BASED OPTICAL NETWORKS WITH ELASTIC BANDWIDTH ALLOCATION* (SPP-OFDM) [21]

Em [21], Shao e colaboradores propõem um algoritmo heurístico para tráfego dinâmico de proteção compartilhada e comparam duas políticas de compartilhamento: agressiva e conservativa. No compartilhamento conservativo, também chamado de tradicional pelos autores, dois caminhos de proteção cujos caminhos primários sejam disjuntos entre si podem apenas compartilhar seus espectros de proteção se estes tiverem o mesmo número de FS requeridos para transmissão. Na política agressiva, dois caminhos de proteção podem compartilhar seus espectros de proteção contanto que seus respectivos caminhos primários sejam disjuntos entre si. O compartilhamento agressivo tem como benefício o maior número de possibilidades de compartilhamento. O algoritmo calcula inicialmente K-SP candidatos como caminho primário e utiliza o método FF para alocação espectral. Depois de estabelecer o caminho primário, o algoritmo remove os enlaces utilizados por ele da rede e os FS utilizados por outros caminhos primários e por caminhos de proteção cujos caminhos primários não são disjuntos do caminho primário da conexão atual. Caso a política selecionada seja a conservativa, também são removidos da rede os FS dos caminhos de proteção que não possuam a mesma demanda da conexão atual. Só então, o algoritmo calcula *k* menores rotas, disjuntas do caminho primário, para a seleção do caminho de proteção e avalia onde alocar a faixa de espectro daquela demanda, seguindo as restrições de continuidade espectral. A alocação espectral também é feita com FF. A probabilidade de bloqueio (PB) e a probabilidade de bloqueio de banda (PBB) das duas versões do algoritmo são comparadas com um algoritmo de proteção dedicada. Em todas as redes testadas para diferentes valores de carga (em erlangs), o algoritmo de proteção compartilhada apresenta menores PB e PBB em relação ao de proteção dedicada. A política agressiva de compartilhamento obteve melhor desempenho do que a conservativa, pois o número FS compartilhados é maior, consequentemente diminuindo o total de FS necessários para proteção e liberando mais FS para novas conexões.

5.2 *SHARED BACKUP PATH PROTECTION* (SBPP) [22, 23]

Shen e colaboradores propõem em [22] um algoritmo MILP (*mixed integer linear programming*) de proteção compartilhada para tráfego estático, que chamaram de SBPP. No algoritmo, dado um conjunto de conexões a serem alocadas com suas demandas previamente definidas, a rota do caminho primário é calculada usando Dijkstra e a do caminho de proteção é escolhida entre *k* menores caminhos, todos disjuntos do caminho primário. A proteção compartilhada permite que dois caminhos de proteção compartilhem FS desde que seus respectivos caminhos primários não tenham nenhum enlace em comum. Os FS de proteção ficam reservados e são apenas utilizados em caso de falha no enlace por onde o caminho primário transmite, quando a conexão é transferida para o caminho de proteção. Como a probabilidade de ocorrência de duas falhas simultâneas em uma rede é muito menor que a probabilidade

de uma única falha, o artigo considera apenas a ocorrência de uma falha por vez. O método de proteção compartilhada possui menor taxa de capacidade reservada (SCR, de *spare capacity ratio*) quando comparado com o método de proteção dedicada 1+1 e também obteve menor número total de FS utilizados para estabelecer todas as conexões. Isso se deve à possibilidade de compartilhamento de FS destinados à proteção. Os algoritmos foram testados em três redes diferentes, com diferentes graus nodais. Redes mais densas, com maior grau nodal, possuem mais possibilidades de caminhos a serem selecionadas, os caminhos de proteção são em média mais curtos, distribuindo melhor as conexões em seus enlaces e diminuindo a SCR.

Em outro artigo, [23], os autores propõem alterações para otimizar o algoritmo proposto anteriormente. O algoritmo ILP de proteção compartilhada agora possui duas versões, de acordo com as características dos transceptores da rede. Para transceptores configuráveis, assume-se que, em caso de falha, estes dispositivos poderão ser reconfigurados para alterar a frequência das subportadoras do sinal, que agora será transmitido pela rota de proteção, permitindo que a faixa de FS selecionada para transmissão no caminho primário seja diferente da selecionada o caminho de proteção. Na segunda versão, a rede não possui transceptores configuráveis, limitando as opções de caminhos de proteção, pois os índices dos FS utilizados tanto para o caminho primário quanto para o secundário devem ser os mesmos.

O algoritmo utiliza Dijkstra para o cálculo da rota mais curta a ser utilizada como caminho primário e K-SP para as rotas de proteção. As duas versões do algoritmo são comparadas em relação à SCR e número de FS totais utilizados para acomodação de todo o tráfego com o algoritmo ILP de proteção dedicada 1+1. Novamente, o método de proteção compartilhada obteve melhores resultados. A versão para transceptores configuráveis obteve melhores resultados em relação a versão não-configurável, pois as possibilidades de compartilhamento são maiores, levando a melhor utilização do espectro.

Por fim, os autores propõem uma versão dos algoritmos com proteção compartilhada parcial, utilizando restauração por compressão de banda (BSR). O objetivo também é minimizar a utilização espectral. É definido um valor de compressão (entre 0 e 1) que indica a porcentagem do número de FS a serem reservados para proteção em relação a demanda utilizada no caminho primário. Por exemplo, para compressão de 0,5 apenas 50% da demanda será recuperada, ou seja, se a conexão utilizar 8 FS para transmissão no caminho primário, apenas 4 FS serão reservados no caminho de proteção. Foram comparadas duas versões do algoritmo com BSR, uma para transceptores configuráveis e outra para não-configuráveis, com duas versões do algoritmo de proteção 1+1, também com BSR. Quanto menor a razão de compressão, menor o número de FS utilizados para proteção. Portanto, menor o valor total de FS utilizados para acomodar o tráfego.

Na segunda etapa de testes, o objetivo é encontrar, para um valor fixo de FS a em cada enlace da rede, qual valor de compressão garante que todo o trafego seja acomodado e protegido. Os autores concluíram que, quanto mais FS disponíveis, menor é a necessidade de compressão, alcançando o valor

de 100% (proteção total) para mais de 120 FS, no teste utilizando a rede NSFNET. Nas versões do algoritmo de proteção 1+1, por outro lado, não foi possível obter porcentagem de restauração 100%, pois o número de FS necessários é maior que os valores testados. Como nos casos anteriores, as versões dos algoritmos para transceptores configuráveis têm melhor desempenho que as versões para transceptores fixos, devido ao maior número de possibilidades de proteção.

5.3 *SHARED-PATH PROTECTED, ROUTING, MODULATION LEVEL AND SPECTRUM ASSIGNMENT* (SPP-RMLSA) [24]

O algoritmo heurístico de proteção compartilhada proposto por Tarhan [24] para tráfego dinâmico, além de realizar o roteamento e alocação espectral, define o formato de modulação a ser empregado de acordo com a distância a ser percorrida pelo sinal. Seis formatos de modulação estão disponíveis: 64-QAM, 32-QAM, 16-QAM, 8-QAM, Q-PSK e B-PSK, com alcances de 125 a 4000 km. O algoritmo emprega a aproximação em duas etapas, resolvendo primeiro o caminho primário e em seguida o de proteção e as fazes de roteamento e alocação espectral são resolvidas sequencialmente. Tanto para o caminho primário quanto para o de proteção, o algoritmo calcula K-SP (usando o algoritmo de Yen [17]). Dentre as rotas, a seleção dos FS a serem utilizados é realizada de acordo com as restrições do algoritmo e é empregado FF para o caminho primário (com os *slots* de frequência numerados em ordem crescente) e uma versão modificada de *last-fit* (MLF, de *modified last-fit*) para o de proteção (com os *slots* de frequência numerados em ordem decrescente). A intenção é agrupar os FS reservados para proteção e evitar que eles se misturem aos FS utilizados pelos caminhos primários. Nessa versão modificada para o caminho de proteção, a seleção dos FS a serem utilizados só é feita após analisar todas as possibilidades em todos os *k* caminhos e escolhe-se a de menor custo, calculado por uma fórmula que relaciona: o número de FS requeridos para transmissão (definido pela modulação a ser utilizada) e um custo do índice do FS. *Slots* de menor índice tem menor custo, de modo a incentivar a seleção dos primeiros FS, diminuindo a fragmentação do espectro.

O algoritmo não restringe que os FS utilizados por uma conexão para seus caminhos de proteção e primário tenham os mesmos índices (equivalente à versão para transceptores configuráveis do algoritmo de Shen [23]).

Para garantir que a conexão possa ser recuperada, os caminhos de proteção e primário devem ser disjuntos. O compartilhamento de FS entre caminhos de proteção só é possível se seus respectivos caminhos primários forem disjuntos entre si.

A performance do algoritmo é comparada com a dos algoritmos propostos em [21] e [25] em relação a probabilidade de bloqueio, probabilidade de bloqueio de banda, taxa de utilização espectral, grau de compartilhamento (*shareability*, segundo os autores) e fragmentação. A rede utilizada nas simulações é a NSFNET. O algoritmo obteve desempenho superior aos outros dois em todos os testes,

com exceção ao grau de compartilhamento, em que obteve valores muito semelhantes aos do algoritmo proposto em [25] mas ainda superior ao algoritmo proposto em [21].

5.4 *MINIMUM FREE SPECTRUM BLOCK* (MFSB) [25]

No artigo [25], Chen e colaboradores propõem um algoritmo heurístico, para tráfego dinâmico, chamado MFSB (de *minimum free spectrum block*), com o objetivo minimizar o número de FS utilizados e garantir uma alta sobrevivência da rede. Para o caminho primário, o algoritmo utiliza K-SP no cálculo das rotas e FF para alocação espectral. Ao computar as rotas dos caminhos de proteção, além de utilizar K-SP, o algoritmo também considera a probabilidade de falha dos enlaces considerados. As rotas devem ter probabilidade de falha abaixo de um valor máximo definido, chamado pelos autores de JFP (de *joint failure probability*). Para definir quais FS utilizar, o algoritmo proposto escolhe, dentre todas as rotas, o bloco que consumir o menor número de *slots* livres possível, daí o nome do algoritmo. Na fase de alocação espectral é levado em consideração também o número máximo de caminhos que podem compartilhar o mesmo espectro de proteção. Esse valor é definido pois, na ocorrência de uma falha, apenas um dos caminhos primários será recuperado. O algoritmo não impede que caminhos primários não disjuntos compartilhem espectros de proteção. Assim como outros algoritmos de proteção compartilhada, os caminhos de proteção e primário devem ser disjuntos.

Para avaliar o algoritmo proposto, os autores comparam seus resultados com dois outros algoritmos também propostos no artigo: CSSB (*common shared spectrum block*) e MSSB (*maximum shared spectrum block*). A diferença deles para o MFSB é o método utilizado na fase de alocação espectral do caminho de proteção. O primeiro, CSSB, escolhe o primeiro bloco de espectro disponível encontrado (FF), independentemente se é um *slot* livre ou compartilhado. Já o segundo algoritmo opta pelo bloco que utilizar o maior número de *slots* de proteção possível. Em caso de empate, o algoritmo escolhe o bloco do caminho de menor número de enlaces.

Nas simulações realizadas, o algoritmo MFSB obteve melhores resultados em relação à probabilidade de bloqueio e utilização espectral, mas o CSSB alcançou menores valores médios de JFP dos pares primário e *backup* de caminhos selecionados. O MFSB, portanto, minimiza o número de FS utilizados como esperado, mas não garante que os caminhos selecionados possuam baixa probabilidade de falha.

5.5 RESTAURAÇÃO POR COMPRESSÃO DE BANDA (BSR) [26, 27]

Em [26, 27], Sone e colaboradores propõem um algoritmo heurístico para tráfego estático que utiliza o conceito de restauração do tráfego por compressão de banda (BSR).

Para calcular o caminho principal de transmissão de uma demanda, o algoritmo busca rotas mais curtas (K-SP) com espectro contínuo disponível e suficiente para aquela demanda. A rota escolhida é

aquela com espectro disponível de menor índice (LSS). Para o caminho de recuperação, o algoritmo utiliza os mesmos procedimentos, mas busca por uma rota que seja disjunta do caminho primário.

O algoritmo lida apenas com restauração do tráfego e não proteção. Quando uma conexão é afetada por uma falha, uma porcentagem de seu tráfego é redirecionada para o caminho reserva e, algumas vezes, derrubando outras conexões ativas consideradas de menor importância.

A BSR é utilizada quando ocorre uma falha na rede e considera as seguintes possibilidades:

- *Recuperação de banda parcial (PBGR, de* partial bandwidth guaranteed recovery*): a largura de banda no caminho de recuperação é menor que no caminho principal (ocorre compressão de banda). O grau de compressão de banda que será utilizado para recuperação é definido antes da decisão do caminho reserva (que é escolhido sem considerar a quantidade de recursos disponíveis). Caso não tenha* slots *de frequência suficientes no momento de falha, a conexão é bloqueada;*
- *Recuperação de banda total (FBGR, de* full bandwidth guaranteed recovery*): Essa situação é considerada um caso especial da PBGR, quando não ocorre compressão de banda. A banda utilizada pela demanda no caminho reserva é a mesma utilizada do caminho principal;*
- *Recuperação com melhor esforço (BER, de* best-effort recovery*): a banda do caminho reserva é menor que a do caminho primário. Aqui, o caminho reserva é selecionado de acordo com a disponibilidade dos recursos e depois é feita a decisão do grau de compressão a ser utilizado.*

Além de utilizar a BSR na recuperação do tráfego afetado por uma falha, os autores propõem o uso de mapeadores e controladores de demandas transportadas pelos caminhos ópticos. Eles são responsáveis por verificar tráfegos de dimensões diferentes e marcá-los de acordo com seu tamanho. Tráfegos com taxas próximas do limite de transmissão do canal são tratados como alta probabilidade de queda e geralmente são descartados no momento de uma falha, pois são mais difíceis de recuperar. A técnica de BSR proposta no artigo é utilizada para comprimir alguns tráfegos com taxas que excedem um dado limite especificado, ou tratá-los como BER, aumentando a capacidade de recuperação dessas demandas e diminuindo a probabilidade de bloqueio.

Os testes do algoritmo proposto indicaram aumento da sobrevivência do tráfego da rede, principalmente quando tratados como BER.

5.6 *ITERATIVE TWO STEP APPROACH* (ITSA) [28]

Kosaka e colaboradores [28] propõem um algoritmo heurístico de RSA com proteção compartilhada para tráfego estático com o intuito de melhorar o compartilhamento de recursos dos caminhos de proteção e diminuir a fragmentação do espectro. O algoritmo proposto utiliza a

aproximação em dois passos iterativa, chamada ITSA (de *iterative two step approach*). Na aproximação em dois passos, o algoritmo lida inicialmente com a atribuição do melhor caminho primário e posteriormente com a do melhor caminho de proteção, de acordo com os critérios desejados. Embora o caminho de proteção selecionado seja a melhor opção para aquele caminho primário, o par primário/proteção pode não ser a melhor escolha no sentido global, levando em consideração todos os demais caminhos já atribuídos, a fragmentação espectral, ou mesmo a utilização dos recursos da rede [35]. A aproximação iterativa tenta melhorar essas escolhas, considerando várias opções de caminhos primários e de proteção e escolhendo o par que melhor atende tais critérios. Além da ITSA, o algoritmo utiliza também um método de realocação dos caminhos (tanto a rota quanto o espectro selecionado). São definidas duas medidas a serem levadas em consideração: eficiência da utilização espectral (EUE), que calcula para cada conexão a média de FS de proteção compartilhados em cada enlace da rota selecionada, e a pior utilização de fibra (PUF), que mede para cada conexão a fragmentação do espectro em cada enlace da rota. Após todas as conexões estarem estabelecidas utilizando o método de ITSA, o algoritmo seleciona um conjunto de conexões (seus caminhos primário e de proteção) com os piores valores de EUE e de PUF e as remove da rede. As conexões restantes são todas reorganizadas (utilizando ITSA). Em seguida as conexões removidas são realocadas também. Os novos valores de EUE e PUF de cada conexão são atualizados e o processo é repetido, removendo as conexões que menos compartilham FS de proteção e de enlaces mais fragmentados, reorganizando as conexões restantes e realocando as conexões removidas. No artigo, os autores definem que esse processo de reorganização das conexões é realizado cem vezes. O algoritmo teve seus resultados comparados com o algoritmo de proteção compartilhada agressiva proposta em [21] e obteve redução de 8% do número de enlaces e 18% do número de FS utilizados como proteção para a média de seis conexões por par de nós da rede. Esse algoritmo, entretanto, apenas garante que os caminhos primário e de proteção de uma conexão sejam disjuntos, mas não impede que caminhos primários não disjuntos compartilhem FS de proteção. Isso o torna mais vulnerável a falhas em enlaces, diminuindo a capacidade de recuperação do tráfego afetado.

5.7 *SINGLE-PATH PROVISIONING MULTI-PATH RECOVERY* (SPPMPR) [29]

O algoritmo MILP proposto em [29] utiliza técnicas de proteção parcial e restauração parcial do tráfego, buscando um equilíbrio entre a velocidade de recuperação da proteção e a alta eficiência (em relação a utilização espectral) da restauração. A restauração e proteção são ditas parciais, pois o algoritmo utiliza a técnica de BSR.

Conhecendo o perfil do tráfego previamente (tráfego estático), o SPPMPR seleciona um caminho primário para cada conexão utilizando toda a banda requerida. Para o caminho de proteção, apenas uma porcentagem da banda é reservada, garantindo que parte do tráfego esteja protegido em caso de falha na rede. O método de proteção parcial, por utilizar menos recursos da rede do que a proteção total, é considerado mais eficiente em relação a utilização de espectro. O algoritmo possui duas opções de

proteção: dedicada e compartilhada. Na proteção compartilhada, é permitido que dois caminhos primários disjuntos compartilhem seus espectros de proteção.

No caso de uma falha, o algoritmo tenta restaurar o restante da porção do tráfego que não estava protegido. Tanto o caminho de proteção quanto o de restauração são disjuntos do caminho principal selecionado.

Para simulações em redes de grande porte, os autores utilizam um algoritmo meta-heurístico baseado no algoritmo BRKGA (*biased random-key genetic algorithm*) de [36], já que a utilização do algoritmo MILP proposto é impraticável em grandes redes.

5.8 *DYNAMIC LOAD BALANCING SHARED-PATH PROTECTION* (DLBSPP) [30, 31]

Esse algoritmo heurístico, proposto em [30, 31], soluciona o problema de RSA para redes com proteção compartilhada levando em consideração o balanceamento dinâmico da quantidade de tráfego transportado entre os enlaces da rede. O algoritmo tem como principal objetivo tolerar múltiplas falhas em enlaces de redes sob tráfego dinâmico.

Durante o cálculo da rota para transmissão do caminho primário, o algoritmo dá preferência para enlaces com menor carga (mais *slots* de frequência livres) e mais curtos, escolhidos por meio de um algoritmo de Dijkstra modificado. Para o caminho reserva a função de custo do enlace relaciona o número de *slots* livres e *slots* com frequência compartilhada entre outras conexões. Enlaces com todos os *slots* de frequência livres só serão selecionados para utilização em caminhos primários. As rotas dos caminhos primário e de proteção devem ser disjuntas. No artigo, foram avaliados dois tipos de alocação de espectro: FF e RF.

O algoritmo tolera falhas simultâneas em diferentes enlaces, que podem afetar o tráfego no caminho primário e de proteção. Casso isso ocorra, o algoritmo possui um mecanismo de restauração com as informações de tráfego da rede (TAR, de *traffic-aware restoration*). Utilizado para o cálculo de novas rotas de transmissão em caso de falhas, esse mecanismo classifica o tráfego afetado em ordem de prioridade (o tráfego de maior demanda é restaurado primeiro). Desse modo, a demanda com maior número de *slots* de frequência é restaurada primeiro, aumentado a probabilidade de o algoritmo encontrar um novo caminho para transmissão. As demandas menores (com menos *slots* de frequência requeridos) são restauradas com menor prioridade, pois é mais fácil encontrar recursos disponíveis para elas.

O algoritmo, utilizando os métodos de alocação de espectro FF e RF, foi comparado ao desempenho do algoritmo de proteção de caminho compartilhado (SPP) clássico (também utilizando os métodos FF e RF). Comparado ao algoritmo SPP, o algoritmo DLBSPP obteve menores probabilidades de bloqueio de tráfego, melhor utilização espectral e maior razão de restauração de falhas, aumentando a sobrevivência da rede. Entre os dois métodos de alocação de espectro em um mesmo algoritmo, os

métodos FF e RF obtiveram resultados muito semelhantes entre si, embora os resultados da versão FF sejam ligeiramente melhores.

Embora o artigo esteja dentro do contexto de proteção compartilhada, o algoritmo está fora do escopo deste trabalho, pois nosso objetivo é proteger a rede contra uma única falha.

5.9 ALGORITMO GENÉTICO COM PROTEÇÃO DEDICADA DE CAMINHO (RSA/DPP/SC) [32]

O algoritmo híbrido (meta-heurístico), proposto por Klinkowski [32], combina um algoritmo de RSA sequencial com um algoritmo genético (GA) com a função de auxiliar na procura por uma solução quase ótima para o problema de RSA que minimize a ocupação espectral. É utilizado em redes *offline*, com demanda de tráfego conhecida e estática. O algoritmo busca por dois caminhos disjuntos (um primário e um de proteção) utilizando a mesma frequência espectral (mesmo canal) para uma dada demanda. A técnica de proteção dedicada (DPP) aumenta a velocidade de recuperação da rede pois os recursos já estão reservados e não há a necessidade de alterar a frequência de transmissão dos transceptores.

O algoritmo de RSA processa uma sequência de demandas, organizada em ordem decrescente de número de *slots* de frequência requisitados, calculando uma rota e alocando uma faixa espectral suficiente para transmissão (isso é feito para o caminho principal e caminho de proteção). Sabe-se que a ordem do processamento dessas demandas altera o resultado final da ocupação do espectro da rede [11, 32], pois altera a ordem de ocupação do espectro de cada caminho óptico.

Utilizando algoritmo genético, o vetor que representa essa sequência de demandas sofre permutações em sua ordem (mutações), ainda respeitando a ordem decrescente de tamanho do espectro requerido, de maneira a encontrar uma ordem de demandas que, quando processadas pelo algoritmo de RSA, minimize a ocupação espectral na rede.

Os resultados das simulações foram comparados aos do algoritmo MSF proposto por Christodoulopoulos e colaboradores em [11] (veja seção 4.5). O algoritmo proposto em [32] obteve menor utilização espectral.

5.10 *SURVIVABLE MULTI-PATH ROUTING AND SPECTRUM ALLOCATION* (SM-RSA) [33, 34]

Lu Ruan e colaboradores propõem um algoritmo de RSA para tráfego estático em [33] e dinâmico em [34] que utiliza a técnica de MPP (*multi-path provisioning*). Nos artigos, são apresentadas as versões ILP e heurística do algoritmo SM-RSA. Os resultados são comparados com o algoritmo SS-RSA (de *survivable single-path routing and spectrum allocation*).

No algoritmo SM-RSA, são computadas $N \geq 2$ rotas disjuntas, utilizando o algoritmo de Bhandari [37] e é definido um valor $0 < q < 1$, referente ao nível de proteção requerido (q igual a um indica proteção total e menor que um, proteção parcial). Cada requisição de conexão possui demanda de largura de banda B. O algoritmo, de acordo com o número de rotas e o nível de proteção, aloca faixas de espectro (respeitando a SCC) em cada rota para aquela conexão de modo que a capacidade total alocada em todas as rotas seja ao menos B e a capacidade total alocada em cada grupo de $N-1$ rotas seja ao menos qB. Desse modo, o algoritmo garante que toda a demanda requerida esteja distribuída entre as N rotas disjuntas selecionadas e, em caso de falha em algum enlace de uma das rotas, os caminhos restantes transmitam uma porcentagem do tráfego de acordo com o nível de proteção desejado. O objetivo é minimizar o número de *slots* de frequência utilizados.

Note que, como todos os caminhos selecionados para cada requisição são caminhos ativos (estão transmitindo dados), não há compartilhamento de espectros de proteção como nos algoritmos descritos anteriormente.

Para demonstrar a eficiência do algoritmo proposto, é criado o algoritmo SS-RSA, que calcula apenas duas rotas disjuntas. Em uma delas, será alocado FS suficientes para transmissão da demanda B daquela requisição de conexão e na outra rota é alocado qB. Esse algoritmo é semelhante à técnica de proteção não compartilhada 1+1, com proteção total caso $q = 1$.

Nas simulações realizadas, é demonstrado que o algoritmo SM-RSA é mais eficiente em relação à utilização espectral e essa eficiência aumenta conforme o número de rotas disjuntas utilizadas para cada conexão aumenta. Isso se deve à distribuição da demanda entre muitas rotas, consumindo menos *slots* de frequência por rota e total do que no caso de proteção 1+1. Por exemplo, para $N = 2$ e $q = 0,8$ o algoritmo SM-RSA seleciona duas rotas disjuntas e aloca $0,8B$ por rota, totalizando $1,6B$. Já o algoritmo SS-RSA utiliza $1B$ em uma das rotas e $0,8B$ em outra, totalizando $1,8B$.

Embora a técnica de proteção utilizando transmissão via múltiplos caminhos apresentada nos artigos e suas vantagens em relação à eficiência espectral sejam importantes, o algoritmo proposto não lida com proteção compartilhada como no algoritmo proposto neste trabalho.

5.11 CONCLUSÕES DO CAPÍTULO

O estudo de artigos publicados é importante na definição de padrões e métodos de avaliação dos algoritmos. Dentre os artigos descritos anteriormente, pode-se ver a complexidade do problema de roteamento e alocação espectral com proteção compartilhada e a grande quantidade de estudos já realizados. Entretanto, notou-se também a dificuldade na obtenção de dados suficientes para a reprodução de tais algoritmos, como pode ser visto pela quantidade de espaços sem informações na Tabela 5-2.

Dentre os artigos estudados, nenhum leva em consideração o tamanho da rota e a posição da demanda dentro do espectro simultaneamente ao definir seus caminhos de proteção e primário. A maioria opta pelo método FF de alocação espectral.

Também pode-se ver que são poucos os estudos envolvendo algoritmos heurísticos de proteção compartilhada para tráfego dinâmico e a maior parte dos estudos trabalha com tráfego estático.

O algoritmo desenvolvido neste trabalho possui características únicas, como as técnicas *smart-fit* e busca lógica, que serão descritas nos capítulos a seguir.

6 DEFINIÇÃO DO PROBLEMA

O objetivo do algoritmo criado é minimizar o custo das conexões atendidas, ou seja, encontrar a melhor combinação entre o comprimento da rota e os *slots* de frequência ocupados, de modo a diminuir a fragmentação do espectro, aumentar a utilização espectral e diminuir o número de conexões bloqueadas.

O problema de roteamento e alocação de espectro com proteção e restauração de rede sob tráfego dinâmico pode ser formulado como se segue.

6.1 A TOPOLOGIA DA REDE

Um conjunto de pontos (vértices) conectados por linhas (arestas) pode ser representado por um grafo. Um grafo *G (V,A)* é definido pelos conjuntos *V* e *A*, em que *V* é o conjunto não vazio dos vértices e *A* é o conjunto de arestas ou pares ordenados $a = (v,u)$, com v e $u \in V$.

A topologia de uma rede conectada é representada por um grafo $Gnet_{N \times N}$, no qual *N* representa o número de nós da rede. O grafo indica quais nós da rede estão conectados e qual a distância D_{ij} entre os nós *i* e *j* (também chamado de custo do enlace). Dois nós não conectados diretamente por um enlace possuem custo infinito.

Como a rede é conectada por enlaces ópticos bidirecionais, cada enlace contém um par de fibras ópticas e cada fibra é utilizada para transmissão em um sentido. O número de *slots* de frequência (ou subportadoras), FS_L, é o mesmo em todas as fibras ópticas. Conexões transmitidas através de um mesmo enlace não são consideradas disjuntas, mesmo que tenham direção diferentes e, portanto, utilizem fibras ópticas diferentes. Uma falha na rede afeta o enlace e, consequentemente, as duas fibras.

Considera-se que a rede possua transceptores suficientes para atender a demanda em todos os seus nós.

6.2 GERAÇÃO DE TRÁFEGO

O tráfego gerado segue a distribuição de Poisson com λ requisições de conexão por unidade de tempo e o tempo de duração das conexões segue uma distribuição exponencial com valor médio unitário $1/\mu=1$. A carga da rede é dada em erlangs ($carga = \lambda/\mu$).

A quantidade de subportadoras requeridas para cada conexão também é gerada aleatoriamente com distribuição uniforme entre 1 e *Rsub_max*. Os nós fonte *s* e destino *d* são escolhidos aleatoriamente entre todos os nós da rede com probabilidade uniforme.

O tráfego gerado é ordenado em ordem crescente de tempo de chegada das conexões, simulando o comportamento de tráfego dinâmico. Porém, para avaliar a influência da ordem das requisições com relação à probabilidade de bloqueio, a função de geração de tráfego permite também ordenar o tráfego

em ordem crescente ou decrescente do número de subportadoras requeridas. Nesse caso, as conexões ainda são tratadas de uma em uma, mas já não possuem comportamento do tráfego dito dinâmico.

Uma requisição de conexão *R* possui as seguintes informações: nós fonte *s*, nó destino *d*, número de subportadoras requeridas *Rsub*, tempo de chegada *ta* e tempo de duração *th*.

6.3 OCORRÊNCIA DE FALHAS

Para avaliação da capacidade de restauração da rede, é possível gerar aleatoriamente uma ou duas falhas (não simultâneas) em enlaces da rede, escolhidos aleatoriamente. Optou-se por trabalhar com uma única falha da rede pois a probabilidade da ocorrência de múltiplas falhas simultâneas é muito pequena em relação à probabilidade de ocorrer uma única falha [23]. As falhas geradas afetam apenas enlaces da rede e não seus nós.

Para que a falha afete a rede por um período de tempo suficiente, o instante de início e de duração da falha está relacionado ao tempo total de todo o tráfego gerado. Caso apenas uma falha seja gerada, a falha ocorre após aproximadamente 10% do tempo total de todo o tráfego gerado e dura até o fim da simulação. Caso opte por gerar duas falhas, o tempo de duração é aproximadamente entre 10-40% e 50-90% da simulação.

A segunda falha foi usada para testar a resposta do programa caso uma falha (após a primeira) afete enlaces de proteção das conexões que foram anteriormente afetadas pela primeira falha e ainda estão transmitindo. Nesse caso, a conexão não pode ser recuperada e é bloqueada.

6.4 CONEXÕES TRATADAS (*OUTPUT*)

O algoritmo lida com eventos discretos. Assume-se que o estado da rede não se altera entre a ocorrência de dois eventos. Os eventos são: chegada de uma conexão, fim de uma conexão, ocorrência de uma falha da rede e fim (conserto) da falha.

Após tratadas pelo algoritmo, as conexões aceitas *C* possuem as seguintes características: nó fonte *s*, nó destino *d*, número de *slots* de frequência necessários *SWsize* (contando com a banda de guarda *G*) e, para os caminhos primário e de proteção respectivamente, as informações de rota *Ppath*, *Bpath*, custo final *Pcost*, *Bcost* (que se refere ao custo da rota e do *slot* de frequência selecionado) e o índice do primeiro *slot* de frequência ocupado pela conexão *Pfs*, *Bfs*.

Além das características das conexões, quando aceitas, o algoritmo também informa caso uma conexão seja bloqueada (seja por falta de rota disponível ou falta de FS livres), o motivo do bloqueio e se a conexão foi afetada pela falha na rede e recuperada (ou não).

6.5 RESTRIÇÕES

Para que o algoritmo heurístico seja capaz de tomar decisões apropriadas ao tratar uma requisição de conexão, ele deve estar sujeito a um conjunto de restrições:

a) Restrição de continuidade espectral (SCC)

Como dito anteriormente, engloba: continuidade espectral, contiguidade espectral e não sobreposição espectral. A restrição de não sobreposição espectral é aplicada para *slots* já ocupados de caminhos primários. Novas conexões de caminhos primários podem apenas utilizar *slots* livres. Também, caminhos de proteção não podem utilizar *slots* já reservados como primários.

b) Compartilhamento de *slots* de frequência de caminhos de proteção

Slots de frequência de caminhos de proteção podem ser compartilhados, contanto que seus respectivos caminhos primários sejam disjuntos em relação aos enlaces utilizados.

c) Caminhos de proteção e primário de uma conexão devem ser disjuntos

Para garantir a recuperação de uma conexão afetada por uma falha, seus caminhos primário e de proteção devem ser disjuntos em enlace. No algoritmo, permite-se que caminhos considerados disjuntos possuam nós em comum pois as falhas geradas afetam apenas enlaces.

d) Limite do índice de *slots* de frequência alocados

Cada enlace da rede possui um número fixo de FS. Uma conexão deve ser alocada de modo a caber inteiramente no espectro disponível, sem extrapolar esse limite.

6.6 AVALIAÇÃO DO ALGORITMO

Para que o algoritmo proposto seja avaliado, é necessário analisar diversas informações e resultados e comparar com outros algoritmos conhecidos. Neste trabalho, foram avaliados:

a) Probabilidade de bloqueio (PB)

Uma conexão é aceita se o algoritmo encontrar recursos suficientes disponíveis na rede para aquela conexão. Caso contrário, ela é bloqueada. A probabilidade de bloqueio é o número de conexões bloqueadas NC_{Bloq} em relação ao total de requisições de conexões NC_{Total}.

$$PB = \frac{NC_{Bloq}}{NC_{Total}} \qquad 6.1$$

b) Probabilidade de bloqueio de banda (PBB)

Quando uma conexão é bloqueada, a demanda de banda requerida para aquela conexão não é atendida. Considerada uma avaliação melhor da performance do sistema que a probabilidade de bloqueio, a PBB mostra a relação entre a soma das demandas das conexões bloqueadas B_{Bloq} e a demanda total requerida de todas as conexões B_{Total}. No programa, esse valor é calculado em função do número de *slots* de frequência requeridos pelas conexões.

$$PBB = \frac{B_{Bloq}}{B_{Total}} \qquad 6.2$$

c) Grau de compartilhamento (Shar, de *shareability*)

É a média temporal da relação entre a quantidade total de *slots* de frequência requeridos para proteção das conexões aceitas (*SWsize* multiplicado pelo número de enlaces do caminho de proteção de cada conexão) e a quantidade de *slots* de frequência reservados para proteção em toda a rede. No caso de proteção dedicada, esse valor é igual a um. Na proteção compartilhada, por outro lado, muitos dos *slots* de frequência destinados à proteção são compartilhados entre várias conexões. Quanto maior esse valor, maior o grau de compartilhamento da rede e menor a quantidade de recursos reservados para proteção.

$$Shar = \frac{total\ FS\ de\ proteção\ requerido\ por\ conexão}{\sum FS_B} \qquad 6.3$$

d) Fragmentação espectral (Frag)

Utilizada em [24] e também para medir a fragmentação em discos rígidos de computadores, essa fórmula mostra a fragmentação espectral dos enlaces da rede durante as simulações. A fragmentação de um enlace, $Frag_L$, é proporcional à relação entre o maior segmento contíguo de *slots* livres, e o número total de *slots* livres no enlace, FS_F. Caso todas as demandas das conexões estejam organizadas lado a lado, sem *slots* livres entre elas, não haveria fragmentação no espectro, pois o único e maior segmento de *slots* livres possui tamanho igual ao número de *slots* livres no enlace. Se não houver *slots* livres em um enlace, ele não é considerado fragmentado, logo $Frag_L = 0$.

Como o algoritmo trabalha em modo dinâmico, com novas conexões entrando e outras saindo, o cálculo da fragmentação dos enlaces é realizado a cada evento e, ao final da simulação, faz-se uma média dos valores obtidos. Com a média por evento do grau de fragmentação dos enlaces é calculada a fragmentação média por enlace $Frag_N$ (Fórmula 6.5), utilizada como representação do grau de fragmentação da rede.

$$Frag_L = 1 - \frac{Maior\ segmento\ livre}{FS_F} \qquad 6.4$$

$$Frag_N = \frac{\sum Frag_L}{L} \quad (6.5)$$

e) Taxa de utilização espectral (SUR, de *spectral utilization ratio*)

É a média temporal (por evento) da relação entre o somatório do número de *slots* ocupados nos enlaces, sejam eles utilizados como primário (FS_P) ou como proteção (FS_B), e o total de *slots* de frequência em toda a rede, que é calculado pela soma do número de *slots* de frequência em cada enlace FS_L.

$$SUR_N = \sum \frac{FS_B + FS_P}{FS_L} \quad (6.6)$$

f) Taxa de capacidade reservada (SCR, de *spare capacity ratio*)

É a média temporal que relaciona a quantidade de *slots* de frequência utilizados como proteção e a quantidade utilizada como primário. Caso o valor seja maior que um, o número de *slots* da rede reservados para proteção supera o número de *slots* utilizados para transmissão de dados. É comum que esse resultado seja obtido, principalmente quando lidamos com proteção dedicada, pois os caminhos de proteção costumam ser maiores (possuem mais enlaces) que os primários, consumindo mais FS. Um dos objetivos da utilização de técnicas de proteção compartilhada ou restauração por compressão de banda é reduzir esse valor, consumindo menos recursos livres da rede.

$$SCR_N = \sum \frac{FS_B}{FS_P} \quad (6.7)$$

g) Porcentagem de restauração (Rest)

É o número de conexões recuperadas em relação ao número de conexões afetadas pela falha. Na proteção dedicada, esse valor é sempre 100%, pois as conexões possuem proteção garantida e única. Na proteção compartilhada, ao permitir que apenas conexões disjuntas em enlace compartilhem FS de proteção e lidando com apenas uma falha, esse valor também deve ser 100%. É um meio de verificar se o algoritmo está correto.

$$Rest = \frac{NC_{Rec}}{NC_{Falha}} \quad (6.8)$$

7 ALGORITMO HEURÍSTICO DE ROTEAMENTO E ALOCAÇÃO DE ESPECTRO "*SMART-FIT*" COM PROTEÇÃO COMPARTILHADA DE CAMINHO (RSA_SF_SPP)

O algoritmo proposto, para tráfego dinâmico, tem como objetivo localizar um caminho primário e um caminho de proteção para cada requisição de conexão, buscando minimizar o consumo de recursos (*slots* de frequência) da rede. Caso a rede não possua recursos disponíveis suficientes ou um dos caminhos (primário ou de proteção) não seja encontrado, a conexão é bloqueada. O algoritmo divide o problema de RSA em duas etapas, procurando inicialmente por rotas entre os nós fonte e destino e em seguida procura nas rotas selecionadas por FS livres a serem alocados.

O RSA_SF_SPP foi desenvolvido, inicialmente, para aplicar o algoritmo proposto em [23] (descrito na seção 5.2), em redes com tráfego dinâmico. O algoritmo também possui duas versões: para transceptores configuráveis e não-configuráveis. Com a proposta do método de *last-fit* modificado em [24] (seção 5.3), o algoritmo desenvolvido foi alterado, passando a utilizar (tanto para o caminho primário quanto para de proteção) uma técnica que compara as rotas e seus respectivos *slots* de frequência disponíveis em cada caminho e seleciona o caminho de menor custo final. Essa nova técnica foi chamada de *smart-fit* (SF), pois consideramos que esse nome descreva melhor o método empregado.

Outra característica importante do algoritmo descrito a seguir é o método de busca lógica, utilizado na procura dos FS livres para os caminhos primário e de proteção. Como será demonstrado, a técnica empregada é mais veloz que a conhecida *spectrum-scan routing* (SSR), descrita na seção 4.4.

7.1 CHEGADA DE UMA NOVA CONEXÃO

O programa estabelece inicialmente o caminho primário. Se for encontrado, passa a procurar o de proteção.

7.1.1 Caminho primário

A Figura 7-1 mostra de maneira simplificada os passos realizados pelo algoritmo durante a busca do caminho primário. As rotas dos caminhos primários são calculadas utilizando o algoritmo de Yen [17], que calcula os *k* caminhos mais curtos, sem *loop*, entre dois nós. Em todas as *k* rotas encontradas, procura-se pela disponibilidade espectral para transmissão de tamanho suficiente (adiciona-se ao número subportadoras requeridas pela conexão o valor da banda de guarda *G*).

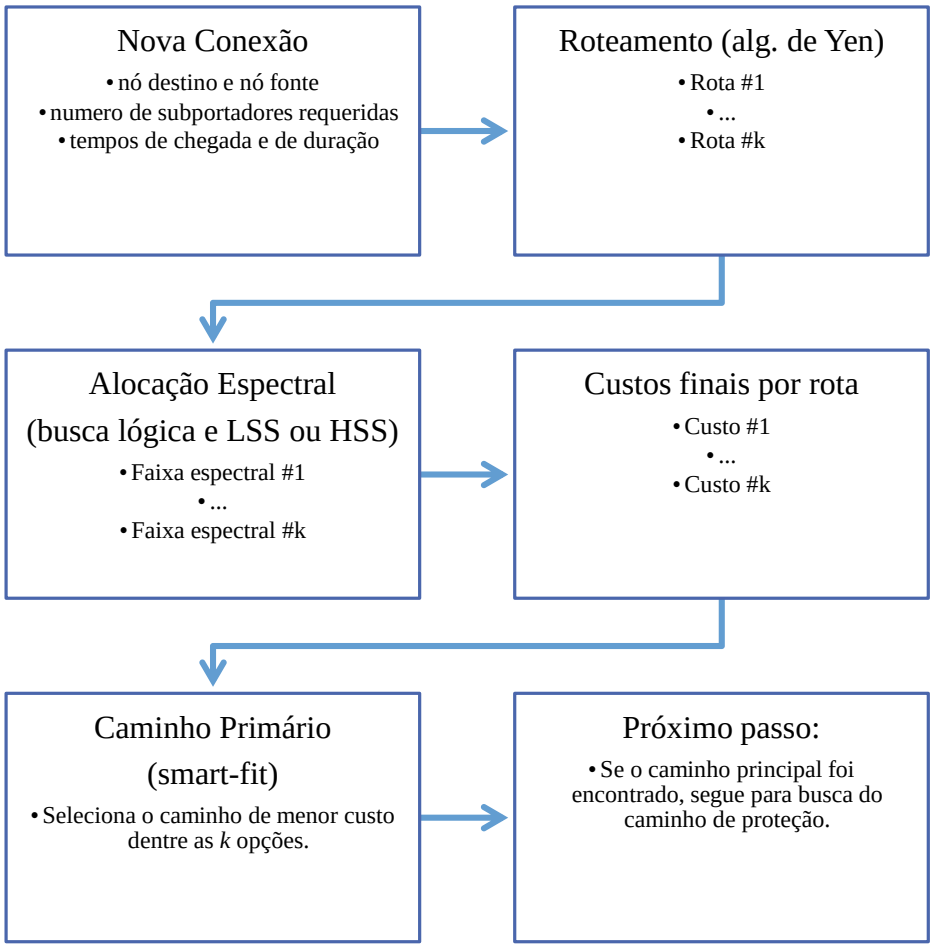

Figura 7-1: Fluxograma dos passos realizados durante a busca pelo caminho primário.

Utilizando a matriz de auxílio *Cmat*, o programa busca *slots* de frequência disponíveis para transmissão como caminho primário. Os *slots* de frequência são contíguos, devem ser os mesmos (mesmo índice) em todos os enlaces da rota e não podem se sobrepor a nenhum outro *slot* já utilizado por outra conexão. Caso haja mais de uma possibilidade de faixa espectral livre, escolhe-se a de menor índice (LSS, *lowest starting slot*) para aquela rota.

O custo final do caminho (Fórmula 7.1) é o custo da rota (definido pela soma das distâncias dos enlaces percorridos) somado ao custo daquele conjunto de *slots* de frequência (seu índice inicial multiplicado por uma constante α). É importante lembrar que, na versão HSS do algoritmo, o índice dos slots deve ser decrescente, para que o algoritmo selecione a faixa mais à direita disponível.

$$FS_{custo}^{k} = Dist_k + FS_1^k . \alpha \qquad 7.1$$

Por fim, dentre todos os *k* caminhos, é escolhido o caminho de menor custo final como caminho primário de transmissão. Esse método foi chamado de *smart-fit*, pois compara as possibilidades de alocação espectral dentre as rotas disponíveis e seleciona a melhor delas, buscando equilibrar os custos entre a posição da faixa espectral dentro do enlace e o tamanho da rota.

Caso não haja rota disponível, ou não sejam encontrados *slots* de frequência livres para a transmissão, a conexão é bloqueada. Com o caminho primário selecionado, o algoritmo segue para a seleção do caminho de proteção.

Na função responsável pela localização de *slots* de frequência disponíveis pode-se alterar o método de seleção do *slot* entre LSS ou HSS (*highest starting slot*). Isso se assemelha ao método utilizado em [24], que busca separar os *slots* de caminhos primários dos de proteção em porções opostas do espectro. Para permitir o uso desse método, os transceptores da rede devem ser configuráveis, pois o conjunto de FS utilizados pelo caminho primário e de proteção não serão os mesmos.

7.1.2 **Caminho de proteção**

A Figura 7-2 mostra de maneira simplificada os passos realizados pelo algoritmo durante a busca do caminho de proteção. Semelhante ao método anterior, calculam-se *k* menores rotas. O caminho de proteção deve ser obrigatoriamente disjunto (em enlace) do primário. Para isso, o grafo que representa a rede é temporariamente alterado, removendo-se os enlaces do caminho primário definido. Isso garante que os caminhos procurados pelo algoritmo sejam todos disjuntos em enlace do caminho primário.

O número de FS a serem reservados para proteção é o mesmo que a requisição para aquela conexão. Por isso é dito que o algoritmo oferece proteção total. A seleção dos *slots* de frequência depende das

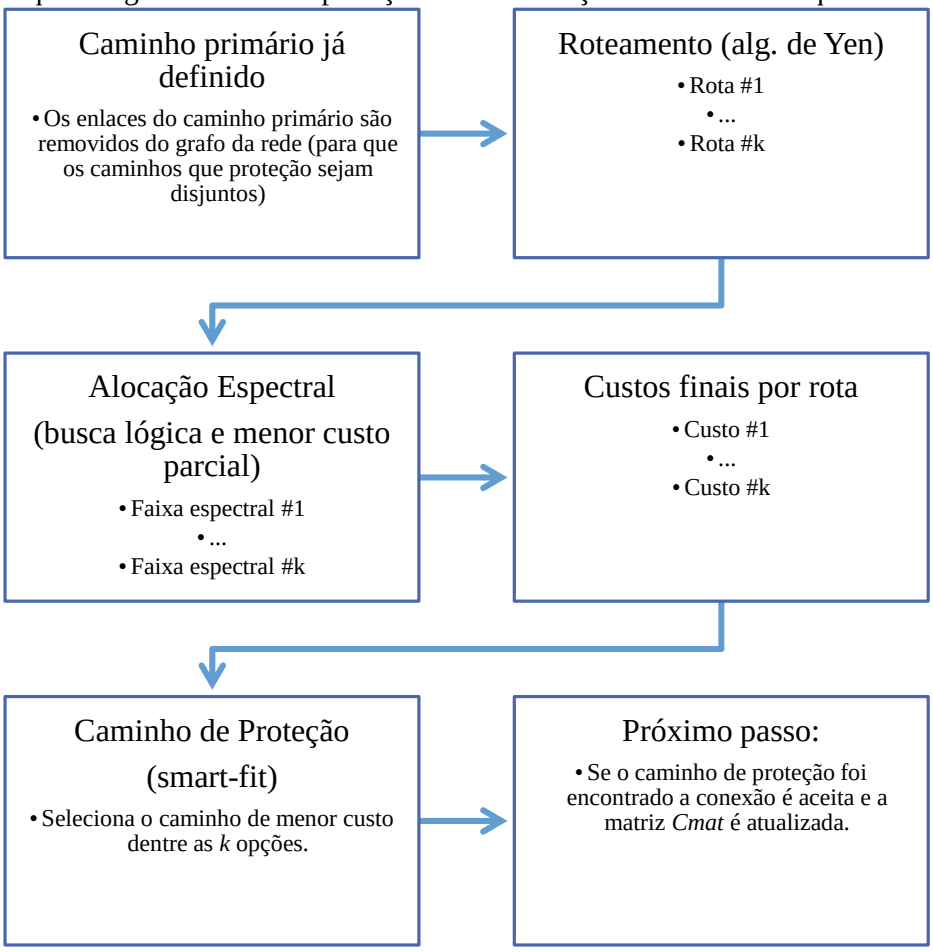

Figura 7-2: Fluxograma dos passos realizados durante a busca pelo caminho de proteção.

características dos transceptores da rede. Se eles forem configuráveis, é possível que os índices dos *slots* reservados para proteção sejam diferentes dos utilizados para transmissão. Nesse caso, quando uma falha afetar a conexão, ela deverá ser reconfigurada para transmitir na nova faixa de espectro. Caso os transceptores não sejam configuráveis, o valor dos índices dos *slots* selecionados para o primário e o caminho de proteção devem ser os mesmos (mesma faixa espectral).

Para cada uma das rotas, procuram-se *slots* de frequência disponíveis (de acordo com o tipo de transponder). Nessa etapa, há duas possibilidades:

(a) o caminho de proteção compartilhará uma faixa de espectro (ou parte dela) com outro caminho de proteção já estabelecido;

(b) o caminho de proteção utilizará uma faixa de espectros livre.

Para o compartilhamento de espectro ser possível, os caminhos primários de todos as conexões envolvidas devem ser disjuntos em enlace. Isso garante que, na ocorrência de uma falha, apenas uma das conexões irá transmitir utilizando essa faixa de frequência. Novamente, os *slots* de frequência são contíguos, devem ser os mesmos (mesmo índice) em todos os enlaces da rota e não podem se sobrepor a nenhum outro *slot* já utilizado por outra conexão como caminho primário, mas há a possibilidade de compartilhamento entre outros caminhos de proteção.

Caso haja mais de uma possibilidade de *slots* de frequência livres e/ou compartilháveis na mesma rota, escolhe-se a de menor custo parcial. Nessa etapa, não são selecionados os *slots* de menor índice como no caso do caminho primário. Para incentivar o compartilhamento espectral, calculam-se os custos parciais: o índice inicial daquele grupo de FS multiplicado por uma constante β (Fórmula 7.2) ou γ (Fórmula 7.3). Essas constantes dependem se a faixa de espectro escolhida é uma faixa compartilhada, situação (a), ou livre, situação (b), e é escolhido o menor entre eles.

$$FS^{k}_{custo\ parcial(comp.)} = FS^{k}_{1}.\beta \quad (7.2)$$

$$FS^{k}_{custo\ parcial(livre)} = FS^{k}_{1}.\gamma \quad (7.3)$$

$$FS^{k}_{custo} = Dist_{k} + FS^{k}_{custo\ parcial} \quad (7.4)$$

Esses custos buscam equilibrar a posição dos *slots* a serem utilizados dentro do espectro e a possibilidade de compartilhamento. Nas simulações, o custo de escolher uma faixa compartilhada é menor que o custo de uma faixa livre ($\beta < \gamma$). O objetivo é dar preferência ao compartilhamento espectral, buscando deixar os espectros livres para a transmissão de caminhos primários.

O custo final dos caminhos é calculado como o custo da rota (definido pela soma das distâncias dos enlaces percorridos) somado ao custo do *slot* de frequência (Fórmula 7.4).

Utilizando novamente a técnica *smart-fit*, é selecionado o caminho de menor custo final como caminho de transmissão de proteção. Caso não haja rota possível, ou não sejam encontrados *slots* de frequência livres ou a serem compartilhados, a conexão é bloqueada.

É importante lembrar que, nos caminhos de proteção não são transmitidos dados, apenas nos primários. Os recursos de proteção ficam previamente selecionados e reservados para serem utilizados apenas em caso de falha. Essa característica é fundamental pois torna possível o compartilhamento de FS entre caminhos de proteção.

7.1.3 **Método de busca por faixa espectral disponível: Busca Lógica**

O método padrão de busca por *slots* de frequência disponíveis para alocação procura em cada um dos enlaces da rota todas as faixas de espectro livres de tamanho suficiente para ser alocada. Para isso, é criada uma janela espectral (SW) que varre todos os *slots* do enlace buscando por uma sequência de *slots* disponíveis. O tempo de processamento desse método depende do número de enlaces da rota e do número total de *slots* de frequência em cada enlace da rede.

Visando acelerar esse processo, o método utilizado neste algoritmo compara todos os enlaces simultaneamente e a busca pelos espectros disponíveis é feita por meio de convolução.

A matriz de auxílio *Cmat* foi desenvolvida pela primeira vez para esse método. A função de busca é diferente caso os *slots* desejados sejam destinados a um caminho primário ou alocados como recursos de proteção. *Cmat* é uma estrutura formada por células que representam cada enlace da rede. Cada célula, por sua vez, é o conjunto de três vetores chamados *Cmat.Freq*, *Cmat.Type* e *Cmat.ID*. O primeiro representa o conjunto de *slots* de frequência de cada enlace e se eles estão sendo utilizados ou não. O segundo indica qual o tipo de uso do respectivo *slot* (1 se primário ou 0 se for um *slot* reservado para proteção). O terceiro vetor possui uma lista de todos os identificadores das conexões que estão utilizando aquele *slot* (no caso de proteção compartilhada, mais de uma conexão poderá estar atribuída a um mesmo *slot* de frequência).

a) Notações importantes

O algoritmo desenvolvido nesse trabalho foi criado utilizando o software MATLAB. Alguns conceitos e definições são importantes para seu entendimento.

- **Célula (*cell*)**

Um vetor célula é um tipo de dado do MATLAB que contém células, que são dados indexados de tipos diferentes, como vetores de tamanhos diversos, textos, caracteres, ou mesmo combinações de letras e números. Por exemplo, um vetor célula:

>> vetorCelula = {1 2 3; 'teste' 'Natália' [99 15 12]};

É um vetor de 6 células (2 linhas e 3 colunas) com a seguinte forma:

1	2	3
teste	Natália	99 15 12

- ***Estrutura (structure)***

Um vetor estrutura é um tipo de dado que contém campos (*fields*) com tipos diferentes de dados. Todas as estruturas de um mesmo vetor tem os mesmos campos e os mesmos tipos. Por exemplo, um vetor estrutura:

```
>> vetorEstrutura(1).nome = 'Natália';
>> vetorEstrutura(1).numero = 1;
>> vetorEstrutura(1).vetor = [99 15 12];
>> vetorEstrutura(2).nome = 'Capelari';
```

Contém os campos: nome, numero e vetor:

vetorEstrutura(1)	vetorEstrutura(2)
•nome: Natalia •numero: 1 •vetor: [99 15 12]	•nome: Capelari •numero: [ ] •vetor: [ ]

- ***Índice***

Dados contidos em um vetor podem ser acessados por meio de seu índice, ou seja, indicando qual o local, dentro do vetor, onde está o dado desejado. Por exemplo, temos o vetor de números com uma linha e três colunas:

```
>> vetor = [99 15 12];
```

Para acessar o primeiro número dentro desse vetor, temos que usar seu índice:

```
>> vetor(1)
        ans = 99
```

É importante lembrar que no MATLAB, diferentemente de outras linguagens de programação, o primeiro índice de um vetor é 1 (e não 0). Essa característica é usada no algoritmo para acessar os *slots* de frequência dentro do espectro de frequência dos enlaces e também durante a função de busca lógica que será explicada a seguir.

b) Seleção de FS para primário

A Figura 7-3 mostra a utilização de um enlace da rede, representado pela matriz de auxílio *Cmat*, utilizado pelas conexões ativas *A*, *B* e *C*. Suponha que uma nova conexão *D* (de tamanho 2) queira utilizar esse enlace como caminho primário. Note que, mesmo utilizado como proteção, os *slots* de 7 a 10 possuem o valor no vetor *Freq* iguais a um. Para uma conexão como caminho primário, o algoritmo precisa apenas buscar por regiões com valor zero no vetor *Freq*. Para a conexão *D*, o algoritmo escolhe o conjunto de *slots* com menor índice inicial possível, no caso os *slots* 4 e 5. A nova célula, após a configuração da conexão *D*, pode ser vista na Figura 7-4.

	1	2	3	4	5	6	7	8	9	10
Freq	1	1	1	0	0	0	1	1	1	1
Type	1	1	1	0	0	0	0	0	0	0
ID	A	A	A				B	BC	BC	BC

- *Slots* 1 a 3: usados como primário pela conexão A.
- *Slots* 4 a 6: livres.
- *Slots* 7 a 10: usados como proteção das conexões B (7 a 10) e C (8 a 10)

Figura 7-3: Exemplo de um enlace da rede representado pela matriz de auxílio Cmat.

	1	2	3	4	5	6	7	8	9	10
Freq	1	1	1	1	1	0	1	1	1	1
Type	1	1	1	1	1	0	0	0	0	0
ID	A	A	A	D	D		B	BC	BC	BC

Figura 7-4: Enlace representado pela matriz Cmat após ocupação de um novo caminho primário.

	1	2	3	4	5	6	7	8	9	10
Freq	1	1	1	1	1	0	1	1	1	1
Type	1	1	1	1	1	0	0	0	0	0
ID	A	A	A	D	D		BE	BC	BC	BC

Figura 7-5: Enlace representado pela matriz Cmat após a ocupação de um novo caminho de proteção.

c) Seleção de FS para proteção

Seguindo o mesmo exemplo, uma outra conexão *E* (de tamanho 1) deseja utilizar o enlace como caminho de proteção. Nesse caso, ela poderá ser alocada no *slot* 6, utilizando um *slot* vazio, ou nos *slots* 7 a 10, compartilhando com as conexões *B* e/ou *C*. O algoritmo verifica se os caminhos primários das conexões *B*, *C* e *E* são disjuntos em enlace e descobre que *E* e *C* não podem compartilhar espectro. Nesse caso, a conexão *E* pode apenas ocupar os *slots* 6 ou 7.

Mesmo possuindo um índice menor, o custo de um *slot* livre é maior que o custo de um *slot* compartilhado, então o algoritmo escolhe colocar a conexão *E* no *slot* 7. A célula é atualizada, resultando na Figura 7-5.

d) Busca lógica

Para buscar simultaneamente em todos os enlaces da rota, a função de busca seleciona os enlaces pertencentes à rota na matriz *Cmat* e faz um OU lógico entre os vetores correspondentes. O exemplo da Figura 7-6 mostra uma rota de dois enlaces sendo preparada para a operação lógica.

O enlace final é utilizado na fase de operação lógica. Para caminhos primários, o algoritmo realiza um não-E lógico entre os vetores não-*Freq_final* e não-*Type_final* (Equação 7.5). Para caminhos de proteção, é realizado a operação E lógico entre *Freq_final* e *Type_final* (Equação 7.6). O resultado, é um vetor único com apenas zeros e uns, chamado *PRI_final* ou *BCK_final*.

$$PRI_f = \overline{\left(\overline{Freq_f} \,.\, \overline{Type_f}\right)} \qquad 7.5$$

$$BCK_f = \left(Freq_f \,.Type_f\right) \qquad 7.6$$

Seguindo o mesmo exemplo anterior, a Figura 7-7 mostra os vetores finais no caso de caminho primário ou de proteção.

A Figura 7-8 mostra como é realizado o cálculo para localização dos índices iniciais dos *slots* de frequência onde a conexão pode ser alocada. Com o vetor final, o algoritmo faz uma convolução com um vetor de valores 1, *SW*, do mesmo tamanho da requisição da conexão (já adicionada a banda de guarda). Supondo que a conexão a ser alocada tenha tamanho 2, o vetor da janela espectral será *SW* = [1 1].

No vetor resultado da convolução, os índices onde são encontrados valores nulos (subtraídos o tamanho do vetor *SW* - 1) são os índices dos *slots* iniciais disponíveis para alocação. Ignoram-se valores nulos, negativos ou muito grandes (tais que a janela espectral não caiba inteira dentro do vetor *Freq*).

Enlace #1

	1	2	3	4	5	6	7	8
Freq	0	0	1	1	1	0	1	1
Type	0	0	1	1	1	0	0	0
ID			A	A	A		B	B

Enlace #2

	1	2	3	4	5	6	7	8
Freq	0	0	0	1	1	1	1	1
Type	0	0	0	1	1	0	0	0
ID				C	C	D	D	D

Enlace final = (Enlace #1) OU (Enlace #2)

	1	2	3	4	5	6	7	8
Freq_final	0	0	1	1	1	1	1	1
Type_final	0	0	1	1	1	0	0	0
ID_final			A	A C	A C	D	B D	B D

Figura 7-6: Preparação dos enlaces de uma rota para a operação lógica.

Caminho primário:

	1	2	3	4	5	6	7	8
Não-Freq_final	1	1	0	0	0	0	0	0
Não-Type_final	1	1	0	0	0	1	1	1
PRI_final	0	0	1	1	1	1	1	1

Caminho de proteção:

	1	2	3	4	5	6	7	8
Freq_final	0	0	1	1	1	1	1	1
Type_final	0	0	1	1	1	0	0	0
BCK_final	0	0	1	1	1	0	0	0

Figura 7-7: Operação lógica para caminho primário (Equação 7.5) ou caminho de proteção (Equação 7.6).

Caminho primário:

i. *Convolução: CONV (PRI_final, SW) = [0 0 1 2 2 2 2 2 1];*

ii. *Índices do vetor CONV onde os valores são nulos: idx = [1 2];*

iii. *Calculo da localização dos* slots *livres: idx – (size (SW) – 1) = [0 1];*

Remove-se valores nulos, negativos ou muito grandes, logo o índice do *slot* inicial onde a conexão pode ser alocada é 1.

Caminho de proteção:

i. *Convolução: CONV (BCK_final, SW) = [0 0 1 2 2 1 0 0 0];*

ii. *Índices do vetor CONV onde os valores são nulos: idx = [1 2 7 8 9];*

iii. *Calculo da localização dos* slots *livres: idx – (size (SW) – 1) = [0 1 6 7 8];*

Remove-se valores nulos, negativos ou muito grandes, logo os índices dos *slots* iniciais onde a conexão pode ser alocada são 1, 6 e 7. Para os *slots* 6 e 7, é necessário verificar a possibilidade de compartilhamento com as conexões B e D.

Figura 7-8: Convolução e seleção dos índices iniciais dos slots de frequência onde a conexão pode ser alocada para caminho primário ou caminho de proteção.

7.2 FALHA EM UM ENLACE

A Figura 7-9 mostra os passos realizados pelo algoritmo e as decisões tomadas ao ocorrer uma falha na rede.

Quando um dos enlaces da rede falha, o primeiro passo é indicar esse enlace como indisponível para todas as conexões futuras (até que a falha seja resolvida). Isso impede que novas conexões usem aquele enlace.

Para as demais conexões já estabelecidas, o programa seleciona todas as conexões que tinham como parte do caminho primário o enlace falho. Essas conexões têm suas rotas primárias redirecionadas para suas respectivas rotas de proteção (a matriz *Cmat* é devidamente atualizada). Os *slots* de frequência em todos os enlaces (inclusive os não danificados) utilizados pelos caminhos primários afetados são liberados para que novas conexões possam utilizá-los.

Os caminhos de proteção que estão agora em uso são atualizados e passam a ser considerados *slots* primários, impedindo que novas conexões utilizem aquele espaço como proteção. Para isso, seus valores no vetor *Cmat.Type* dos enlaces da rota de proteção passam a ser 1. Entretanto, as conexões que anteriormente compartilhavam aqueles *slots* de frequência continuam com as mesmas configurações, pois como apenas uma falha por vez ocorre na rede, esses *slots* de frequência não serão requisitados por novas conexões falhas. Há, porém, a possibilidade da ocorrência de uma segunda falha, após a primeira, que afete uma outra conexão configurada para utilizar os mesmos *slots* de frequência de proteção da conexão anteriormente afetada. Caso a primeira conexão ainda esteja transmitindo, a segunda não

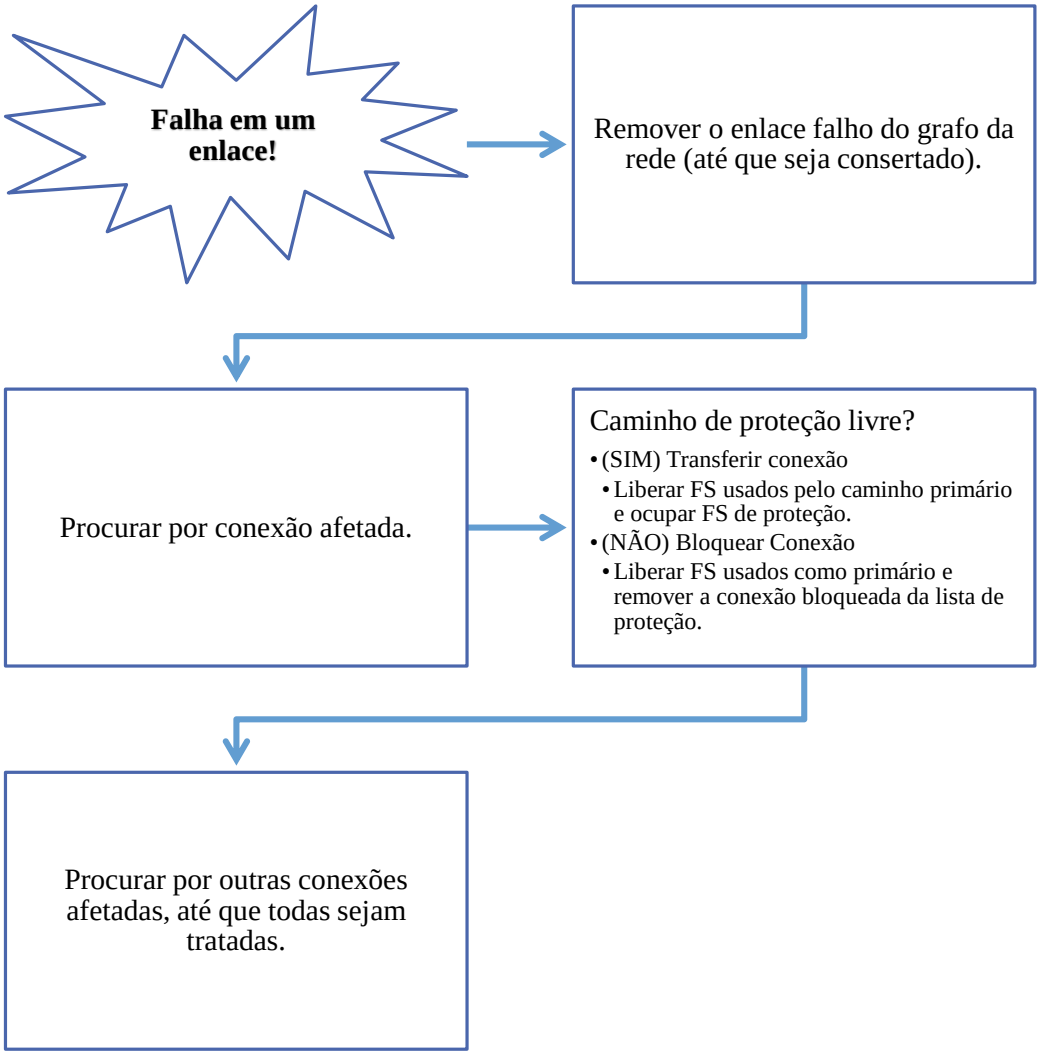

Figura 7-9: Fluxograma dos passos realizados para recuperar as conexões afetadas por uma falha.

poderá ser recuperada. O programa, antes de redirecionar uma conexão afetada por uma falha, verifica se o caminho de proteção reservado ainda está disponível.

Quando uma conexão não pode ser recuperada, ela é bloqueada. Seu caminho primário é liberado, assim como os FS reservados para proteção. É importante lembrar que tais FS podem ser *slots* compartilhados. Nesse caso, o algoritmo apenas remove o identificador da conexão bloqueada da lista de conexões utilizando aqueles *slots*, mas os mantém reservados para as conexões ainda ativas.

Quando uma falha é resolvida (consertada), o programa volta a considerar aquele enlace disponível, mas nenhuma conexão desviada volta ao caminho primário original, pois os *slots* antes utilizados por elas possivelmente já foram ocupados por novas conexões.

7.3 FIM DE UMA CONEXÃO

Quando uma conexão termina, o programa libera os *slots* de frequência de proteção e primários utilizados por ela. Se os FS utilizados para proteção daquela conexão forem compartilhados, o algoritmo

apenas remove o identificador da conexão finalizada da lista de conexões utilizando aqueles *slots*, mas os mantém reservados para as conexões ainda ativas.

Caso seja uma conexão previamente afetada por uma falha, os slots de frequência de proteção que estavam transmitindo são indicados como disponíveis novamente. Para isso, seus valores no vetor *Cmat.Type* dos enlaces da rota de proteção voltam a ser 0. As demais conexões que compartilhavam esses slots poderão ser recuperadas em caso de uma nova falha.

7.4 COMPLEXIDADE COMPUTACIONAL

O problema de RSA com proteção compartilhada possui complexidade de tempo polinomial não determinístico (NP-completo) [21].

Para que a complexidade do algoritmo heurístico RSA_SF_SPP seja reduzida, ele foi dividido em duas etapas. O algoritmo busca o caminho primário e, se for encontrado, busca o caminho de proteção em seguida. Também foi dividido em duas etapas o problema de RSA, como explicado anteriormente.

A primeira etapa do algoritmo é a busca pelas rotas candidatas ao caminho primário. Sabe-se que o algoritmo de Yen possui complexidade computacional $O[K\,N\,(M + N \log N)]$ [38]. Onde K é número de rotas desejadas, N é o número de nós da rede e M o número de enlaces.

A segunda etapa é a busca pelos *slots* de frequência disponíveis, realizada pelo algoritmo de busca lógica. Como a versão do algoritmo para caminhos de proteção é mais complexa que a versão para caminho primário por lidar também com as listas de todas as conexões utilizando cada slot, essa versão foi utilizada para o cálculo de complexidade. O algoritmo realiza operações OU lógicas com todos os enlaces da rota. A complexidade desse passo é $O[C\,N]$, onde C é o número de *slots* de frequência nos enlaces e a operação será realizada para cada enlace da maior rota sem *loop* possível, passando por todos os N nós da rede. A função que remove valores repetidos e organiza a lista de identificadores das conexões utilizando cada slot do enlace é bastante complexa, pois no pior caso possui todas as B conexões disjuntas ativas, compartilhando os mesmos *slots*, e foi concatenada um número de vezes igual ao tamanho da maior rota sem *loop* possível. Além disso, ela seria realizada em todos os *slots* do enlace. Logo, essa função possui complexidade $O[CBN \log BN]$. Por fim, a convolução realizada possui complexidade $O[C \log C]$ no pior caso, onde a demanda teria o mesmo tamanho do enlace.

Após encontrar todas as posições no espectro, a função que realiza a verificação se duas conexões são disjuntas em enlace possui complexidade $O[C\,B\,M]$, pois ela será chamada no máximo para todos os C *slots*, para verificar todas as B rotas disjuntas possíveis da rede, e o vetor de representação dos enlaces utilizados para cada rota possui tamanho M.

Logo o algoritmo possui complexidade computacional:

$$O[\,(K\,N\,(M + N \log N)) + (C\,(N + B\,N \log BN + \log C)) + (C\,B\,M)\,]$$

Note que, para um valor muito alto de nós na rede (N), o trecho do algoritmo de maior peso computacional é o algoritmo de Yen. Para enlaces com muitos *slots* (C), o algoritmo de Busca Lógica

se torna a função de maior peso, mas seu comportamento não é polinomial. É importante ressaltar que, embora o cálculo de complexidade seja realizado para o pior caso, o comportamento do algoritmo, em média, não alcança tais valores.

8 OUTROS ALGORITMOS UTILIZADOS PARA COMPARAÇÃO

8.1 ALGORITMO HEURÍSTICO DE ROTEAMENTO E ALOCAÇÃO DE ESPECTRO *FIRST-FIT* COM PROTEÇÃO COMPARTILHADA DE CAMINHO (RSA_FF_SPP)

Esse algoritmo de proteção compartilhada para tráfego dinâmico utiliza o método de FF na seleção dos FS para os caminhos primário e de proteção. O algoritmo não precisa avaliar a disponibilidade espectral em todas as *k* rotas encontradas, apenas seleciona a primeira rota em que encontrar tais recursos disponíveis. A comparação entre os algoritmos RSA_SF_SPP e RSA_FF_SPP *tem como objetivo avaliar a eficiência na utilização de recursos do método smart-fit desenvolvido.* O algoritmo também possui versões para transceptores configuráveis e não configuráveis.

8.2 ALGORITMO HEURÍSTICO DE ROTEAMENTO E ALOCAÇÃO DE ESPECTRO *FIRST-FIT* E *LAST-FIT* MODIFICADO COM PROTEÇÃO COMPARTILHADA DE CAMINHO (RSA_FFMLF_SPP)

Baseado no algoritmo proposto em [24], porém sem a técnica de seleção da modulação do sinal, esse algoritmo utiliza o método FF *na seleção dos slots de frequência para os caminhos primários e* MLF *na seleção dos caminhos de proteção (para mais detalhes, veja a seção 5.3). A principal diferença entre esse algoritmo e o* RSA_SF_SPP *é que os custos dos slots de proteção, sejam livres ou compartilhados, são os mesmos. O cálculo do custo dos slots de proteção segue a Fórmula* 8.1. *A constante c1 multiplica a diferença entre o valor total de slots de frequência do enlace (FS_L) e o índice do primeiro slot de frequência (FS_1) a ser ocupado pelo caminho de proteção. Essa constante influencia no custo daquele slot a ser ocupado. No artigo [24], o valor de c1 é 0,34. Note que, quanto maior o índice da faixa espectral a ser utilizada, menor o seu custo. Desse modo, o algoritmo reúne os caminhos de proteção na porção final do espectro.*

A segunda constante, c2, multiplica o tamanho (em número de slots) da requisição. No artigo [24], c2 pode ser 0 ou 1, indicando se o tamanho da requisição afeta ou não seu custo. Utilizaremos c2 igual a 1, embora seu valor influencie muito pouco no resultado final do algoritmo, como foi demonstrado em [24].

$$FS_{custo} = (FS_L - FS_1).c_1 + SW_{size}.c_2 \qquad 8.1$$

9 RESULTADOS E DISCUSSÕES

Os algoritmos heurísticos citados foram submetidos a simulações no MATLAB, utilizando as redes NSFNET [23] (grau nodal 3, com comprimento dos enlaces em km) da Figura 9-1 e COST239 [23] (grau nodal 4,7, com comprimento dos enlaces em km) da Figura 9-2. Como especificado na seção 6.2, o tráfego gerado segue a distribuição de Poisson e o tempo de duração das conexões segue uma distribuição exponencial com valor médio unitário $1/\mu=1$.

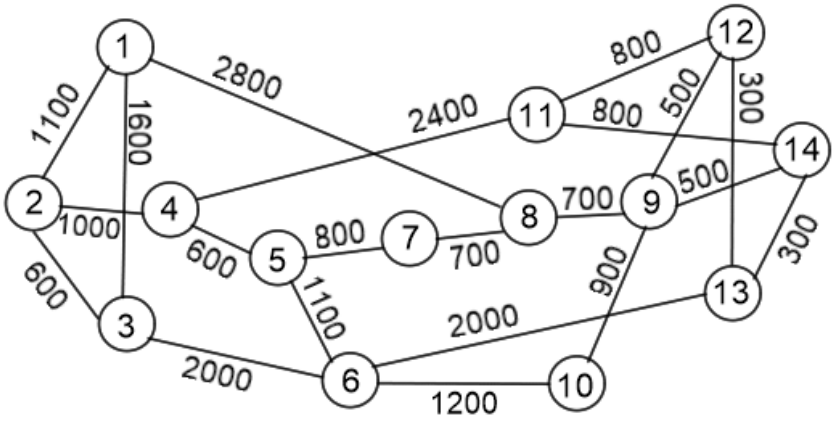

Figura 9-1: Rede NSFNET, 14 nós e 21 enlaces bidimensionais (adaptada de[22]).

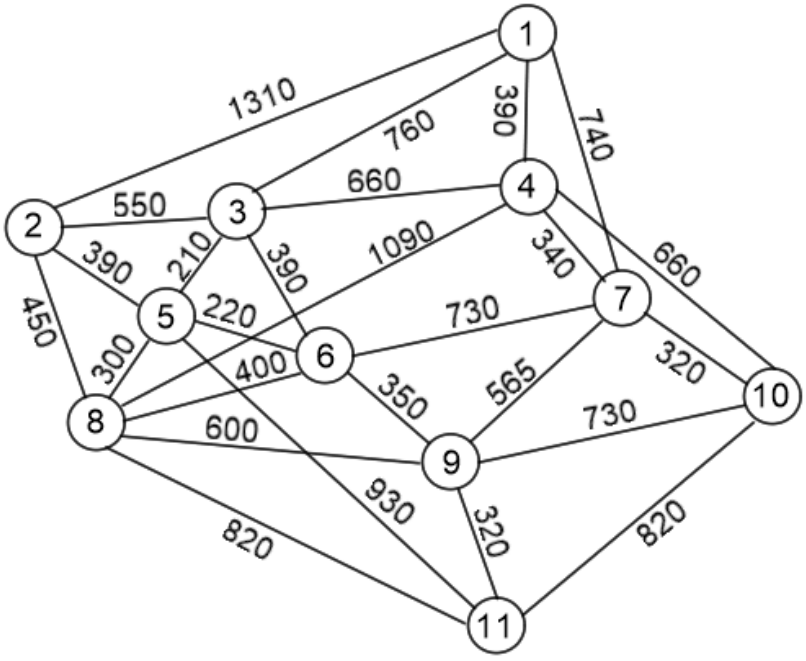

Figura 9-2: Rede COST239, 11 nós e 26 enlaces bidimensionais (adaptada de[22]).

9.1 COMPARAÇÃO ENTRE DIFERENTES ALGORITMOS

O algoritmo criado neste trabalho, RSA_SF_SPP (Capítulo 0), em suas versões de alocação espectral do caminho primário LSS e HSS foi comparado aos algoritmos RSA_FF_SPP (seção 8.1) e RSA_FFMLF_SPP (seção 8.2). Os valores das constantes utilizadas nas simulações, para cada rede, estão na Tabela 9-1. Para cada valor de carga na rede, foram realizadas 10 simulações com tráfego aleatório e os gráficos gerados utilizam a média dos valores obtidos nas simulações.

Tabela 9-1: Valores das constantes utilizadas nas simulações.

Constantes	Símbolo	Valor (NSFNET)	Valor (COST239)
Número de conexões simuladas	NC_{Total}	10000	10000
Número de subportadoras (*slots*) por enlace	FS_L	100	100
Banda de guarda (*slots*)	G	1	1
Número de caminhos primários procurados	num_paths	3	3
Número de caminhos de proteção procurados	num_bpaths	3	3
Número de subportadoras requeridas por conexão	Rsub	[1,5]	[1,5]
Custo dos *slots* de frequência para caminho primário do algoritmo RSA_SF_SPP	α	100	500
Custo dos *slots* de frequência compartilhados para caminho de proteção do algoritmo RSA_SF_SPP	β	100	500
Custo dos *slots* de frequência livres para caminho de proteção do algoritmo RSA_SF_SPP	γ	300	1500
Constantes de custo do algoritmo RSA_FFMLF_SPP	c1 e c2	0,34 e 1	0,34 e 1

9.1.1 Probabilidade de bloqueio e probabilidade de bloqueio de banda

Para diferentes valores de carga da rede, comparou-se a probabilidade de bloqueio e probabilidade de bloqueio de banda dos algoritmos nas redes NSFNET e COST239.

Para a rede NSFNET, o algoritmo RSA_FF_SPP obteve os piores resultados, com maiores valores de PB (Figura 9-3) e PBB (Figura 9-4). Isso é esperado devido à simplicidade do algoritmo, em que nenhum esforço é feito no sentido de diminuir a fragmentação da rede ou incentivar o aumento do grau de compartilhamento. O algoritmo RSA_FFMLF_SPP, obteve bons resultados, principalmente para altos valores de carga, mas sua performance para cargas menores é pior do que ambas as versões do algoritmo desenvolvido RSA_SF_SPP. A separação dos *slots* de caminhos primários na porção final do espectro e os *slots* de proteção no início (versão HSS) melhora o desempenho do algoritmo em relação a quantidade de conexões bloqueadas.

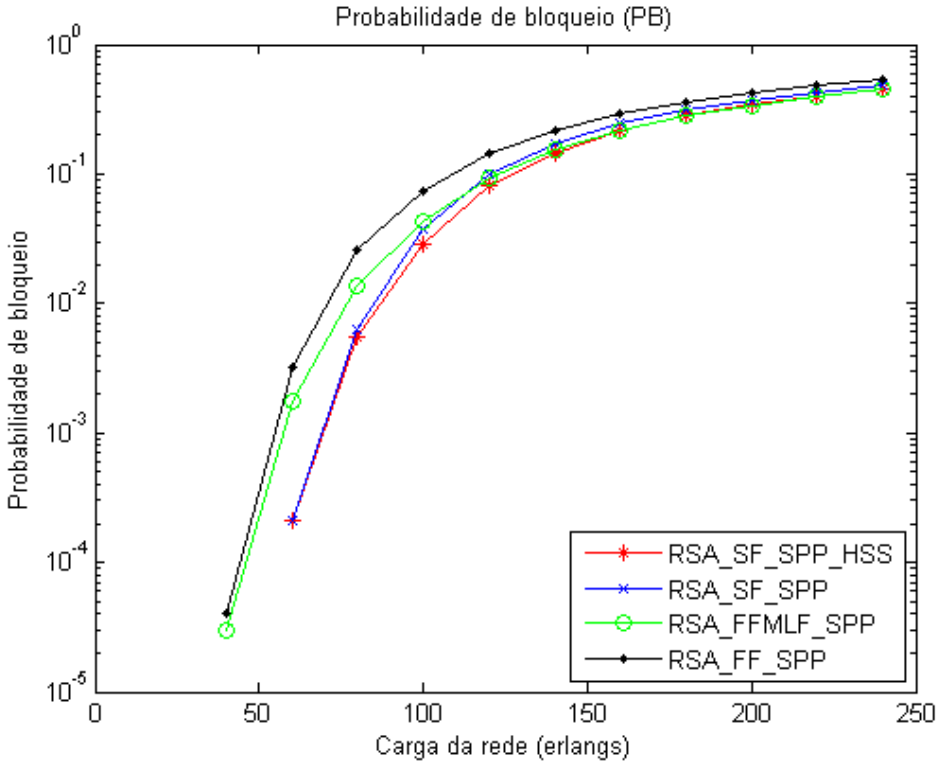

Figura 9-3: Probabilidade de bloqueio para rede NSFNET.

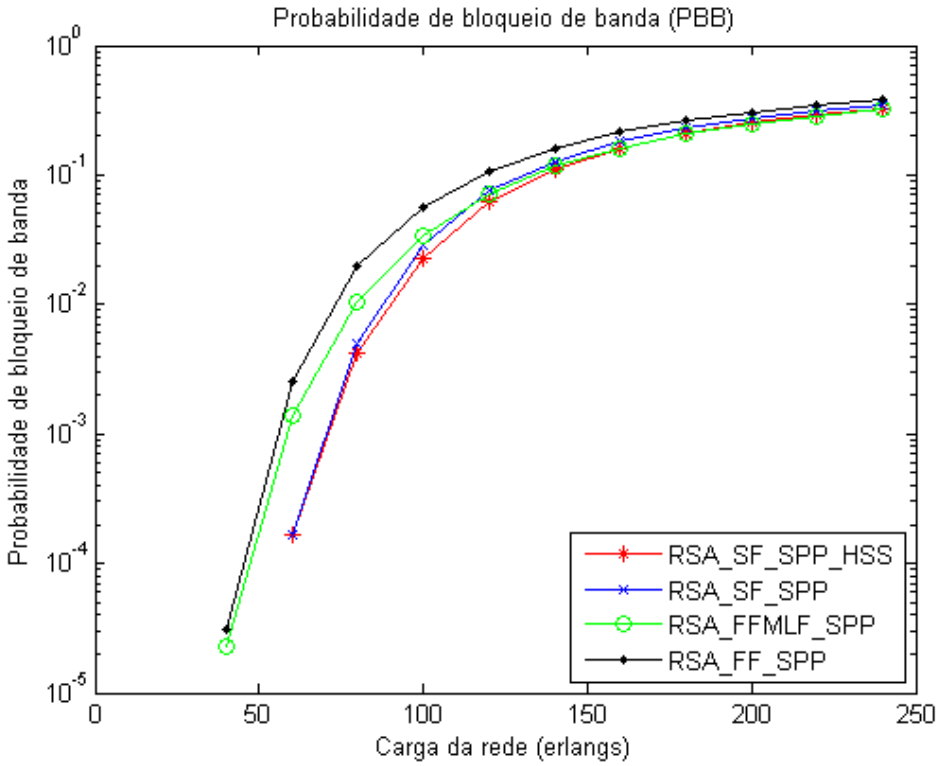

Figura 9-4: Probabilidade de bloqueio de banda para rede NSFNET.

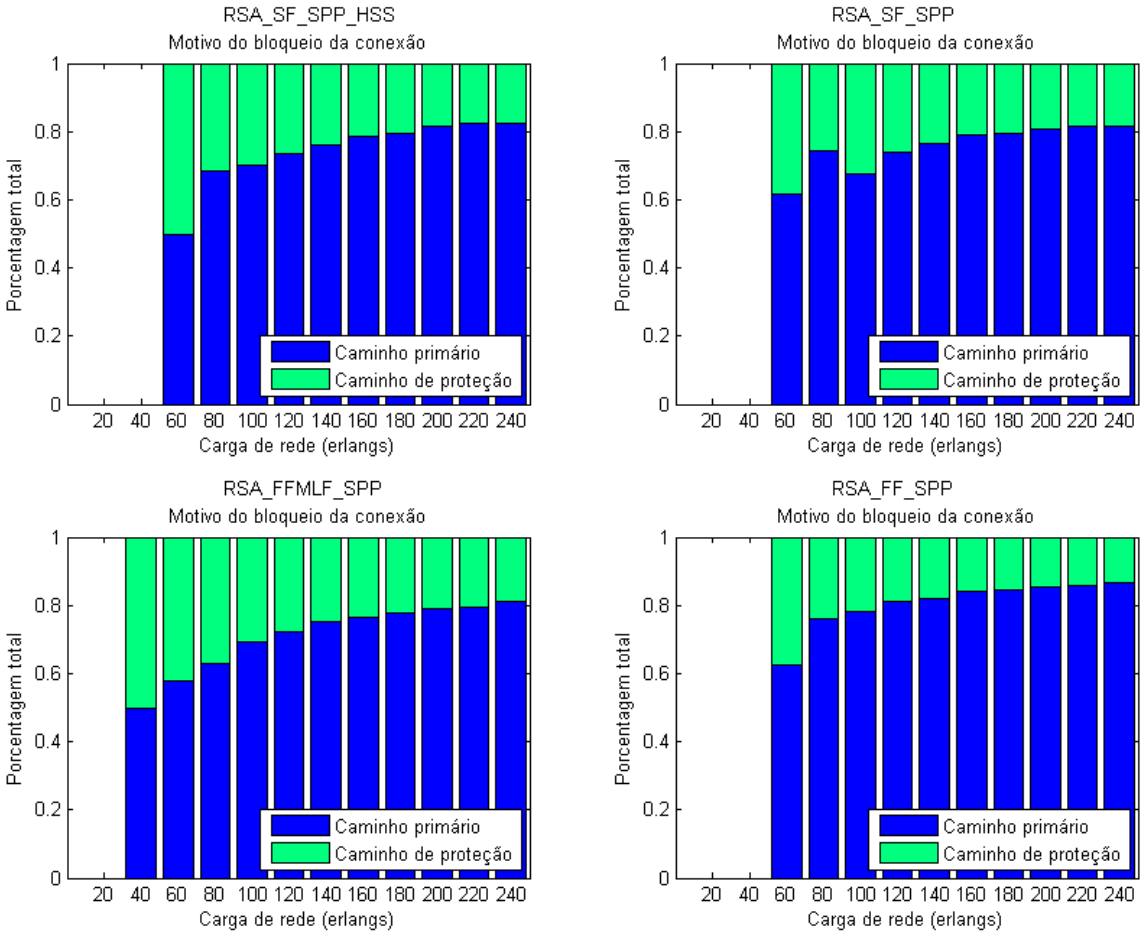

Figura 9-5: Motivo de bloqueio da conexão para rede NSFNET.

A Figura 9-5 mostra o motivo dos bloqueios ocorridos durante as simulações dos algoritmos para diferentes valores de carga na rede NSFNET. Como dito anteriormente, uma conexão é bloqueada se o algoritmo não localizar uma rota ou *slots* de frequência suficientes para o caminho primário ou de proteção. Conforme a carga da rede aumenta, o comportamento dos algoritmos converge para um valor similar, com cerca de 80% dos bloqueios ocorrendo devido a impossibilidade de a rede suportar mais uma conexão ativa. Na rede NSFNET, para 240 E, o algoritmo RSA_FF_SPP obteve a pior porcentagem de bloqueio devido à falha ao encontrar um caminho primário para conexões (86,5%), seguido pelos algoritmos RSA_SF_SPP versão HSS (82,4%), RSA_SF_SPP (81,6%) e RSA_FFMLF_SPP (81%).

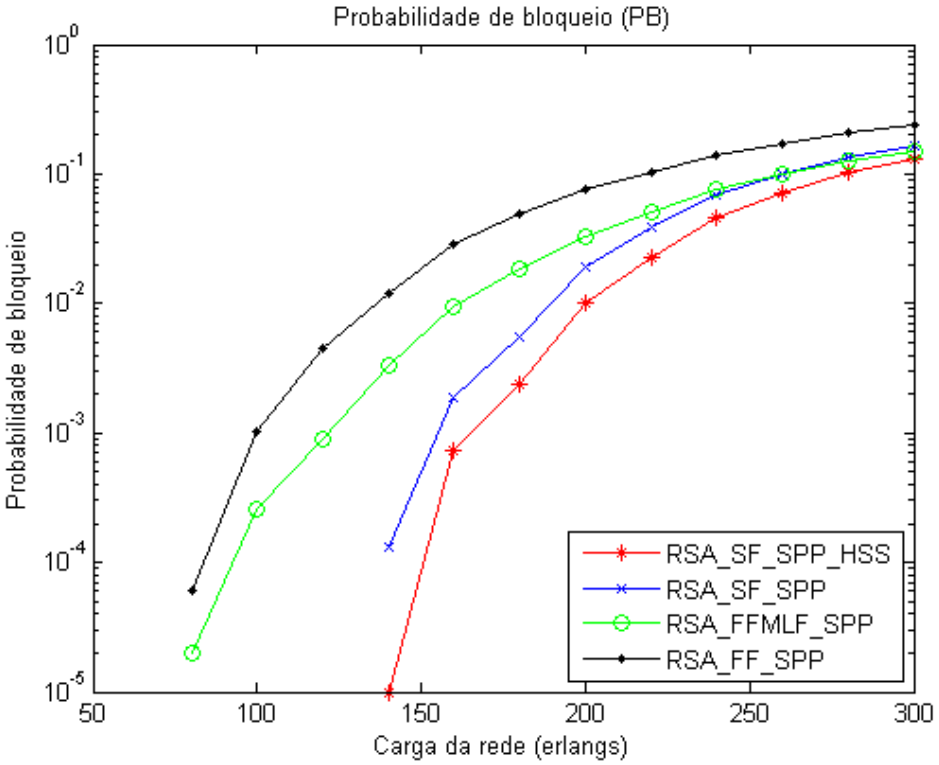

Figura 9-6: Probabilidade de bloqueio para rede COST239.

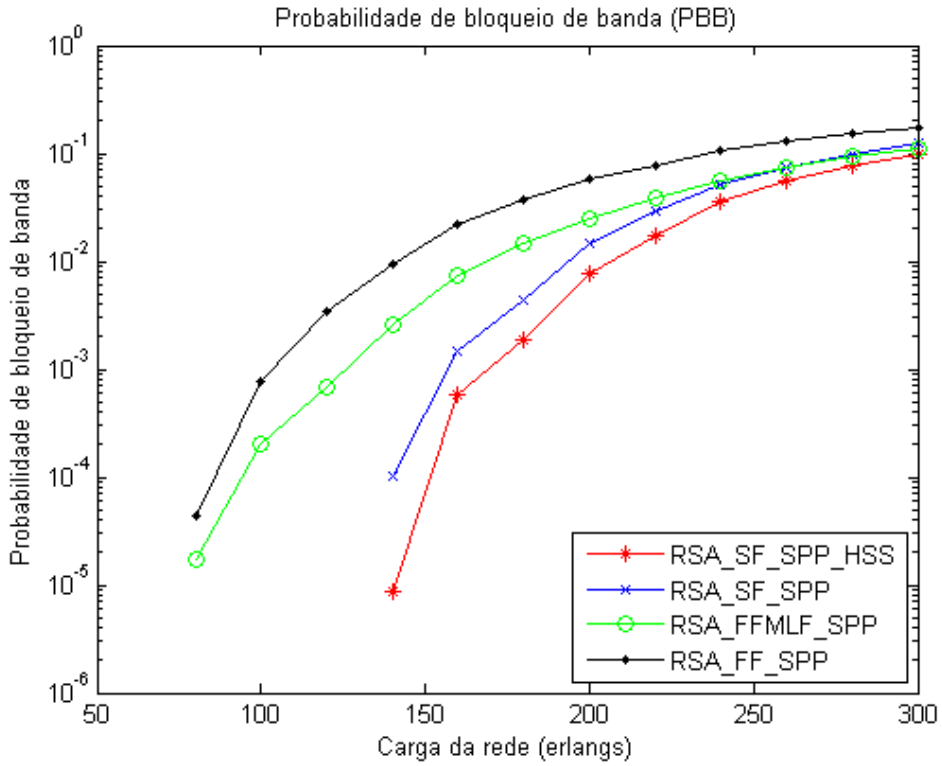

Figura 9-7: Probabilidade de bloqueio de banda para rede COST239.

Para a rede COST239, Figura 9-6 e Figura 9-7, as probabilidades de bloqueio e bloqueio de banda dos algoritmos avaliados diferem mais entre sí. Novamente, o algoritmo RSA_FF_SPP obteve os maiores valores de PB e PBB. Nessa rede, com maior grau nodal que a NSFNET, pode-se notar o melhor desempenho das versões LSS e HSS do algoritmo RSA_SF_SPP em relação ao RSA_FFMLF_SPP. Apenas para valores de carga superiores a 220 E os algoritmos RSA_FFMLF_SPP e RSA_SF_SPP possuem valores semelhantes de PB e PBB, mas a versão HSS do algoritmo RSA_SF_SPP supera os demais. Como a rede COST239 possui maior grau nodal que a NSFNET, há mais possibilidades de existirem caminhos primário e de proteção, permitindo que o algoritmo distribua melhor as conexões por toda a rede e, consequentemente, diminuindo o numero de conexões bloqueadas.

Em relação ao motivo dos bloqueios ocorridos, na Figura 9-8, para 300 E, o algoritmo RSA_FF_SPP bloqueou 89,9% das conexões por não encontrar um caminho primário para transmissão. O algoritmo RSA_FFMLF_SPP bloqueou 84,3% das conexões pelo mesmo motivo, seguido pelo algoritmo RSA_SF_SPP em sua versão LSS, com 83,8%. Na versão HSS do algoritmo desenvolvido em 85% das conexões bloqueadas não foi encontrado caminho primário para transmissão. Embora seja

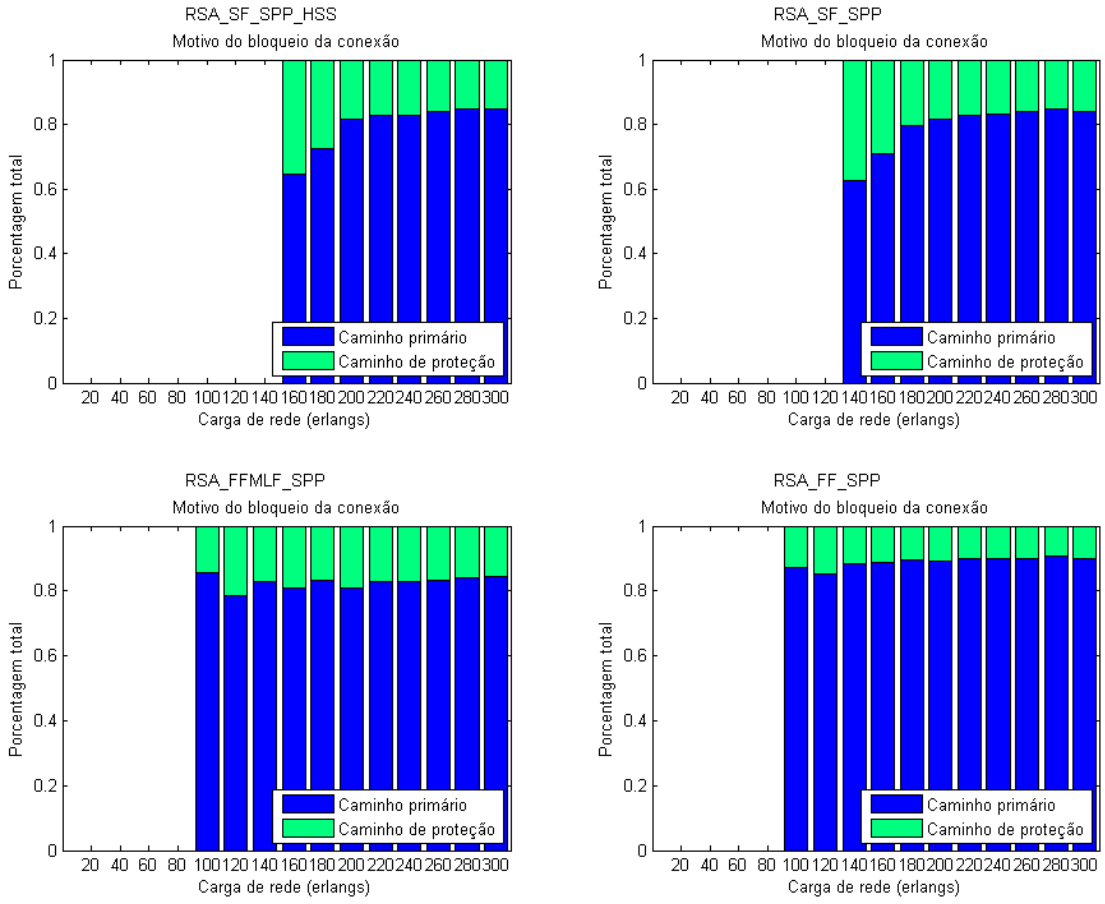

Figura 9-8: Motivo de bloqueio da conexão para rede COST239.

um valor mais alto que aquele da versão LSS, é importante lembrar que este foi o algoritmo com menor PB e PBB, indicando que a rede foi melhor utilizada.

9.1.2 Grau de compartilhamento

O grau de compartilhamento indica a capacidade do algoritmo em encontrar espectros de proteção a serem compartilhados e a quantidade de recursos da rede destinados a proteção que são economizados. No pior caso, para *Shar=1*, todos os *slots* de proteção requeridos foram alocados na rede e não há compartilhamento espectral, como em proteção dedicada. Logo, quanto maior esse valor, menor é a quantidade de *slots* de proteção reservados na rede em relação à quantidade requerida pelas conexões.

O grau de compartilhamento cresce conforme a carga da rede aumenta, pois com o número maior de conexões simultâneas, maiores são as possibilidades de compartilhamento de *slots*.

Na Figura 9-9, temos as simulações realizadas na rede NSFNET. Com o algoritmo RSA_FF_SPP, para valores baixos de carga, o grau de compartilhamento é apenas 1,1 e para cargas altas esse valor é de 1,56, enquanto o algoritmo RSA_SF_SPP obteve entre 1,21 e 1,71 e, na versão HSS, entre 1,26 e 1,85. O algoritmo RSA_FFMLF_SPP obteve valores entre 1,26 e 1,96. Note que o baixo grau de compartilhamento do algoritmo RSA_FF_SPP condiz com os resultados obtidos anteriormente em relação à alta probabilidade de bloqueio, pois o algoritmo é menos eficiente, consumindo mais espectro com proteção.

O algoritmo RSA_SF_SPP não possui grau de compartilhamento tão alto quanto o RSA_FFMLF_SPP, pois seu objetivo é buscar o equilíbrio entre o custo de compartilhar um *slot* de proteção ou utilizar outro de menor índice, mesmo que ainda livre. O algoritmo RSA_FFMLF_SPP, como esperado, prioriza o compartilhamento espectral ao separar os *slots* de frequência de caminhos primário e de proteção em porções diferentes do espectro. A separação dos recursos de proteção e primário é essencial para incentivar o compartilhamento espectral, como pode ser visto ao comparar as versões LSS e HSS do algoritmo RSA_SF_SPP.

Para a rede COST239, na Figura 9-10, o desempenho do algoritmo RSA_SF_SPP na versão HSS supera todos os demais, com valores entre 1,36 e 2,15. A versão LSS do algoritmo RSA_SF_SPP obtém valores entre 1,27 e 1,89. Seu desempenho é semelhante ao do algoritmo RSA_FFMLF_SPP até 150 E, mas depois é superado. Os valores obtidos do algoritmo RSA_FFMLF_SPP estão entre 1,23 e 2,07.

Novamente, pode-se ver a importância na separação em porções diferentes do espectro dos *slots* destinados a proteção e a caminhos primários.

O algoritmo RSA_FF_SPP obteve os piores valores de grau de compartilhamento, entre 1,09 e 1,49, indicando que o algoritmo não prioriza o compartilhamento espectral entre caminhos de proteção.

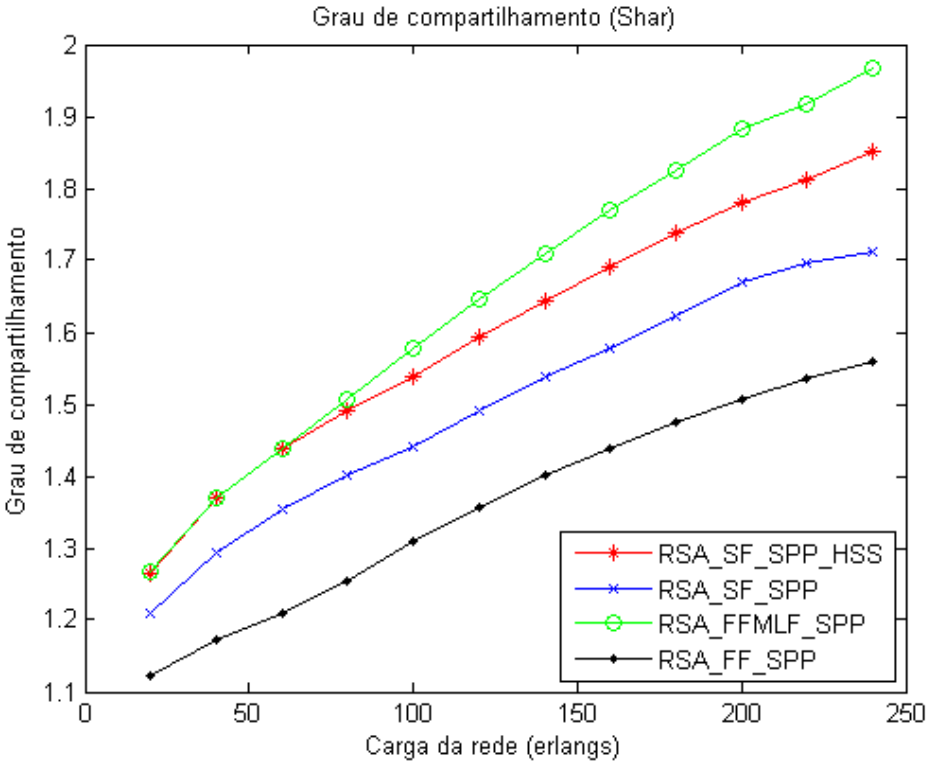

Figura 9-9: Grau de compartilhamento para rede NSFNET.

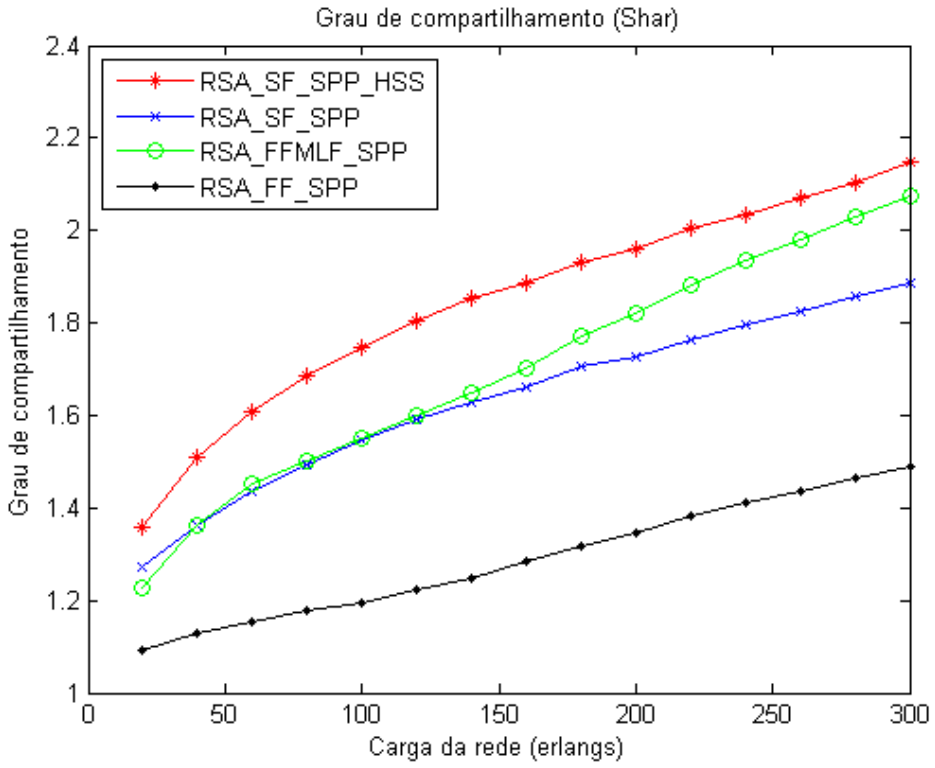

Figura 9-10: Grau de compartilhamento para rede COST239.

9.1.3 **Fragmentação espectral**

Avaliando o desempenho dos algoritmos em relação à fragmentação espectral média da rede, o algoritmo RSA_FF_SPP obteve pior nível de fragmentação em ambas as redes (Figura 9-11 e Figura 9-12). O algoritmo RSA_FFMLF_SPP organiza melhor os *slots* de proteção e primários na rede, diminuindo a fragmentação espectral. No algoritmo RSA_SF_SPP, versão LSS, os *slots* de proteção e primários ficam misturados e com a chegada e saída das conexões o espectro se torna mais fragmentado. Esse problema diminui com a versão HSS, que supera o desempenho do RSA_FFMLF_SPP. Tais resultados indicam que a separação dos caminhos de proteção e primário em porções diferentes do espectro ajuda a diminuir a fragmentação espectral.

Como o algoritmo RSA_SF_SPP busca equilibrar a utilização dos *slots* de proteção em relação ao seu índice e à distância percorrida, de acordo com os custos envolvidos, o algoritmo pode selecionar a opção de ocupar um *slot* livre de menor índice mesmo que haja a possibilidade de compartilhamento em outra posição do espectro. Isso diminui ainda mais a fragmentação espectral da rede.

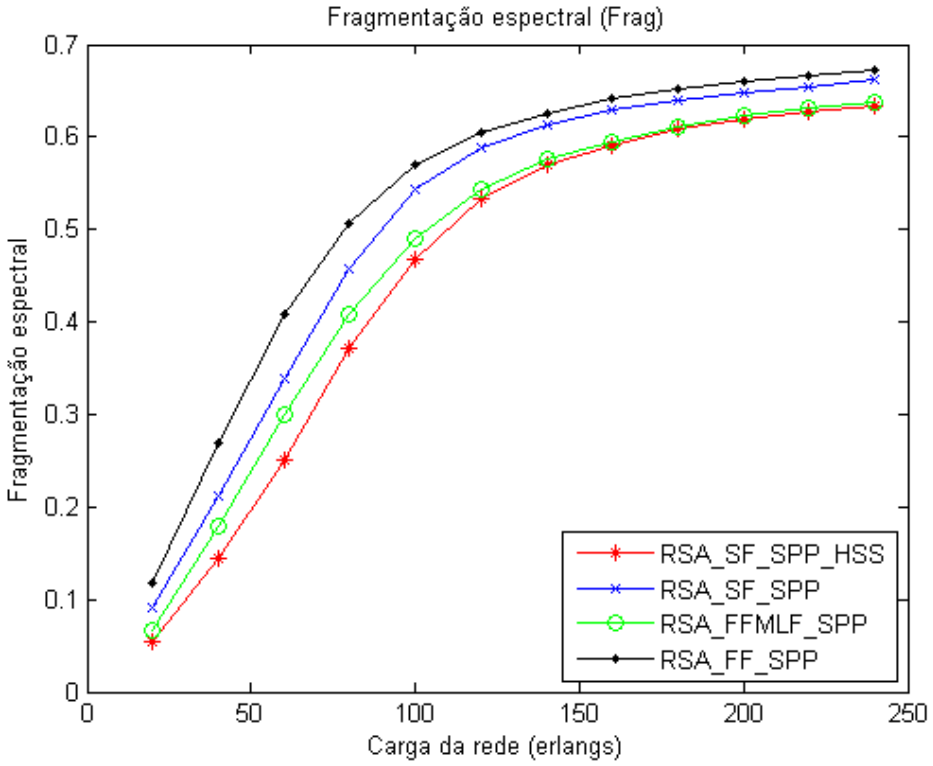

Figura 9-11: Fragmentação espectral para rede NSFNET.

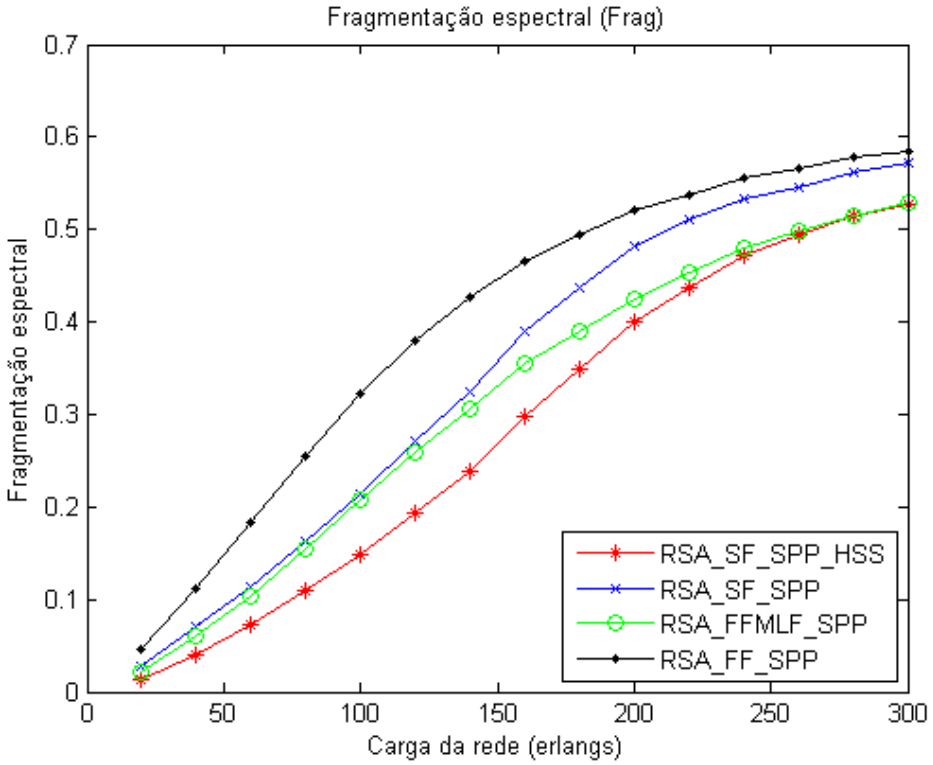

Figura 9-12: Fragmentação espectral para rede COST239.

9.1.4 **Taxa de utilização espectral**

A SUR mostra a relação entre o número de *slots* utilizados da rede em relação ao total. Entretanto, analisar apenas a porcentagem de ocupação da rede com determinado algoritmo não indica seu desempenho. É necessário observar também os gráficos de probabilidade de bloqueio e grau de compartilhamento.

Para valores baixos de carga (aproximadamente até 100 E na Figura 9-13 e 180 E na Figura 9-14), o algoritmo com menor SUR é a versão HSS do RSA_SF_SPP, seguido pelos algoritmos RSA_FFMLF_SPP, RSA_SF_SPP (versão LSS) e por último o RSA_FF_SPP.

Embora todos os algoritmos foram capazes de ocupar os *slots* disponíveis na rede, o algoritmo RSA_FF_SPP obteve pior desempenho em relação ao modo que esses recursos foram utilizados, principalmente em relação ao compartilhamento de recursos de proteção, consequentemente levando a alta probabilidade de bloqueio.

A SUR do algoritmo RSA_FFMLF_SPP é menor que a dos demais para altas cargas, utilizando menos recursos da rede, principalmente recursos de proteção, como indicado pelo alto grau de compartilhamento obtido.

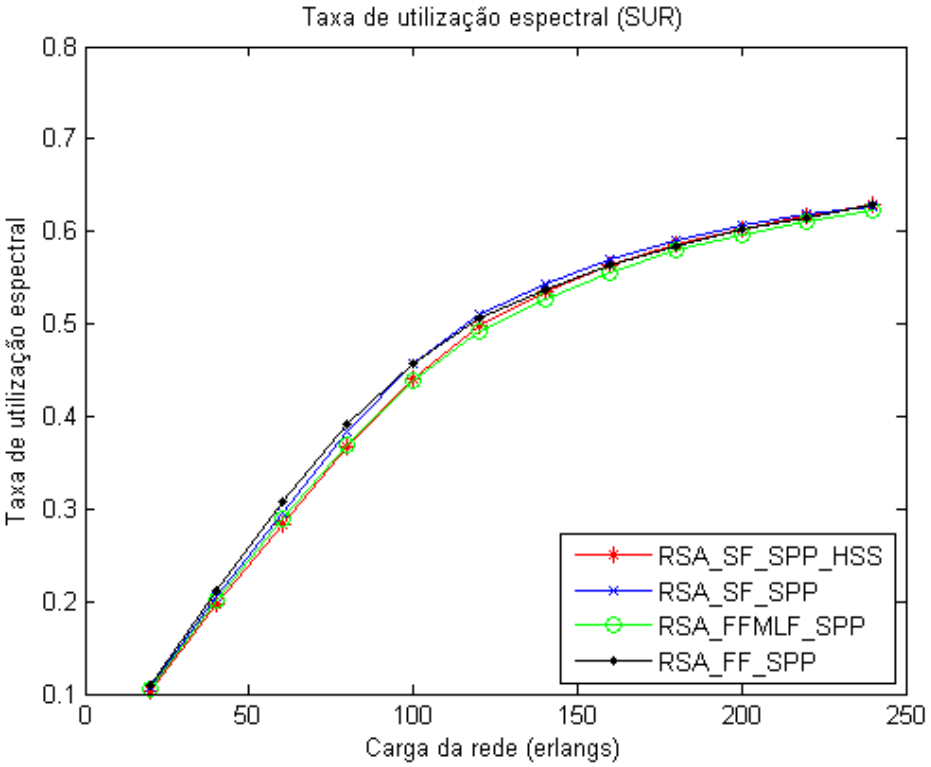

Figura 9-13: Taxa de utilização espectral para rede NSFNET.

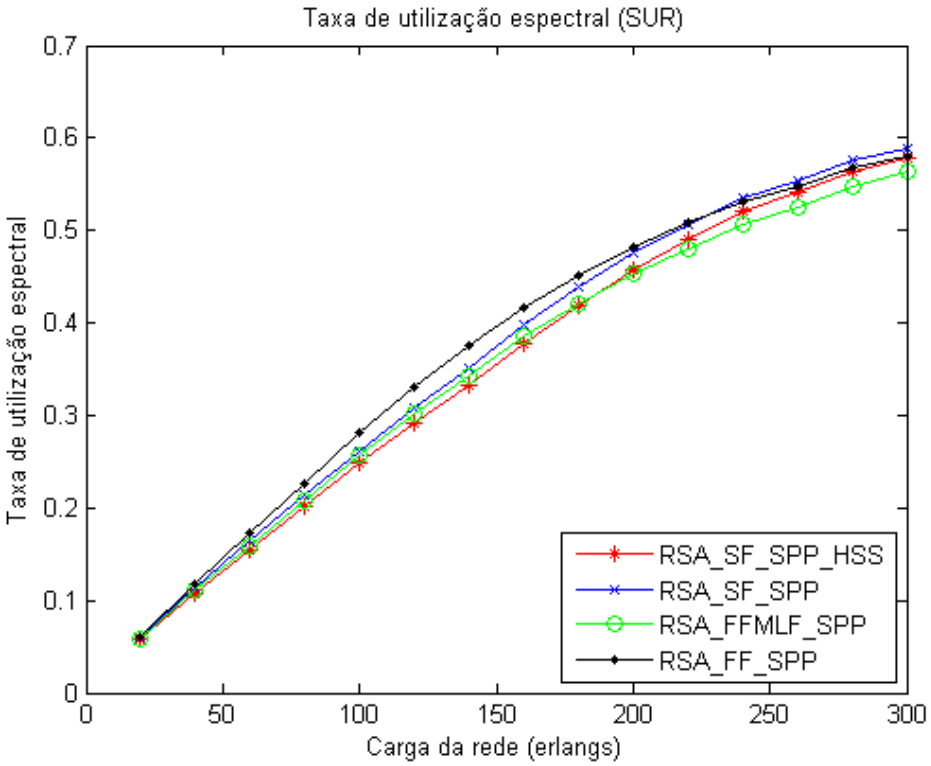

Figura 9-14: Taxa de utilização espectral para rede COST239.

Entretanto, na rede COST239, Figura 9-14, mesmo consumindo menos recursos da rede do que a versão HSS do algoritmo RSA_SF_SPP, o algoritmo RSA_FFMLF_SPP obteve maior probabilidade de bloqueio e maior fragmentação espectral. Isso indica que os recursos utilizados estavam mal organizados na rede, aumentando o número de conexões bloqueadas e consequentemente diminuindo a SUR.

Também para altas cargas, o alto valor de utilização espectral do algoritmo RSA_SF_SPP (em ambas as versões) não indica o mau uso de recursos da rede. Como o algoritmo obteve as menores taxas de bloqueio, a alta SUR indica que a rede possui maior número de conexões ativas.

9.1.5 **Taxa de capacidade reservada**

Como a SCR indica a relação entre o número de *slots* de proteção e o número de *slots* primários da rede, esse valor tende a ser maior que um, pois, geralmente, o caminho primário escolhido para uma conexão possui menos enlaces do que o caminho de proteção selecionado. O compartilhamento espectral tem o objetivo de diminuir essa relação, economizando recursos da rede. Conforme o valor da carga aumenta, mais conexões estão ativas simultaneamente e maiores são as possibilidades de compartilhamento. Desse modo, é esperado que o valor da SCR diminua para altas cargas.

Na Figura 9-15, para rede NSFNET, note que o algoritmo RSA_FF_SPP possui os maiores valores de SCR, indicando o baixo grau de compartilhamento e alto gasto de recursos da rede com a proteção de suas conexões. Isso leva também à alta probabilidade de bloqueio. O algoritmo RSA_FFMLF_SPP, novamente obteve bons resultados, mas foi superado pelo RSA_SF_SPP em sua versão HSS. Novamente, comparando as duas versões do algoritmo desenvolvido, pode-se ver que a separação de caminhos primário e de proteção em porções diferentes do espectro é essencial no desempenho do algoritmo, principalmente para cargas acima de 120 E, onde a SCR obteve valores menores do que um, indicando que há mais *slots* na rede utilizados como primário do que reservados para proteção. Para 240 E, o algoritmo RSA_SF_SPP, versão HSS, obteve SCR 0,92, valor próximo de 0,94, obtido pelo RSA_FFMLF_SPP.

O mesmo comportamento pode ser observado nas simulações da rede COST239, da Figura 9-16. Os algoritmos RSA_FFMLF_SPP e RSA_SF_SPP obtiveram melhor desempenho que o RSA_FF_SPP. Nessa rede, por possuir maior grau nodal e, consequentemente, mais possibilidades de caminhos distintos para as conexões, os recursos utilizados foram mais bem espalhados por toda a rede e os caminhos de proteção das conexões geralmente não possuíam mais enlaces que os caminhos primários. Desse modo, a SCR de todos os algoritmos possui valores menores do que os obtidos na rede NSFNET. Além disso, o alto grau de compartilhamento obtido pelo algoritmo RSA_SF_SPP versão HSS possibilitou o uso muito eficaz dos recursos destinados à proteção, com valor de SCR igual a 0,65 para 300 E em comparação a 0,71 obtido com o algoritmo RSA_FFMLF_SPP.

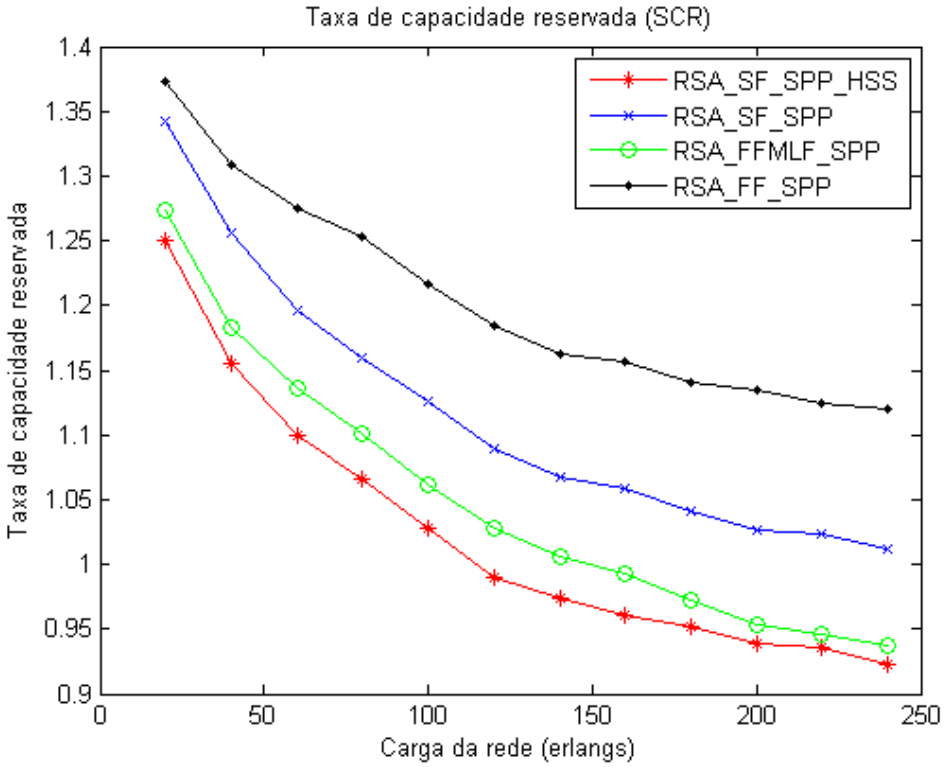

Figura 9-15: Taxa de capacidade reservada para rede NSFNET.

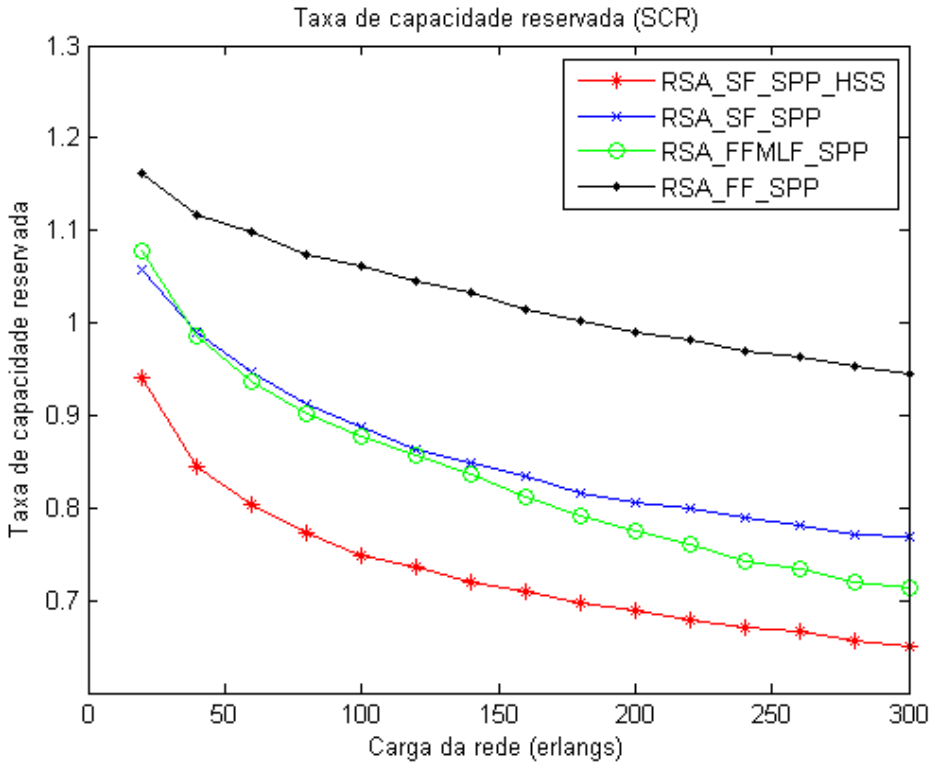

Figura 9-16: Taxa de capacidade reservada para rede COST239.

9.1.6 **Porcentagem de restauração**

Quando uma falha em um enlace ocorre, o algoritmo desenvolvido deve ser capaz de identificar as conexões afetadas pela falha e passar a transmitir seus dados através dos caminhos de proteção determinados para aquelas conexões. O caminho de proteção deve estar livre, ou seja, não pode estar em uso por nenhuma outra transmissão.

Quando todas as conexões afetadas pela falha são recuperadas com sucesso, então a porcentagem de recuperação é de 100%. Como todas as conexões aceitas possuem o caminho de proteção já definido e reservado, é esperado que esse resultado ocorra sempre. A falha ao recuperar uma conexão indica que há erros no algoritmo, pois este não foi capaz de reservar um caminho de proteção adequado para cada conexão. Em todas as simulações realizadas, após uma falha em um dos enlaces aleatórios da rede, e para diversos valores de carga, o algoritmo foi sempre capaz de recuperar com sucesso todas as conexões afetadas.

9.2 **DEFINIÇÃO DOS VALORES DE CUSTO DOS *SLOTS* DE FREQUÊNCIA**

O efeito dos valores de α, β e γ no algoritmo é essencial na tomada de decisão de qual *slot* de frequência será escolhido para aquela conexão. Muitos dos artigos estudados não citam quais valores de suas constantes foram utilizados nos cálculos de custo, tão pouco a motivação para escolha de tais valores. Outros, como [24], apenas fornecem o valor utilizado, mas não debatem sobre ele.

Neste trabalho, a motivação ao escolher o método K-SP na busca por rotas de transmissão não foi apenas reduzir a probabilidade de bloqueio de uma conexão caso não fossem encontrados *slots* livres no menor caminho. O método *smart-fit* deve avaliar todos os *slots* onde aquela conexão poderia ser alocada, em todas as rotas candidatas, e escolher o melhor deles. Para que isso seja possível, dependendo do custo de um *slot*, o algoritmo poderá escolher outro, de outra rota, com menor custo.

Como o comprimento da rota também influencia no cálculo do custo final, as constantes de custo dos *slots* deveriam ter valores tais que pudessem ser tão significativas quanto o tamanho dos enlaces percorridos.

Foram testados vários valores de α, β e γ e avaliou-se os resultados obtidos com cada um. Nessas simulações, Figura 9-17 até Figura 9-22, a rede testada é a NSFNET e o algoritmo RSA_SF_SPP não separa os *slots* de caminhos primários e de proteção em porções diferentes do espectro, ou seja, o caminho primário tem seus *slots* alocados utilizando o método LSS.

Para valores pequenos das constantes, como há apenas 100 *slots* de frequência em cada enlace, o custo desses *slots* sempre é menor que a distância de qualquer enlace da rede. O algoritmo geralmente não escolhe outro caminho mesmo se o *slot* disponível na rota atual tenha índice muito alto. Nesse caso, o custo dos *slots* tem menor peso em relação ao custo das rotas. Note que os piores resultados das simulações são para valores pequenos das constantes.

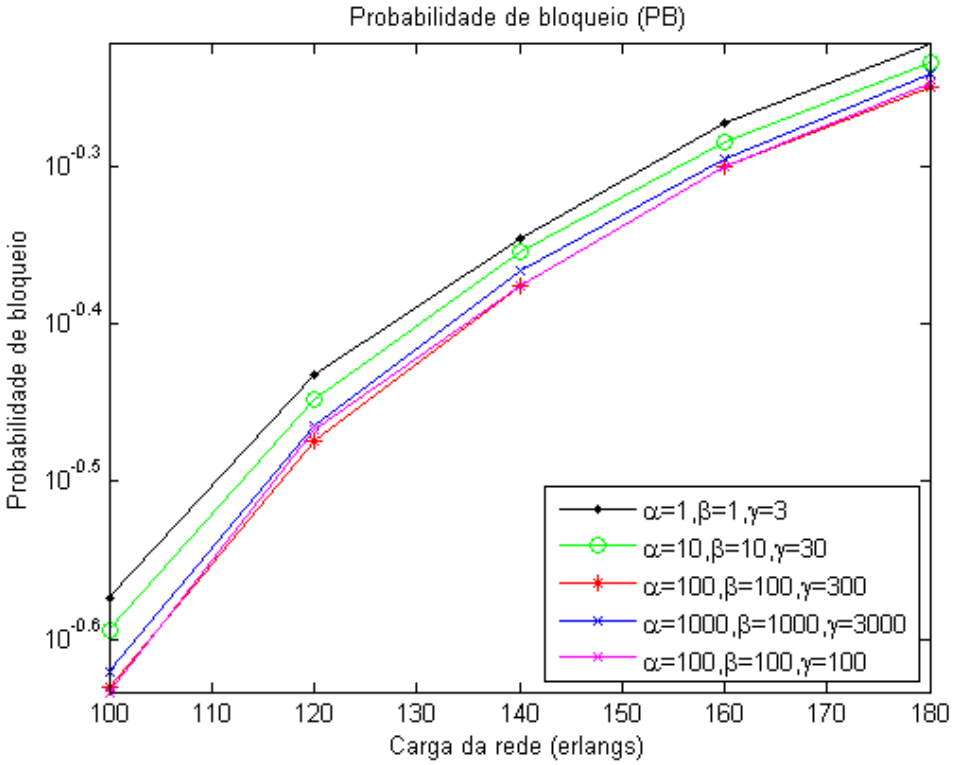

Figura 9-17: Probabilidade de bloqueio para diferentes valores de constantes para rede NSFNET.

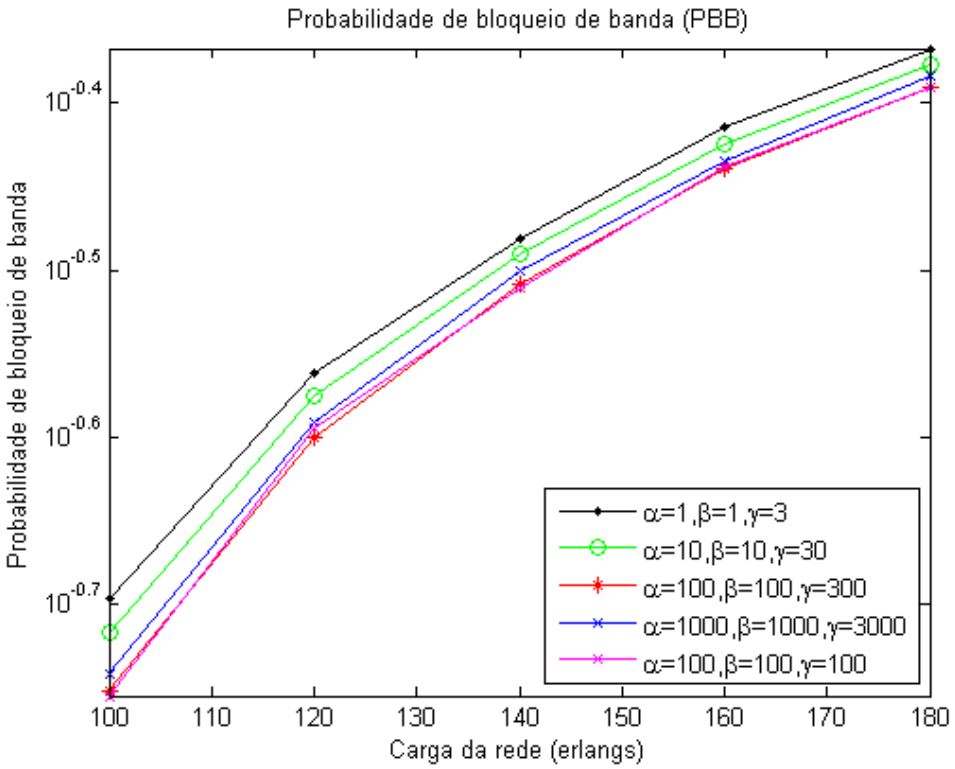

Figura 9-18: Probabilidade de bloqueio de banda para diferentes valores de constantes para rede NSFNET.

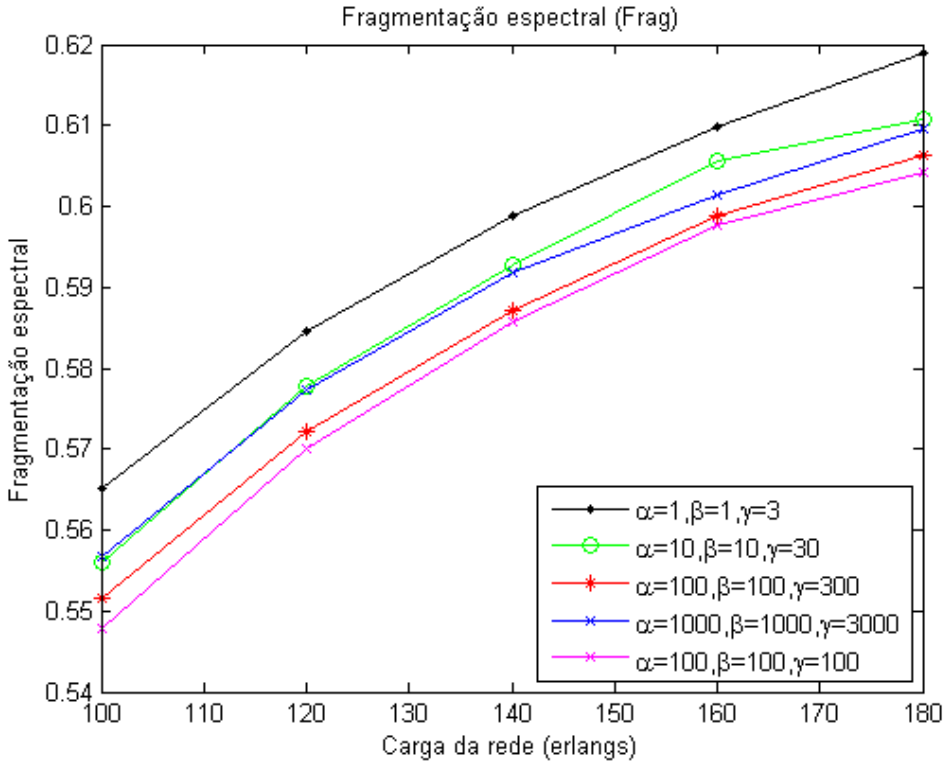

Figura 9-19: Fragmentação espectral para diferentes valores de constantes para rede NSFNET.

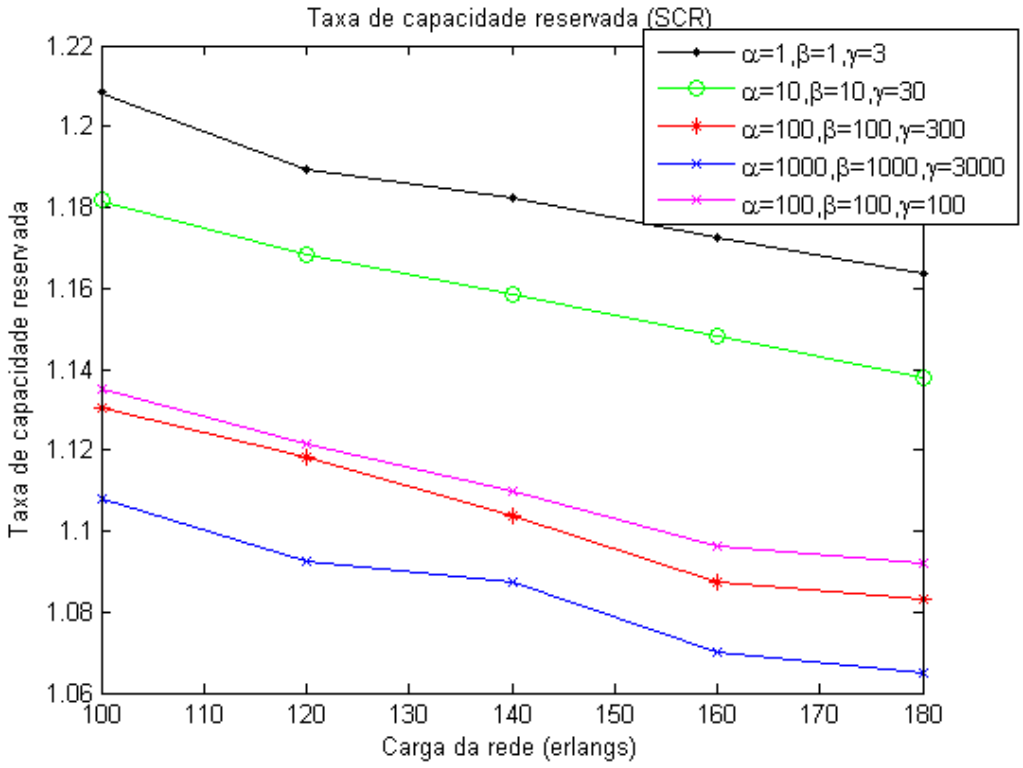

Figura 9-20: Taxa de capacidade reservada para diferentes valores de constantes para rede NSFNET.

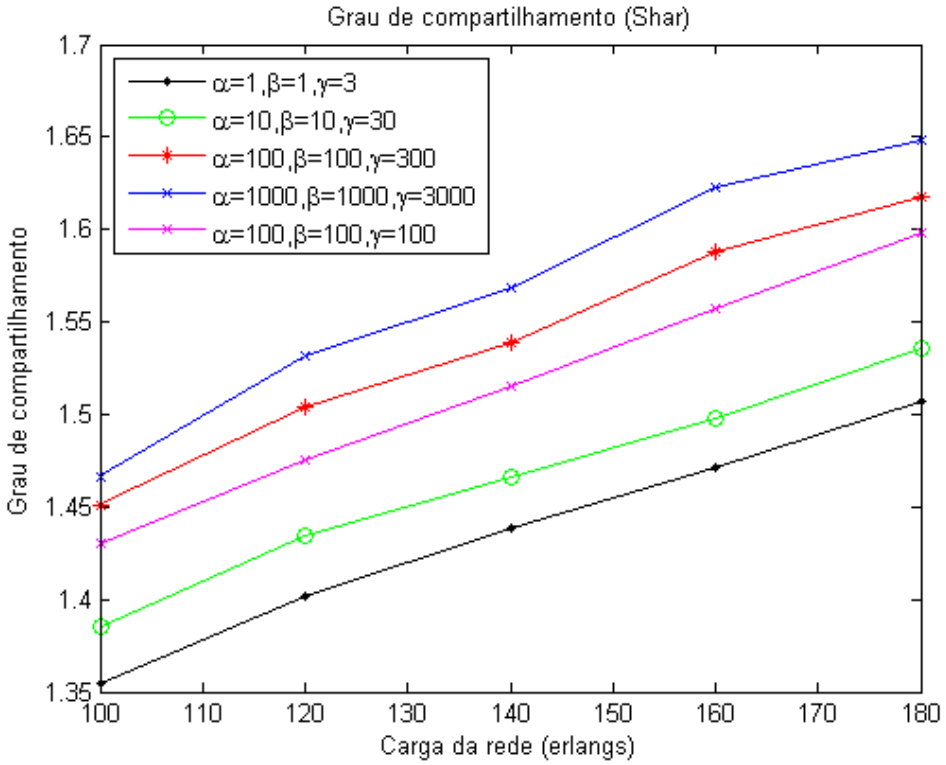

Figura 9-21: Grau de compartilhamento para diferentes valores de constantes para rede NSFNET.

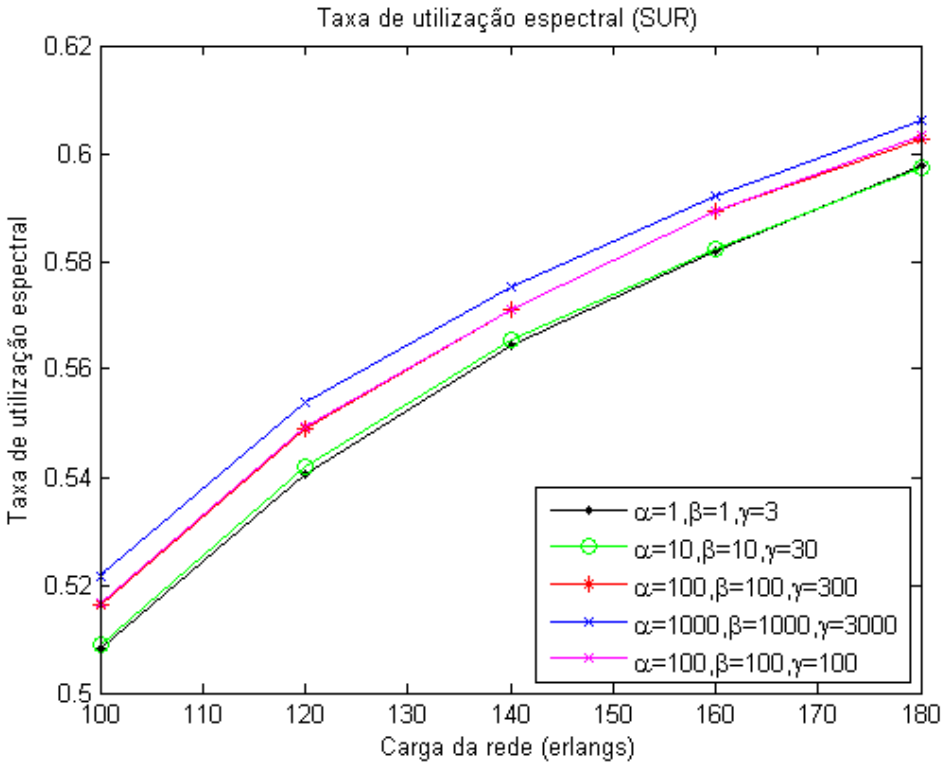

Figura 9-22: Taxa de utilização espectral para diferentes valores de constantes para rede NSFNET.

Para valores muito grandes das constantes, por outro lado, a posição do *slot* se torna muito mais significativa do que a rota escolhida. O algoritmo acaba escolhendo a rota em que o *slot* disponível tenha índice menor, mesmo que possua muitos enlaces. Nesse caso, o tamanho das rotas se torna menos significativo em relação ao custo dos *slots*. Como pode ser visto, tal prioridade melhora o grau de compartilhamento dos *slots*, mas aumenta a probabilidade de bloqueio de conexões.

Para a rede NSFNET, os melhores resultados possuem constantes α = 100, β = 100 e γ = 300. O valor da constante γ escolhido é três vezes maior que β, para desencorajar o uso de *slots* livres no caminho de proteção. Esses valores das constantes foram os que proporcionaram menor probabilidade de bloqueio de banda e menor fragmentação espectral da rede, pois equilibram a escolha das rotas e *slots*.

Para as simulações com α = 100, β = 100 e γ = 300 e α, β e γ = 100, note que o valor da constante γ maior que β incentiva de fato o compartilhamento espectral (Figura 9-21).

A rede NSFNET possui grau nodal 3 e o tamanho médio de seus enlaces é 1080,95 km. A partir do décimo *slot*, o algoritmo tende a priorizar a escolha dos *slots* de frequência de menor índice à escolha da rota. É importante ressaltar que essa rede possui menor número de rotas distintas que a COST239, pois possui menor grau nodal. Durante as simulações, notou-se que os nós 7 e 10, com apenas dois enlaces cada um, eram aqueles com maior índice de bloqueio de conexões, pois ficavam rapidamente com todos os *slots* utilizados.

Para a rede COST239, Figura 9-23 até Figura 9-28, os melhores resultados obtidos possuem constantes entre α = 100, β = 100 e γ = 300 e α = 1000, β = 1000 e γ = 3000. As simulações possuem resultados muito semelhantes, com valores da ordem de 100 obtendo menores probabilidades de bloqueio e valores da ordem de 1000 com melhor grau de compartilhamento e SCR. Por isso, buscando encontrar um equilíbrio entre todos os resultados, optou-se por utilizar constantes com valores α = 500, β = 500 e γ = 1500.

A rede COST239 possui maior grau nodal (4,7) e o tamanho médio de seus enlaces é de 578,65 km. O valor final definido das constantes prioriza a escolha dos *slots* de frequência à rota. É comum que um caminho de tamanho semelhante ou um pouco maior seja escolhido por possuir *slots* disponíveis de menor índice. Entretanto, o alto grau nodal da rede indica que esta oferece muitas possibilidades de rotas, distribuindo melhor a utilização dos *slots* por toda a rede.

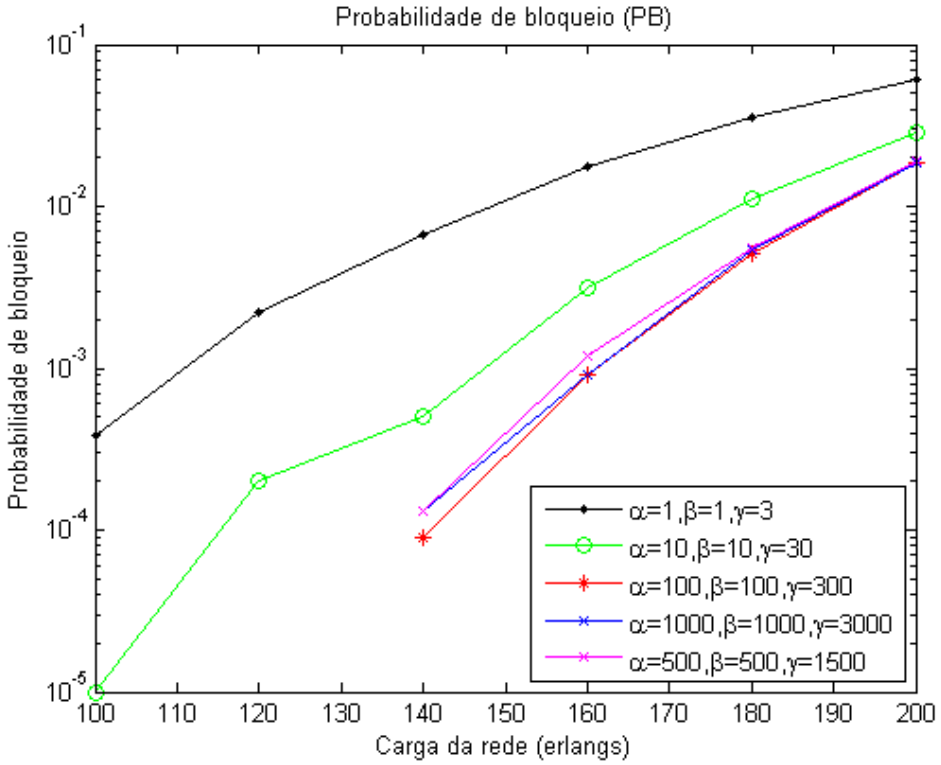

Figura 9-23: Probabilidade de bloqueio para diferentes valores de constantes para rede COST239.

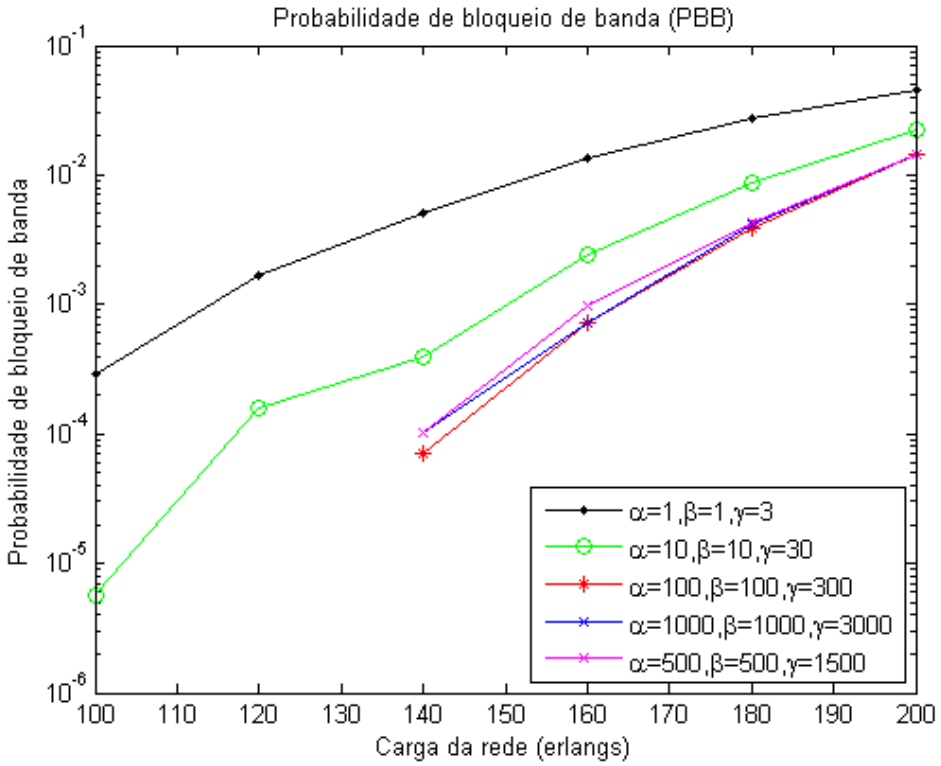

Figura 9-24: Probabilidade de bloqueio de banda para diferentes valores de constantes para rede COST239.

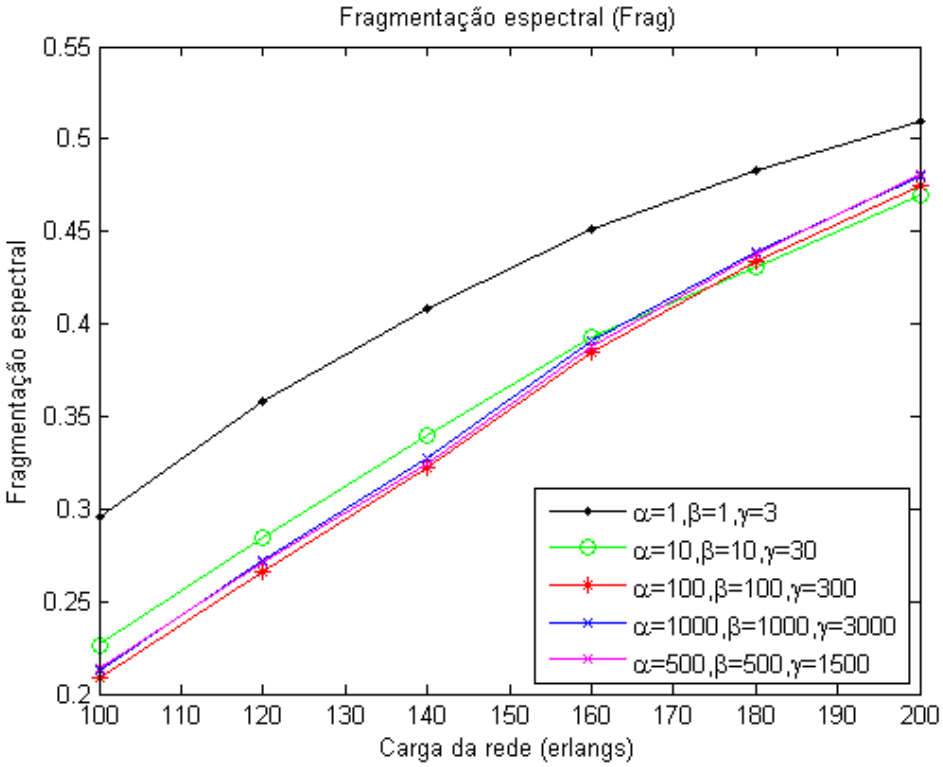

Figura 9-25: Fragmentação espectral para diferentes valores de constantes para rede COST239.

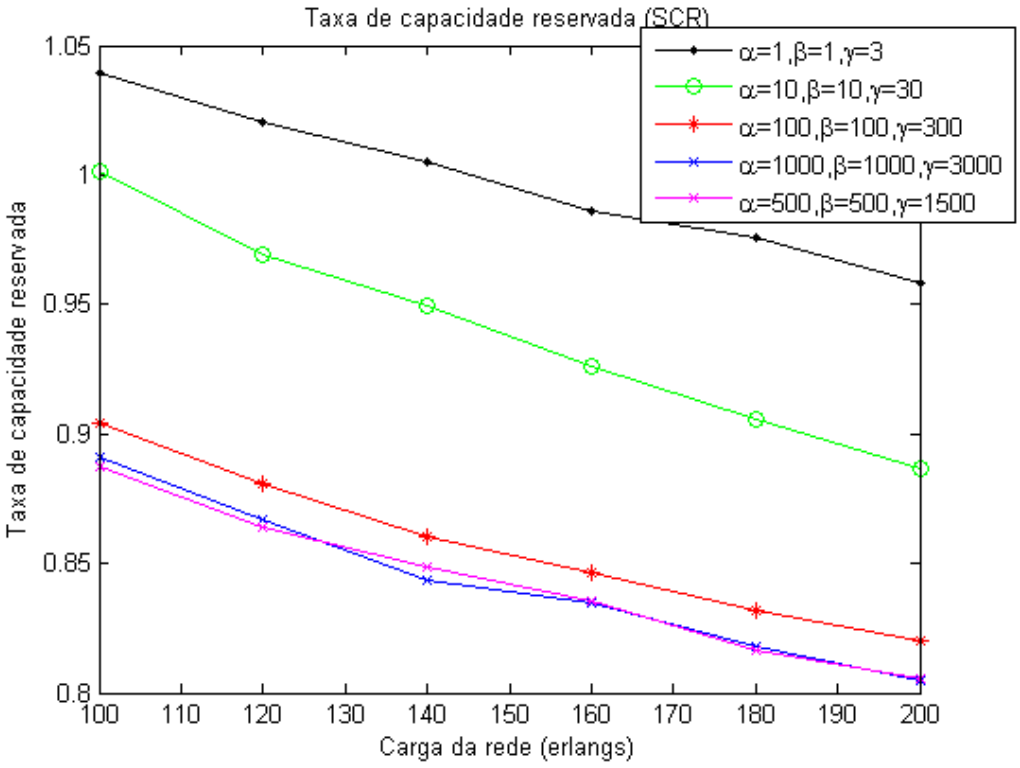

Figura 9-26: Taxa de capacidade reservada para diferentes valores de constantes para rede COST239.

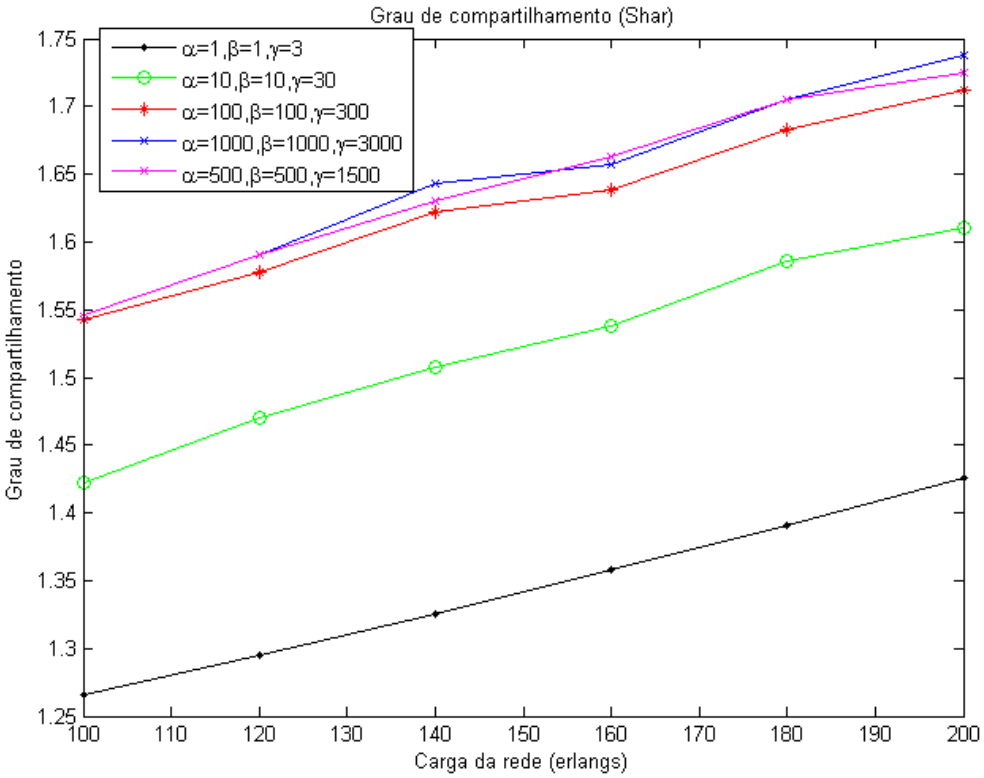

Figura 9-27: Grau de compartilhamento para diferentes valores de constantes para rede COST239.

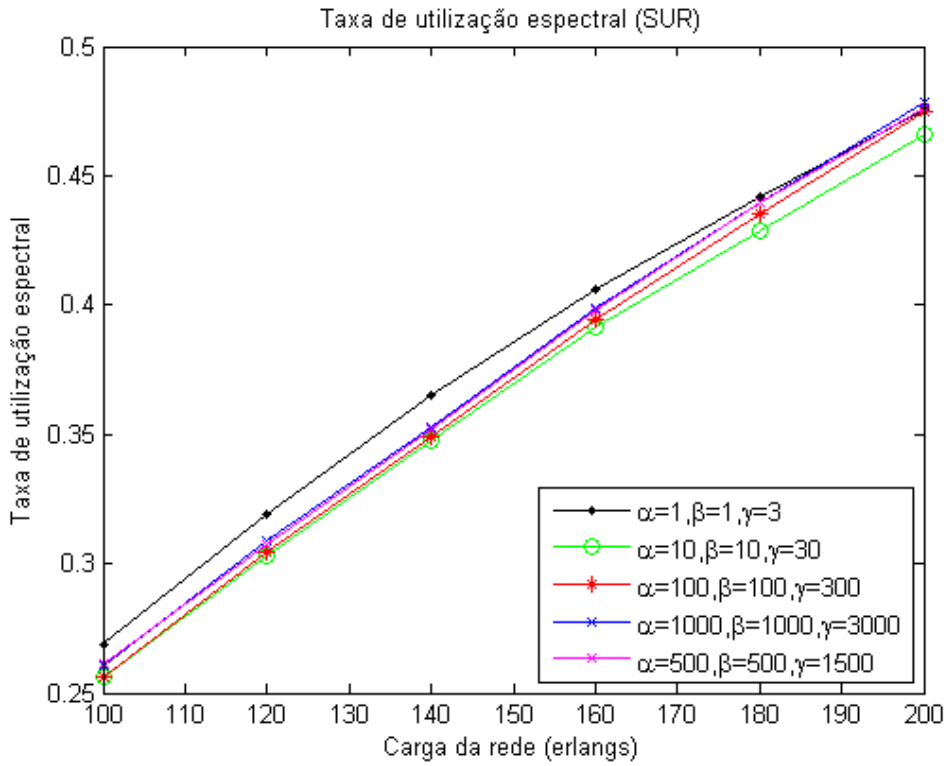

Figura 9-28: Taxa de utilização espectral para diferentes valores de constantes para rede COST239.

9.3 TÉCNICAS DE ALOCAÇÃO ESPECTRAL SW E BUSCA-LÓGICA COM CONVOLUÇÃO

Na Tabela 5-2, foi comparado o tempo médio de duração das simulações de um algoritmo dinâmico simples de RSA usando Dijkstra para calcular a menor rota entre dois nós e FF como método de alocação espectral. O algoritmo não realiza proteção das conexões. Nas simulações, foi utilizada a técnica tradicional de busca por *slots* de frequência utilizando varredura com janela espectral (SW) descrito na seção 4.4 e o método de busca lógica desenvolvido (seção 7.1.3). Com o tempo médio de dez simulações com 10 mil e 100 mil conexões, foi calculado o tempo médio gasto para tratar cada conexão. O método desenvolvido é mais veloz que o tradicional, pois a busca por *slots* não precisa ser repetida a cada enlace da rota. Ela é realizada apenas uma vez, por meio de uma convolução.

Tabela 9-2: Comparação do tempo de simulação dos métodos de alocação espectral SW e busca lógica.

Método	Tempo médio de simulação (em segundos)		
	10 mil conexões	100 mil conexões	Por conexão
SW	30,24	291,05	0,00297
Busca lógica	6,37	68,78	0,00066

9.4 SIMULAÇÕES PARA REDES COM TRANSCEPTORES NÃO CONFIGURÁVEIS

Assim como o algoritmo em [22, 23], o algoritmo RSA_SF_SPP desenvolvido pode ser utilizado em redes cujos transceptores não sejam configuráveis. Nesse caso, os *slots* de frequência dos caminhos primário e de proteção deverão possuir os mesmos índices, pois os transceptores não poderão ser reconfigurados para transmitir o tráfego afetado pela falha em um faixa diferente do espectro. Para avaliar o desempenho do algoritmo, foi simulado o algoritmo RSA_FF_SPP, também para redes com transceptores não configuráveis. O intuito dessas simulações é avaliar como o método *smart-fit* de seleção do caminho primário afeta a seleção do caminho de proteção quando há restrição de posicionamento no espectro. Note que a diferenciação dos custos entre *slots* livres ou compartilhados para o caminho de proteção afeta muito pouco a decisão do algoritmo, a não ser quando é encontrado dois caminhos de proteção candidatos, um deles com *slots* livres na posição do espectro já definido e outro com *slots* que possam ser compartilhados. Nessa versão do algoritmo, também não há separação dos *slots* destinados a proteção e primários em porções diferentes do espectro.

9.4.1 Probabilidade de bloqueio e probabilidade de bloqueio de banda

Nas redes NSFNET (Figura 9-29 e Figura 9-30), e COST239 (Figura 9-32 e Figura 9-33), o algoritmo RSA_SF_SPP com constantes α = 100, β = 100 e γ = 300 obteve maiores taxas de bloqueio do que o algoritmo RSA_FF_SPP. A busca pelo melhor caminho, que equilibre entre o custo do *slot* a ser utilizado e a distância percorrida, leva o algoritmo RSA_SF_SPP a selecionar como caminho

primário rotas que permitam utilizar *slots* de menor índice. Entretanto, ao procurar pelo caminho de proteção com *slots* de mesmo índice, o algoritmo não os encontra, pois geralmente estes já estão sendo utilizados por outros caminhos primários. O método *first-fit*, por outro lado, dá preferência para utilização de rotas menores, mesmo que os *slots* utilizados possuam maiores índices. Essa escolha aumenta a probabilidade de encontrar um caminho de proteção com *slots* disponíveis. Ao alterarmos as constantes do algoritmo RSA_SF_SPP para $\alpha = 1$, $\beta = 1$ e $\gamma = 3$, este passa a dar prioridade à escolha de rotas mais curtas, pois o valor dos *slots* passa a ter menor peso na fórmula de custo e o algoritmo tende a escolher os primeiros slots disponíveis na rota mais curta, semelhante ao método *first-fit*. Pode-se ver que o algoritmo passa a ter performance melhor, alcançando probabilidade de bloqueio semelhante à do RSA_FF_SPP.

Na Figura 9-31 e na Figura 9-34, note que, diferentemente dos resultados para redes com transceptores configuráveis, a maior causa de bloqueio das conexões é pela não localização do caminho de proteção. Para rede NSFNET, Figura 9-31, o algoritmo RSA_SF_SPP para $\alpha = 1$, $\beta = 1$ e $\gamma = 3$, obteve apenas 15,7% de bloqueio de caminhos primários, seguido pelo algoritmo RSA_SF_SPP com $\alpha = 100$, $\beta = 100$ e $\gamma = 300$, com 18,7% e o algoritmo RSA_FF_SPP com 19,2%.

Para rede COST239, Figura 9-34, o algoritmo RSA_SF_SPP com valores de constantes $\alpha = 1$, $\beta = 1$ e $\gamma = 3$, obteve 0,8% de bloqueio de caminhos primários, o algoritmo RSA_SF_SPP com $\alpha = 100$, $\beta = 100$ e $\gamma = 300$, obteve 3,5% e o algoritmo RSA_FF_SPP 4%. Como a rede COST239 possui maior grau nodal, os algoritmos têm menos dificuldades para localizar um caminho primário e as restrições de localização da faixa espectral para o caminho de proteção são responsáveis pelas altas probabilidades de bloqueio.

A alta porcentagem de conexões bloqueadas devido à não localização de um caminho de proteção que respeite as restrições de índice dos slots a serem reservados indica que a rede, mesmo vazia e com poucas conexões ativas, deixa de aceitar novas conexões por não poder garantir sua proteção. A rede com transceptores não configuráveis não é ideal para esquemas de proteção, pois limita as opções de caminhos de proteção, ocasionando bloqueio de muitas conexões.

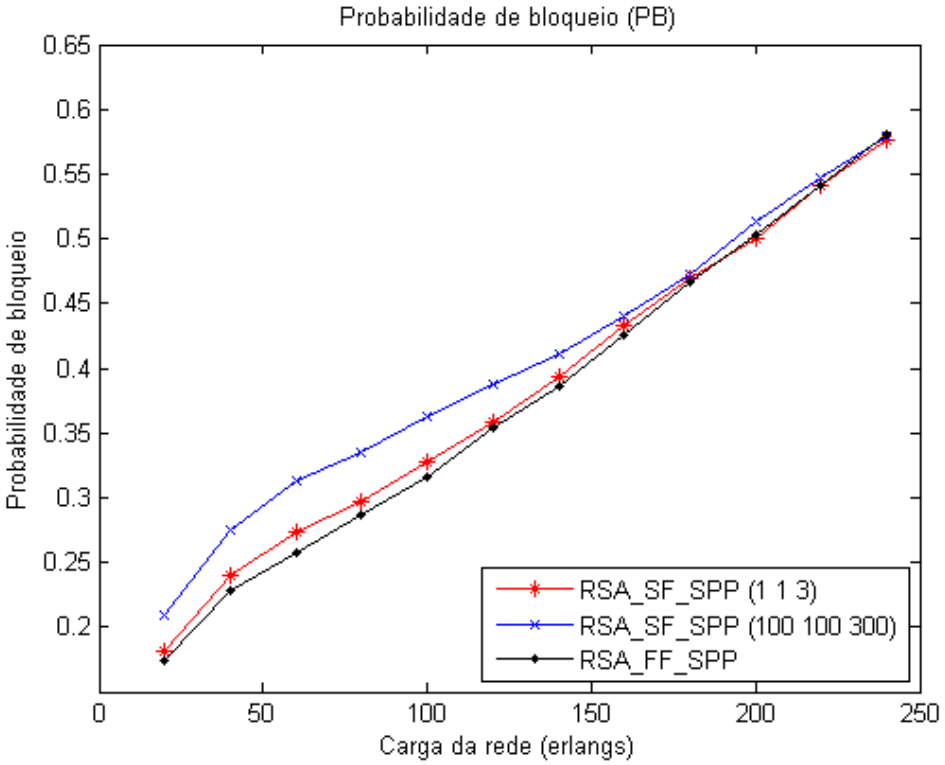

Figura 9-29: Probabilidade de bloqueio para rede NSFNET com transceptores não configuráveis.

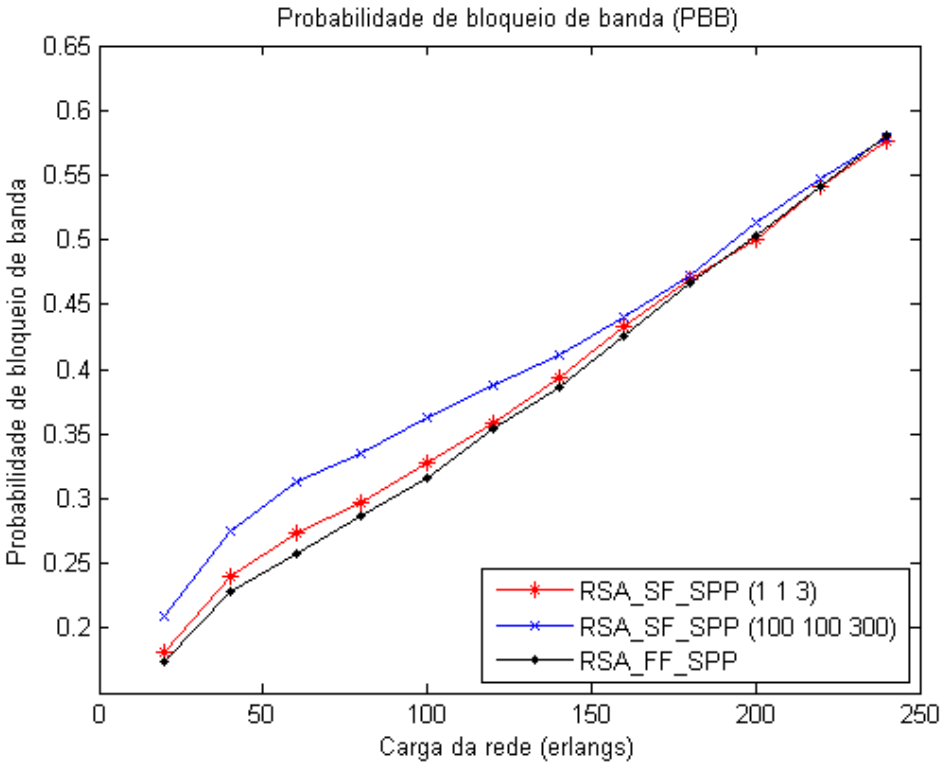

Figura 9-30: Probabilidade de bloqueio de banda para rede NSFNET com transceptores não configuráveis.

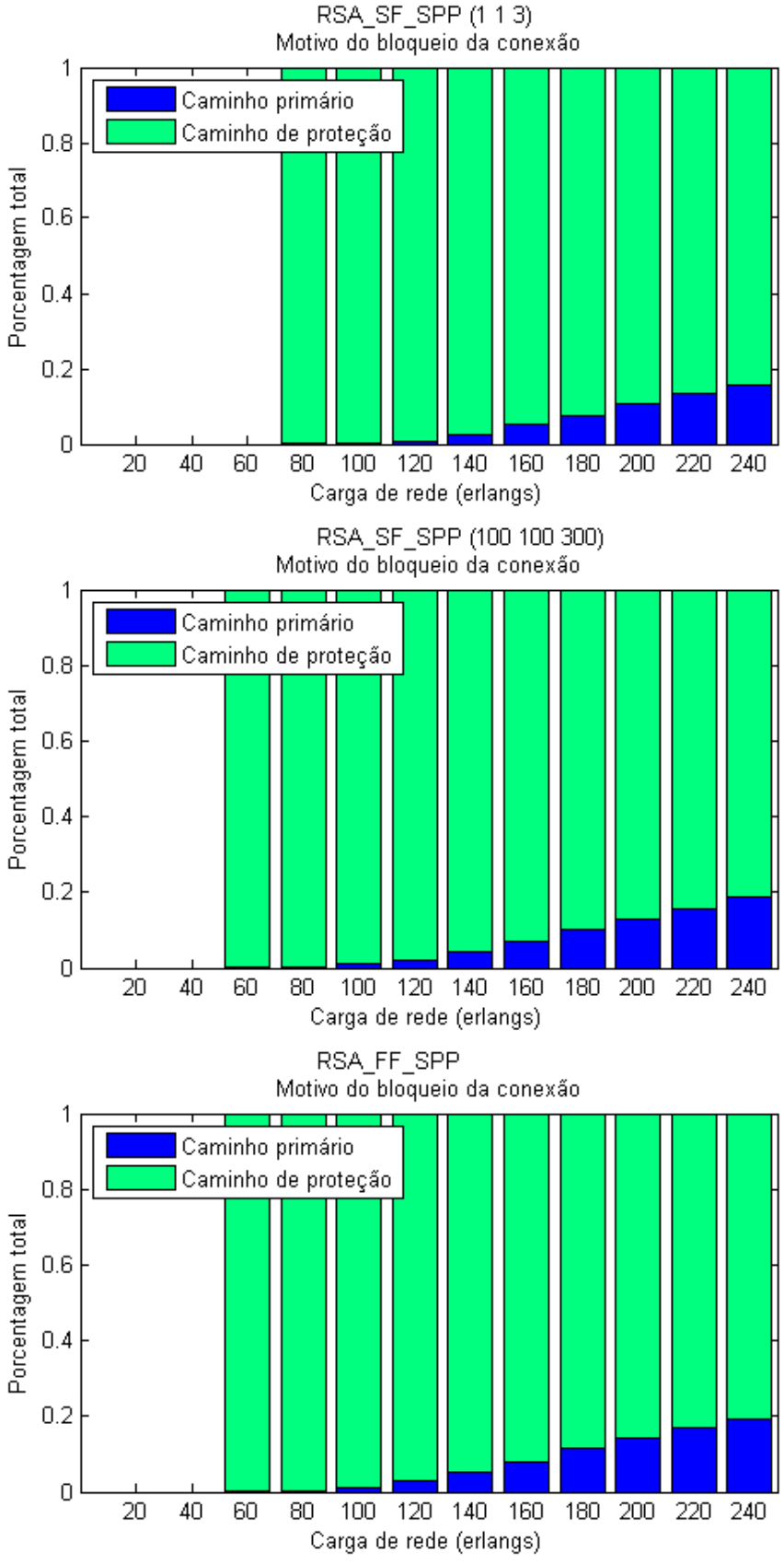

Figura 9-31: Motivo de bloqueio de conexões para rede NSFNET com transceptores não configuráveis.

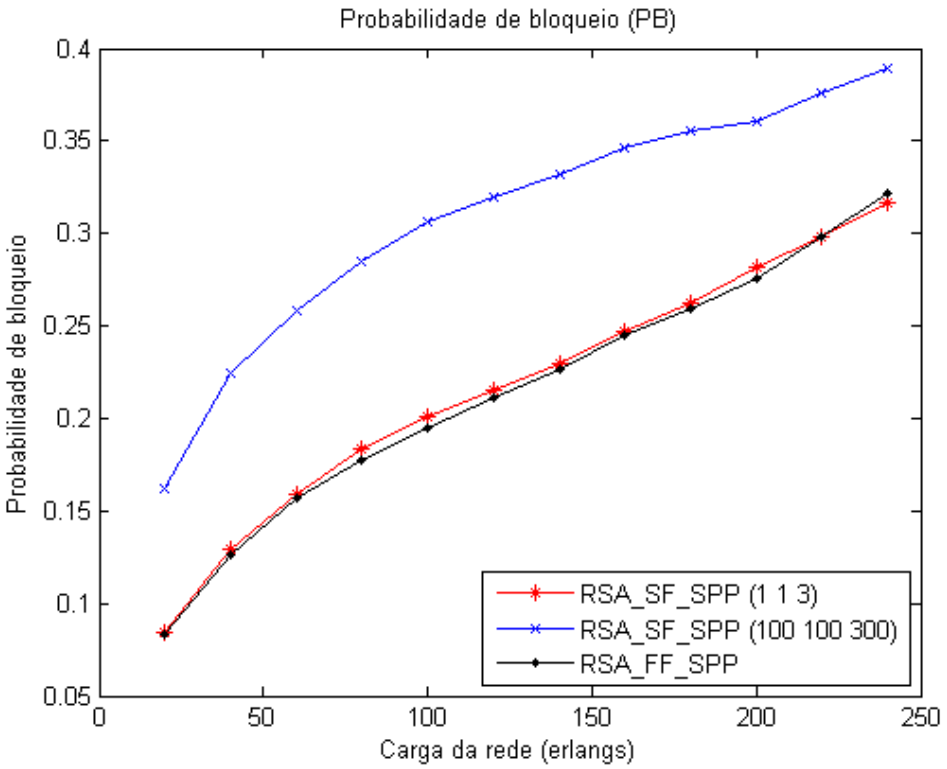

Figura 9-32: Probabilidade de bloqueio para rede COST239 com transceptores não configuráveis.

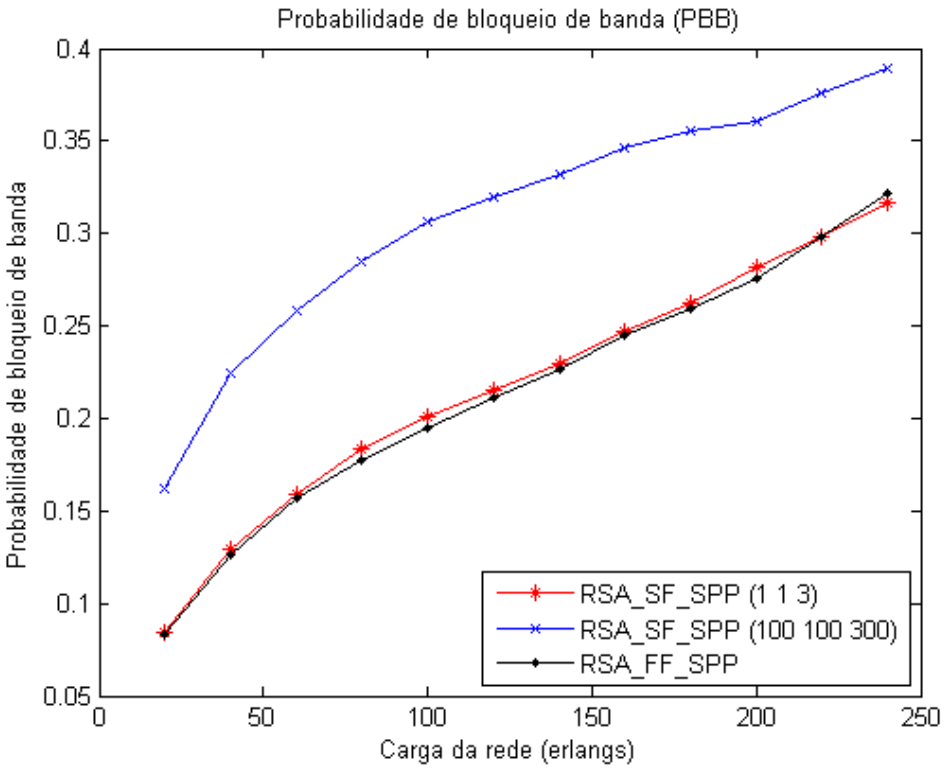

Figura 9-33: Probabilidade de bloqueio de banda para rede COST239 com transceptores não configuráveis.

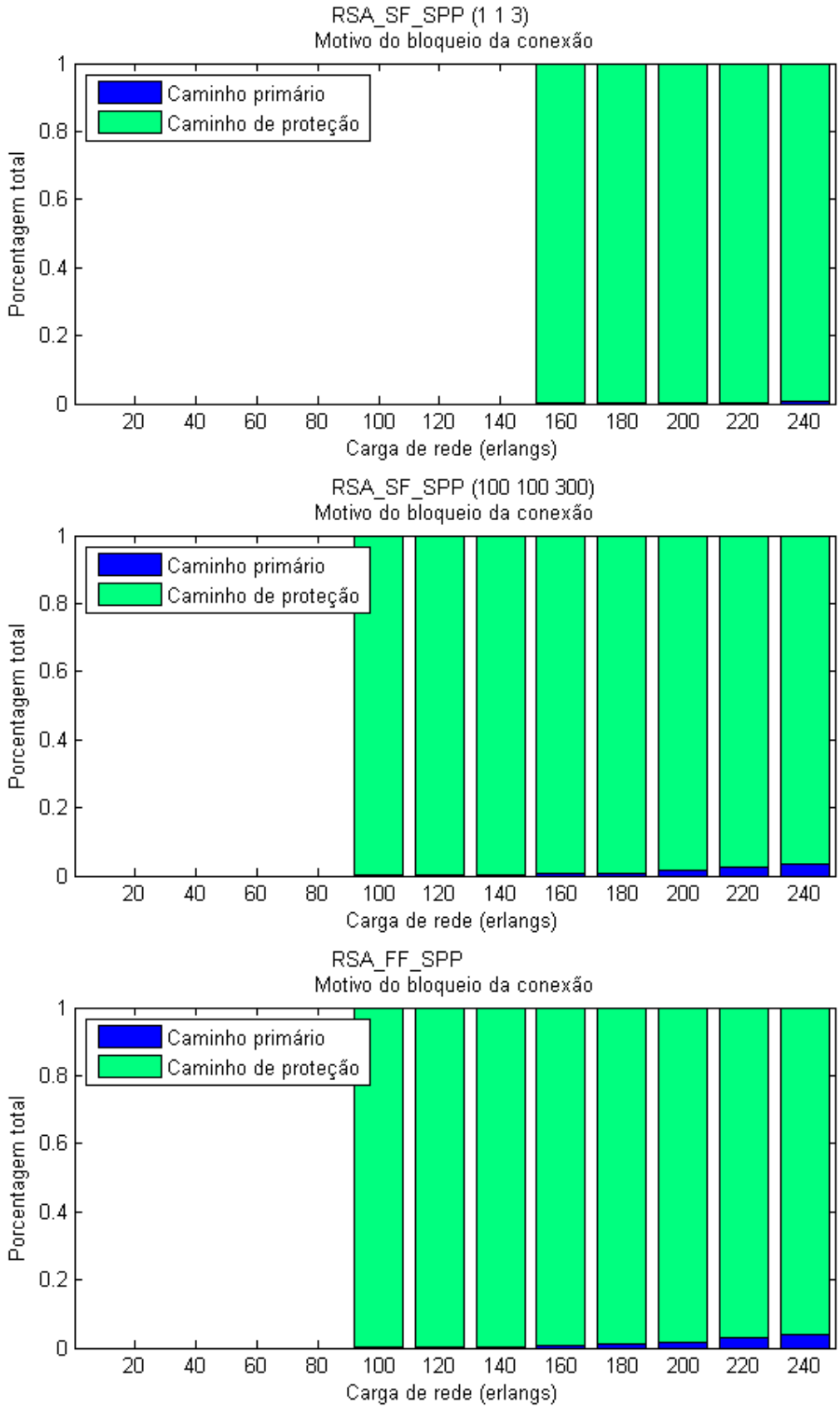

Figura 9-34: Motivo de bloqueio de conexões para rede COST239 com transceptores não configuráveis.

9.4.2 **Grau de compartilhamento**

Ainda que o algoritmo RSA_FF_SPP possua menor probabilidade de bloqueio, pode-se ver nos gráficos da Figura 9-35 e da Figura 9-36 que as duas versões do algoritmo RSA_SF_SPP são capazes organizar melhor as conexões ativas na rede, aumentando o grau de compartilhamento entre caminhos de proteção. Como esperado, o valor menor das constantes prioriza a seleção de caminhos menores à seleção de *slots* de menor índice, diminuindo o grau de compartilhamento. Embora o algoritmo RSA_SF_SPP, com constantes $\alpha = 1$, $\beta = 1$ e $\gamma = 3$, obteve probabilidades de bloqueio semelhantes à do algoritmo RSA_FF_SPP, pode-se ver que ele foi capaz de organizar melhor os recursos de proteção da rede, incentivando o compartilhamento espectral.

9.4.3 **Fragmentação espectral**

A fragmentação espectral obtida em ambas as redes (Figura 9-37 e Figura 9-38) é menor com o uso do algoritmo RSA_SF_SPP com constantes $\alpha = 100$, $\beta = 100$ e $\gamma = 300$, que prioriza a utilização de *slots* de menor índice, ocupando a rede de forma sequencial e organizada. Entretanto, é possível ver também que o algoritmo RSA_FF_SPP obteve maior fragmentação espectral do que a versão do algoritmo RSA_SF_SPP com constantes $\alpha = 1$, $\beta = 1$ e $\gamma = 3$.

9.4.4 **Taxa de utilização espectral**

Embora os três algoritmos obtiveram valores de SUR próximos, novamente pode-se ver que, para as duas redes (Figura 9-39 e Figura 9-40), o valor maior das constantes do algoritmo RSA_SF_SPP obteve menor SUR, enquanto o algoritmo RSA_FF_SPP obteve valores mais elevados. Semelhante à situação anterior, com transceptores configuráveis, o algoritmo RSA_FF_SPP é utiliza mais recursos das redes, mas de maneira menos organizada e mais fragmentada.

Note que o comportamento do algoritmo RSA_SF_SPP com constantes $\alpha = 1$, $\beta = 1$ e $\gamma = 3$ é bastante semelhante ao RSA_FF_SPP.

9.4.5 **Taxa de capacidade reservada**

Na Figura 9-41 e na Figura 9-42, é possível ver que o algoritmo RSA_SF_SPP, com valores de constantes $\alpha = 100$, $\beta = 100$ e $\gamma = 300$, é capaz de reduzir a quantidade de recursos utilizados na rede para proteção das conexões, mas essa redução não é tão significativa quanto nas simulações realizadas para transceptores configuráveis. Apenas para cargas acima de 160 E, na rede COST239 da Figura 9-42, a SCR cai abaixo de um. Para os demais valores de carga, e para rede NSFNET, a SCR dos algoritmos é sempre maior que um, indicando que a restrição da faixa espectral a ser utilizada pelo caminho de proteção prejudica as possibilidades de compartilhamento e aumenta a quantidade de recursos de proteção necessários. Logo, o gasto de recursos destinados à proteção em uma rede com transceptores não configuráveis é maior.

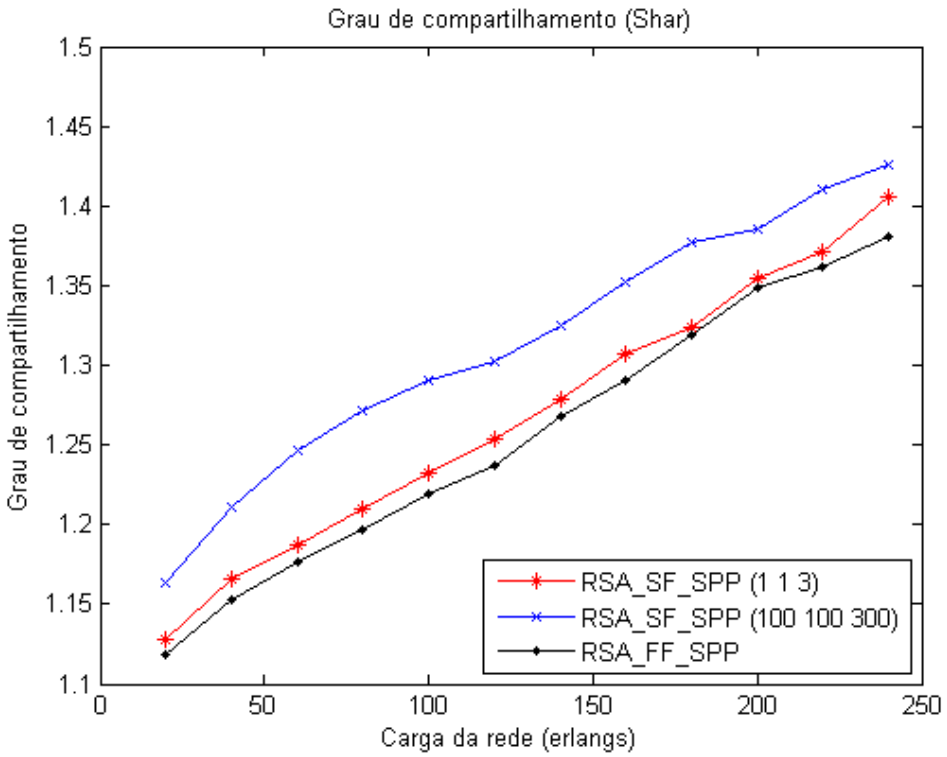

Figura 9-35: Grau de compartilhamento para rede NSFNET com transceptores não configuráveis.

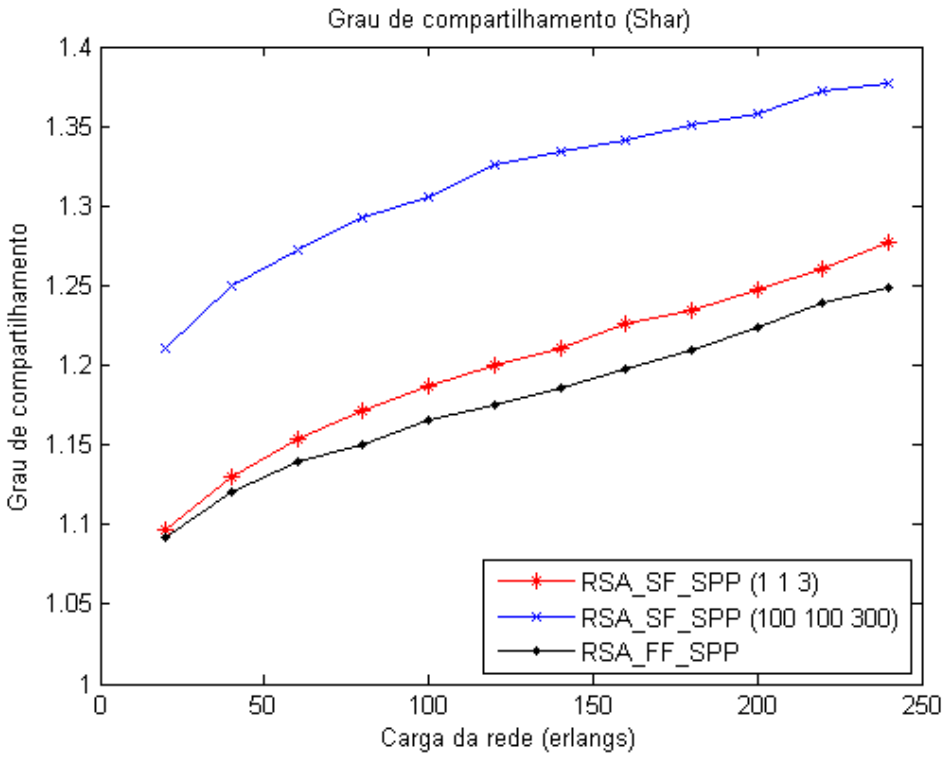

Figura 9-36: Grau de compartilhamento para rede COST239 com transceptores não configuráveis

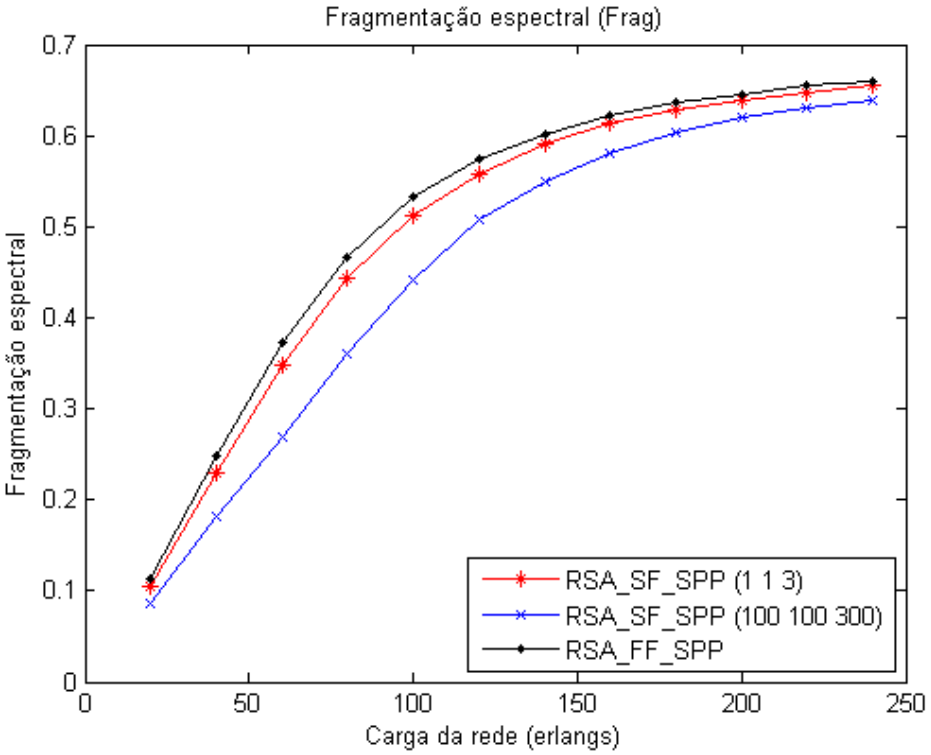

Figura 9-37: Fragmentação espectral para rede NSFNET com transceptores não configuráveis.

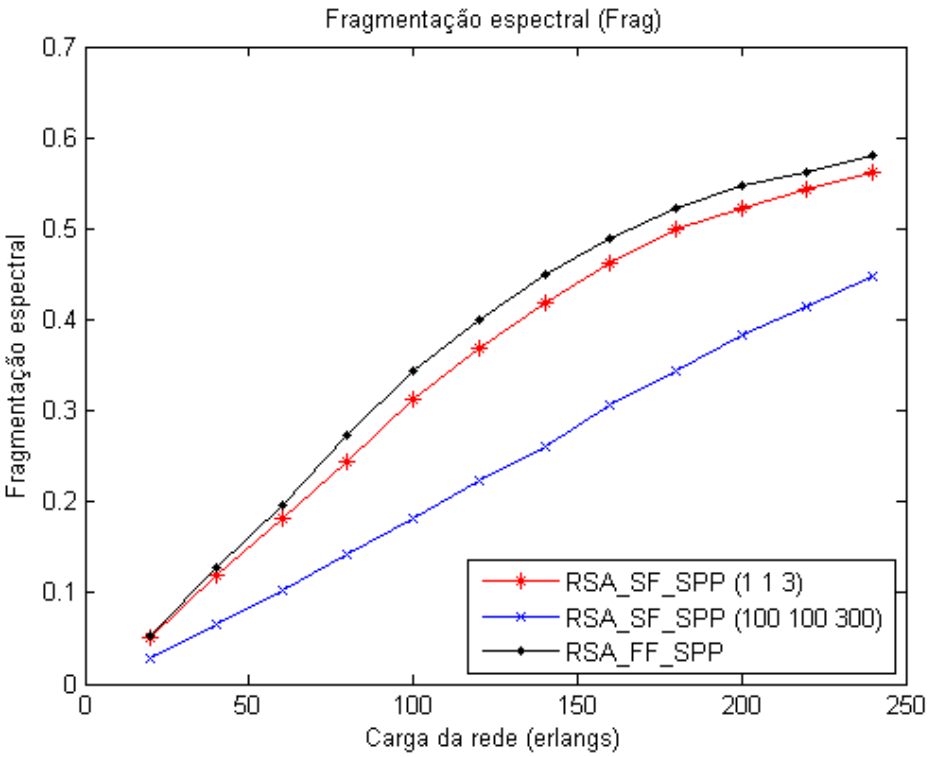

Figura 9-38: Fragmentação espectral para rede COST239 com transceptores não configuráveis.

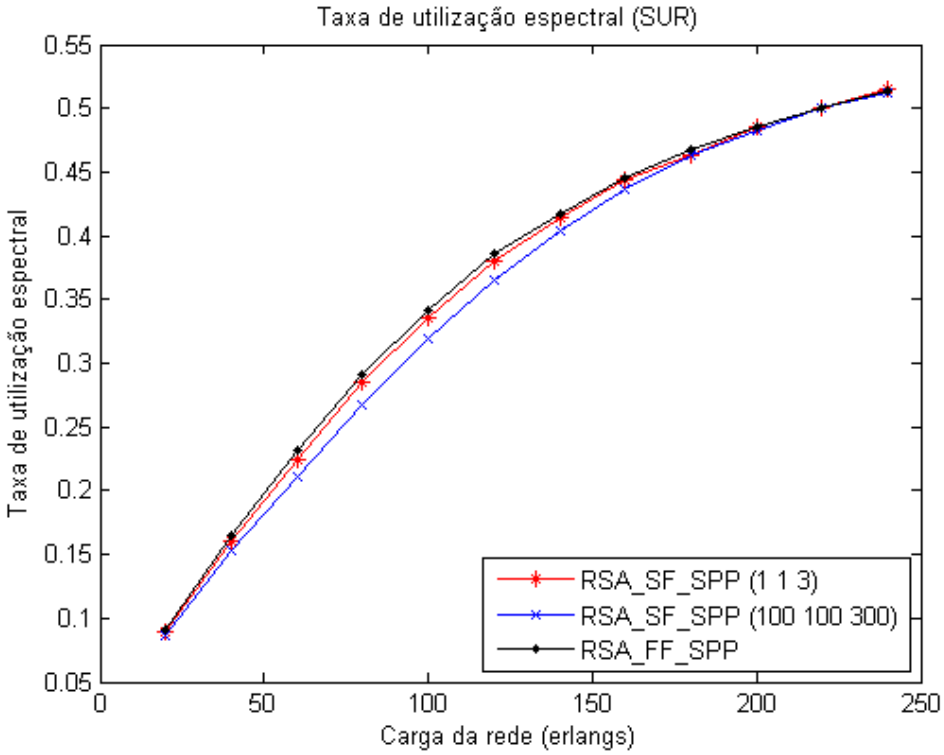

Figura 9-39: Taxa de utilização espectral para rede NSFNET com transceptores não configuráveis.

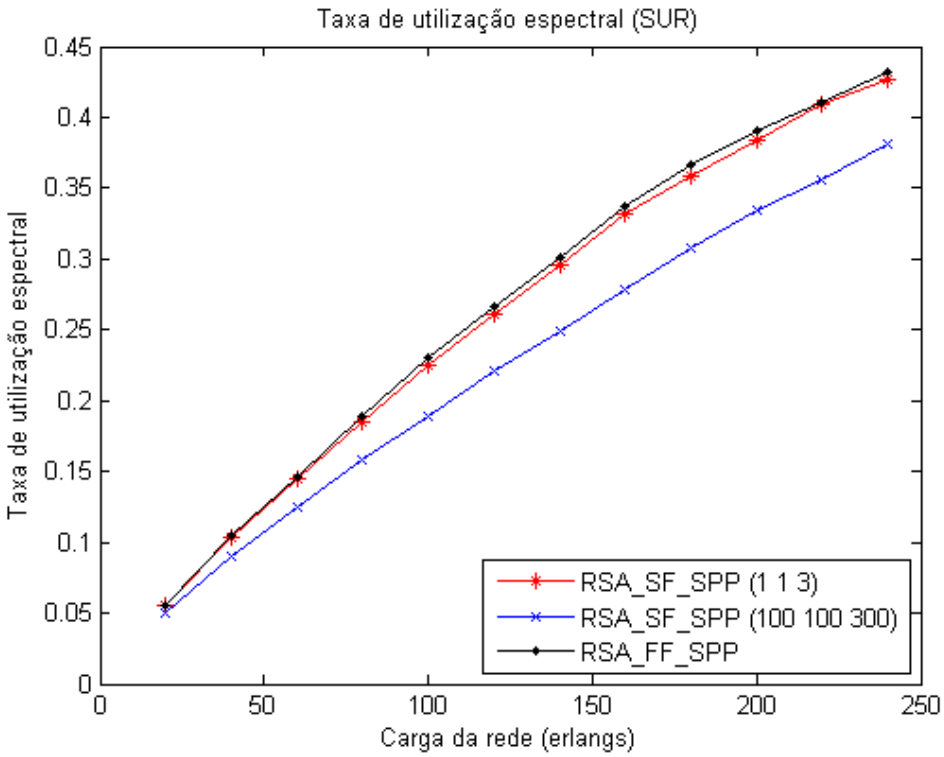

Figura 9-40: Taxa de utilização espectral para rede COST239 com transceptores não configuráveis.

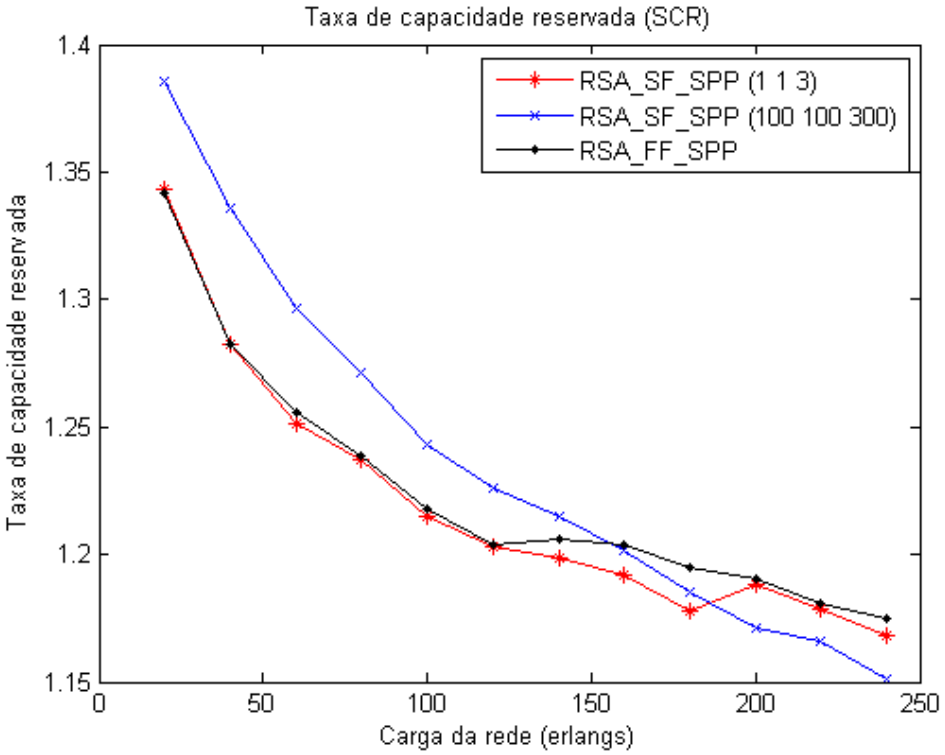

Figura 9-41: Taxa de capacidade reservada para rede NSFNET com transceptores não configuráveis.

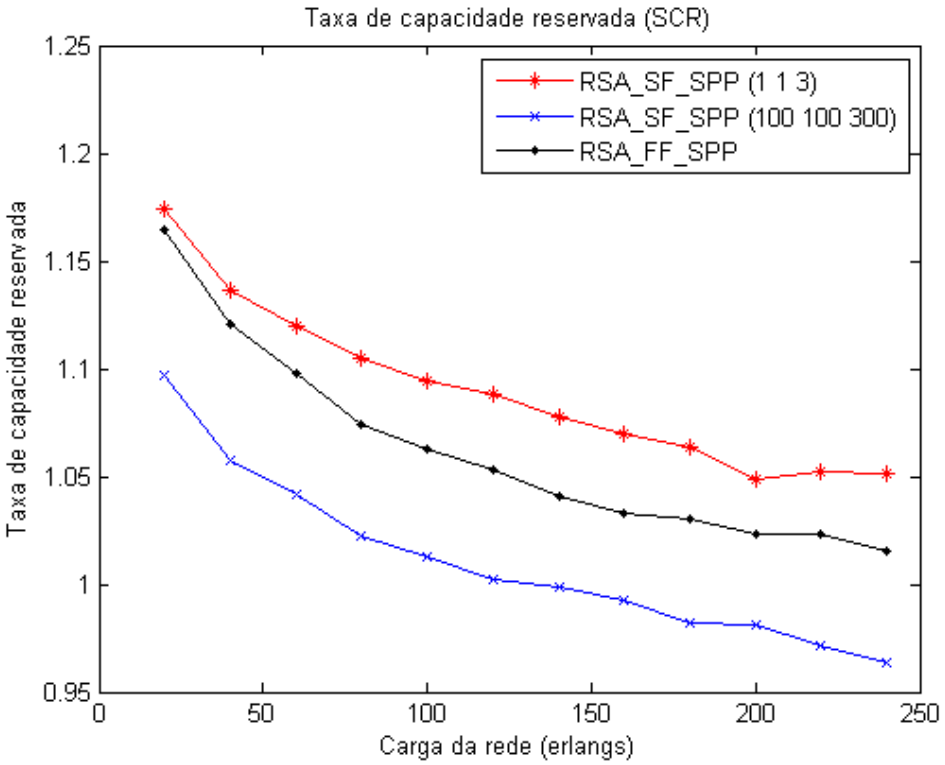

Figura 9-42: Taxa de capacidade reservada para rede COST239 com transceptores não configuráveis.

Para o algoritmo RSA_SF_SPP com constantes $\alpha = 1$, $\beta = 1$ e $\gamma = 3$, a SCR é um pouco menor na rede NSFNET (Figura 9-41) do que do algoritmo RSA_FF_SPP, mas o mesmo não ocorre na rede COST239 (Figura 9-42). O valor baixo das constantes prejudica as decisões do algoritmo, levando ao desperdício de recursos de proteção. É possível que o algoritmo utilize rotas mais longas como caminho de proteção com a faixa espectral exigida, aumentando o valor de SCR, embora o alto grau de compartilhamento diminua a utilização de recursos da rede, como pode ser observado nos gráficos de SUR.

9.4.6 **Porcentagem de restauração**

Mesmo com menor número de conexões ativas na rede, a porcentagem de restauração do algoritmo RSA_SF_SPP, para os dois valores de constantes, quando uma falha aleatória ocorre em um dos enlaces para diferentes valores de carga, é 100% em ambas as redes. Isso indica que o algoritmo foi capaz de proteger todas as conexões ativas, como esperado.

10 CONCLUSÕES

Com o aumento da demanda por tráfego nas redes de comunicações ópticas, é essencial que busquemos por novas tecnologias que tornem possível o atendimento a essa demanda. A OFDM é uma técnica de multiplexação robusta à interferência entre símbolos, modulável e com alta eficiência espectral, tornando-se uma candidata promissora nas novas gerações de redes de transmissão ópticas de banda larga. Utilizando essa técnica de multiplexação, foi proposta a rede SLICE, uma rede óptica com espectro flexível, em que o caminho óptico se adapta de acordo com a quantidade de dados a serem transmitidos, aumentando a eficiência espectral e economizando banda de transmissão. A rede SLICE, entretanto, tem como desvantagem a fragmentação espectral. Conforme as conexões chegam e saem, o espectro se torna cada vez mais fragmentado, diminuindo a eficiência da utilização do espectro e aumentando a probabilidade de bloqueio de uma nova conexão.

O aumento do tráfego de dados exige também que as redes de transmissão se tornem mais robustas a falhas. Para redes de pequena escala, técnicas que utilizam caminhos de proteção são uma solução interessante para garantir a sobrevivência do tráfego da rede. O problema se torna mais complexo em redes de grande escala, pois a utilização de recursos de proteção dedicados se torna dispendiosa. Buscando diminuir a utilização desses recursos, diversas técnicas de proteção compartilhada são estudadas, permitindo que algumas conexões compartilhem os mesmos recursos de proteção. As conexões que podem compartilhar seus caminhos de proteção devem ser escolhidas de modo a evitar que mais de uma conexão necessite do mesmo recurso de proteção. Em caso de uma única falha na rede, é necessário garantir que os caminhos primários dessas conexões sejam disjuntos em enlace.

As técnicas de restauração da rede não utilizam reserva prévia de recursos. Ao ocorrer uma falha, o algoritmo busca por caminhos e segmentos de espectros disponíveis para desviar o tráfego afetado e evitar a queda da conexão. Embora utilize menos recursos da rede, a restauração não garante que o tráfego será recuperado, pois pode não encontrar recursos disponíveis na rede. A restauração também é mais lenta que a proteção.

A decisão de quais técnicas serão utilizadas para garantir a sobrevivência da rede deve levar em consideração uma série de fatores, não só a quantidade de recursos que a rede oferece, ou o tempo de recuperação esperado, mas também as características do tráfego transmitido e os acordos de qualidade de serviço. Pode-se definir quais demandas tem maior prioridade para serem protegidas e deixar demandas de menor prioridade serem tratadas apenas por técnicas de restauração. Como a reserva de recursos de proteção ocupa parte do espectro da rede, pode-se, por exemplo, criar novos acordos de qualidade de serviço que cubram essa proteção. Neste trabalho, optou-se pelo estudo de técnicas de proteção compartilhada pelo desafio que estas apresentam, principalmente em relação à eficiência na utilização dos recursos.

O algoritmo desenvolvido busca avaliar as principais características dos demais algoritmos estudados, além de oferecer uma solução para o problema de RSA e proteção compartilhada em uma rede de grande porte com tráfego dinâmico. Sua primeira característica é a técnica de escolha do caminho primário. Assim como os demais algoritmos estudados, as rotas candidatas a caminho primário são as menores rotas entre o nó fonte e o destino. A principal diferença, entretanto, é o método de seleção dos *slots* de frequência a serem utilizados. As técnicas mais comuns são FF ou LSS, mas os algoritmos param a busca assim que uma faixa disponível e suficiente é encontrada. O algoritmo RSA_SF_SPP, por outro lado, analisa todas as rotas candidatas e escolhe aquela com melhor relação entre o índice dos *slots* a ser utilizado e a distância da rota percorrida. Essa técnica foi chamada *smart-fit*.

Para a seleção do caminho de proteção, a maioria dos algoritmos estudados realiza uma nova busca das menores rotas, dessa vez disjuntas do caminho primário, e escolhe os *slots* de frequência via FF ou LSS. No artigo [24], porém, o método é diferente. Chamado pelo autor de *last-fit* modificado, o algoritmo analisa todas as faixas de frequência encontradas nas rotas candidatas antes de tomar a decisão pela faixa de menor índice inicial encontrada. Esse método é bastante semelhante ao *smart-fit*, mas não leva em consideração o tamanho a rota a ser escolhida. Para o caminho de proteção, o algoritmo RSA_SF_SPP possui ainda outro diferencial importante. A escolha da rota e dos *slots* a serem utilizados como proteção leva em consideração também se tais *slots* serão ou não compartilhados entre outras conexões. O custo de se utilizar um *slot* livre é maior que o custo de um *slot* compartilhado. O intuito é incentivar o compartilhamento espectral, buscando um equilíbrio entre a posição da faixa de frequências dentro do espectro e a distância da rota percorrida.

Diversas avaliações foram realizadas para testar o algoritmo desenvolvido e comparar seu desempenho com o algoritmo clássico de proteção compartilhada RSA_FF_SPP, que seleciona os *slots* via FF, e uma versão semelhante ao algoritmo de [24], mas sem modulação adaptativa, que chamamos de RSA_FFMLF_SPP.

Uma característica importante do algoritmo em [24] é a separação dos *slots* destinados a transmissão do caminho primário e reservados como caminho de proteção em porções diferentes do espectro de frequência. No artigo, o algoritmo coloca os caminhos primários na porção inicial do espectro e os de proteção na porção final. Para avaliar os efeitos dessa separação, o algoritmo desenvolvido RSA_SF_SPP possui duas versões. Na primeira, não há separação dos *slots* e os caminhos primários e de proteção são alocados do menor para o maior índice. Na segunda versão, o algoritmo separa os caminhos primários na porção final do espectro e os de proteção na porção inicial. Essa ordem foi escolhida simplesmente pela facilidade ao alterar o método de escolha dos *slots* primários entre LSS e HSS no código do algoritmo.

Foi possível avaliar também que as características dos transceptores da rede influenciam nos resultados e capacidade dos algoritmos. Redes com transceptores não configuráveis restringem as opções de seleção de caminhos de proteção, aumentado a fragmentação da rede, as probabilidades de

bloqueio e os gastos de recursos destinados à proteção. Foi necessário diminuir os valores das constantes de custo dos *slots* de frequência para que o algoritmo RSA_SF_SPP tivesse melhor desempenho.

Outro importante método desenvolvido nesse trabalho, é a técnica de busca pelos *slots* disponíveis que chamamos de busca lógica. Como no método *smart-fit* todos os *slots* em todas as rotas candidatas são avaliados, é necessário que tal avaliação seja veloz, pois o algoritmo lida com tráfego dinâmico. O método de busca logica foi chamado dessa maneira por tratar os *slots* de frequência livres ou ocupados como uma sequência lógica de zeros e uns. De acordo com o tipo de *slots* desejado, para ser utilizado como caminho primário ou de proteção, o algoritmo realiza operações lógicas entre todos os enlaces da rota selecionada e procura apenas uma vez por posições do espectro resultante onde a conexão possa ser alocada. Essa busca é feita por meio de uma convolução entre o espectro resultante e uma janela espectral de mesmo tamanho da requisição da conexão. Como foi demonstrado, o método de busca lógica é mais veloz que o método de varredura clássico, pois independe do número de enlaces da rota, apenas no número de *slots* no espectro. Para que o algoritmo de busca lógica seja capaz de diferenciar slots livres, slots reservados para proteção e slots de caminhos primários, foi proposta aqui pela primeira vez a matriz Cmat, que representa cada enlace da rede como um conjunto de três vetores: Cmat.Freq, Cmat.Type e Cmat.ID, que indicam se os slots estão livres ou não, se são slots usados por caminhos de proteção ou caminhos primários e os identificadores das conexões ativas em cada slot.

A fragmentação espectral é uma importante ferramenta para analisar as tomadas de decisões dos algoritmos. Como o algoritmo desenvolvido busca equilibrar a utilização dos *slots* de proteção em relação ao seu índice e à distância percorrida, o algoritmo opta muitas vezes por ocupar um *slot* livre de menor índice mesmo que haja a possibilidade de compartilhamento em outra posição do espectro. Isso diminui a fragmentação do espectro da rede, mesmo prejudicando o grau de compartilhamento. Entretanto, tal decisão não afeta o desempenho final do algoritmo em relação à probabilidade de bloqueio e probabilidade de bloqueio de banda. Além disso, a separação de *slots* em porções diferentes do espectro melhora a fragmentação espectral e incentiva o compartilhamento de *slots* de frequência entre caminhos de proteção que utilizam o mesmo enlace.

A avaliação de um algoritmo não pode ser feita por apenas um ponto de vista. É necessário analisar várias informações, inclusive acompanhar de perto cada tomada de decisão do algoritmo. Os gráficos de SUR, por exemplo, indicam quanto dos recursos da rede foram utilizados pelo algoritmo, mas não avalia como esses recursos foram utilizados. Nos testes realizados, o algoritmo RSA_FF_SPP obteve altos valores de SUR, mas alto bloqueio de conexões e alta fragmentação. Isso indica que os recursos da rede foram consumidos de forma desorganizada e fragmentada. Por outro lado, o algoritmo RSA_SF_SPP, na versão HSS, o alto valor obtido nos gráficos de SUR, em adição à baixa fragmentação e baixa probabilidade de bloqueio, indica que os recursos da rede estavam bem ocupados, de maneira organizada, permitindo mais conexões ativas simultaneamente.

A porcentagem de restauração é outro método para avaliar o algoritmo desenvolvido. No caso de falhas, se algumas conexões não pudessem ser recuperadas, indicaria que o algoritmo não era capaz de

proteger devidamente as conexões ativas. Ao obter recuperação total em todas as simulações, sabe-se que o algoritmo compartilha corretamente apenas caminhos de proteção entre conexões cujos caminhos primários são disjuntos em enlace e organiza corretamente os *slots* destinados a proteção e a transmissão.

É possível perceber o desempenho superior do algoritmo criado principalmente ao analisar os gráficos de taxa de capacidade reservada (SCR). Ao equilibrar a decisão entre as rotas utilizadas e a posição no espetro, o algoritmo (na versão HSS) foi capaz de utilizar os recursos de proteção de forma extremamente eficiente, de maneira que a rede tenha menor número de *slots* destinados à proteção do que primários. Obter um algoritmo que use de maneira eficiente os recursos de proteção é o principal objetivo deste trabalho. Ao utilizar menos *slots* de proteção, mais recursos da rede passam a estar disponíveis para novas conexões, diminuindo a probabilidade de bloqueio.

Como trabalhos futuros, o algoritmo RSA_SF_SPP desenvolvido pode ser atualizado e ter sua complexidade aumentada a fim de englobar, por exemplo, uma importante característica das redes SLICE: a restauração por compressão de banda BSR (descrita na seção 5.5). Para diminuir a probabilidade de bloqueio de uma conexão por falta de caminho de proteção, pode-se definir métodos que envolvam a seleção entre diferentes níveis de modulação do sinal de acordo com a distância percorrida na rota e definir a quantidade de slots de proteção necessários de acordo com a modulação selecionada. Assim, uma demanda de muitos slots, por exemplo, de acordo com a distância percorrida, poderia ter sua modulação alterada para utilizar menos slots de proteção. Seguindo mais adiante, a seleção do nível de modulação do sinal pode ser aplicada tanto para caminhos de proteção quanto para caminhos primários, economizando ainda mais espectro e aumentando a eficiência da rede.

Também pode-se atualizar o algoritmo de modo a adicionar esquemas de restauração de conexões de baixa prioridade. A reserva de slots de proteção seria utilizada apenas para conexões com tráfego de alta prioridade, diminuindo a quantidade de slots reservados para proteção na rede. Conexões de menor prioridade, quando afetadas pela falha, seriam tratadas por esquemas de restauração, que buscariam por novos caminhos primários em que tais conexões pudessem continuar transmitindo.

Há muitas possibilidades para novos estudos na área de sobrevivência de redes de comunicações ópticas. A eficiência tanto do algoritmo quanto da utilização dos recursos da rede deve ser sempre levada em consideração, buscando o equilíbrio entre os prós e contras de cada uma das técnicas utilizadas.

REFERÊNCIAS

[1] Cisco. (2015). *Cisco Visual Networking Index: Forecast and Methodology, 2014–2019 White Paper*. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html

[2] W. Shieh and I. Djordjevic, *OFDM for Optical Communications*: Elsevier Science, 2009.

[3] E. Pincemin, M. Song, Y. Loussouarn, G. Thouenon, and C. Betoule, "Towards 400G/1T flexible optical transport networks," in *2013 15th International Conference on Transparent Optical Networks (ICTON)*, 2013, pp. 1-13.

[4] D. Y. Zhou and S. Subramaniam, "Survivability in optical networks," *Ieee Network*, vol. 14, pp. 16-23, Nov-Dec 2000.

[5] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka, "Spectrum-Efficient and Scalable Elastic Optical Path Network: Architecture, Benefits, and Enabling Technologies," *Ieee Communications Magazine*, vol. 47, pp. 66-73, Nov 2009.

[6] G. Y. Zhang, M. De Leenheer, A. Morea, and B. Mukherjee, "A Survey on OFDM-Based Elastic Core Optical Networking," *Ieee Communications Surveys and Tutorials*, vol. 15, pp. 65-87, 2013.

[7] J. Proakis and M. Salehi, *Digital Communications*: McGraw-Hill Education, 2007.

[8] W. Stallings, *Data and Computer Communications*: Pearson/Prentice Hall, 2007.

[9] M. Seimetz, *High-Order Modulation for Optical Fiber Transmission*: Springer, 2009.

[10] M. Jinno, B. Kozicki, H. Takara, A. Watanabe, Y. Sone, T. Tanaka, *et al.*, "Distance-Adaptive Spectrum Resource Allocation in Spectrum-Sliced Elastic Optical Path Network," *Ieee Communications Magazine*, vol. 48, pp. 138-145, Aug 2010.

[11] K. Christodoulopoulos, I. Tomkos, and E. A. Varvarigos, "Elastic Bandwidth Allocation in Flexible OFDM-Based Optical Networks," *Lightwave Technology, Journal of*, vol. 29, pp. 1354-1366, 2011.

[12] O. Gerstel, M. Jinno, A. Lord, and S. J. Ben Yoo, "Elastic Optical Networking: A New Dawn for the Optical Layer?," *Ieee Communications Magazine*, vol. 50, pp. S12-S20, Feb 2012.

[13] J. Armstrong, "OFDM for Optical Communications," *Lightwave Technology, Journal of*, vol. 27, pp. 189-204, 2009.

[14] M. Ilyas and H. T. Mouftah, *The Handbook of Optical Communication Networks*: Taylor & Francis, 2003.

[15] K. Christodoulopoulos, I. Tomkos, and E. A. Varvarigos, "Routing and Spectrum Allocation in OFDM-Based Optical Networks with Elastic Bandwidth Allocation," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, 2010, pp. 1-6.

[16] N. Hua, Y. Liu, X. Wan, X. P. Zheng, Z. G. Liu, and Ieee, *Dynamic Routing and Spectrum Assignment Algorithms in Flexible Optical Networks: An Overview*, 2012.

[17] J. Y. Yen, "Finding the K Shortest Loopless Paths in a Network," *Management Science*, vol. 17, pp. 712-716, 1971.

[18] H. Zang, J. P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," *Optical Networks Magazine*, vol. 1, pp. 47-60, 2000.

[19] Y. Liu, N. Hua, X. Wan, X. Zheng, and Z. Liu, "A Spectrum-Scan Routing Scheme in Flexible Optical Networks," in *Network Architectures, Management, and Applications*, Shanghai, 2011, p. 83100B.

[20] Y. Liu, N. Hua, X. Zheng, H. Zhang, and B. Zhou, "Discrete Spectrum-Scan Routing Based on Spectrum Discretization in Flexible Optical Networks," in *Optical Fiber Communication Conference*, Los Angeles, California, 2012, p. JTh2A.49.

[21] X. Shao, L. Zhou, Y.-K. Yeo, X. Cheng, and Z. Xu, "Shared-path protection in OFDM-based optical networks with elastic bandwidth allocation," 2012, pp. 1-3.

[22] G. Shen, Y. Wei, and Q. Yang, "Shared Backup Path Protection (SBPP) in Elastic Optical Transport Networks," in *Asia Communications and Photonics Conference*, Guangzhou, 2012, p. PAF4C.6.

[23] S. Gangxiang, W. Yue, and S. Bose, "Optimal design for shared backup path protected elastic optical networks under single-link failure," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 6, pp. 649-659, 2014.

[24] A. Tarhan and C. Cavdar, "Shared path protection for distance adaptive elastic Optical Networks under dynamic traffic," 2013, pp. 62-67.

[25] B. Chen, Z. Jie, Z. Yongli, L. Jinyan, H. Shanguo, G. Wanyi, *et al.*, "Minimum spectrum block consumption for shared-path protection with joint failure probability in flexible bandwidth optical networks," 2013, pp. 1-3.

[26] Y. Sone, A. Watanabe, W. Imajuku, Y. Tsukishima, B. Kozicki, H. Takara, *et al.*, "Highly survivable restoration scheme employing optical bandwidth squeezing in spectrum-sliced elastic optical path (SLICE) network," in *Optical Fiber Communication - incudes post deadline papers, 2009. OFC 2009. Conference on*, 2009, pp. 1-3.

[27] Y. Sone, A. Watanabe, W. Imajuku, Y. Tsukishima, B. Kozicki, H. Takara, *et al.*, "Bandwidth Squeezed Restoration in Spectrum-Sliced Elastic Optical Path Networks (SLICE)," *Journal of Optical Communications and Networking*, vol. 3, pp. 223-233, Mar 2011.

[28] S. Kosaka, H. Hasegawa, K. Sato, T. Tanaka, A. Hirano, and M. Jinno, "Shared protected elastic optical path network design that applies iterative re-optimization based on resource utilization efficiency measures," 2012, pp. 1-3.

[29] A. Castro, L. Velasco, M. Ruiz, and J. Comellas, "Single-path provisioning with multi-path recovery in flexgrid optical networks," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on*, 2012, pp. 745-751.

[30] B. W. Chen, J. Zhang, Y. L. Zhao, C. H. Lv, W. Zhang, S. G. Huang, *et al.*, "Multi-link failure restoration with dynamic load balancing in spectrum-elastic optical path networks," *Optical Fiber Technology*, vol. 18, pp. 21-28, Jan 2012.

[31] B. W. Chen, J. Zhang, Y. L. Zhao, C. H. Lv, W. Zhang, Y. Gu, *et al.*, "A Novel Recovery Algorithm for Multi-link Failures in Spectrum-Elastic Optical Path Networks," in *Conference on Network Architectures, Management, and Applications IX*, Shanghai, PEOPLES R CHINA, 2011.

[32] M. Klinkowski, "A Genetic Algorithm for Solving RSA Problem in Elastic Optical Networks with Dedicated Path Protection," *International Joint Conference Cisis'12 - Iceute'12 - Soco'12 Special Sessions*, vol. 189, pp. 167-176, 2013.

[33] L. Ruan and N. Xiao, "Survivable Multipath Routing and Spectrum Allocation in OFDM-Based Flexible Optical Networks," *Journal of Optical Communications and Networking*, vol. 5, pp. 172-182, Mar 2013.

[34] R. Lu and Z. Yanwei, "Dynamic survivable multipath routing and spectrum allocation in OFDM-based flexible optical networks," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 6, pp. 77-85, 2014.

[35] H. Pin-han and H. T. Mouftah, "Shared protection in mesh WDM networks," *Communications Magazine, IEEE*, vol. 42, pp. 70-76, 2004.

[36] J. Gonçalves and M. C. Resende, "Biased random-key genetic algorithms for combinatorial optimization," *Journal of Heuristics*, vol. 17, pp. 487-525, 2011/10/01 2011.

[37] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing*: Springer, 1999.

[38] E. Bouillet, G. Ellinas, J. F. Labourdette, and R. Ramamurthy, *Path Routing in Mesh Optical Networks*: Wiley, 2007.

APÊNDICE – PSEUDOCÓDIGOS DOS ALGORITMOS

ALGORITMO 1 ALGORITMO DE YEN

```
1  ENTRADA: grafo da rede, nó fonte, nó destino, número de rotas
2  SAÍDA: menores rotas, distâncias das rotas
3  Ver [17].
```

ALGORITMO 2 BUSCA LÓGICA PRIMÁRIO

```
1  ENTRADA: rota, tamanho da demanda, Cmat, tipo de SA
2  SAÍDA: bloqueado, Fslot
3  PARA cada enlace da rota FAÇA
4          CALCULAR OU lógico entre os vetores Cmat.Freq
5          CALCULAR OU lógico entre os vetores Cmat.Type
6  FIM PARA
7  CALCULAR vetor PRI_final - Fórmula (7.5)
8  CRIAR vetor SW de valores unitários de mesmo tamanho da demanda
9  CALCULAR convolução entre PRI_final e SW
10 ENCONTRAR índices do vetor resultante da convolução onde os valores são nulos
11 CALCULAR localização dos slots livres, subtraindo do vetor de índices o tamanho do vetor SW -1
12 REMOVER valores negativos, nulos e muito grandes (tais que a demanda não caiba dentro do enlace)
13 SE encontrar ao menor um slot disponível ENTÃO
14         SE tipo de SA for LSS ENTÃO % lowest starting slot
15                 DEFINIR Fslot igual ao menor índice encontrado
16                 DEFINIR bloqueado igual a zero
17         SENÃO SE tipo de SA for HSS ENTÃO % highest starting slot
18                 DEFINIR Fslot igual ao maior índice encontrado
19                 DEFINIR bloqueado igual a zero
20         FIM SE
21 SENÃO
22         DEFINIR Fslot vazio
23         DEFINIR bloqueado igual a um
24 FIM SE
25 RETORNAR
26 FIM ALGORITMO
```

ALGORITMO 3 BUSCA LÓGICA PROTEÇÃO

```
1  ENTRADA: rota, tamanho da demanda, Cmat, tipo de SA
2  SAÍDA: bloqueado, Flista, IDlista
3  PARA cada enlace da rota FAÇA
4      CALCULAR OU lógico entre os vetores Cmat.Freq
5      CALCULAR OU lógico entre os vetores Cmat.Type
6      CONCATENAR listas das conexões usando cada slot dos vetores Cmat.ID
7  FIM PARA
8  CALCULAR vetor BCK_final - Fórmula (7.6)
9  CRIAR vetor SW de valores unitários de mesmo tamanho da demanda
10 CALCULAR convolução entre BCK_final e SW
11 ENCONTRAR índices do vetor resultante da convolução onde os valores são nulos
12 CALCULAR localização dos slots livres, subtraindo do vetor de índices o tamanho do vetor SW -1
13 REMOVER valores negativos, nulos e muito grandes (tais que a demanda não caiba dentro do enlace)
14 SE encontrar ao menor um slot disponível ENTÃO
15     DEFINIR Flista como todas as opções de faixas de slots que podem ser ocupadas pela demanda
16     DEFINIR IDlista como lista de todas as conexões que utilizam cada faixa em Flista
17     DEFINIR bloqueado igual a zero
18 SENÃO
19     DEFINIR Flista vazio
20     DEFINIR IDlista vazio
21     DEFINIR bloqueado igual a um
22 FIM SE
23 RETORNAR
24 FIM ALGORITMO
```

ALGORITMO 4 TESTE CAMINHOS DISJUNTOS

```
1  ENTRADA: vetor de enlaces da rota 1, vetor de enlaces da rota 2
2  % O vetor de enlaces é constituído de zeros e uns, indicando quais enlaces da rede são utilizados na rota
3  % Exemplo: a rede possui 6 enlaces no total. A rota 1 utiliza os enlaces 1, 2 e 3.
4  % Vetor de enlaces da rota 1 = [1 1 1 0 0 0]
5  SAÍDA: Disjuntos
6  CALCULAR soma dos dois vetores de enlaces
7  SE alguma posição do vetor resultante possuir valor maior do que um ENTÃO
8      DEFINIR Disjuntos igual a zero % rotas não são disjuntas
9  SENÃO
10     DEFINIR Disjuntos igual a um % rotas são disjuntas
11 FIM SE
12 RETORNAR
13 FIM ALGORITMO
```

ALGORITMO 5 RSA_SF_SPP

```
1  ENQUANTO novo evento
2      SE evento for de chegada de conexão ENTÃO
3      % Caminho Primário
4      CALCULAR k menores caminhos - Algoritmo (1)
5      PARA cada caminho encontrado FAÇA
6          CALCULAR slots para alocação - Algoritmo (2)
7          SE encontrar slots disponíveis ENTÃO
8              CALCULAR custo final para esse caminho primário - Fórmula (7.1)
9          SENÃO
10             DEFINIR custo final desse caminho primário como infinito
11         FIM SE
12     FIM PARA
13     CALCULAR menor custo final entre todos os caminhos primários candidatos
14     SE menor custo for infinito ENTÃO
15         DEFINIR caminho primário bloqueado
16         INTERROMPER % próximo evento
17     SENÃO
18         DEFINIR caminho primário encontrado
19     FIM SE
20     % Caminho de Proteção
21     SE caminho primário encontrado ENTÃO
22         REMOVER temporariamente enlaces do caminho primário do grafo da rede
23         CALCULAR k menores caminhos - Algoritmo (1)
24         PARA cada caminho encontrado FAÇA
25             CALCULAR slots para alocação - Algoritmo (3)
26             SE encontrar slots disponíveis ENTÃO
27                 PARA cada faixa utilizada como proteção por outras conexões FAÇA
28                     SE conexões tem c. primários disjuntos – Algoritmo (4) ENTÃO
29                         DEFINIR custo parcial do caminho de proteção - Fórmula (7.2)
30                     SENÃO
31                         DEFINIR custo parcial do caminho de proteção como infinito
32                     FIM SE
33                 FIM PARA
34                 PARA cada faixa de slots livres FAÇA
35                     DEFINIR custo parcial do caminho de proteção - Fórmula (7.3)
36                 FIM PARA
37                 CALCULAR menor custo parcial para esse caminho de proteção candidato
38                 CALCULAR custo final para esse caminho de proteção - Fórmula (7.4)
39             SENÃO
40                 DEFINIR custo final desse caminho de proteção como infinito
41             FIM SE
42         FIM PARA
43         CALCULAR menor custo final entre todos os caminhos de proteção candidatos
44         SE menor custo for infinito ENTÃO
45             DEFINIR caminho de proteção bloqueado
46             INTERROMPER % próximo evento
47         SENÃO
48             DEFINIR caminho de proteção encontrado
49         FIM SE
50     FIM SE
51     SE encontrar caminho de proteção e caminho primário ENTÃO
52         DEFINIR conexão aceita
53         ATUALIZAR matriz Cmat
54     FIM SE
55 FIM SE
56 SE evento for de fim de conexão ENTÃO
57     VERIFICAR se conexão foi aceita e/ou afetada por falha e ATUALIZAR matriz Cmat
58 FIM SE
```

```
59          SE evento for de falha em enlace ENTÃO
60              REMOVER enlace falho do grafo da rede até que este seja consertado
61              PARA cada conexão utilizando o enlace falho como caminho primário FAÇA
62                  SE caminho de proteção está livre ENTÃO
63                      REDIRECIONAR conexão
64                  SENÃO
65                      DEFINIR conexão bloqueada
66                  FIM SE
67                  ATUALIZAR matriz Cmat
68              FIM PARA
69          FIM SE
70          SE evento for de conserto de falha em enlace ENTÃO
71              ATUALIZAR grafo da rede, liberando enlace falho para uso
72          FIM SE
73      FIM ENQUANTO
74      FIM ALGORITMO
```

ALGORITMO 6 RSA_FF_SPP

```
1  ENQUANTO novo evento
2      SE evento for de chegada de conexão ENTÃO
3          % Caminho Primário
4          CALCULAR k menores caminhos - Algoritmo (1)
5          PARA cada caminho encontrado FAÇA
6              CALCULAR slots para alocação - Algoritmo (2)
7              SE encontrar slots disponíveis ENTÃO
8                  CALCULAR custo final para esse caminho primário - Fórmula (7.1)
9                  INTERROMPER % método FF! Vai para linha 14
10             SENÃO
11                 DEFINIR custo final desse caminho primário como infinito
12             FIM SE
13         FIM PARA
14         SE custo final for infinito ENTÃO
15             DEFINIR caminho primário bloqueado
16             INTERROMPER % próximo evento
17         SENÃO
18             DEFINIR caminho primário encontrado
19         FIM SE
20         % Caminho de Proteção
21         SE caminho primário encontrado ENTÃO
22             REMOVER temporariamente enlaces do caminho primário do grafo da rede
23             CALCULAR k menores caminhos - Algoritmo (1)
24             PARA cada caminho encontrado FAÇA
25                 CALCULAR slots para alocação - Algoritmo (3)
26                 SE encontrar slots disponíveis ENTÃO
27                     PARA cada faixa de slots disponível FAÇA
28                         SE for faixa utilizada como proteção por outras conexões ENTÃO
29                             SE conexões c. primários disjuntos – Algoritmo (4) ENTÃO
30                                 DEFINIR custo parcial do c. de proteção - Fórmula (7.2)
31                                 INTERROMPER % método FF! Vai para linha 40
32                             SENÃO
33                                 DEFINIR custo parcial do caminho de proteção infinito
34                             FIM SE
35                         SENÃO SE cada faixa de slots livres ENTÃO
36                             DEFINIR custo parcial do caminho de proteção - Fórmula
37                             (7.3)
38                             INTERROMPER % método FF! Vai para linha 40
39                         FIM SE
40                     FIM PARA
41                     CALCULAR custo final para esse caminho de proteção - Fórmula (7.4)
42                     INTERROMPER % método FF! Vai para linha 46
43                 SENÃO
44                     DEFINIR custo final desse caminho de proteção como infinito
45                 FIM SE
46             FIM PARA
47             SE custo final for infinito ENTÃO
48                 DEFINIR caminho de proteção bloqueado
49                 INTERROMPER % próximo evento
50             SENÃO
51                 DEFINIR caminho de proteção encontrado
52             FIM SE
53         FIM SE
54         SE encontrar caminho de proteção e caminho primário ENTÃO
55             DEFINIR conexão aceita
56             ATUALIZAR matriz Cmat
57         FIM SE
       FIM SE
```

```
58          SE evento for de fim de conexão ENTÃO
59              VERIFICAR se conexão foi aceita e/ou afetada por falha e ATUALIZAR matriz Cmat
60          FIM SE
61          SE evento for de falha em enlace ENTÃO
62              REMOVER enlace falho do grafo da rede até que este seja consertado
63              PARA cada conexão utilizando o enlace falho como caminho primário FAÇA
64                  SE caminho de proteção está livre ENTÃO
65                      REDIRECIONAR conexão
66                  SENÃO
67                      DEFINIR conexão bloqueada
68                  FIM SE
69                  ATUALIZAR matriz Cmat
70              FIM PARA
71          FIM SE
72          SE evento for de conserto de falha em enlace ENTÃO
73              ATUALIZAR grafo da rede, liberando enlace falho para uso
74          FIM SE
75      FIM ENQUANTO
76      FIM ALGORITMO
```

ALGORITMO 7 RSA_FFMLF_SPP

```
1  ENQUANTO novo evento
2      SE evento for de chegada de conexão ENTÃO
3          % Caminho Primário
4          CALCULAR k menores caminhos - Algoritmo (1)
5          PARA cada caminho encontrado FAÇA
6              CALCULAR slots para alocação - Algoritmo (2)
7              SE encontrar slots disponíveis ENTÃO
8                  CALCULAR custo final para esse caminho primário - Fórmula (7.1)
9                  INTERROMPER % método FF! Vai para linha 14
10             SENÃO
11                 DEFINIR custo final desse caminho primário como infinito
12             FIM SE
13         FIM PARA
14         SE custo final for infinito ENTÃO
15             DEFINIR caminho primário bloqueado
16             INTERROMPER % próximo evento
17         SENÃO
18             DEFINIR caminho primário encontrado
19         FIM SE
20         % Caminho de Proteção
21         SE caminho primário encontrado ENTÃO
22             REMOVER temporariamente enlaces do caminho primário do grafo da rede
23             CALCULAR k menores caminhos - Algoritmo (1)
24             PARA cada caminho encontrado FAÇA
25                 CALCULAR slots para alocação - Algoritmo (3)
26                 SE encontrar slots disponíveis ENTÃO
27                     PARA cada faixa de slots disponível FAÇA
28                         SE for faixa de slots utilizada como proteção por outras conexões
29                         ENTÃO
30                             SE conex. não tem c. primários disjuntos-Algoritmo(4)ENTÃO
31                                 DEFINIR slot não está disponível para uso
32                             FIM SE
33                         FIM SE
34                     FIM PARA
35                     CALCULAR maior índice de slot disponível
36                     CALCULAR custo final para esse caminho de proteção - Fórmula (8.1)
37                 SENÃO
38                     DEFINIR custo final desse caminho de proteção como infinito
39                 FIM SE
40             FIM PARA
41             CALCULAR menor custo final entre todos os caminhos de proteção candidatos
42             SE custo final for infinito ENTÃO
43                 DEFINIR caminho de proteção bloqueado
44                 INTERROMPER % próximo evento
45             SENÃO
46                 DEFINIR caminho de proteção encontrado
47             FIM SE
48         FIM SE
49         SE encontrar caminho de proteção e caminho primário ENTÃO
50             DEFINIR conexão aceita
51             ATUALIZAR matriz Cmat
52         FIM SE
       FIM SE
53     SE evento for de fim de conexão ENTÃO
54         VERIFICAR se conexão foi aceita e/ou afetada por falha e ATUALIZAR matriz Cmat de
55         acordo
       FIM SE
```

```
56          SE evento for de falha em enlace ENTÃO
57              REMOVER enlace falho do grafo da rede até que este seja consertado
58              PARA cada conexão utilizando o enlace falho como caminho primário FAÇA
59                  SE caminho de proteção está livre ENTÃO
60                      REDIRECIONAR conexão
61                  SENÃO
62                      DEFINIR conexão bloqueada
63                  FIM SE
64                  ATUALIZAR matriz Cmat
65              FIM PARA
66          FIM SE
67          SE evento for de conserto de falha em enlace ENTÃO
68              ATUALIZAR grafo da rede, liberando enlace falho para uso
69          FIM SE
70      FIM ENQUANTO
71      FIM ALGORITMO
```