

**UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
CAMPUS CHAPECÓ
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**DIMENSIONAMENTO DE TOPOLOGIAS LÓGICAS EM
REDES ÓPTICAS ELÁSTICAS DE TELECOMUNICAÇÕES
COM PROTEÇÃO DEDICADA**

TAILAN RODRIGO BONASSI

**CHAPECÓ
2017**

TAILAN RODRIGO BONASSI

**DIMENSIONAMENTO DE TOPOLOGIAS LÓGICAS EM
REDES ÓPTICAS ELÁSTICAS DE TELECOMUNICAÇÕES
COM PROTEÇÃO DEDICADA**

Trabalho de conclusão de curso de graduação
apresentado como requisito parcial para obten-
ção do grau de Bacharel em Ciência da Compu-
tação da Universidade Federal da Fronteira Sul.
Orientador: Prof. Ph.D. Claunir Pavan

CHAPECÓ
2017

PROGRAD/DBIB - Divisão de Bibliotecas

Bonassi, Tailan Rodrigo

Dimensionamento de Topologias Lógicas em Redes Ópticas Elásticas de Telecomunicações com Proteção Dedicada/ Tailan Rodrigo Bonassi. -- 2017.

70 f.:il.

Orientador: Ph.D. Claunir Pavan.

Trabalho de conclusão de curso (graduação) - Universidade Federal da Fronteira Sul, Curso de Ciência da Computação , Chapecó, SC, 2017.

1. RSA. 2. Roteamento. 3. Alocação de Espectro. 4. Redes Ópticas Elásticas. 5. Algoritmo Genético. I. Pavan, Ph.D. Claunir, orient. II. Universidade Federal da Fronteira Sul. III. Título.

TAILAN RODRIGO BONASSI

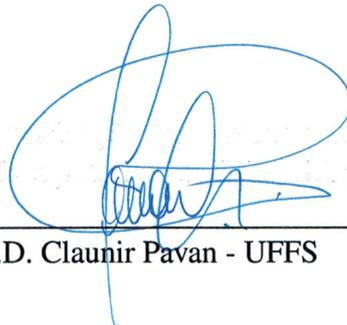
**DIMENSIONAMENTO DE TOPOLOGIAS LÓGICAS EM REDES
ÓPTICAS ELÁSTICAS DE TELECOMUNICAÇÕES COM PROTEÇÃO
DEDICADA**

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Prof. Ph.D. Claunir Pavan

Aprovado em: 02/03/2017

BANCA EXAMINADORA:



Ph.D. Claunir Pavan - UFFS



Ph.D. Marco Aurélio Spohn - UFFS



Me. Adriano Sanick Padilha - UFFS

AGRADECIMENTOS

Inicialmente agradeço a Deus ao dom da vida, à minha saúde e a força para a caminhada nessa etapa da minha vida. Agradeço a minha esposa Rubia A. Bonassi pelo apoio, amor, força, encorajamento e compreensão durante a elaboração desse trabalho e durante todo o curso, agradeço também à sua ajuda com as palavras para completar uma frase ou parágrafo. Agradeço aos meus pais, pela força e apoio em me encorajar a mudar de cidade para cursar esse curso e me manter perseverante na busca do meu objetivo.

Agradeço ao meu orientador, professor Ph. D. Claunir Pavan por toda dedicação e empenho em me auxiliar na elaboração desse trabalho e principalmente por ampliar o horizonte do meu conhecimento. Agradeço aos professores Me. Adriano Sanick Padilha e Ph.D. Marco Aurélio Spohn por aceitarem participar da banca e construção desse trabalho. Agradeço ao corpo docente do curso de Ciência da Computação da Universidade Federal da Fronteira Sul, pelos ensinamento e me capacitarem à elaborar esse trabalho. Por fim, agradeço aos meus colegas e amigos pelo companheirismo durante a longa caminhada. A todos, muito obrigado.

“Jamais considere seus estudos como uma obrigação, mas como uma oportunidade invejável para aprender a conhecer a influência libertadora da beleza do reino do espírito, para seu próprio prazer pessoal e para proveito da comunidade à qual seu futuro trabalho pertencer.”

— ALBERT EINSTEIN

RESUMO

Nos últimos anos, o tráfego de dados nas redes de telecomunicações tem aumentado de forma expressiva, devido a diversas aplicações emergentes na Internet, como transmissão de vídeos em alta e ultra definição, compartilhamento de imagens e dados nas redes sociais, exploração do mercado de equipamentos móveis e *smartphones*, grandes centros de dados e a computação em nuvem. Esse aumento na demanda de tráfego, exige que as redes de telecomunicações aprimorem suas estruturas e seus gerenciamentos para manter a qualidade do serviço prestado, bem como, melhor utilizar seus recursos e aumentar sua capacidade de suporte ao tráfego. Nesse contexto, esse trabalho propõe, o desenvolvimento de novos métodos para auxílio à engenharia de tráfego, para geração *off-line* de topologias lógicas de redes de transporte com proteção dedicada, para falhas de um enlace, em redes ópticas elásticas. O dimensionamento da topologia lógica em redes elásticas, consiste em estabelecer os caminhos em que o tráfego de uma determinada demanda irá percorrer, através do roteamento e a alocação do espectro no domínio óptico da rede, definindo o comprimento de onda e a largura de banda necessários para atender ao tráfego. O problema de roteamento e alocação de espectro é conhecido como RSA. Combinando heurísticas já existentes, para políticas de roteamento e alocação de espectro, com inteligência artificial, mais especificamente algoritmo genético, são apresentados dois métodos para geração e otimização das topologias lógicas, apresentando uma redução na probabilidade de bloqueio de até 35%, nos experimentos realizados.

Palavras-chave: RSA. Roteamento. Alocação de Espectro. Redes Ópticas Elásticas. Algoritmo Genético.

ABSTRACT

In recent years, data traffic on telecommunications networks has increased significantly, due to several emerging Internet applications, such as streaming videos in high and ultra definition, images and data sharing on social networks, mobile equipment market exploration and smartphones, data centers and cloud computing. This increase in traffic demand, requires that telecommunications networks improve their structures and their management schemes to maintain a certain level of quality of the service. Additionally, it is required a better use of resources and support to traffic load. In this context, this work proposes the development of new methods for traffic engineering support, for offline generation of logical topologies for transport networks that operates with dedicated protection in elastic optical networks. The dimensioning of logical topology in elastic networks, involve the establishment of lightpaths and spectrum allocation in the optical domain. The routing and spectrum allocation problem is known as RSA. We present two methods to generate optimized logical topologies from heuristics for routing and allocation of espectrum and considering genetic algorithms. Experiments show a reduction of up to 35% in the blocking probability.

Keywords: RSA, Routing, Spectrum Allocation, Elastic Optical Networks, Genetic Algorithm.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 2.1 – Topologias lógica e física..... | 16 |
| Figura 2.2 – Exemplo de matriz de demanda (Gbps). | 17 |
| Figura 2.3 – Comparação entre grade fixa e grade flexível | 18 |
| Figura 2.4 – Distribuição do espectro com banda de guarda. | 20 |
| Figura 2.5 – Continuidade e contiguidade do espectro..... | 21 |
| Figura 2.6 – Exemplo de alocação do espectro..... | 21 |
| Figura 2.7 – Probabilidade de bloqueio de políticas de alocação de espectro em RSA | 25 |
| Figura 2.8 – Esquemas de proteção. | 26 |
| Figura 2.9 – Representação gráfica do algoritmo de Suurballe..... | 28 |
| Figura 3.1 – Exemplo de cruzamento de ponto único..... | 33 |
| Figura 3.2 – Exemplo de iteração do algoritmo genético. | 34 |
| Figura 4.1 – Algoritmo Genético | 38 |
| Figura 5.1 – Arquitetura do nó da rede. | 42 |
| Figura 5.2 – Representação do Indivíduo. | 50 |
| Figura 5.3 – Cruzamento..... | 52 |
| Figura 5.4 – Mutação | 52 |
| Figura 6.1 – topologias de Redes | 56 |
| Figura 6.2 – Probabilidades de boqueio - NSFNET..... | 59 |
| Figura 6.3 – Probabilidades de boqueio - COST37 | 61 |
| Figura 6.4 – Probabilidades de boqueio - INTERNET2 | 62 |
| Figura 6.5 – Probabilidades de boqueio - USA100 | 63 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 2.1 – Larguras de bandas necessárias por taxa de transferência | 19 |
| Tabela 2.2 – Probabilidade de bloqueio de políticas de alocação de espectro em RSA | 24 |
| Tabela 3.1 – Exemplo de aptidão acumulada..... | 32 |
| Tabela 6.1 – Comparação das médias dos resultados | 64 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|--------|--|
| AG | Algoritmo Genético |
| BER | <i>Bit Error Rate</i> |
| BV-WXC | <i>Bandwidth-Variable Wavelength Cross-Connects</i> |
| BVT | <i>Bandwidth-Variable Transponder</i> |
| EF | <i>Exact Fit</i> |
| FF | <i>First Fit</i> |
| FLF | <i>First-Last Fit</i> |
| ILP | <i>Integer Linear Programming</i> |
| KDPPs | <i>K - Disjoint-Paths Pair</i> |
| LF | <i>Last Fit</i> |
| LPF | <i>Longest Path First,</i> |
| MSF | <i>Most Subcarriers First</i> |
| NMPCD | NxM Pares de Caminhos Disjuntos |
| RF | <i>Random Fit</i> |
| Gbps | Giga Bits por Segundo |
| RMLSA | <i>Routing, Modulation Level and Spectrum Allocation</i> |
| RSA | <i>Routing and Spectrum Allocation</i> |
| TE | <i>Traffic Engineering</i> |
| WDM | <i>Wavelength Division Multiplexing</i> |

LISTA DE SÍMBOLOS

| | |
|--------------|---|
| G | Grafo |
| V | Vértices |
| E | Arestas |
| \leftarrow | Atribuição |
| \cup | União |
| \cap | Intersecção |
| D | Conjunto das demandas |
| B | Conjunto das demandas bloqueadas |
| Pbd | Probabilidade de bloqueio de demanda |
| $ X $ | Tamanho do conjunto X |
| Pbs | Probabilidade de bloqueio de <i>slot</i> |
| \sum | Somatório |
| S | Conjunto de <i>slots</i> de um enlace |
| \in | Pertence |
| P | Conjunto de pares de caminhos disjuntos |
| \subset | Contido |
| \emptyset | Conjunto vazio |
| C_d | Conjunto dos pares de caminhos disjuntos para uma determinada demanda |
| \neq | Diferente |
| $>$ | Maior que |
| Ind | Indivíduo do algoritmo genético |
| O | Vetor da ordem das demandas |
| Sel | Vetor dos índices dos caminhos selecionados |
| Pop | População do algoritmo genético |
| $<$ | Menor que |

SUMÁRIO

| | |
|--|----|
| 1 INTRODUÇÃO | 13 |
| 2 REDES ÓPTICAS DE TELECOMUNICAÇÕES | 16 |
| 2.1 Redes elásticas | 17 |
| 2.1.1 RSA | 20 |
| 2.1.1.1 Roteamento | 22 |
| 2.1.1.2 Alocação de espectro | 23 |
| 2.2 Proteção de rede | 25 |
| 2.2.1 Proteção dedicada por caminho em redes elásticas | 28 |
| 3 ENGENHARIA DE TRÁFEGO | 30 |
| 3.1 Algoritmos Genéticos | 30 |
| 4 ESTUDOS RELACIONADOS | 35 |
| 5 PROPOSTA | 41 |
| 5.1 Formulação do problema | 41 |
| 5.2 Implementação | 44 |
| 5.2.1 Roteamento | 44 |
| 5.2.2 Alocação de Espectro | 47 |
| 5.2.3 Algoritmo Genético Proposto | 49 |
| 6 EXPERIMENTOS E RESULTADOS | 55 |
| 6.1 Apresentação dos Resultados | 58 |
| 7 CONSIDERAÇÕES FINAIS | 66 |
| REFERÊNCIAS | 67 |

1 INTRODUÇÃO

O tráfego de dados nas redes ópticas de telecomunicações tem aumentando ano após ano com as novas aplicações de redes emergentes, e a expectativa é que continue aumentando exponencialmente [Yu et al., 2014]. São muitas as novas tendências de aplicativos de rede, pode-se citar a transmissão de vídeo em alta ou ultra definição, transmissão de vídeo em tempo real, redes sociais, a *Internet Protocol TeleVision* (IPTV), computação em nuvem, gerenciamento de grandes bancos de dados (*Big Data*), jogos *on-line* dentre outros.

Com esse crescimento acelerado, as redes se tornaram cada vez maiores e complexas, dificultando seu dimensionamento por engenheiros de tráfego, que precisam de ferramentas inteligentes para auxiliar nas tomadas de decisões. As escolhas feitas na implantação de uma rede impactam diretamente no seu custo e na qualidade de serviço prestado, portanto devem ser profundamente analisadas. Para isso, novos métodos de dimensionamento devem ser estudados e propostos, para que as mesmas tenham o melhor desempenho ao menor custo possível, através da otimização do uso de seus recursos.

O uso das grades fixas da tecnologia WDM (*Wavelength Division Multiplexed*) com *slots*¹ de 50 ou 100 GHz, têm possibilitado um crescimento acelerado nas técnicas de transmissão de dados em redes de telecomunicações por meio ótico, um aumento de 2.5 Gbps para até 100 Gbps, [Wright et al., 2013]. Entretanto, essa evolução nos meios de transmissão óticos WDM já não acompanham a evolução da demanda.

Uma evolução emergente no campo das redes de telecomunicações são as redes elásticas. Nos equipamentos tradicionais de telecomunicações em meio ótico WDM, os *slots* de frequência possuem um tamanho fixo de largura de banda, já nas redes elásticas os *slots* são flexíveis, assim é possível alocar larguras de bandas maiores ou menores de acordo com a necessidade do tráfego dimensionado em cada caminho [ITU, 2012]. Dessa forma o espectro é melhor utilizado, evitando o desperdício de espectro não ocupado, pois as alocações de *slots* tendem a se adequar às necessidades das demandas.

Com o advento das redes elásticas, novos desafios são propostos, como o problema do RSA (*Routing and Spectrum Allocation*), roteamento e alocação de espectro, que consiste em encontrar os caminhos entre todos os pontos de origem aos pontos de destino demandados, alocando o número de *slots* necessários para atender às demandas em cada enlace dos caminhos

¹ Faixa de frequência para transmissão de dados no meio ótico

encontrados. Esse problema não pode ser tratado através de uma análise de todas as possibilidades, pois se trata de um problema *NP-Hard* [Morais et al., 2011], sendo assim, é necessário o uso de algoritmos de otimização por aproximação, como algoritmos genéticos que exploram os conceitos evolucionistas para encontrar uma aproximação à solução ótima.

O presente trabalho aborda o problema do RSA para redes ópticas elásticas com proteção dedicada, para falha em um enlace. Serão apresentadas três técnicas de dimensionamento *off-line*, sendo uma implementação de uma das soluções apresentada por [Chatterjee et al., 2015], utilizando a política de roteamento fixo e *First-last Fit* (FLF) como política de alocação de espectro, que consiste em alocar, para cada demanda, o par de caminhos mínimos disjuntos (caminhos com menor número de saltos), considerando a disponibilidade de faixas de espectro nos enlaces para atender a demanda. As outras duas, são novas implementações propostas e apresentadas nesse trabalho, baseadas no uso de algoritmos genéticos para otimizar o uso dos recursos da rede, minimizando o número de demandas não atendidas, os chamados bloqueios.

As duas implementações propostas, utilizam a política de roteamento chamada de Roteamento Fixo Alternativo [Chatterjee et al., 2015], com o uso de um algoritmo genético para escolha de um dos caminhos candidatos para atender a cada uma das demandas solicitadas. O algoritmo genético também leva em consideração a heurística do algoritmo genético apresentado em [Klinkowski, 2012], onde alterna a ordem de processamento de demandas pelo algoritmo de roteamento e alocação de espectro, para encontrar uma permutação na ordem de processamento que minimize a probabilidade de bloqueio. Assim, o algoritmo genético proposto, possui duas características para gerar combinações a fim de minimizar a probabilidade de bloqueio, (i) a ordem de processamento da demanda para alocação do espectro, como em [Klinkowski, 2012]; e (ii) a escolha do caminho entre os caminhos candidatos de cada demanda, como nova heurística, com FLF como política de alocação de espectro, assim como em [Chatterjee et al., 2015].

A diferença entre as duas implementações propostas, está em seus métodos de geração de caminhos candidatos. Uma utiliza a geração de k pares de caminhos candidatos utilizando o algoritmo proposto em [Phung et al., 2005], onde primeiramente, um caminho mínimo entre os nós origem e destino é encontrado, sendo o caminho de trabalho, e posteriormente são encontrados k caminhos disjuntos, ao caminho de trabalho, para serem caminhos de *backup*. E um novo algoritmo de geração de caminhos candidatos foi proposto, onde são gerados $N \times M$ pares de caminhos disjuntos. Esse algoritmo busca N caminhos mínimos para cada par origem-destino como caminhos de trabalho, e para cada um destes, é encontrado M caminhos disjuntos para o

caminhos de *backup*.

Finalmente, os resultados de cada solução são comparados, considerando-se como métricas a probabilidade de bloqueio de demandas e a probabilidade de bloqueio por *slots*.

2 REDES ÓPTICAS DE TELECOMUNICAÇÕES

Nas últimas décadas vemos um aumento significativo no tráfego de dados na internet, vivemos na era da informação onde se busca compartilhar e transmitir dados de vídeo, voz, imagens, informações de grandes centros de dados dentre outros. Isso cria uma demanda de tráfego que exige a evolução dos métodos de transmissão de dados em redes de telecomunicações, forçando que novas tecnologias de transmissões sejam criadas e exploradas em larga escala. Dentre as tecnologias que avançaram para atender essas exigências, a fibra óptica trouxe enorme evolução no meio de transmissão de dados em alta largura de banda, trazendo o potencial de taxa de transferência para a casa de centenas ou milhares de Gbps (Giga Bits por Segundo), além de prover altíssima confiabilidade na transmissão, sendo que seu *Bit Error Rate* (BER) é inferior a 10^{-12} [Bouillet et al., 2007].

As redes de telecomunicações são representadas por topologias que definem seus esquema de ligações, entre seus nós. A topologia física representa como os nós são ligados entre si no meio físico, através dos enlaces, oferecendo informações sobre os enlaces e os nós, como por exemplo, o comprimento da fibra do enlace que liga dois nós, o número de *slots* do enlace, etc. Já a topologia lógica, representa como os fluxos de dados ocorrem, ou seja, modela os caminhos alocados no meio físico para atender uma demanda de tráfego. A topologia lógica também define a largura de banda dos caminhos, o número de saltos no meio físico de cada caminho, etc.

Conforme representado na Figura 2.1, a topologia física modela as ligações entre os nós da rede, nomeados de A a F. Já a topologia lógica representa como os fluxos de dados são modelados sobre a camada física da rede, por exemplo, o fluxo do nó A para o nó C corresponde ao caminho formado pelos enlaces físicos A->B e B->C.

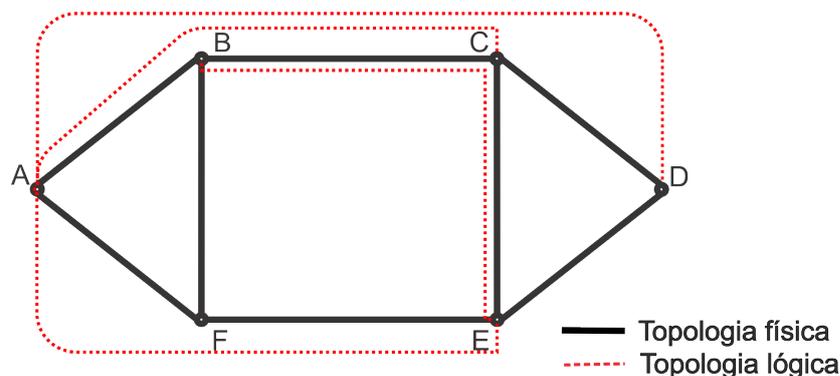


Figura 2.1 – Topologias lógica e física

A topologia lógica precisa além dos caminhos, as demandas de tráfego, ou seja, a largura de banda ou a taxa de transferência que serão utilizados no caminho. Para isso, comumente utiliza-se uma matriz de demandas, onde cada posição representa a demanda necessária para o caminho lógico entre o nó representado pelo índice da linha e o nó representado pelo índice da coluna.

A Figura 2.2, mostra um exemplo de matriz de demanda, onde foi representada uma demanda no caminho do nó A para o nó C de 40 Gbps; do nó A para o nó D de 100 Gbps; do nó A para o nó E de 10 Gbps e do nó B para o nó E de 400 Gbps. Portanto, para melhor utilizar os recursos físicos de uma rede óptica de telecomunicação, é necessário o dimensionamento de uma topologia lógica que suporte a demanda indicada, consumindo o mínimo de recursos possível da topologia física.

| | A | B | C | D | E | F |
|---|---|---|----|-----|-----|---|
| A | x | 0 | 40 | 100 | 10 | 0 |
| B | x | x | 0 | 0 | 400 | 0 |
| C | x | x | x | 0 | 0 | 0 |
| D | x | x | x | x | 0 | 0 |
| E | x | x | x | x | x | 0 |
| F | x | x | x | x | x | x |

Figura 2.2 – Exemplo de matriz de demanda (Gbps).

Para aumentar ainda mais a capacidade do tráfego de dados em fibras ópticas, foi proposto o *Wavelength Division Multiplexed* (WDM), que consiste em transmitir em uma única fibra diversos comprimentos de onda simultaneamente, fornecendo uma utilização maior do espectro disponível para a transmissão de dados no meio ótico. Para isso o espectro é dividido em *slots* controlados através da multiplexação e demultiplexação das faixas de frequência em cada nó da rede [Bouillet et al., 2007]. Recentemente esses *slots* se tornaram flexíveis, abrindo novas possibilidades para configuração e otimização na alocação e ocupação do espectro, as chamadas redes elásticas.

2.1 Redes elásticas

Uma evolução recente no campo de redes ópticas são as redes elásticas, que tem como características a possibilidade de personalizar o tamanho de faixas de frequência na divisão do

espectro em fibras ópticas. O conceito é alocar *slots* com tamanhos adequados de acordo com a largura de banda fim-a-fim, de modo a utilizar de forma eficiente todo o espectro. Dessa forma, minimiza-se o desperdício de faixas de frequência, pois os *slots* podem se expandir ou contrair para se adequar ao fluxo de cada largura de banda, diferentemente dos equipamentos com *slots* rígidos cujas divisões são fixas, assim quando uma largura de banda alocada é menor que o tamanho da largura de banda do *slot* o excedente é desperdiçado [Jinno et al., 2009].

Na Figura 2.3, pode-se observar a comparação entre a distribuição em uma grade fixa e em uma grade flexível com o conceito de rede elástica, enquanto que na primeira, para alocar larguras de banda para taxas de transferências de 10 Gbps, é utilizado um *slot* de 50 GHz, sobrando parte do espectro sem uso, na rede elástica as larguras de banda são acomodadas de forma a maximizar a utilização do espectro, sendo possível diminuir os *slots* para larguras de banda menores e alocar maiores *slots* para larguras de banda maiores, como por exemplo, p a de 1 Tbps que está alocada dinamicamente num *slot* maior, sendo que numa grade fixa não seria possível ter essa divisão, sendo necessário particionar a mesma em diversos *slots*.

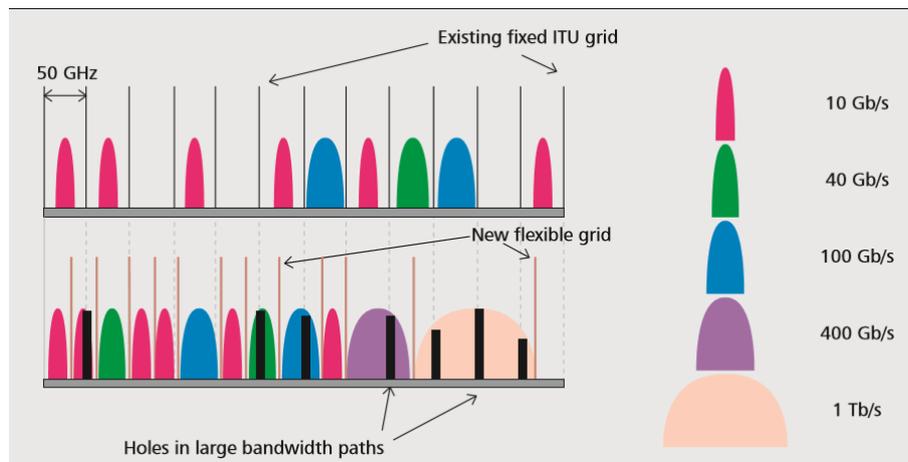


Figura 2.3 – Comparação entre grade fixa e grade flexível. Fonte: [Gerstel et al., 2012]

As redes elásticas possuem uma granularidade de 6.25, 12.5 ou 25 GHz, ou seja, inferior às utilizadas nas redes WDM que utilizam *slots* de 50 GHz. A chave da tecnologia da rede elástica, consiste em gerar larguras de bandas variáveis, múltiplos de sua granularidade, alocando um número variado de *slots*, com tamanho de sua granularidade, adjacentes para formar uma largura de banda única, representada pelo seu comprimento de onda central e a quantidade de *slots* adjacentes, conforme explica [Wright et al., 2013]. Assim, em num mesmo enlace, utilizando uma granularidade de 12.5 GHz, pode-se configurar uma largura de banda de 12.5 GHz com um *slot*, e um largura de banda de 125 GHz, alocando 10 *slots* adjacentes. Dessa forma é

possível dimensionar melhor a utilização do espectro na transmissão de dados nas redes ópticas de telecomunicações, evitando que larguras de banda sejam alocadas desnecessariamente.

De acordo com a taxa de transferência que deve ser dimensionada na rede, uma largura de banda diferente é necessária para transmissão, conforme Tabela 2.1. Para dimensionar um *link* com taxa de transferência de 400 Gbps é necessário uma largura de banda de 75 GHz, em uma rede convencional WDM com *slots* de 50 GHz, seria necessário dividir a demanda em 4 *links* de 100 Gbps, o máximo que pode ser alocado em um *slot*, ou seja, seriam necessários 4 *slots* ocupando um total de 200 GHz de largura de banda no espectro disponível. Já utilizando uma grade flexível com granularidade de 12.5 GHz, seria possível alocar 6 *slots* adjacentes de 12.5 GHz, totalizando 75 GHz. Comparando com a arquitetura WDM, isso gera uma sobra de 125 GHz, onde, por exemplo, é possível alocar mais um *link* de 400 Gbps.

| Taxa de transferência (Gbps) | Largura de banda do espectro (GHz) | Formato da modulação |
|------------------------------|------------------------------------|----------------------|
| 40 | 25 | DP-QPSK |
| 100 | 37.5 | DP-QPSK |
| 100 | 25 | DP-16QAM |
| 400 | 75 | DP-16QAM |
| 1000 | 187.5 | DP-16QAM |

Tabela 2.1 – Larguras de bandas necessárias por taxa de transferência. Adaptada de: [Wright et al., 2013]

Entretanto, de acordo com [Gerstel et al., 2012], para conseguir enviar múltiplas larguras de banda em uma mesma fibra óptica com grade flexível, é necessário deixar um ou mais *slots* entre os *links* como banda de guarda, para que o emissor e o receptor, não sofram interferência no sinal provocado por larguras de bandas adjacentes. Dessa forma, ao alocar uma largura de banda de 25 GHz, serão necessários de 3 a 4 *slots* de 12.5 GHz, sendo 2 para atender à taxa de transferência e um ou dois *slots* para banda de guarda à esquerda ou à direita. Conforme exemplificado na Figura 2.4, a cada largura de banda, um espaço entre larguras de banda adjacentes é deixada para servir como banda de guarda.

Encontrar a melhor forma de alocar o espectro de uma topologia lógica é um desafio para as futuras redes de telecomunicações. Diversos estudos sobre dimensionamento de rede são focos atuais de estudos, por exemplo, fragmentação e desfragmentação do espectro, agregação e divisão de tráfego, etc. No âmbito deste trabalho, o foco se dá ao problema de roteamento e alocação de espectro (RSA) em redes ópticas elásticas.

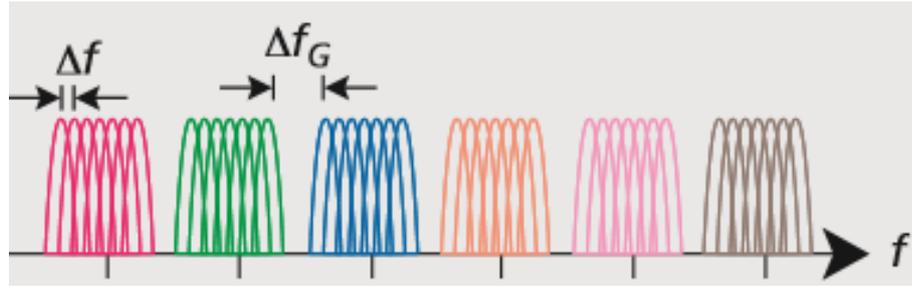


Figura 2.4 – Distribuição do espectro com banda de guarda. Fonte: Adaptado de [Gerstel et al., 2012]

2.1.1 RSA

Em redes ópticas elásticas de telecomunicações, um dos maiores desafios é o dimensionamento da demanda de tráfego sobre a mesma. A possibilidade de flexibilizar as faixas de frequência do espectro, com o uso das redes elásticas, amplia o poder para ocupar de forma massiva toda a faixa do espectro disponível para transmissão de dados, aumentando assim sua capacidade de tráfego. Encontrar a melhor forma de dimensionar a topologia lógica em redes ópticas elásticas pode ser descrito como o problema de roteamento e alocação de espectro, o RSA. Consiste em encontrar para cada demanda, um caminho e a alocação de faixas de frequência, *slots*, nos enlaces pertencentes ao caminho, buscando uma melhor utilização dos recursos e ampliando a capacidade de fluxo de tráfego de dados.

Segundo [Chatterjee et al., 2015], o problema do RSA em redes ópticas elásticas é equivalente ao problema do *Routing and Wavelength Assignment*(RWA) para redes ópticas WDM. O RWA também é um problema baseado em encontrar os caminhos e a seleção apropriada do comprimento de onda para ser utilizados nos enlaces do caminho.

Nas redes ópticas WDM transparentes, não há conversão de comprimentos de ondas ao longo do caminho ótico, portanto um caminho mantém o mesmo comprimento de onda em todos os enlaces do caminho, essa restrição é conhecida com continuidade.

Ainda de acordo com [Chatterjee et al., 2015], a diferença entre o RSA e o RWA é a capacidade das redes ópticas elásticas de oferecerem flexibilidade da grade de slots de frequência [ITU, 2012]. Para isso, é necessário que seja definido o número de *slots* para compor o espectro que atenda adequadamente à taxa de transferência a ser alocada. Assim, uma nova restrição é adicionada ao problema do RSA, a contiguidade do espectro. Essa restrição define que os *slots* necessários para atender uma determinada demanda sejam alocados de forma contígua em todos os enlaces do caminho, ou seja, os *slots* devem ser faixas de frequências adjacentes

no espectro.

Para resolver o problema do RSA, as restrições de continuidade e de contiguidade do espectro devem ser respeitadas. A representação da Figura 2.5 exemplifica essas restrições, a contiguidade que garante que *slots* adjacentes em um mesmo enlace sejam alocados e a restrição de continuidade que garante que os *slots* de mesmo comprimento de onda sejam alocados em todos os enlaces do caminho.

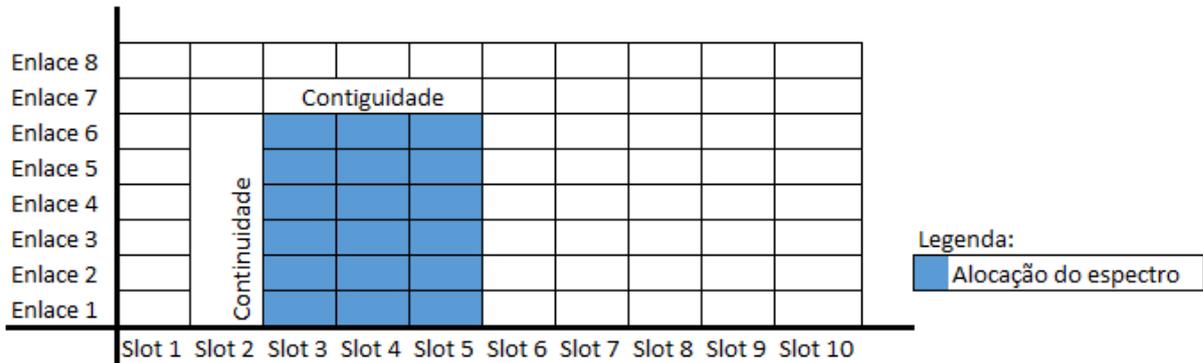


Figura 2.5 – Continuidade e contiguidade do espectro.

Tomando a alocação de espectro da Figura 2.6 para exemplificar as restrições de continuidade e contiguidade, onde está alocado um caminho do nó D para o nó E, em verde, com uma demanda de 3 *slots*; um caminho do nó A para o nó B, passando pelos enlaces F<->A e B<->F, em azul, com uma demanda de 2 *slots*; e um caminho do nó F ao nó C, passando pelos enlaces E<->F, D<->E e C<->D, em laranja, com uma demanda de um *slot*.

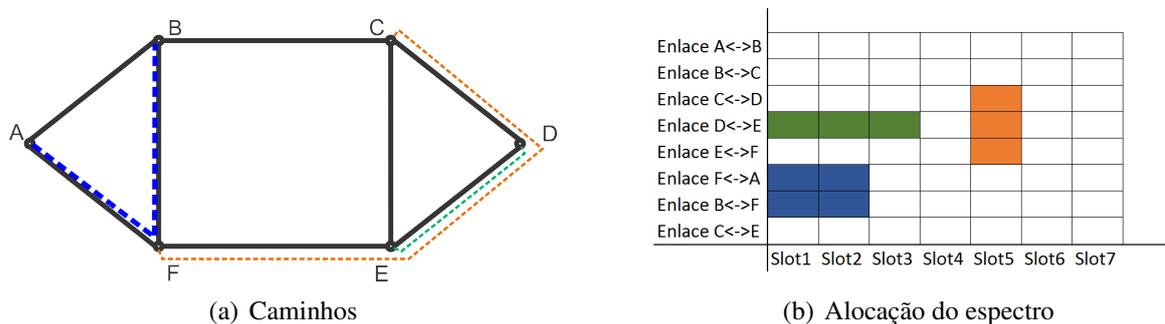


Figura 2.6 – Exemplo de alocação do espectro.

Observa-se que para alocar uma nova demanda de 2 *slots* do nó A ao nó C, através de um caminho definido pelos enlaces F<->A, E<->F e D<->E, não é possível alocar os *slots* 3 e 4 pois os mesmos não são contínuos em todos os enlaces, pois o *slot* 3 já está alocado no enlace D<->E. Também não é possível alocar os *slots* 4 e 6, pois apesar de serem contínuos, estes não

são contíguos. Assim a alocação deve ser feita nos *slots* 6 e 7 para satisfazer às restrições.

Quando uma demanda for solicitada, e não é possível encontrar em todos os enlaces do caminho roteado o número de *slots* solicitados, que satisfaçam as restrições de continuidade e contiguidade, então essa demanda é bloqueada. O problema do RSA também estuda formas de minimizar a probabilidade de bloqueio, que é o número de bloqueios que ocorrem em relação ao número de requisições totais.

Os problemas do RSA e do RWA são de complexidade *NP-Hard* [Morais et al., 2011], portanto não há uma solução para os mesmos em tempo hábil de computação, dessa forma métodos probabilísticos, heurísticos e de inteligência artificial são amplamente utilizados para solucioná-los. Devido à complexidade do problema do RSA, a abordagem da solução pode ser dividida em dois subproblemas: o roteamento e a alocação do espectro.

2.1.1.1 Roteamento

O subproblema do roteamento consiste em encontrar caminhos sobre uma topologia física, entre os nós origem-destino das demandas. Em [Chatterjee et al., 2015] o autor lista as principais políticas de roteamentos no meio ótico:

- **Roteamento Fixo:** Uma rota única é pré-computada para cada origem-destino, usando um algoritmo de caminho mínimo, por exemplo o algoritmo de Dijkstra. Para estabelecer esse caminho é verificado se há *slots* necessários em todos os enlaces deste caminho para atender a demanda, utilizando alguma política de alocação de espectro, se não houver então esse caminho é bloqueado;
- **Roteamento Fixo Alternativo:** É uma versão melhorada do roteamento fixo, nesse método para cada nó da rede, são pré-computados um conjunto rotas para os demais nós da rede. Assim, ao solicitar uma demanda, a partir do nó de origem busca-se encontrar a disponibilidade de espectro, em número de *slots*, em sua lista de caminhos alternativos. Se em nenhum dos caminhos candidatos houver a disponibilidade de espectro, a demanda será bloqueada. Apesar desse método minimizar a probabilidade de bloqueio em relação ao roteamento de rota fixa, sua solução ainda não é ideal, pois o método pode não encontrar todas as possibilidades de rotas entre um par origem-destino;
- **Roteamento Menos Congestionado:** Através de um algoritmo de roteamento, são pré-determinados um conjunto de caminhos para cada par origem-destino, no momento de

fazer a alocação de espectro os caminhos com mais *slots* disponíveis são escolhidos. A desvantagem dessa estratégia em relação às anteriores, é sua complexidade computacional, com uma probabilidade de bloqueio muito próxima do roteamento fixo alternativo;

- **Roteamento Adaptativo:** Nesta política, os caminhos são definidos no momento de sua alocação do espectro, observando o estado atual dos enlaces na rede. Dessa forma, a probabilidade de bloqueio é inferior às estratégias anteriores, porém sua implantação é complexa e necessita de amplos protocolos de controle e gestão contínua das tabelas de roteamentos nos nós.

2.1.1.2 Alocação de espectro

Neste subproblema, o tema é a alocação de espectro nos caminhos definidos para a topologia lógica. Em redes elásticas, a alocação de espectro consiste em definir quais *slots* serão alocados ao longo dos enlaces para cada demanda, respeitando as restrições de continuidade e contiguidade de espectro. As seguintes políticas de alocação de espectro foram listadas por [Chatterjee et al., 2015]:

- **First Fit (FF):** Nessa política, os *slots* são indexados em ordem numérica crescente e uma lista desses índices é mantida, marcando os *slots* disponíveis e ocupados. Para alocação, são escolhidos os *slots* disponíveis com menor índice que satisfaçam as restrições de continuidade e contiguidade em todos os enlaces do caminho a ser alocado. A implementação dessa política não requer informações globais da rede, pois é possível estabelecer o caminho óptico através das listas indexadas dos *slots* dos enlaces pertencentes ao caminho. O FF é considerado uma das melhores políticas de alocação de espectro, devido à baixa probabilidade de bloqueio com baixo custo computacional;
- **Randon Fit (RF):** Uma lista de *slots* disponíveis é mantida, e quando chega uma solicitação de demanda, esta política escolhe aleatoriamente um conjunto de *slots* para atendê-la. Após a alocação dos *slots*, os mesmos são removidos das listas de disponíveis, mantendo assim essa lista sempre atualizada com os *slots* disponíveis. Selecionando o espectro de forma aleatória, minimiza a probabilidade de múltiplas conexões escolherem o mesmo comprimento de onda quando a alocação de espectro é executada de forma distribuída;
- **Last Fit (LF):** Esta política é similar a FF, porém busca alocar os *slots* disponíveis com maior índice;

- **First-Last Fit (FLF)**: Todos os *slots* de cada enlace, são agrupados em partições. Dessa forma a política consiste em: (i) nas partições de número ímpar, tenta-se alocar os *slots* com menor índice, como o FF; (ii) nas partições de número par os *slots* com maior índices são alocados, como a política LF. Dessa forma, busca-se manter mais regiões do espectro com *slots* disponíveis contíguos, reduzindo assim a probabilidade de bloqueio em relação às políticas anteriores;
- **Exact Fit (EF)**: Iniciando do *slot* com menor índice, essa política busca encontrar um intervalo com o mesmo número de *slots* disponíveis que a demanda a ser alocada. Se encontrar um intervalo com o exato tamanho necessário para a demanda, alocará os *slots* para atender a mesma, se não encontrar, utiliza a política FF. Selecionando o espectro dessa forma, reduz a fragmentação do espectro e conseqüentemente, reduz a probabilidade de bloqueio.

As estratégias são comparadas na Tabela 2.2, onde através de testes implementados na rede NSFNET, obteve-se os números de probabilidade de bloqueios para as políticas de alocação de espectro demonstradas.

| políticas de Alocação de Espectro | 20 Requisições | 40 Requisições | 60 Requisições | 80 Requisições | 100 Requisições |
|-----------------------------------|----------------|----------------|----------------|----------------|-----------------|
| <i>First-last Fit</i> | 0.005 | 0.007 | 0.010 | 0.017 | 0.025 |
| <i>First-exact Fit</i> | 0.006 | 0.008 | 0.012 | 0.020 | 0.027 |
| <i>First Fit</i> | 0.006 | 0.009 | 0.014 | 0.021 | 0.029 |
| <i>Last Fit</i> | 0.007 | 0.008 | 0.015 | 0.020 | 0.030 |
| <i>Randon Fit</i> | 0.017 | 0.022 | 0.031 | 0.043 | 0.055 |

Tabela 2.2 – Probabilidade de bloqueio de políticas de alocação de espectro em RSA. Adaptada de: [Chatterjee et al., 2015]

Pode-se observar que a política FLF se mostrou mais eficiente em relação às demais políticas, proporcionando uma baixa probabilidade de bloqueio nos cenários testados. Em seguida, a política EF ligeiramente mais bloqueante que a FLF. As políticas FF e FL em sequência com probabilidades de bloqueios similares. E finalmente a política RF que possui probabilidade de bloqueio bem superior às demais.

No gráfico da Figura 2.7, pode-se observar que a política FLF teve a menor probabilidade de bloqueio em todos os casos de testes, já a política RF foi a com probabilidade de bloqueio mais alta.

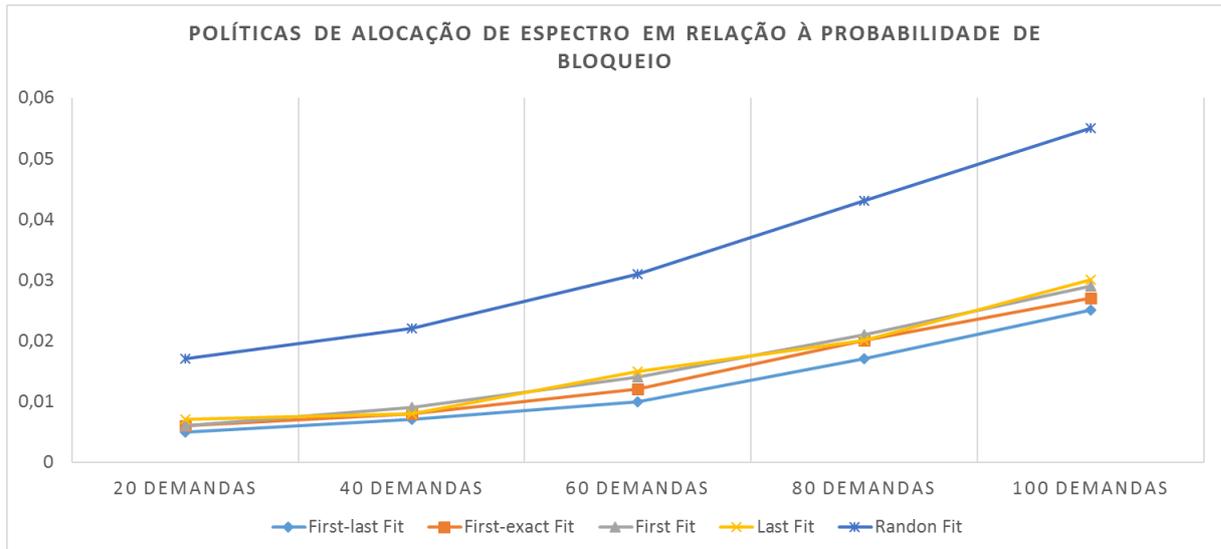


Figura 2.7 – Probabilidade de bloqueio de políticas de alocação de espectro em RSA.

2.2 Proteção de rede

As redes de telecomunicações possuem suas redes físicas expostas a diversos agentes que podem danificar a mesma. Para garantir o funcionamento da rede após uma falha em algum enlace, métodos de proteção são propostos para garantir um caminho alternativo aos tráfegos que ali passavam, também chamado de métodos de sobrevivência da rede.

Os esquemas de proteção de rede podem atuar contra alguns tipos específicos de falhas, as falhas nos enlaces, e falhas nos nós da rede. Nesse trabalho, foi abordado apenas as falhas nos enlaces. Para [Ramamurthy and Mukherjee, 1999], as falhas nos enlaces podem ser tratados com dois esquemas de sobrevivência da rede, a proteção por caminho e a proteção por enlace. Na proteção por enlace, quando ocorre a falha em um enlace, então um caminho alternativo entre os nós do enlace é estabelecido, re-estabelecendo assim os caminhos que passavam pelo mesmo. Na proteção por caminho, ao ocorrer uma falha em um enlace, um caminho novo é estabelecido (desde o nó origem até o nó destino) para cada caminho afetado por essa falha.

A Figura 2.8 demonstra os esquemas de proteção por caminho (Figura 2.8(a)) e proteção por enlace (Figura 2.8(b)), simulando uma falha no enlace B-C, onde o caminho primário, também chamado de caminho de trabalho, A-B-C-D estava alocado. No método de proteção por caminho, um novo caminho A-F-E-D é estabelecido para manter a demanda ativa. Na proteção por enlace, um caminho alternativo B-F-E-C é criado para desviar o enlace B-C, também re-estabelecendo o caminho.

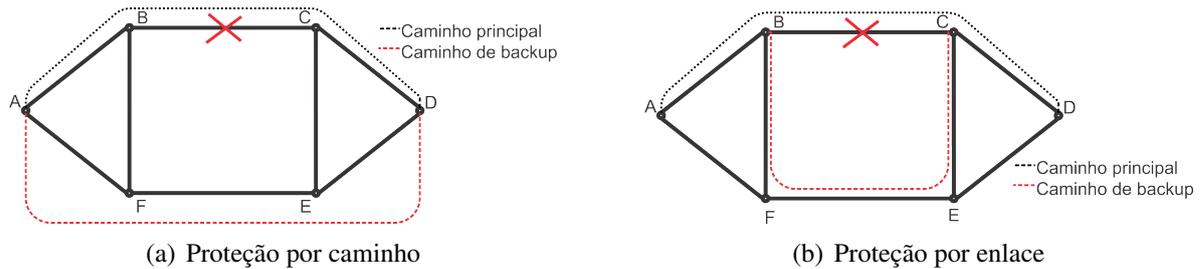


Figura 2.8 – Esquemas de proteção.

Neste trabalho utilizou-se a abordagem de proteção por caminho. Segundo [Ramamurthy and Mukherjee, 1999] existem três esquemas de proteção por caminho:

- Proteção dedicada por caminho:** Para proteção dedicada, a cada caminho de trabalho para atender à uma determinada demanda, um caminho disjunto em enlaces² deve ser reservado para o caminho de *backup*. O caminho de *backup* dimensionado deve reservar a largura de banda nos enlaces unicamente para o seu caminho de trabalho, para quando ocorrer uma falha utilizar esse *backup* para re-estabelecer a conexão;
- Proteção compartilhada por caminho:** No dimensionamento do caminho de trabalho, similar à proteção dedicada, um caminho disjunto em enlaces é reservado para caminho de *backup*. Entretanto, esse caminho reservado pode ser compartilhado com outros caminhos de *backup* da rede. Nessa abordagem, se ocorrer uma falha em um determinado caminho ele irá passar a utilizar o caminho de *backup*, assim se ocorrer falha em outro caminho que compartilha enlaces com esse *backup* que está em uso, essa segunda conexão será bloqueada. Contudo, este esquema requer menos recursos que o anterior;
- Restauração de caminho:** Esse esquema de proteção não reserva caminhos de *backup*, a restauração do caminho ocorre dinamicamente, dessa forma, quando ocorre uma falha em determinado caminho, um algoritmo de roteamento busca um novo caminho observando o estado atual da rede, caso não seja possível encontrar um novo caminho para a conexão, a mesma é bloqueada.

No esquema de proteção dedicada é preciso encontrar dois caminhos disjuntos em enlaces, para isso os algoritmos de Suurballe [Suurballe, 1974] e [Suurballe and Tarjan, 1984] são amplamente utilizados na literatura. O algoritmo de Suurballe [Suurballe, 1974] inicialmente

² Caminho disjuntos em enlaces são caminhos cujo a intersecção dos conjuntos de enlaces de cada caminho é vazio, ou seja, não possuem enlaces em comum. Isso não significa necessariamente que não possam compartilhar nós do caminho.

foi proposto para encontrar caminhos disjuntos em nós da rede, ou seja, os caminhos não possuem nós da rede em comum. Posteriormente em [Suurballe and Tarjan, 1984], o algoritmo foi estendido a fim de criar uma solução para encontrar caminhos disjuntos em enlaces. Como o presente trabalho foi desenvolvido acerca de proteção dedicada por enlace, foi utilizado esse segundo algoritmo de Suurballe.

Algoritmo 1: Algoritmo de Suurballe

Data: Um grafo dirigido, com enlaces bidirecionais
 $G = (V, E)$ e o par de nós origem e destino (o, d) .

Result: Um par de caminhos disjuntos.

```

1  $p \leftarrow$  buscaCaminhosMinimo( $G, o, d$ );
2 removeArestasDirigidas( $G, p$ );
3 atualizaPesosArestasSentidoOposto( $G, p$ );
4  $q \leftarrow$  buscaCaminhosMinimo( $G, o, d$ );
5  $G' \leftarrow p \cup q - p \cap q$ ;
6  $t \leftarrow$  buscaCaminhoMinimos( $G', o, d$ );
7 removeArestasDirigidas( $G', t$ );
8  $b \leftarrow$  buscaCaminhoMinimo( $G', o, d$ );
9 return  $\{(t, b)\}$ ;
```

Tomando como exemplo a Figura 2.9, especificando uma rede representado por um grafo $G = (V, E)$ com pesos nos enlaces, com enlaces bidirecionais e uma demanda d com origem no nó A e destino no nó D, o Algoritmo 1 executa a seguinte sequencia: (i) primeiramente encontra um caminho mínimo p , cujo a soma dos pesos de seus enlaces seja o menor possível entre os nós origem e destino, conforme exemplificado na Figura 2.9(a); (ii) posteriormente remove os enlaces direcionados no sentido do caminho p encontrado, permanecendo apenas os enlaces no sentido oposto, esses enlaces opostos ao caminho têm seus pesos atualizados para 0, induzindo a escolha do enlace para o próximo caminho, conforme Figura 2.9(b); (iii) novamente busca-se um menor caminho q entre a origem e destino, com os enlaces já atualizados, Figura 2.9(c); (iv) após encontrar os dois caminhos p e q , os enlaces dos mesmos são unidos formando um novo grafo G' , e os enlaces que eram repetidos nos dois caminhos são removidos, conforme exemplificado na Figura 2.9(d), o enlace $C \leftrightarrow E$ aparece nos dois caminhos, portanto o mesmo é removido; (v) por fim, conforme Figura 2.9(e), busca-se o menor caminho, na soma de seus pesos, para trabalho t , remove-se os enlaces dirigidos de t no grafo G' e busca o segundo caminho b para *backup*, retornando um par (t, b) de caminhos disjuntos.

O esquema de proteção dedicada é o único que garante que as conexões serão mantidas mesmo em qualquer falha de um enlace, para garantir a proteção em falha com mais de um

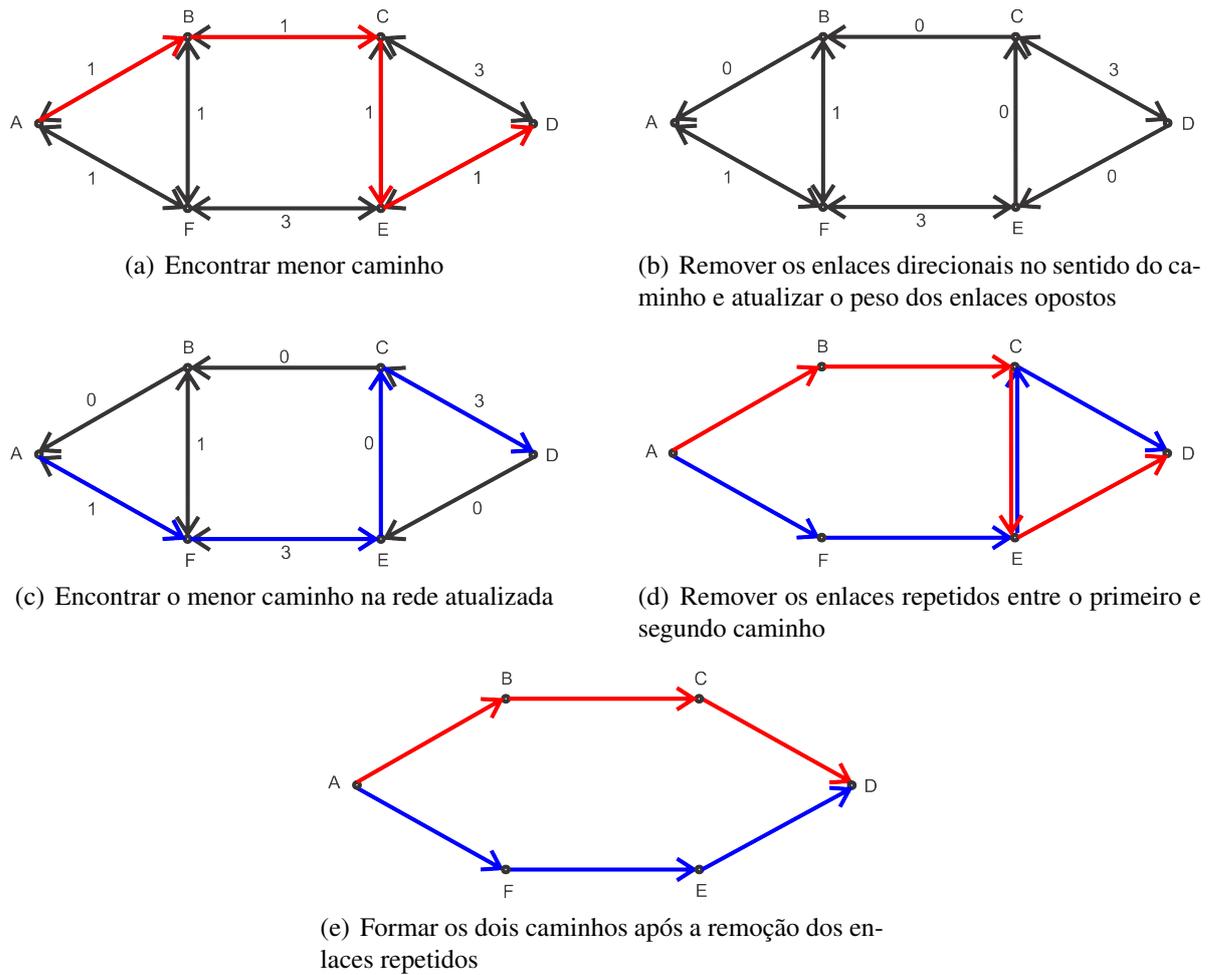


Figura 2.9 – Representação gráfica do algoritmo de Suurballe para busca de caminhos disjuntos por enlace

enlace, mais caminhos de *backup* devem ser reservados. No entanto, a proteção dedicada exige mais recursos ociosos na rede, aqueles reservados para *backup*. Na proteção compartilhada, minimiza-se a utilização de recursos ociosos, mantendo um bom nível de proteção da rede, pois para bloquear um caminho é necessário que ocorra falha em dois caminhos de trabalho que compartilhem um mesmo enlace.

2.2.1 Proteção dedicada por caminho em redes elásticas

Em redes ópticas elásticas, segundo [Walkowiak et al., 2014] existem dois cenários distintos para proteção dedicada: (i) alocação com o mesmo comprimento de onda e (ii) alocação com comprimentos de onda diferentes.

Na alocação de mesmo comprimento de onda, uma nova restrição é adicionada à alocação de espectro com proteção dedicada, a igualdade de comprimento de onda escolhidos para

o caminho de trabalho e para o caminho de *backup*. Dessa forma, ao alocar o caminho de trabalho em um determinado conjunto de *slots*, o caminho de *backup* deve possuir os *slots* com as mesmas frequências do caminho de trabalho, permitindo assim que sejam compartilhados os emissores e receptores ópticos entre ambos os caminhos. Essa solução oferece uma redução no custo de implementação da rede, pois compartilham equipamentos na emissão e recepção dos dados, e também reduz o tempo de computação para restauração do caminho em caso de falha. Por outro lado, essa abordagem torna mais complexa a alocação do espectro, pois deve-se garantir a continuidade e contiguidade do espectro no mesmo comprimento de onda, tanto para o caminho principal quanto para o caminho de *backup*, gerando assim uma maior probabilidade de bloqueio.

Para o caso de se utilizar a alocação com comprimentos de onda diferentes, não se tem a restrição de igualdade de comprimento de onda entre caminho de trabalho e caminho de *backup*, reduzindo a complexidade da alocação do espectro e por consequência, a probabilidade de bloqueio. Nesse cenário, a implementação carece de receptores e emissores dedicados para os caminhos de trabalho e *backup*, aumentando assim os custos de sua implementação. No artigo [Walkowiak et al., 2014], o autor cita uma abordagem alternativa nesse cenário, utilizando equipamentos com laser sintonizável, que são capazes de modificar sua frequência central reduzindo os custos de sua implementação. As propostas de dimensionamento de topologias lógicas apresentadas nesse trabalho, visam a redução do número de bloqueio, portanto, não foi requerida a alocação com mesmo comprimento de onda para caminhos de trabalho e *backup*.

3 ENGENHARIA DE TRÁFEGO

Segundo [Pavan, 2011], podemos classificar os problemas de redes de transportes em engenharia de tráfego, engenharia de rede e planejamento de rede.

A engenharia de tráfego de rede consiste em tratar problemas de fluxo para distribuí-lo na rede existente com o objetivo de otimizar o uso dos recursos. Este trabalho possui como foco principal a engenharia de tráfego, pois aborda o dimensionamento de topologias lógicas sobre redes ópticas elásticas de telecomunicações.

A engenharia de rede trabalha com problemas de alocação dos recursos de rede, provendo a minimização do custo dos equipamentos para atender uma demanda esperada; já o planejamento de rede é similar ao anterior, porém sem conhecer a demanda. Para esse trabalho, foi estudado essas duas últimas, pois a estrutura física já estava alocada para o problema proposto.

Os problemas relacionados à engenharia de tráfego, como o RSA, muitas vezes possuem uma representação matemática de complexidade exponencial, *NP-Completo*, e para solucioná-los em tempo hábil, algoritmos probabilísticos ou de inteligência artificial tem sido constantemente usados. Para o trabalho proposto, a modelagem de soluções para o problema do RSA possui complexidade *NP-Difícil* [Morais et al., 2011], por isso foi abordado através de uma técnica de inteligência artificial conhecida como algoritmos genéticos.

3.1 Algoritmos Genéticos

Segundo [Goldschmidt, 2010], algoritmos genéticos são modelos computacionais baseados na teoria evolucionista de Charles Darwin e da reprodução genética, e buscam a otimização de soluções de problemas matemáticos complexos. A base de seu funcionamento é a evolução de uma população codificada em cromossomos artificiais, onde segundo o princípio básico de Darwin, a evolução ocorre pelo processo de seleção natural, onde os mais aptos têm maior probabilidade de sobrevivência e de reprodução, passando seus genes para as futuras gerações, dessa forma, a população através dessa seleção natural, tende a se adaptar ao ambiente em que se encontra, para modelagem dos algoritmos genéticos a aptidão pode ser o melhor resultado de otimização e o ambiente é a função de custo proposta.

Para exemplificação da modelagem de um algoritmo genético são apresentados seus

componentes e suas estruturas de acordo com [Goldschmidt, 2010], que são:

- **Problema:** O problema é a caracterização do objetivo do algoritmo genético, tais como, a busca de uma solução ótima para o problema, maximizando ou minimizando o resultado;
- **Representação do indivíduo:** Um indivíduo representa uma modelagem de uma possível solução no espaço de busca. O gene representa a menor unidade do algoritmo genético sendo a representação de uma informação no domínio do problema, assim juntando uma série de genes temos um cromossomo, e esse cromossomo representa a estrutura de um indivíduo, os cromossomos podem ser palavras binárias, vetores, matrizes, etc;
- **Codificação e Decodificação do cromossomo:** Esse processo representa uma função bidirecional capaz de converter a solução real do problema em um cromossomo e vice-versa, como por exemplo, um número inteiro convertido em binário para representar um cromossomo;
- **Função de avaliação:** Tem por objetivo fornecer uma medida que seja capaz de mensurar a aptidão de um determinado indivíduo. Tal função, recebe como entrada um cromossomo decodificado para o problema real, e utiliza as informações genéticas do mesmo para fornecer um valor que mensura o indivíduo, para que possa ser classificado dentro da população. Por exemplo, a função de aptidão pode ser a função matemática x^2 , e o cromossomo fornece em sua informação genética o valor de x , assim a aptidão do indivíduo será o resultado da expressão;
- **Seleção:** O processo de seleção consiste em selecionar indivíduos para reprodução, através de métodos de escolhas dos quais pode-se citar:
 - Método da roleta: Nesse método um indivíduo é selecionado através da sua aptidão relativa à população, dessa forma preservando o princípio de Darwin, o mais apto terá maior probabilidade de passar o seu gene para as futuras gerações. Primeiramente é calculada a aptidão relativa de cada indivíduo, dividindo a sua aptidão pelo somatório da aptidão de toda população, posteriormente a aptidão relativa é ordenada em ordem decrescente, como demonstrado na Tabela 3.1. Em seguida, é sorteado um valor entre 0 e 1 e então o indivíduo que possuir a faixa de aptidão acumulada sorteada será selecionado para cruzamento. No exemplo da Tabela 3.1,

sorteando um valor maior que 0.4 e menor ou igual a 0.7, o indivíduo I2 será selecionado;

| Indivíduo | Aptidão | Aptidão acumulada |
|------------------|----------------|--------------------------|
| I1 | 0.4 | 0.4 |
| I2 | 0.3 | 0.7 |
| I3 | 0.2 | 0.9 |
| I4 | 0.1 | 1 |

Tabela 3.1 – Exemplo de aptidão acumulada.

- Método do torneio: Nesse método indivíduos são selecionados aleatoriamente para compor chaves de competição, assim divide-se os indivíduos selecionados em grupos, onde dentro dos grupos são sorteados os vencedores para próxima fase, na qual novamente são sorteados os vencedores até encontrar um par de vencedores para cruzamento. Esse método é puramente aleatório e não leva em consideração a aptidão do indivíduo para cruzamento.
- **Operadores genéticos:** São operações de combinação ou alterações no cromossomo para gerar novos indivíduos para a população, através de cruzamento ou mutação.
 - Cruzamento: para um cruzamento são selecionados dois indivíduos da população, de acordo com os métodos de seleção, e então são aplicadas operações de combinação genética entre esses dois indivíduos para gerar novos indivíduos. Há inúmeras formas de fazer o cruzamento de cromossomo, ele pode ser modelado de acordo com a aplicação do algoritmo genético, os mais clássicos são:
 - * Cruzamento do ponto único: a partir da representação do cromossomo um ponto da cadeia de gene é escolhido, assim o cromossomo do genitor A é quebrado em duas partes, uma do início até o ponto escolhido e outra do ponto escolhido em diante, a primeira parte é concatenada com a segunda do genitor B, e vice e versa, conforme exemplificado na Figura 3.1;
 - * Cruzamento de múltiplos pontos: Assim como o cruzamento de ponto único, diferenciando pelo número de corte, nesse método n pontos são selecionados, os cromossomos são quebrados em pedaços nos pontos selecionados e concatenados alternando entre os cromossomos genitores;

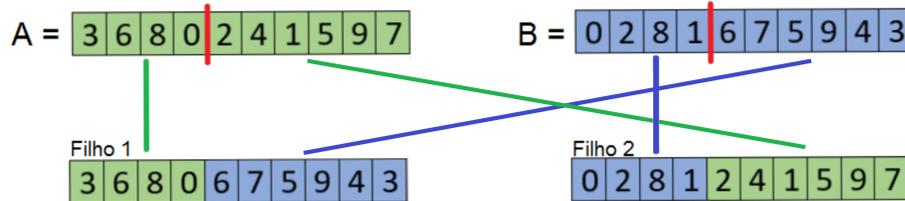


Figura 3.1 – Exemplo de cruzamento de ponto único.

- * Cruzamento uniforme: Nesse método é elaborado uma máscara com o tamanho equivalente ao número de genes de um cromossomo, marcando as posições no cromossomo cujos os genes devem ser trocados. Dessa forma, em cada posição marcada na máscara os genes dos genitores são trocados entre si formando seus descendentes com a composição entre os dois genes.
- Mutação: A mutação é uma alteração genética aleatória no cromossomo de determinado indivíduo, assim é possível inserir uma nova informação genética na população que devido a evolução, possui um material genético que não permite que o algoritmo convirja para mais otimização da função de avaliação, podendo estar preso a um ponto de otimização local. Por isso, em determinadas gerações, mutações são inseridas para que o processo evolutivo continue a otimizar as aptidões dos indivíduos.
- **Inserção de novos indivíduos e remoção:** Após selecionar os indivíduos para o cruzamento, os indivíduos são cruzados gerando novos indivíduos, estes são inseridos à população para a próxima geração, dessa forma também podem ser removidos da população os indivíduos menos aptos, preservando a seleção natural onde os menos aptos são eliminados de forma natural, equivalente ao número de novos indivíduos inseridos. Assim, o tamanho da população se mantém durante as gerações, pois sendo um algoritmo iterativo, o número de iterações pode ser alto e o crescimento da população comprometeria o desempenho do algoritmo.

Para modelagem do algoritmo genético, inicialmente é preciso modelar o problema, gerar as funções de codificação do problema em um cromossomo (indivíduo) e vice-versa, inserir uma população inicial com possíveis soluções no intervalo de busca; definir a função de avaliação e calcular as aptidões de todos os indivíduos para dar início ao ciclo iterativo, como representado na Figura 3.2. No ciclo, primeiramente é realizado o processo de escolha dos genitores, depois é realizado o cruzamento entre os genitores selecionados ou inserção de

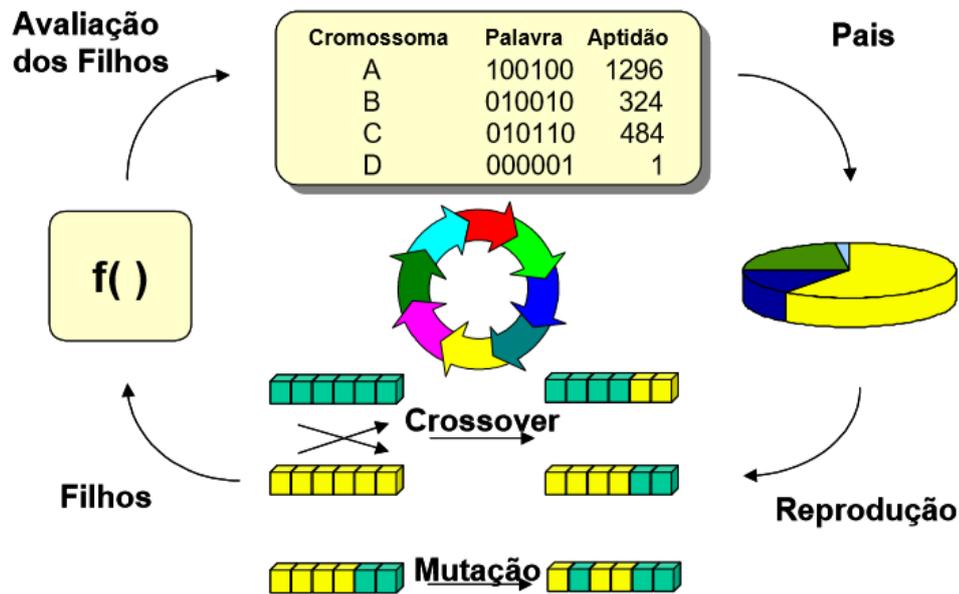


Figura 3.2 – Exemplo de iteração do algoritmo genético. Fonte: [Goldschmidt, 2010]

mutações, posteriormente é executada a função de aptidão dos novos indivíduos inserindo-os na população para a próxima geração. Esse ciclo evolutivo é repetido até que critérios de parada predefinidos sejam atingidos, como por exemplo, o número de gerações ou percentual evolutivo entre uma geração e outra.

Essa abordagem do uso de algoritmos genéticos para geração de topologias é utilizada em [Morais et al., 2011] e em [Klinkowski, 2012], onde o algoritmo genético é utilizado para solucionar os problemas complexos de otimização no dimensionamento de topologias físicas e lógicas, respectivamente. Neste trabalho, foi desenvolvida uma modelagem de um algoritmo genético, para encontrar soluções otimizadas de dimensionamento de topologias lógicas, modelando a ordem em que as demandas serão processadas pelo algoritmo do RSA, e as escolhas entre caminhos candidatos de cada demanda, levando em consideração sua possível alocação de espectro, buscando a minimização da probabilidade de bloqueios.

4 ESTUDOS RELACIONADOS

Em [Chatterjee et al., 2015] é apresentado um tutorial sobre o RSA, explorando técnicas para soluções do problema. O trabalho começa descrevendo os conceitos básicos de redes ópticas elásticas, sua arquitetura, seus princípios operacionais, além de ser esclarecidas as funcionalidades da largura de banda variável. O autor também explora o problema do RSA dividindo-o em dois subproblemas: (i) roteamento e (ii) alocação de espectro. Nesse tutorial é apresentado diferentes métodos de roteamento e diferentes métodos de alocação de espectro, analisando os seus desempenhos.

Os métodos de roteamento explorados são: (i) Roteamento fixo; (ii) roteamento fixo alternativo; (iii) roteamento menos congestionado e (iv) roteamento adaptativo. Ambos abordados na Seção 2.1.1.1 do presente trabalho. Para alocação do espectro, o autor trabalha com as técnicas de: (i) *first fit*; (ii) *randon fit*; (iii) *last fit*; (iv) *first-last fit* e (v) *exact fit*, que servem para alocação individual por demanda, ambas descritas na Seção 2.1.1.2. O autor também utiliza outros métodos de alocação de espectro, para demandas dinâmicas que comumente são utilizadas para alocação *on-line*, esses métodos não foram relacionados ao estudo desse trabalho, pois o mesmo propõe abordagens *off-line* de dimensionamento.

As soluções para o problema do RSA propostas em [Chatterjee et al., 2015], podem ainda ser otimizada através do uso de heurísticas, já que encontrar a solução ótima para o problema do RSA é NP-completo.

Em [Christodoulopoulos et al., 2011], os autores apresentam algumas heurísticas para o problema do *Routing, Modulation Level and Spectrum Allocation* (RMLSA). O RMLSA adiciona ao problema do RSA a modulação do sinal utilizada nos caminhos, dessa forma, o RMLSA consiste em (i) rotear os caminhos para cada demanda; (ii) definir a modulação adequada para cada caminho; (iii) encontrar e alocar os *slots* necessário para acomodar a demanda de tráfego.

No artigo, os autores propõem heurísticas para o RMLSA considerando o plano em fase *off-line* quando as demandas de tráfego são pré-estabelecidas. Eles têm como objetivo, encontrar uma solução que minimize a utilização do espectro da rede para atender a todas as demandas.

Em um primeiro momento, os autores apresentam um *integer linear programming* (ILP) como solução do problema do RMLSA, posteriormente o problema do RMLSA é subdividido em dois subproblema (i) roteamento e modulação do sinal; (ii) e alocação do espectro, deno-

minado RML+SA, e um ILP é proposto para cada subproblema. Em ambas soluções os ILP's tinham como função a minimização do uso do espectro. Entretanto, esses algoritmos não podem resolver com eficiência redes com grandes quantidades de nós e enlaces. Para isso, foram apresentadas propostas heurísticas.

Em uma das abordagens heurísticas apresentadas, as demandas são estabelecidas na rede de forma sequencial, uma a uma. Para isso, o algoritmo pré-calcula um conjunto limitado de caminhos mínimos, candidatos, para cada par origem-destino. Para cada caminho, uma função calcula, de acordo com o maior comprimento de enlace, a modulação adequada para aquele caminho.

Assim, para cada demanda a ser estabelecida na rede, é calculado a disponibilidade de espectro para todos os caminhos candidatos, selecionando-se o caminho disponível com menor índice de *slot*, como no método de alocação FF. Ao selecionar o caminho e o índice de *slot* inicial da alocação de espectro, o algoritmo marca os *slots*, necessários para atender à demanda, de todos os enlaces do caminho como ocupado. Posteriormente, vai estabelecer a próxima demanda, até que todas as demandas sejam alocadas.

A sequência com que as demandas são alocadas na rede interferem diretamente na quantidade de espectro utilizada. Segundo os autores, as demandas de tráfego servidas uma a uma para o algoritmo de alocação de espectro, são importantes para o processo e resultam em utilizações diferentes do espectro, portanto, uma heurística pode otimizar essa ordenação para que o uso do espectro seja minimizado. Nesse sentido, são apresentados no artigo duas políticas de ordenação das demandas: (i) *Most Subcarriers First (MSF)*, que ordena as demandas de acordo com o número de *slots* requeridos, ordenando primeiramente as demandas que requerem maior número de *slots*; (ii) *Longest Path First (LPF)*, ordena as demandas de acordo com o número de enlaces em seu caminho, trazendo primeiramente as demandas que possuem o caminho com maior número de enlaces.

Baseado no trabalho [Christodoulopoulos et al., 2011], uma outra proposta de heurística é apresentada em [Klinkowski, 2012]. Nesse artigo o autor propõe a construção de um algoritmo genético *off-line* que busca uma solução otimizada, em uso do espectro, para o problema do RSA em redes ópticas elásticas com proteção dedicada, considerando que a demanda de tráfego seja estática.

Em sua modelagem do problema, o esquema de sobrevivência da rede utiliza a proteção dedicada em enlaces utilizando o mesmo comprimento de onda para o caminho de trabalho e

o de *backup*. Portanto, as restrições na alocação do espectro são: (i) contiguidade do espectro; (ii) continuidade do espectro; (iii) possuir *slots* o suficiente para atender a demanda e (iv) para todas as demandas o caminho de trabalho e *backup* devem possuir o mesmo comprimento de onda e largura de banda.

O número de *slots* necessários para atender a uma determinada demanda pode ser calculado através de uma função que leva em consideração a técnica de modulação utilizada, a largura de banda de cada *slot*, a quantidade de *slots* utilizados como banda de guarda, entre outros. Ainda segundo [Klinkowski, 2012], levando em consideração um cenário onde essas informações são fornecidas na entrada, há uma relação direta entre a largura de banda solicitada e o número de *slots*. Logo, no artigo o autor assumiu, sem perda de generalidade, que as demandas são expressas em número de *slots*. Para cada demanda, o autor assume ainda, que é dado um conjunto de pares de caminhos disjuntos em enlaces.

A heurística proposta por [Klinkowski, 2012], baseia-se em uma solução híbrida, entre sua proposta de um algoritmo genético e a heurística de alocação de demandas sequencial descrito em [Christodoulopoulos et al., 2011]. O algoritmo genético atua no vetor da ordem em que as demandas são alocadas na rede, ou seja, o algoritmo busca uma permutação na ordem em que as demandas são alocadas pelo algoritmo sequencial de RSA, que minimize a ocupação de espectro. Em [Christodoulopoulos et al., 2011], a ordem das demandas foi dada através do MSF e LPF como heurísticas sequenciais. Destaca-se que em [Klinkowski, 2012], o autor adicionou à heurística sequencial do RSA a restrição da proteção dedicada, e para isso foi necessário adaptar sua modelagem para oferecer a proteção dedicada de mesmo comprimento de onda.

Para atender à restrição, os caminhos de trabalho e *backup* de cada demanda são unidos em um caminho único. Essa união, dá-se através da concatenação dos enlaces do caminho principal e o caminho de *backup*, formando um anel, onde o início e o fim desse caminho é o nó de origem. Assim, utilizando esse caminho unificado, o mesmo pode ser processado pelo algoritmo de RSA [Christodoulopoulos et al., 2011], recebendo o mesmo comprimento de onda na alocação do espectro, pois os caminhos de trabalho e *backup* são processados como um único caminho. Portanto, na formulação dessa nova heurística, o autor utiliza a heurística sequencial propostas em [Christodoulopoulos et al., 2011], porém o conjunto de caminhos candidatos de cada demanda é dado pelos caminhos formados através dessas uniões.

Na modelagem do algoritmo genético proposto, a representatividade do cromossomo do indivíduo é um vetor com os índices das demandas, em uma determinada ordenação. A

população é formada por indivíduos com diversificadas permutações na ordem no vetor de índices de demandas, para isso, a população inicial é formada por sequências aleatórias.

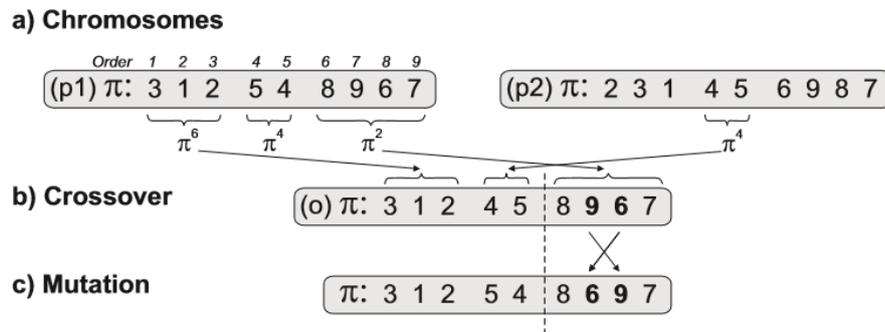


Figura 4.1 – Algoritmo Genético - a) Cromossomos b) Cruzamento c) Mutação. Fonte: [Klinkowski, 2012]

A função de avaliação dos indivíduos é obtida através da execução do algoritmo sequencial do RSA, posteriormente calculando o número de *slots* utilizados em relação ao número total de *slots*, cujo objetivo é ser minimizado. A Figura 4.1 (a) representa os cromossomos selecionados para cruzamento, (b) o cruzamento é realizado pelo método uniforme para geração dos novos membros, apenas trocando valores das permutações, (c) a mutação ocorre trocando a ordem de dois valores da sequência. Após a geração dos novos membros, esses são inseridos na população e os menos aptos são removidos, finalmente, são atualizadas as aptidões dos novos indivíduos para a seleção nas próximas gerações.

Assim, o algoritmo genético evolui a população ao longo de um número pré-determinado de gerações, e o indivíduo mais apto é a solução apresentada que mais otimizou o uso do espectro. O número de gerações é determinado como critério de parada do algoritmo genético.

Para demonstração dos resultados, o autor [Klinkowski, 2012], roda o algoritmo proposto para 4 redes, uma de exemplo de 6 nós, e outras 3 que representam redes reais de referência de 9, 15 e 24 nós. Em sua modelagem, o autor utiliza nas topologias lógicas das redes, um número de *slots* disponíveis maior que o número de *slots* requisitados, dessa forma não ocorrem bloqueios. As demandas são geradas a partir de uma distribuição uniforme, escolhendo aleatoriamente um par de nós distintos como origem e destino. Para as mesmas configurações de redes e demandas, compara seus resultados com a heurística do MSF [Christodoulopoulos et al., 2011], obtendo uma melhora de 5% a 10% em relação ao uso do espectro.

Nota-se que nos artigos [Christodoulopoulos et al., 2011] e [Klinkowski, 2012], as redes

físicas disponíveis são suficientes para atender a todas as demandas, desconsiderando a possibilidade de bloqueio. Porém, em [Zhao et al., 2012], o autor traz a probabilidade de bloqueio como uma das principais métricas para dimensionamento de uma topologia lógica. A probabilidade de bloqueio é calculada através do número de demandas bloqueadas em relação ao número total de demandas solicitadas, assim, quanto menor for a probabilidade de bloqueio, mais demandas são atendidas. Nesse sentido, encontrar uma solução para o problema do RSA que minimize a probabilidade de bloqueio otimiza o uso da rede, atendendo ao maior número de demandas possíveis para uma topologia física já dimensionada.

As técnicas demonstradas em [Chatterjee et al., 2015], citadas anteriormente, também são avaliadas de acordo com sua probabilidade de bloqueio, essa métrica também é utilizada em [Gerstel et al., 2012], que à utiliza, dentre outras métricas, para demonstrar a eficiência das redes ópticas elásticas em relação às redes ópticas transparentes WDM.

De forma similar à probabilidade de bloqueio de demandas, é possível também calcular a probabilidade de bloqueio de *slots*, essa métrica é utilizada em [SANTOS, 2015] juntamente com a métrica de probabilidade de bloqueio de demandas, para medir seus resultados alcançados. As fórmulas de cálculo dessas métricas foram explanadas na Seção 5.1.

Neste trabalho, a probabilidade de bloqueio de demanda e de *slots* foram as métricas empregadas para avaliação das propostas de heurísticas para solução do problema RSA. Foram utilizadas as técnicas de roteamento e alocação de espectro apresentadas em [Chatterjee et al., 2015], explorando a heurística através da utilização do algoritmo genético apresentado em [Klinkowski, 2012], formulando uma nova heurística com o uso do algoritmo genético para o problema do RSA.

Para a formulação de novas soluções para o problema do RSA com proteção dedicada, é necessário também algumas heurísticas afim de encontrar pares de caminhos disjuntos candidatos como possível solução para cada demanda. Nesse sentido em [Phung et al., 2005], uma heurística para encontrar um número k de pares de caminhos disjuntos em redes WDM é apresentada, sua abordagem é similar ao algoritmo de Suurballe [Suurballe and Tarjan, 1984]. Primeiramente encontra um caminho mínimo entre a origem e o destino, depois remove as arestas dirigidas do caminho encontrado e atualiza os pesos dos enlaces no sentido oposto do caminho principal, como no algoritmo de Suurballe, porém com as informações dos enlaces atualizadas, busca k caminhos mínimos entre a origem e o destino e para cada um dos k caminhos, remove os enlaces repetidos entre o k -ésimo caminho e o caminho principal, formando

assim um k -ésimo par de caminhos disjuntos.

Nota-se que nessa técnica, o caminho principal é igual para todos os k pares de caminhos disjuntos. No final o autor compara seu algoritmo com um ILP, buscando soluções otimizadas para minimizar o número de *slots* utilizados em uma arquitetura WDM, porém a solução ótima é encontrada ao calcular todos os caminhos possíveis para cada demanda, sendo complexo em tempo. Dessa forma a heurística apresenta resultados satisfatórios com um baixo tempo computacional, sendo que o número de soluções ótimas encontradas aumenta proporcionalmente ao aumento do número k de caminhos candidatos.

Além do algoritmo para encontrar k pares de caminhos disjuntos apresentadas em [Phung et al., 2005], para compor a heurística de solução do problema RSA, objetivo desse trabalho, foi proposto um novo algoritmo para gerar caminhos disjuntos candidatos afim de flexibilizar mais as opções de escolha dos caminhos de cada demanda e seu impacto na otimização dos resultados.

5 PROPOSTA

Neste capítulo é proposto um algoritmo genético para solucionar o RSA em redes ópticas elásticas com proteção dedicada, visando minimizar a probabilidade de bloqueio.

Baseado na proposta de [Klinkowski, 2012], um novo algoritmo genético foi implementado, estendendo a caracterização da solução do problema para que, além de alternar a ordem com que as demandas serão processadas pelo algoritmo do RSA, alterne as escolhas dos caminhos candidatos de cada demanda. Dessa forma, uma nova característica é adicionada ao algoritmo genético de Klinkowski, dando à heurística mais possibilidades de soluções e consequentemente maior interferência na otimização do processo de roteamento e alocação do espectro.

5.1 Formulação do problema

O problema consiste em encontrar um roteamento, definindo os caminhos através da rede física, para cada demanda solicitada e definir em quais *slots* dos enlaces o *lightpath* será transmitido no meio ótico, utilizando redes elásticas. As demandas são estáticas e pré-determinadas na entrada do problema, que o tratará com uma abordagem *off-line*.

As demandas devem possuir uma alocação com proteção para falha de um enlace, sendo dedicada por caminho. Assim, além de definir o caminho de trabalho, deve-se alocar, na rede física, um caminho de *backup* dedicado para cada demanda. A restrição de mesmo comprimento de onda, descrita em [Walkowiak et al., 2014], não foi utilizada nesse trabalho, e sim a abordagem de comprimentos de onda diferentes. Essa abordagem reduz a probabilidade de bloqueio, portanto tem mais aplicabilidade no escopo desse trabalho.

A arquitetura do nó da rede utilizada, representado na Figura 5.1, é composta por dois transmissores de largura de banda variáveis (*Bandwidth-Variable Transponder* - BVT) e um comutador óptico de banda variável (*Bandwidth-Variable Wavelength Cross-Connects* - BV-WXC). Os BVTs recebem os dados do cliente e criam os *lightpaths* na largura de banda e o comprimento de onda configurados. Como o comprimento de onda pode ser diferente entre o caminho de trabalho e o de *backup*, são utilizados dois BVTs com configurações distintas. O BV-WXC junto com os demais nós da rede, estabelecem os caminhos fim-a-fim acomodando os recursos de espectro definidos pelos BVTs [Zhang et al., 2013].

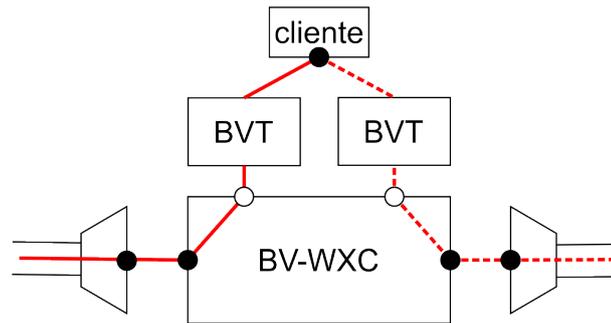


Figura 5.1 – Arquitetura do nó da rede.

A entrada do método é dada por:

- Uma topologia física de uma rede óptica elástica, representada por um grafo $G = (V, E)$, onde V representa o conjunto de nós das redes, e E o conjunto de enlaces, em que cada um conecta dois nós u e v , sendo $u, v \in G$;
- A divisão da frequência do espectro, sendo a mesma em todos os enlaces $e \in E$, dada através do número de *slots* e a largura de banda de cada um deles (eg. 12.5 GHz);
- Um conjunto D de demandas de tráfego, cada demanda d é representada por uma tupla (o_d, d_d, n_d) , onde o_d e d_d representam o nó de origem e o nó de destino respectivamente, n_d representa o número de *slots* necessários para atender à demanda.

Para cada demanda, deve-se encontrar um caminho de trabalho e um caminho de *backup* que satisfaçam as seguintes restrições:

- Continuidade no espectro: Em todos os enlaces de cada um dos caminhos, de trabalho ou *backup*, deve ser utilizado o mesmo comprimento de onda, entretanto o caminho de trabalho pode possuir comprimento de onda diferente do caminho de *backup*;
- Contiguidade no espectro: Os *slots* necessários para atender a uma determinada demanda, devem ser alocados de forma adjacente para garantir a contiguidade no espectro;
- Capacidade nos enlaces: Todos os enlaces dos caminhos de trabalho e *backup* devem conter disponibilidade de espectro para atender a demanda.

Caso não seja possível encontrar um caminho na topologia física que atenda às restrições, a demanda é bloqueada. O conjunto de todas as demandas bloqueadas será denotada por B . O objetivo é encontrar uma solução que minimize probabilidade de bloqueio de demanda,

que será calculada pela Equação 5.1. A probabilidade de bloqueio de demanda Pbd será a razão entre a quantidade de demandas bloqueadas $|B|$ pela quantidade total das demandas $|D|$.

$$Pbd = \frac{|B|}{|D|} \quad (5.1)$$

Similarmente à Pbd , será calculada a probabilidade de bloqueio de *slots*, utilizando a Equação 5.2. A probabilidade de bloqueio por *slots* Pbs será a razão entre o somatório do número de *slots* das demandas bloqueadas $n_d \in B$, pelo somatório dos *slots* das demandas solicitadas $n_d \in D$.

$$Pbs = \frac{\sum n_d \in B}{\sum n_d \in D} \quad (5.2)$$

A demanda de tráfego será dada em número de *slots*, assumindo que uma função calcule previamente o número de *slots* necessários para atender ao tráfego de dados, levando em consideração a modulação utilizada e a granularidade dos enlaces, assim como em [Klinkowski, 2012]. Será considerada uma banda de guarda de um *slot* para separar duas demandas adjacentes, esses *slots* de banda de guarda não estão incluídos no conjunto de demandas de entrada, logo, devem ser alocadas no método de alocação do espectro.

A definição da rede elástica utilizada, está especificada na recomendação da ITU-T G.694.1 [ITU, 2012]. É definido que para alocação do espectro em redes elásticas, deve ser indicado a frequência central da largura de banda, a granularidade utilizada e a quantidade de *slots*. A alocação de espectro se dará com metade da frequência solicitada com frequência inferior à frequência central e metade da superior. A recomendação da ITU-T G.694.1 [ITU, 2012], especifica a frequência central do espectro utilizável de 193.1 THz, dessa forma, utiliza-se a Equação 5.3, para calcular a frequência central desejada F_c , onde n é um número inteiro indicando o deslocamento em *slots* em relação a frequência central do espectro disponível e g a granularidade do espectro em THz (0.0125 representam granularidade de 12.5 GHz).

$$F_c = 193.1 + n \times g \quad (5.3)$$

A especificação G.694.1 [ITU, 2012], traz ainda uma tabela de exemplo de possíveis comprimentos de ondas em uma faixa de 184.5500 THz à 195.9375 THz. Com uma granularidade de 12.5 GHz é possível obter 911 *slots*. Nos experimentos desse trabalho, foi utilizado uma frequência de 190.6 THz à 195.6 THz, com granularidade de 12.5 GHz, formando assim uma grade de 800 *slots* por enlace. Essa grade está representada por um ve-

tor $S = \{s_0, s_1, \dots, s_{798}, s_{799}\}$ em cada enlace, onde na posição s é preenchido pelo índice da demanda que alocou o respectivo *slot*. E a banda de guarda é anotada por uma constante MAX_INT (o maior número inteiro positivo representável com 32 bits) e os *slots* disponíveis são preenchidos com a constante -1 .

A solução do problema se dá pelo conjunto das grades S de todos os enlaces da rede física, que identificam a alocação do espectro das demandas e os caminhos de trabalho e *backup* selecionados de cada demanda.

5.2 Implementação

Para a implementação dos métodos propostos para a solução do problema do RSA, utilizou-se a abordagem da divisão do problema em dois subproblemas, o roteamento e a alocação do espectro. Dessa forma, em um primeiro momento é gerado um conjunto de pares de caminhos disjuntos, sendo um caminho de trabalho e um de *backup*, candidatos para cada demanda.

Para o subproblema da alocação do espectro, as demandas são alocadas uma a uma de forma sequencial, como em [Christodoulopoulos et al., 2011]. Um algoritmo genético manipula a ordem em que as demandas são alocadas e o par de caminhos disjuntos selecionado para cada demanda, alternando-os em busca de uma solução que minimize a probabilidade de bloqueio.

5.2.1 Roteamento

No subproblema do roteamento, é necessário gerar um conjunto de pares de caminhos disjuntos (trabalho e *backup*) para cada demanda. Sendo P o conjunto de pares de caminhos disjuntos, $(p, q) \in P$, tal que p e q são caminhos disjuntos em enlaces, sendo $(p \cup q) \subset E$ e $(p \cap q) = \emptyset$. O conjunto C_d , denota os pares de caminhos disjuntos candidatos para atender a uma determinada demanda d , sendo $C \subset P$.

Os conjuntos de pares de caminhos candidatos por demanda C_d , são gerados demanda a demanda, através do uso de dois algoritmos de roteamento. A representação de um caminho será um vetor com os índices dos nós, na ordem de suas adjacências no caminho, assim, um caminho entre o nó 0 ao nó 2 passando pelo nó 1 será o vetor $[0, 1, 2]$.

Em uma das implementações, foi utilizado algoritmo *K Disjoint-Paths Pairs* (KDPPs) [Phung et al., 2005], para geração de k pares de caminhos disjuntos.

Nesse método, baseado no algoritmo de Suurbale [Suurballe and Tarjan, 1984], o autor propõe encontrar um caminho mínimo para trabalho (p), entre os nós origem e destino da demanda, e depois encontrar k caminhos mínimos disjuntos em enlaces ao caminho de trabalho para caminhos de *backup* (q_i), gerando pares entre o caminho de trabalho e um dos caminhos de *backup*. Dessa forma obtêm-se, k pares de caminhos disjuntos para a demanda (d), $C_d = \{(p, q_1), (p, q_2), \dots, (p, q_{k-1}), (p, q_k)\}$.

Baseado no algoritmo KDPPs, esse trabalho propõe um novo algoritmo de geração de pares de caminhos disjuntos por demanda, denominado NxM Pares de Caminhos Disjuntos (NMPCD). Nessa proposta, modelada no Algoritmo 2, o método recebe na entrada, o grafo dirigido, com enlaces bidirecionais $G = (V, E)$, que representa a topologia física da rede; os nós de origem e destino da demanda que está roteando; e dois números n e m , onde n será o número de caminhos de trabalho que o método deve retornar, e m é o número de caminhos de *backup* para cada um dos n caminhos de trabalho.

Algoritmo 2: NxM Pares de Caminhos Disjuntos (NMPCD)

Data: Um grafo dirigido, com enlaces bidirecionais $G = (V, E)$; o par de nós origem e destino (o, d) ; e um par de inteiros positivos (n, m) .

Result: Um Conjunto de pares de caminhos disjuntos.

```

1 Result  $\leftarrow \emptyset$ ;
2  $P \leftarrow \text{buscaCaminhosMinimos}(G, o, d, n)$ ;
3 foreach caminho  $p \in P$  do
4    $G' \leftarrow G.\text{clone}()$  ;
5    $\text{removeArestasDirigidas}(G', p)$  ;
6    $Q \leftarrow \text{buscaCaminhosMinimos}(G', o, d, m)$  ;
7   foreach caminho  $q \in Q$  do
8      $G'' \leftarrow p \cup q - p \cap q$ ;
9      $t \leftarrow \text{buscaCaminhosMinimos}(G'', o, d, 1)$  ;
10     $\text{removeArestasDirigidas}(G'', t)$  ;
11     $b \leftarrow \text{buscaCaminhosMinimos}(G'', o, d, 1)$  ;
12     $\text{Result} \leftarrow \text{Result} \cup \{(t, b)\}$ ;
13  end
14 end
15 return Result;

```

No Algoritmo 2 (NMPCD), (i) inicializa-se o resultado com conjunto vazio; (ii) busca n caminhos mínimos entre a origem (o) e o destino (d), no grafo G de entrada; (iii) para cada um dos p_i caminhos encontrados; (iv) faz uma cópia de G e atribui o grafo G' ; (v) remove os enlaces dirigidos de G' no sentido do caminho p_i ; (vi) em G' busca m caminhos mínimos entre a origem

e o destino; (vii) para cada um dos caminhos q_j encontrados; (viii) gera um novo grafo G'' com a união dos caminhos p_i e q_j , removendo a intersecção entre p_i e q_j , ou seja, une os dois caminhos removendo os enlaces repetidos, assim como o algoritmo de Suurballe [Suurballe and Tarjan, 1984]; (ix) no grafo G'' busca o caminho mais curto e o atribui como caminho de trabalho (t); (x) remove as arestas de t em G'' ; (xi) busca o outro caminho atribuindo ao caminho de *backup* (b); (xii) gera um par de caminhos disjuntos formados por t e b e o adiciona ao resultado da função.

A saída desse método será um conjunto de $n * m$ pares de caminhos disjuntos para a demanda (d), assim $C_d = \{(p_1, q_{(1,1)}), (p_1, q_{(1,2)}), \dots, (p_1, q_{(1,m-1)}), (p_1, q_{(1,m)})\} \cup \{(p_2, q_{(2,1)}), (p_2, q_{(2,2)}), \dots, (p_2, q_{(2,m-1)}), (p_2, q_{(2,m)})\} \cup \{\dots\} \cup \{(p_{n-1}, q_{(n-1,1)}), (p_{n-1}, q_{(n-1,2)}), \dots, (p_{n-1}, q_{(n-1,m-1)}), (p_{n-1}, q_{(n-1,m)})\} \cup \{(p_n, q_{(n,1)}), (p_n, q_{(n,2)}), \dots, (p_n, q_{(n,m-1)}), (p_n, q_{(n,m)})\}$.

O método para encontrar k caminhos mínimos, utilizado no algoritmo de roteamento foi implementado com base em [Eppstein, 1994]. Para isso, na entrada o algoritmo recebe o grafo $G = (V, E)$ que representa a topologia física, os nós de origem (o) e destino (d) e um número k que indica até quantos caminhos mínimos devem ser retornados.

O Algoritmo 3, (i) inicializa o resultado com um conjunto vazio; (ii) declara um vetor contador (c) com o tamanho do número de nós da rede; (iii) e uma fila de prioridade (F) que ordena seus elementos a partir do atributo distância entre o nó e a origem, sendo o primeiro da fila o nó com menor distância até a origem; (iv) a fila de prioridade é inicializada com o nó origem e distância 0; (v) enquanto a fila de prioridade não estiver vazia; (vi) é removido o primeiro nó da fila e atribuído a variável no ; (vii) se esse nó for o destino; (viii) então adiciona ao resultado o caminho armazenado no nó, que representa o caminho da origem até ele; (ix) nesse caso ainda, verifica se já encontrou k resultados, se sim; (x) então sai do laço para retornar a resposta; (xi) se não saiu do laço; (xiii) incrementa o contador no índice de no ; (xiv) caso esse nó já foi selecionado k vezes então; (xv) ele é descartado, retornando ao início do laço para pegar um novo nó, pois já há k caminhos com distância menores que foram relaxados na fila de prioridade em iterações anteriores, portanto esse novo caminho teria distância maior que os anteriores, esse corte otimiza a busca, pois não permite buscas desnecessárias sabendo que já existem k caminhos menores; (xvi) se o no ainda não foi relaxado k vezes; (xvii) então percorre todos seus nós adjacentes v ; (xviii) atribui ao caminho do nó adjacente v a união do caminho da origem até o no ao enlace do no à v ; (xix) adiciona o nó v na fila de prioridade, sendo sua

Algoritmo 3: *buscakCaminhosMinimos*

Data: Um grafo dirigido, com enlaces bidirecionais $G = (V, E)$; o par de nós origem e destino (o, d) ; e o número de caminhos desejados (k) .

Result: Um Conjunto de caminhos.

```

1 Result  $\leftarrow \emptyset$ ;
2 Contador[|V|] c;
3 Fila de Prioridade<no, distancia> F;
4 F  $\leftarrow F \cup (o, 0)$ ;
5 while F  $\neq \emptyset$  do
6     no  $\leftarrow F.inicio()$ ;
7     if no.indice = d then
8         Result  $\leftarrow Result \cup no.caminho$ ;
9         if |Result| = k then
10            | break;
11        end
12    end
13    c[no.indice] ++;
14    if c[no.indice] > k then
15        | continue;
16    end
17    foreach noVisinho v  $\in getNosAdjacentes(G, no.indice)$  do
18        | v.caminho  $\leftarrow no.caminho \cup v.indice$ ;
19        | F  $\leftarrow F \cup$ 
20            |  $\{(v, no.distancia + getComprimentoEnlace(no.indice, v.indice))\}$ ;
21    end
22 return Result;

```

distância do nó origem o , a distância entre o e no adicionado da distância de no e v .

Na saída do método, tem-se um conjunto com até k caminhos mínimos. Nota-se que se não houver k caminhos entre a origem e o destino, então somente os caminhos possíveis serão retornados, inclusive um conjunto vazio, caso a origem e o destino estejam desconectados na rede física de entrada. Isso impacta diretamente nos métodos de roteamentos, pois se não for possível encontrar todos os caminhos solicitados, logo eles também não gerarão todos os pares de caminhos disjuntos solicitados.

5.2.2 Alocação de Espectro

A política de alocação de espectro utilizada nesse trabalho, foi a política *First-Last Fit* (FLF), por demonstrar melhor desempenho em testes realizados em [Chatterjee et al., 2015].

Nessa política, a grade de 800 *slots* para cada enlace é agrupada em 10 subconjuntos de 80 *slots* cada. Assim a grade de *slots* dos enlaces é composta pela união desses subconjuntos, $S = \{\{s_0, s_1, \dots, s_{79}\} \cup \{s_{80}, \dots, s_{159}\} \cup \{\dots\} \cup \{s_{720}, \dots, s_{799}\}\}$.

Algoritmo 4: Alocação de Espectro FLF

Data: Um grafo dirigido, com enlaces bidirecionais $G = (V, E)$; um caminho (p); uma demanda (d) para ser alocado.

Result: Verdadeiro caso foi possível alocar a demanda, ou Falso caso o caminho foi bloqueado.

```

1 indice  $\leftarrow$  0;
2  $n \leftarrow$  getNumreoSlotNecessario( $d$ ) + 1;
3 for  $i \leftarrow 0$  to  $|p| - 1$  do
4    $S \leftarrow$  getGradeSlots( $p_i$ );
5   if nao temDisponibilidadeEspectro( $S, n, indice$ ) then
6      $indice++$ ;
7      $i \leftarrow 0$ ;
8     if  $indice + n > |S|$  then
9       return falso;
10    end
11  end
12 end
13  $indiceSubConjunto \leftarrow$  getIndiceSubconjunto( $indice$ );
14 if  $indiceSubConjunto$  é par then
15    $ocupado \leftarrow$  falso;
16   while nao ocupado do
17     for  $i \leftarrow 0$  to  $|p| - 1$  do
18        $S \leftarrow$  getGradeSlots( $p_i$ );
19        $S' \leftarrow S[indiceSubConjunto]$ ;
20       if nao temDisponibilidadeEspectro( $S', n, indice$ ) then
21          $ocupado \leftarrow$  true;
22       end
23     end
24     if  $indice + n > |S'|$  then
25        $ocupado \leftarrow$  true;
26     end
27     if nao ocupado then
28        $indice++$ ;
29     end
30   end
31 end
32 marcaSlotsDasGrades( $G, p, d, indice$ );
33 return verdadeiro;

```

O Algoritmo 4 de alocação recebe como entrada o grafo dirigido, com enlaces bidirecionais $G = (V, E)$, que representa a topologia física da rede; um caminho p e a demanda d que

deve ser alocada na rede. Inicialmente, (i) inicializa o índice (*indice*), que representa a posição do *slot* inicial da alocação da demanda na grade S dos enlaces, com 0; (ii) atribui à variável n o número de *slots* necessários para atender a demanda e mais um *slot* como banda de guarda; (iii) inicia um laço percorrendo todos os enlaces do caminho e para cada enlace; (iv) armazena em S a grade de *slots* do enlace p_i ; (v) verifica a disponibilidade de espectro em S , onde a partir do índice *indice* deve conter n *slots* consecutivos disponíveis, garantindo a restrição de contiguidade de espectro. Se não houver disponibilidade, (vi) incrementa o *indice*; (vii) zera o índice i do laço para inicia-lo novamente, assim tentará um novo índice de *slot* inicial para a alocação de espectro, garantindo a restrição de continuidade do espectro em todos os enlaces; (viii) caso $indice + n$ for maior que o número de *slots* da grade; (ix) então retorna falso, pois já se chegou ao maior índice inicial de alocação possível e não será possível alocar a demanda, logo ela é bloqueada.

Após encontrar o índice inicial com menor índice, verifica-se a qual subconjunto da grade o *indice* pertence, depois verifica se o índice do subconjunto selecionado, de 0 a 9, é par. Sendo par, desloca a alocação do espectro para o índice mais alto possível dentro do subconjunto S' de *slots*. Enquanto não encontrar um índice já ocupado, vai incrementando o *indice* e verificando as restrições de continuidade e contiguidade nos enlaces do caminho, até encontrar um índice que não é possível alocar o espectro, ou o fim do subconjunto S' de *slots*.

Dessa forma o índice do *slot* inicial da alocação de espectro será com menor índice em subconjuntos de S ímpares, e maior possível em subconjuntos de S pares, conforme propõe o método de alocação FLF.

Encontrado o índice do *slot* inicial, o método atualiza as grades de *slots* S de todos os enlaces do grafo que pertencem ao caminho, marcando as posições dos *slots* de $S[indice]$ à $S[indice + n - 1]$ com o identificador da demanda alocada, e $S[indice + n]$ com a constante MAX_INT indicando a banda de guarda.

5.2.3 Algoritmo Genético Proposto

O método de dimensionamento de topologias lógicas, proposto nesse trabalho, tem como objetivo encontrar uma solução para o problema do RSA que minimize a probabilidade de bloqueio. Baseado no algoritmo genético de Klinkowski [Klinkowski, 2012], um novo algoritmo genético foi modelado para escolher a ordem em que as demandas serão processadas pelo algoritmo RSA e para escolher entre os pares de caminhos candidatos para cada demanda.

O indivíduo do algoritmo genético possui duas características, a ordem de processamento das demandas, e os caminhos escolhidos para cada demanda. Essas características são representadas por um par de cromossomos por indivíduo, $Ind = \{O, Sel\}$, onde $O = \{d_x, d_y, \dots, d_z\}$, que representa a ordem em que as demandas serão processadas, sendo que $(d_x, d_y, \dots, d_z) \in D$. E $Sel = \{i_0, i_1, \dots, i_{|d|}\}$, que representa os índices dos pares de caminhos disjuntos escolhidos entre os candidatos de cada demanda. Conforme apresentado na Seção 5.2.1, C_d representa o conjunto de pares de caminhos disjuntos candidatos para a demanda d . Assim, $C_{d_i} \in C_d$, representa um par de caminhos disjuntos, dentre os candidatos para uma demanda $d \in D$, e i seu índice no conjunto C_d .

A Figura 5.2, exemplifica a representação de um indivíduo, levando em consideração um roteamento de 10 demandas, e que para cada demanda há 6 pares de caminhos disjuntos. Dessa forma, a representação do cromossomo $O = \{d_3, d_6, d_8, d_0, d_2, d_4, d_1, d_5, d_9, d_7\}$, representa a ordem em que as demandas d_i serão processadas, assim, o algoritmo genético processará demanda por demanda, iniciando pela demanda d_3 na ordem do vetor O finalizando na demanda d_7 . O cromossomo $Sel = \{0, 1, 0, 0, 3, 5, 2, 2, 0, 1\}$, representa os caminhos selecionados dentre os candidatos para cada demanda, dessa forma os caminhos a serem utilizados no roteamento será o conjunto de pares $\{C_{0_0}, C_{1_1}, C_{2_0}, C_{3_0}, C_{4_3}, C_{5_5}, C_{6_2}, C_{7_2}, C_{8_0}, C_{9_1}\}$. Nesse exemplo, para a demanda d_0 é utilizado o par de caminho disjunto com índice 0 dentro do conjunto de pares de caminhos disjuntos candidatos, para a demanda d_1 utiliza o par candidato de índice 1, para a demanda d_2 o par candidato de índice 0, etc.

$$\begin{array}{l}
 O = \begin{array}{|c|c|c|c|c|c|c|c|c|c|}
 \hline
 3 & 6 & 8 & 0 & 2 & 4 & 1 & 5 & 9 & 7 \\
 \hline
 \end{array} \\
 Sel = \begin{array}{|c|c|c|c|c|c|c|c|c|c|}
 \hline
 0 & 1 & 0 & 0 & 3 & 5 & 2 & 2 & 0 & 1 \\
 \hline
 \end{array}
 \end{array}$$

Figura 5.2 – Representação do Indivíduo.

A população é denotada por $Pop = \{Ind_0, Ind_1, \dots, Ind_{|Pop|}\}$. Inicialmente a população Pop é gerada de forma randômica. Para cada indivíduo Ind_i , o vetor O é gerado a partir de uma sequência aleatória das demandas, e o vetor Sel , sorteia um número randômico por posição, de 0 ao número de pares de caminhos candidatos ($|C_d|$). O número de indivíduos gerados para população inicial é recebido por parâmetro, na configuração do algoritmo genético.

A avaliação da aptidão do indivíduo é feita através do Algoritmo 5, onde recebe como entrada o grafo G , que representa a topologia física da rede e um conjunto P de todos os pares de

Algoritmo 5: Calcula Aptidão do Indivíduo

Data: Um grafo dirigido, com enlaces bidirecionais $G = (V, E)$; e o conjunto dos pares de caminhos disjuntos P .

```

1  $B \leftarrow \emptyset$ ;
2 foreach  $d_i \in O$  do
3    $j \leftarrow Sel[i]$ ;
4    $c \leftarrow \text{getCaminhoTrabalho}(C_{d_j})$ ;
5    $c' \leftarrow \text{getCaminhoBackup}(C_{d_j})$ ;
6   if alocação de Espectro FLF( $G, c, d$ ) then
7     if nao (alocação de Espectro FLF)( $G, c', d$ ) then
8        $\text{desaloca}(c)$ ;
9        $B \leftarrow B \cup d$ ;
10    end
11  end
12  else
13     $B \leftarrow B \cup d$ ;
14  end
15 end
16  $Pb \leftarrow \text{CalculaProbabilidadeDeBloqueio}(D, B)$ ;
17  $Aptidao \leftarrow 1 - Pb$ ;

```

caminhos disjuntos roteados. O algoritmo: (i) inicializa o conjunto das demandas bloqueadas B com conjunto vazio; (ii) para cada demanda, na ordem do conjunto O de ordem das demandas do indivíduo; (iii) armazena em j o índice do par de caminhos selecionado para a demanda, obtido do vetor Sel do indivíduo; (iv) armazena em c o caminho de trabalho selecionado em C_{d_j} ; (v) armazena em c' o caminho de *backup*; (vi) tenta alocar o caminho c na rede física, utilizando o Algoritmo 4, se o método retornar verdadeiro, indicando que a alocação foi realizada; (vii) tenta alocar o caminho c' , caso não seja possível alocar c' ; (viii) então desaloca o caminho c . Se um dos dois caminhos não consiga alocar na rede; (ix) adiciona a demanda d ao conjunto das demandas bloqueadas. Por fim, calcula a probabilidade de bloqueio, por demanda utilizando a Equação 5.1 ou por *slots* utilizando a Equação 5.2. A aptidão do indivíduo será $1 - Pb$, quanto menor a probabilidade de bloqueio maior será sua aptidão, variando de $0 - 1$.

A escolha dos indivíduos para cruzamento será através do método da roleta [Goldschmidt, 2010]. Um par de indivíduos selecionados, fará a operação de cruzamento cromossomo a cromossomo. O cruzamento dos cromossomos da ordem das demandas, representado na Figura 5.3(a), os cromossomos O_1 e O_2 , utilizam o método de corte de ponto único. Para isso, uma posição do vetor é escolhida aleatoriamente para o corte. Então, são gerados dois novos cromossomos, O'_1 composto pelo segmento do vetor O_1 , da posição inicial ao corte, seguido das

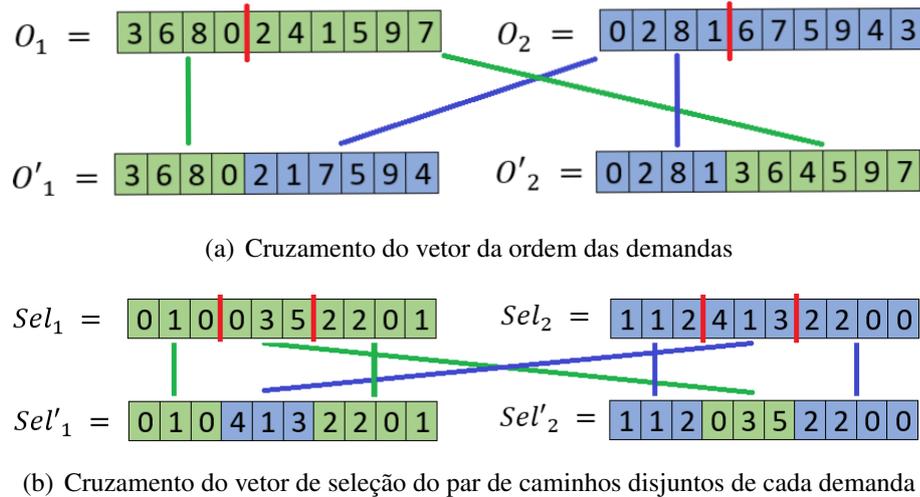


Figura 5.3 – Cruzamento

demandas faltantes na ordem que aparecem em O_2 , e vice-versa para O'_2 . Garantindo que todas as demandas são representadas no vetor dos cromossomos descendentes, apenas alternando a ordem.

O cruzamento do cromossomo de seleção dos pares de caminhos candidatos, conforme Figura 5.3(b), utiliza o método de cruzamento de múltiplos pontos. Para isso, são sorteados dois pontos de corte para os genitores Sel_1 e Sel_2 , para geração de dois novos cromossomos Sel'_1 e Sel'_2 . Após o cruzamento dos cromossomos, dois novos indivíduos são formados $\{O'_1, Sel'_1\}$ e $\{O'_2, Sel'_2\}$.

O operador genético de mutação, também tem tratamento diferenciado para cada um dos cromossomos do indivíduo. Sorteia-se 10% dos genes de cada cromossomo para sofrerem mutação. A Figura 5.4(a), demonstra o método de mutação para o cromossomo O , selecionando dois genes para sofrerem mutação, cada gene troca a ordem com seu gene vizinho. A Figura 5.4(b) demonstra o método de mutação para o cromossomo Sel , selecionando dois genes para mutação, sorteia um número randômico, de 0 ao total de pares de caminhos candidatos para a demanda, para substituí-lo.

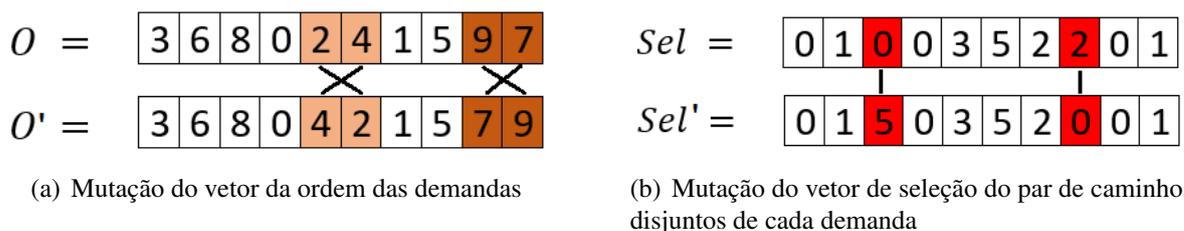


Figura 5.4 – Mutação

Algoritmo 6: Algoritmo Genético

Data: Um grafo dirigido, com enlaces bidirecionais $G = (V, E)$; e o conjunto dos pares de caminhos disjuntos P ; um conjunto de demandas D ; e três números QT_POP , $MAX_ITERACAO$ e $PC_MIN_EVOLUCAO$

Result: Indivíduo mais apto.

```

1  $Pop \leftarrow \text{geraPopulacaoInicial}(G, P, D, QT\_POP)$ ;
2  $AptidaoAnt \leftarrow \text{calculaAptidao}(Pop)$ ;
3 for  $i \leftarrow 1$  to  $MAX\_ITERACAO$  do
4   for  $j \leftarrow 0$  to  $(10\%|Pop|)$  do
5      $\{IndA, IndB\} \leftarrow \text{selecionaParIndividuosCruzamento}$ ;
6      $\{IndA', IndB'\} \leftarrow \text{cruzamento}(IndA, IndB)$ ;
7      $Pop \leftarrow Pop \cup \{IndA', IndB'\}$ ;
8      $\text{removeDoisIndividuosMenosAptos}$ ;
9   end
10   $Aptidao \leftarrow \text{calculaAptidao}(Pop)$ ;
11  if  $\text{calculaEvolucao}(AptidaoAnt, Aptidao) < 2 * PC\_MIN\_EVOLUCAO$ 
12    then
13      for  $j \leftarrow 1$  to  $(1\%|Pop|)$  do
14         $Ind \leftarrow Pop[\text{randomico}]$ ;
15         $\text{geraMutacao}(Ind)$ ;
16      end
17       $Aptidao \leftarrow \text{calculaAptidao}(Pop)$ ;
18    end
19    if  $\text{calculaEvolucao}(AptidaoAnt, Aptidao) < PC\_MIN\_EVOLUCAO$ 
20      then
21         $\text{break}$ ;
22      end
23     $AptidaoAnt \leftarrow Aptidao$ ;
24 end
25 return  $\text{getIndividuoMaisApto}(Pop)$ ;

```

Após as definições da estrutura do indivíduo e operadores genéticos, o Algoritmo 6 representa a formulação do algoritmo genético proposto. Recebe na entrada do método o grafo $G = (V, E)$, que representa a topologia física, o conjunto P de pares de caminhos disjuntos para as demandas, roteados no subproblema de roteamento, o conjunto D de demandas solicitadas e três números QT_POP , $MAX_ITERACAO$ e $PC_MIN_EVOLUCAO$ como parâmetros de controle do algoritmo genético. Inicialmente a população Pop inicial é gerada com QT_POP indivíduos, recebido por parâmetro, armazena em $AptidaoAnt$ o valor da aptidão total da população inicial, gerada a partir da soma da aptidão dos indivíduos. Inicia-se um laço com $MAX_ITERACAO$ de gerações da população, em cada geração, seleciona 10% dos

indivíduos da população e realiza o cruzamento de par em par destes selecionados. Os novos indivíduos gerados a partir do cruzamento são inseridos na população, removendo os menos aptos, aqueles com maior probabilidade de bloqueio.

Com a população atualizada, após as operações de cruzamento e substituição dos indivíduos, calcula-se a aptidão total da população para essa geração, armazenando em *Aptidao*. Posteriormente, calcula-se a evolução da população entre a *Aptidao* e a *AptidaoAnt*, se ela for inferior ao dobro do mínimo de evolução *PC_MIN_EVOLUCAO* esperada para cada geração, tentará inserir mutação no genes de 1% da população. Realizando as mutações, atualiza a *Aptidao* calculando novamente a aptidão atual da população. Se mesmo após as mutações, a população ainda não evoluiu o mínimo esperado, então sai do laço. Caso a evolução foi superior a mínima, atualiza a variável $AptidaoAnt \leftarrow Aptidao$, para gerar uma nova geração da população. Por fim, o indivíduo mais apto da última geração da população será retornado como solução do RSA que menos gerou bloqueios, trazendo a informação genética da ordem em que as demandas devem ser alocadas, e as escolhas dos caminhos de cada demanda.

6 EXPERIMENTOS E RESULTADOS

Os experimentos envolveram a implementação de 3 algoritmos para solução do problema do RSA. O objetivo foi validar se as propostas minimizam a probabilidade de bloqueio por demanda (Pbd) e por *slot* (Pbs), em relação a solução do RSA sem o uso de inteligência artificial, em uma das abordagens apresentadas em [Chatterjee et al., 2015]. Esses algoritmos recebem como entrada (i) a topologia física da rede, uma lista de enlaces, representados por 5 valores: os nós origem e destino do enlace, o custo do enlace, o número de *slots* que o enlace possui e a sua granularidade; e (ii) a matriz de demandas representada por três números: os nós origens e destinos da demanda e o número de *slots* necessários para atender a demanda. Tais algoritmos são:

- Roteamento Fixo + FLF: Nessa implementação, foi reproduzido uma das possíveis soluções do RSA apresentados em [Chatterjee et al., 2015], foi selecionado o algoritmo de roteamento fixo, com o algoritmo de Suurballe [Suurballe and Tarjan, 1984] para encontrar o menor par de caminhos disjuntos para cada demanda. E para alocação do espectro foi implementado a política FLF, demonstrado no Algoritmo 4 na Seção 5.2.2. O algoritmo recebe demanda a demanda sequencialmente, na ordem em que as demandas são lidas na entrada do programa. Para cada demanda, encontra o menor par de caminhos disjuntos, e tenta alocar espectro na rede para o caminho utilizando a política FLF. Caso não seja possível realizar a alocação, a demanda é marcada como bloqueada. Como resultado, o algoritmo retorna a matriz de alocação do espectro e as estatísticas de Pbd e Pbs ;
- Roteamento KDPPs + AG: Essa implementação, utiliza o algoritmo KDPPs [Phung et al., 2005] para gerar um conjunto de k pares de caminhos disjuntos para cada demanda. Fornecendo como entrada para o algoritmo genético proposto. Nos experimentos realizados, foi utilizado $k = 25$, ou seja, até 25 pares de caminhos disjuntos candidatos para cada demanda, sendo que o número de candidatos pode ser inferior se não houverem 25 caminhos disjuntos possíveis para a demanda. Assim, o algoritmo genético vai alternando na escolha do candidato, buscando os candidatos que minimizem a probabilidade de bloqueio;
- Roteamento NMPCD + AG: Por fim, nessa implementação foi utilizado o Algoritmo 2

(NMPCD), proposto na Seção 5.2.1, para gerar um conjunto de $n \times m$ pares de caminhos disjuntos candidatos para cada demanda. Assim como na implementação anterior, esse conjunto de pares de caminhos disjuntos candidatos, é utilizado como entrada no algoritmo genético proposto. Nos experimentos realizados, foram utilizados os valores $n = 5$ e $m = 5$, dessa forma, para cada demanda obteve-se até 5 caminhos mínimos de trabalho, e para cada um destes, foi gerado até 25 caminhos de *backup*. Ressalta-se que para uma determinada demanda, podem não haver 5 caminhos de trabalho ou 5 caminhos disjuntos para *backup* para cada um dos caminhos de trabalho, gerando um valor inferior à esses números. Num total, para cada demanda, são gerados um conjunto de até 25 pares de caminhos disjuntos candidatos, para que o algoritmo genético possa selecionar o candidato que minimize a probabilidade de bloqueio.

Foram selecionadas para o experimento, 4 topologias físicas de redes de referência, em uma base de dados disponível em [Pavan, 2016], ilustradas na Figura 6.1. As redes possuem diferenças em seu tamanho, inicializando com uma rede que possui um número inferior de nós, crescendo gradativamente até a rede com maior número de nós. Dessa forma é possível analisar a eficiência das propostas em cenários diferente, verificando se a otimização ocorre em qualquer tamanho de rede.

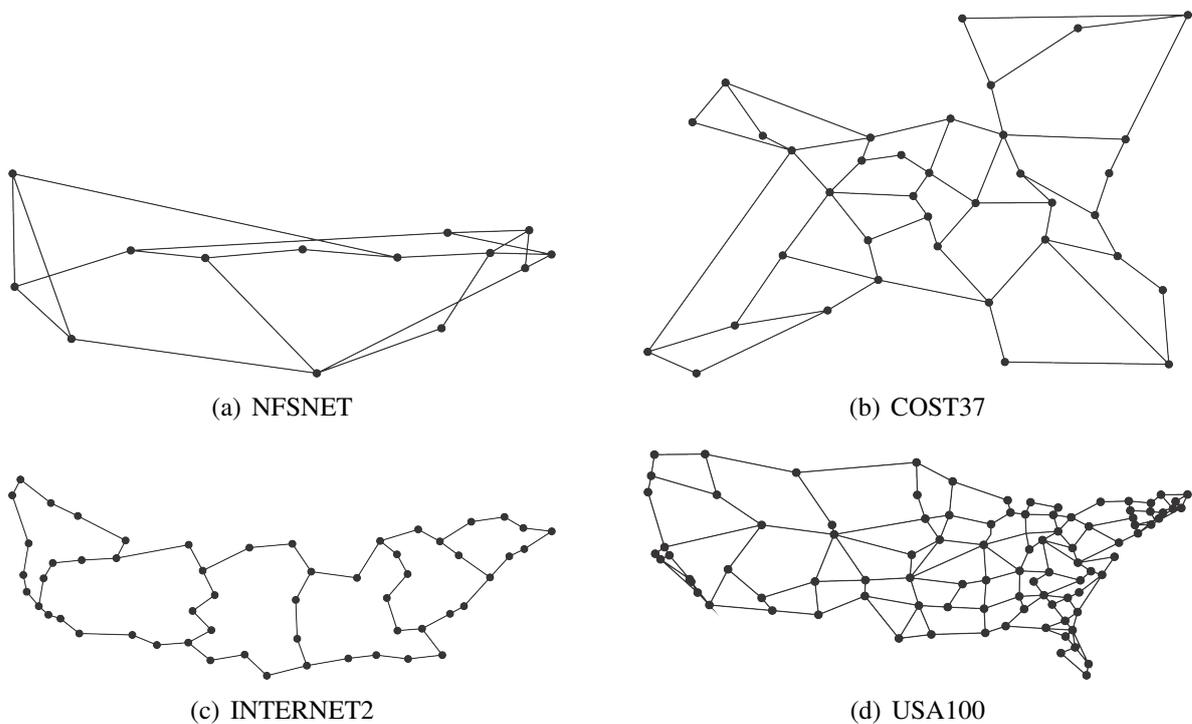


Figura 6.1 – Topologias de redes.

As redes selecionadas foram:

- *National Science Foundation Network* (NSFNET, Figura 6.1(a)): Essa rede localizada nos Estados Unidos da América, possui 14 nós e 21 enlaces, o seu grau médio de adjacências nos nós é 3;
- COST37, Figura 6.1(b): Essa rede está localizada na Europa, contendo nós em diversos países, possui 37 nós e 57 enlaces, o seu grau médio de adjacências nos nós é 3.08;
- Internet2 Network (INTERNET2, Figura 6.1(c)): Localizada nos Estados Unidos da América, possui 56 nós e 61 enlaces, o seu grau médio de adjacências nos nós é 2.18;
- *Korotky's Prototypical Backbone Network* (USA100, Figura 6.1(d)): Por fim, a maior rede da base de dados selecionada, essa rede também está localizada nos Estados Unidos da América, possui 100 nós e 171 enlaces, o seu grau médio de adjacências nos nós é 3.42.

Nessas topologias, não são fornecidas as informações dos comprimentos dos enlaces, portando, como valor de custo de cada enlace foi fixado o valor 1. Dessa forma, assume-se que o menor caminho entre dois nós é medido em número de enlaces, ou seja, número de saltos, métrica amplamente utilizada em estudos da área. As redes selecionadas também não possuem o número de *slots* e granularidade das redes elásticas. Para essas informações, foram assumidos 800 *slots*, com granularidade de 12.5 GHz, disponível em cada enlace, conforme descrito na Seção 5.1. Com as informações das topologias físicas da base de dados, unidas aos valores fixados, montou-se um arquivo para cada rede, modelado para a entrada nas 3 implementações experimentadas.

As demandas de tráfego são representadas em número de *slots* necessários entre um par de nós distintos. Para os experimentos foram geradas 10 listas de demandas para cada uma das redes. Para cada rede, inicialmente foram distribuídos uma quantidade total de demanda de 500 *slots*, e aumentando de 500 em 500 até 5000 *slots*, gerando assim as 10 listas de demandas. A distribuição das quantidades de demandas, de cada uma das listas, ocorreu de forma randômica. Na geração de cada lista de demandas, por exemplo na primeira lista de 500 *slots*, foram sorteados dois números distintos para cada um dos 500 *slots* demandados. Esses números sorteados, representam os nós de origem e destino da demanda. Essa demanda será incrementada ao valor constante na lista de demandas para o par de nós. Dessa forma, cada par de nós, origem e destino, demandará o número de *slots* igual ao número de vezes que foi sorteado na distribuição dos

slots. Assim os arquivos possuindo as listas de demandas, um para cada rede, foram modelados para entrada das 3 implementações experimentais.

O algoritmo genético proposto, utilizando o roteamento KDPPs e utilizando o roteamento NMPCD, foram configurados da seguinte forma:

- Número máximo de iterações: 5000;
- Percentual mínimo de evolução esperada entre as gerações: 0.001%;
- Tamanho da população: 1000 indivíduos.

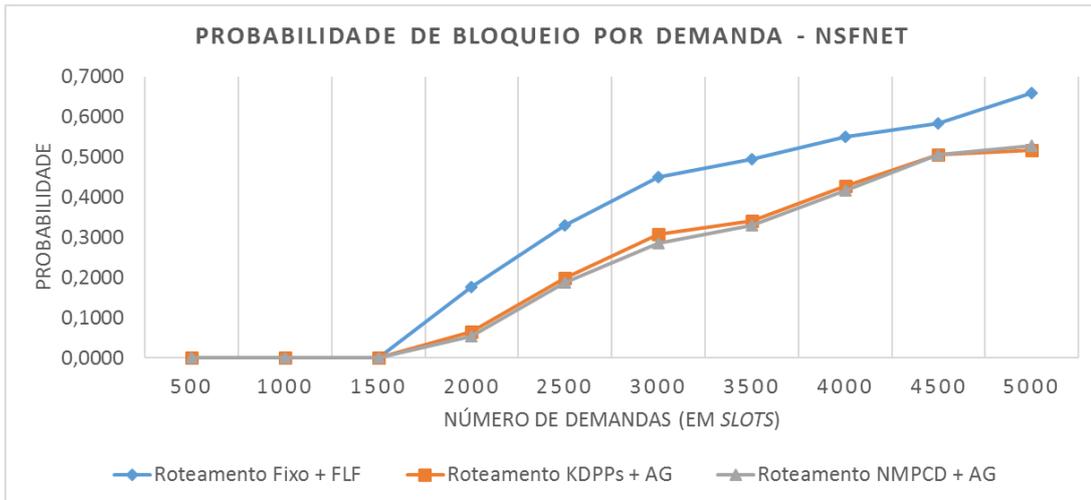
Estabelecidos os algoritmos e valores de entrada e configuração dos experimentos, para cada rede selecionada e cada métrica, estes foram realizados da seguinte forma: (i) gerou-se 10 listas de demandas aleatórias; (ii) para cada uma das listas de demandas, primeiramente foi rodado o algoritmo Roteamento Fixo + FLF, posteriormente o Roteamento KDPPs + AG, e por fim, Roteamento NMPCD + AG; (iii) Os valores de Pbd ou Pbs de cada algoritmo foram gravados em arquivos pela saída do programa; (iv) os valores gravados, foram organizados em uma planilha onde foi possível gerar gráficos para comparar a probabilidade de bloqueio entre as mesmas. Ao todo, foram montados 8 experimentos, nas combinações das quatro redes selecionadas, e as duas métricas de probabilidade de bloqueio, por demanda ou por *slot*.

As implementações foram realizadas na linguagem Java, utilizando paralelismo (*multi-thread*) para cálculos de aptidão, geração e mutação dos indivíduos da população no algoritmo genético. Os experimentos ocorreram em um computador com a seguinte configuração: (i) Processador Intel i5 com 2.8 Ghz, 2 núcleos físicos, virtualizando 4 núcleos lógicos (*Threads*); (ii) 8 Gb de memória RAM, DDR3 1.333 GHz.

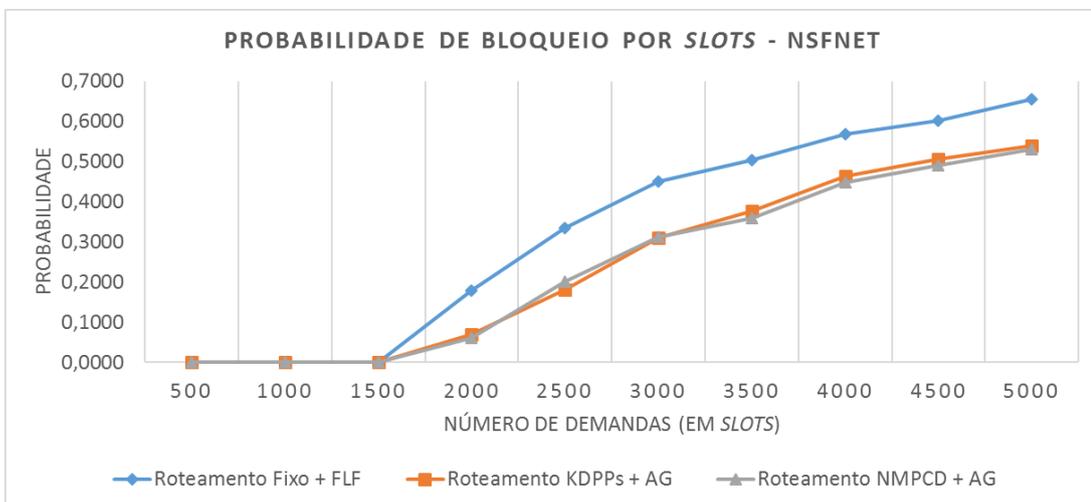
6.1 Apresentação dos Resultados

Para apresentar os resultados obtidos no experimento, dois gráficos foram gerados para cada uma das redes selecionadas, um para exibir a métrica de probabilidade de bloqueio por demanda (Pbd) e outro, para a métrica de probabilidade de bloqueio por *slot* (Pbs). Os gráficos comparam os 3 algoritmos, em cada uma das 10 listas de demandas, iniciando pela lista de demanda onde foram distribuídos uma necessidade de 500 *slots*, até a de 5000 *slots*.

O gráfico na Figura 6.2, representa os resultados obtidos dos experimentos na rede NSF-NET. Para essa rede, o número total de *slots* de cada lista de demandas, foi distribuído em 91



(a) Probabilidade de bloqueio por demanda



(b) Probabilidade de bloqueio por slots

Figura 6.2 – Probabilidades de boqueio - NSFNET

pares de nós, origem e destino, distintos, gerando assim 91 demandas, com número de *slots* distintos.

Para a P_{bd} , Figura 6.2(a), as listas de demandas de 500 a 1500 *slots* não tiveram bloqueios, ou seja, todas as demandas foram acomodadas na rede. A partir da lista de demanda de 2000 *slots*, houve uma diferença entre os algoritmos, sendo que o algoritmo de Roteamento Fixo + FLF, apresentou maior probabilidade de bloqueio, seguido pelo Roteamento KDPPs + AG e o Roteamento NMPCD + AG, respectivamente. Em média, o Roteamento KDPPs + AG, apresentou uma probabilidade de bloqueio 31.74% menor em relação ao Roteamento Fixo + FLF, considerando apenas as listas de demandas que apresentaram bloqueio. Já o Roteamento NMPCD + AG, teve 4.72% menos bloqueio em relação ao Roteamento KDPPs + AG, sendo 34.17% inferior ao Roteamento Fixo + FLF. Em sua maior diferença, na lista de 3000 *slots*,

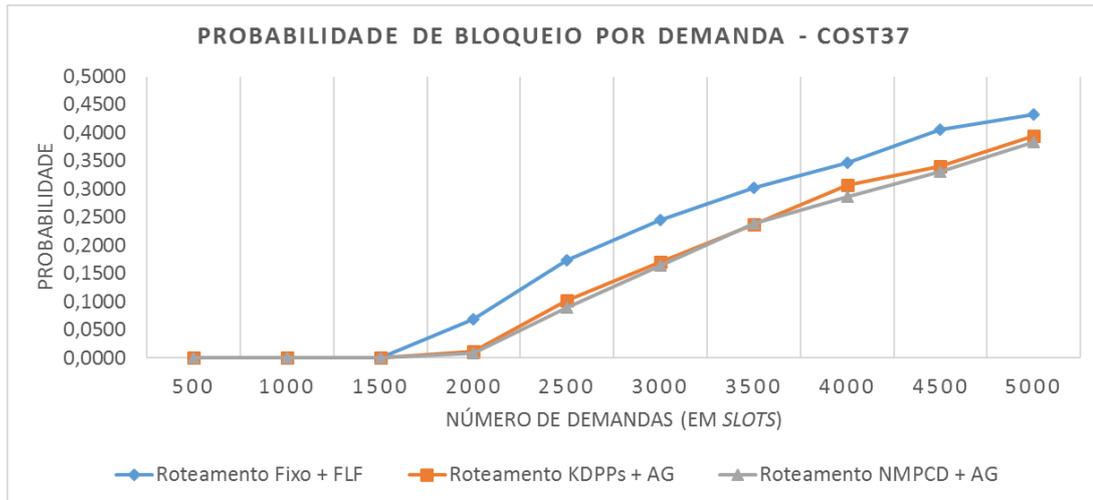
enquanto o Roteamento Fixo + FLF teve 41 demandas bloqueadas, o Roteamento NMPCD + AG teve 26 demandas bloqueadas, ou seja, nessa implementação foram atendidas 15 demandas a mais, reduzindo a probabilidade de bloqueio, dessa lista de demandas, de 0.45 para 0.28.

Também não houve bloqueios nas listas de demandas de 500 a 1500 *slots*, para *Pbs*, Figura 6.2(b). A diferença entre os algoritmos foi similar à apresentada na *Pbd*, ficando com maior probabilidade de bloqueio com o Roteamento Fixo + FLF, seguido do Roteamento KDPPs + AG e Roteamento NMPCD + AG. Na média, o algoritmo Roteamento KDPPs + AG, apresentou uma probabilidade de bloqueio 30.76% menor que o Roteamento Fixo + FLF. O Roteamento NMPCD + AG, apresentou uma probabilidade de bloqueio 1.84% menor que o Roteamento KDPPs e 31.95% menor que o Roteamento Fixo + FLF. Essa diferença, na *Pbs*, representa na lista de 5000 *slots*, um total de 619 *slots* atendidos a mais no Roteamento NMPCD + AG em relação ao Roteamento Fixo + FLF.

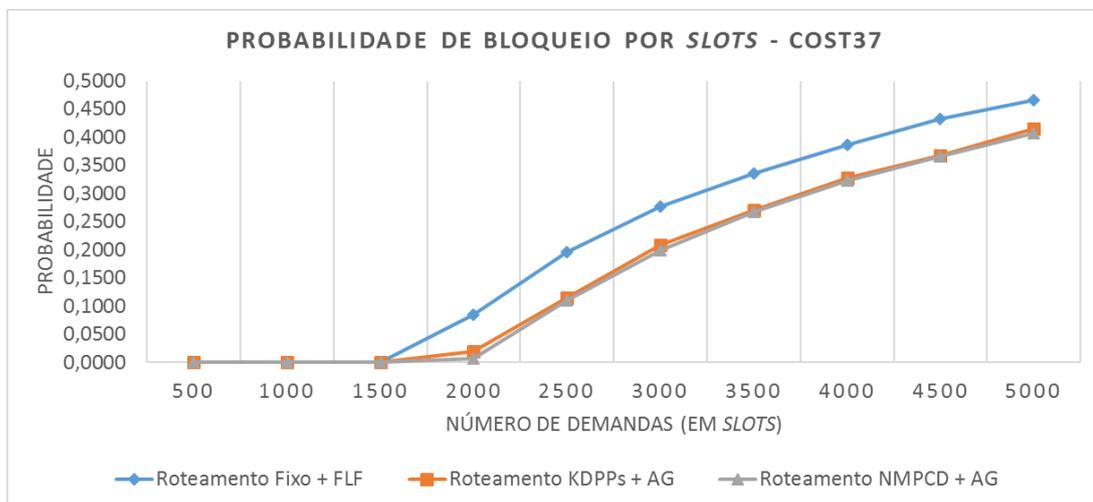
Os resultados obtidos da rede COST37, são exibidos no gráfico da Figura 6.3. O número total de *slots* de cada lista de demandas, foram distribuídos em 666 demandas.

Para os experimentos de *Pbd*, Figura 6.3(a), as listas de demandas de 500 a 1500 *slots* não tiveram bloqueios. Nas demais listas, o algoritmo de Roteamento Fixo + FLF, apresentou maior probabilidade de bloqueio. O Roteamento KDPPs + AG, apresentou uma probabilidade de bloqueio 30.38% menor em relação ao Roteamento Fixo + FLF, considerando apenas as listas de demandas que apresentaram bloqueio. O Roteamento NMPCD + AG, teve 8.28% menos probabilidade de bloqueio em relação ao Roteamento KDPPs + AG, e 34.1% menor em relação ao Roteamento Fixo + FLF. Na lista de 2500 *slots*, o gráfico apresenta a maior diferença, o Roteamento Fixo + FLF apresentou 113 demandas bloqueadas, enquanto que o Roteamento NMPCD + AG teve 58 demandas bloqueadas, reduzindo a probabilidade de bloqueio, dessa lista de demandas, de 0.17 para 0.089.

Também não houve bloqueios nas listas de demandas de 500 a 1500 *slots*, para *Pbs*, Figura 6.3(b). O Roteamento Fixo + FLF continuou tendo a maior probabilidade de bloqueio, seguido do Roteamento KDPPs + AG e Roteamento NMPCD + AG. Na média, o Roteamento KDPPs + AG, apresentou uma probabilidade de bloqueio 29.07% menor que o Roteamento Fixo + FLF. Enquanto, o Roteamento NMPCD + AG ficou com probabilidade de bloqueio, 10.81% menor que o Roteamento KDPPs e 32.6% menor que o Roteamento Fixo + FLF. Na lista de 5000 *slots*, o Roteamento NMPCD + AG, atendeu a um total de 296 *slots* a mais que o Roteamento Fixo + FLF.



(a) Probabilidade de bloqueio por demanda



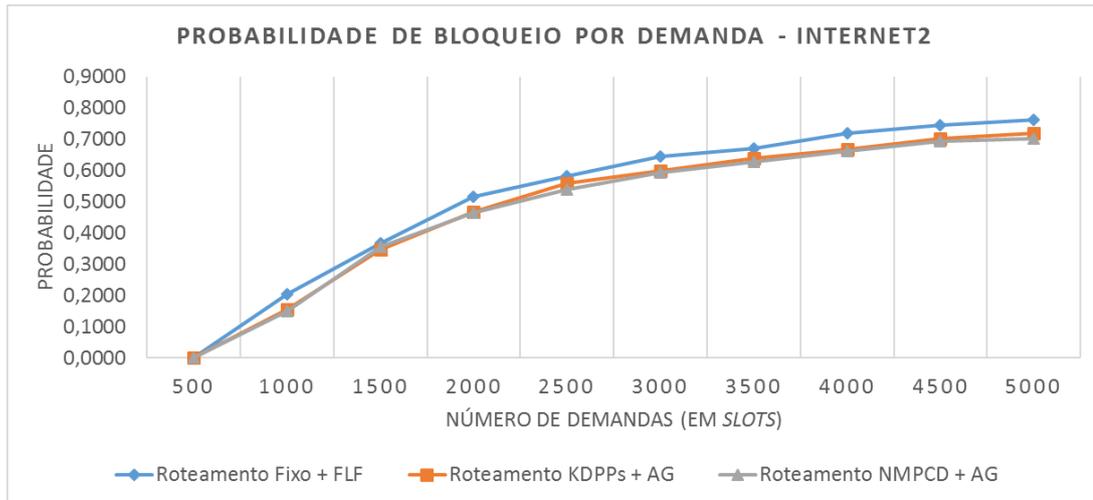
(b) Probabilidade de bloqueio por slots

Figura 6.3 – Probabilidades de boqueio - COST37

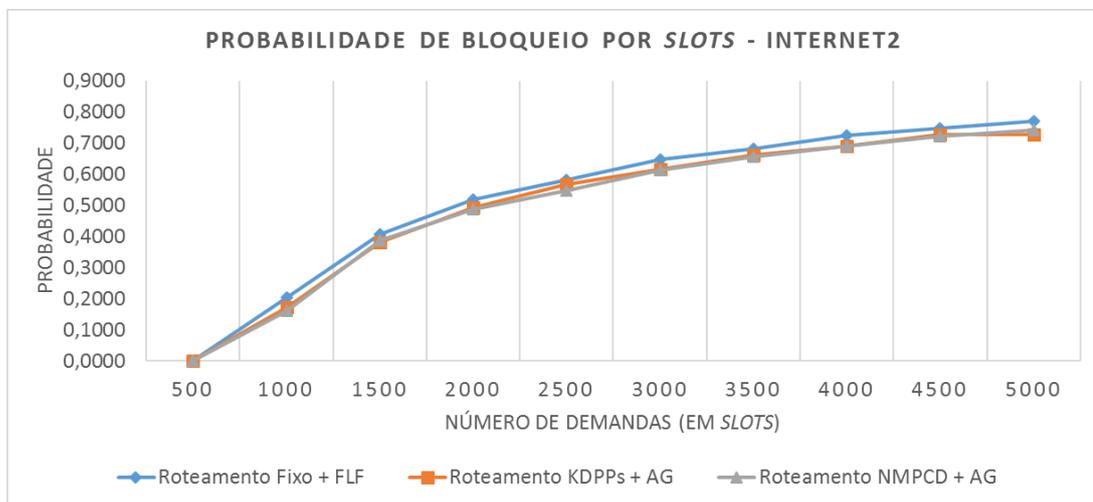
A rede INTERNET2 apresentou resultados mais próximos entre os algoritmos, em relação às demais, conforme pode ser visualizado no gráfico da Figura 6.4. Os slots das listas de demandas forma distribuídas em 1540 demandas.

O Roteamento Fixo + FLF permaneceu com maior probabilidade de bloqueio, seguido do Roteamento KDPPs + AG, em média, com Pbd 8.16% menor e Pbs 5.03% menor; e do Roteamento NMPDC + AG, com 9.4% menos Pbd e 5.9% menos Pbs .

Observou-se que essa rede possui grau médio menor que as demais, apenas 2.18. Isso significa que essa rede possui, em média, menos ligações por nó da rede, como pode ser observado na Figura 6.1(c). Isso impactou diretamente nos resultados de otimização das probabilidades de bloqueio, apresentando diferenças entre os métodos menor que às demais. Isso deve-se ao fato de que o roteamento fixo escolhe o menor par de caminhos disjuntos entre dois nós,



(a) Probabilidade de bloqueio por demanda



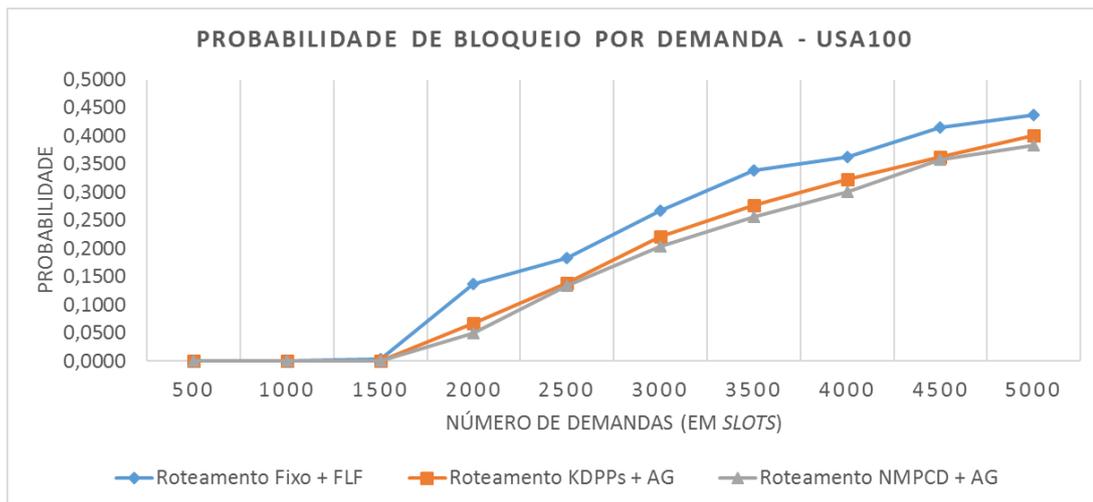
(b) Probabilidade de bloqueio por slots

Figura 6.4 – Probabilidades de boqueio - INTERNET2

restando poucos outros caminhos alternativos, ou os caminhos alternativos com maior número de enlaces utilizados, ocupando rapidamente os *slots* disponíveis. A falta de alternativas que minimizem o uso de recurso da rede, faz com que o algoritmo genético não consiga resultados mais expressivos, como nas demais redes.

Apesar de possuir 56 nós, a rede INTERNET2, apresentou maior probabilidade de bloqueio em relação às demais redes. Com o algoritmo Roteamento NMPCD + AG, que apresentou resultados melhores em P_{bd} e P_{bs} , ficou abaixo de 0.5 nas demais redes até para as listas de demandas de 5000 *slots*, entretanto na INTERNET2 apresentou 0.74, ultrapassando os 0.5 já na lista de demandas de 2500 *slots*. O baixo número de enlaces em relação aos nós dessa rede, tornam-na mais ineficientes ao uso de proteção dedicada, pois o caminho de *backup* utiliza mais recursos por haver menos caminhos alternativos ou estes possuem maior número de enlaces.

Por fim, o gráfico na Figura 6.5, apresenta os resultados da maior rede selecionada, a USA100. Contendo 100 nós, os *slots* das listas de demandas foram distribuídos em 4950 demandas. Para essa rede, não houve bloqueios para as listas de demandas de 500 a 1500 *slots*, para os algoritmos Roteamento KDPPs + AG e Roteamento NMPCD + AG. Portanto esses valores não são considerados na comparação.



(a) Probabilidade de bloqueio por demanda

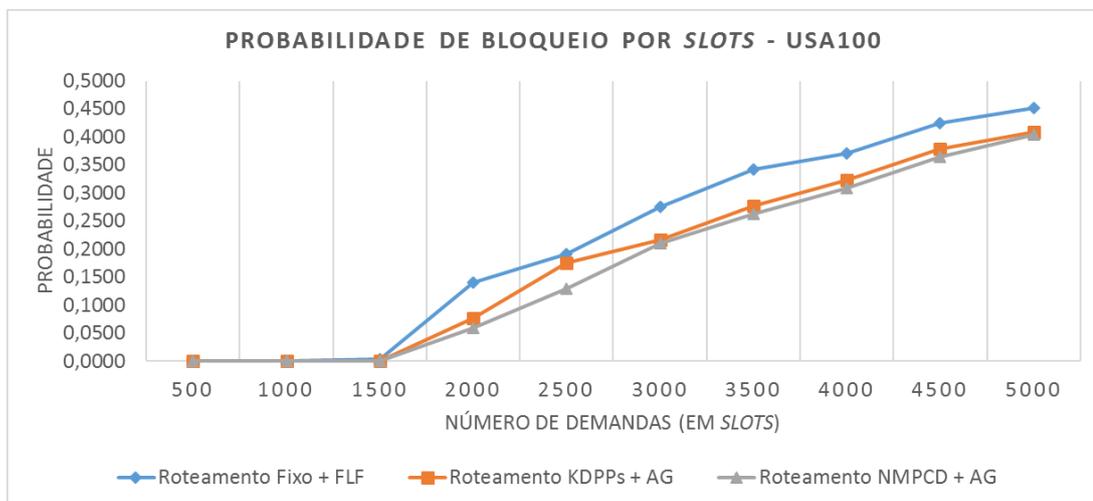
(b) Probabilidade de bloqueio por *slots*

Figura 6.5 – Probabilidades de boqueio - USA100

A P_{bd} , apresentada no gráfico da Figura 6.5(a), mostra o Roteamento Fixo + FLF com maior probabilidade de bloqueio. Em sequência, o Roteamento KDPPs + AG com uma P_{bd} em média 30.36% menor, seguido do Roteamento NMPCD + AG, com P_{bd} em média 35.13% menor que o Roteamento Fixo + FLF.

Na Figura 6.5(b), o gráfico mostra a P_{bs} da rede USA100. Assim como a P_{bd} , o Roteamento Fixo + FLF teve maior P_{bs} . O Roteamento KDPPs + AG, apresentou, em média,

uma Pbs 18.06% menor que o Roteamento Fixo + FLF, seguido do Roteamento NMPCD + AG, com probabilidade 25.42% menor que o Roteamento Fixo + FLF.

A Tabela 6.1 apresenta uma comparação das médias das diferenças entre os algoritmos para cada rede e métrica. Para composição da média das diferenças, só foram considerados os intervalos das listas de demandas em que houve bloqueio.

| Topologia da rede | Probabilidade de bloqueio por | Roteamento Fixo + FLF | Roteamento KDPPs + AG | Roteamento Fixo + FLF |
|-------------------|-------------------------------|-----------------------------|-----------------------------|-----------------------------|
| | | x Roteamento KDPPs + AG (%) | x Roteamento NMPCD + AG (%) | x Roteamento NMPCD + AG (%) |
| NSFNET | Demanda | -31.74 | -4.72 | -34.17 |
| | Slot | -30.76 | -1.84 | -31.95 |
| COST37 | Demanda | -30.38 | -8.28 | -34.10 |
| | Slot | -29.07 | -10.81 | -32.60 |
| INTERNET2 | Demanda | -8.16 | -1.39 | -9.40 |
| | Slot | -5.03 | -0.97 | -5.90 |
| USA100 | Demanda | -30.36 | -6.93 | -35.13 |
| | Slot | -18.06 | -9.47 | -25.42 |
| Média | | -22.94 | -5.55 | -26.08 |

Tabela 6.1 – Comparação das médias dos resultados

A diferença entre o Roteamento Fixo + FLF e o Roteamento KDPPs + AG, para os experimentos realizados, ficou em média -22.94% , ou seja, o Roteamento KDPPs + AG obteve 22.94% menos probabilidade de bloqueio que o Roteamento Fixo + FLF. Nessa comparação, a diferença chegou a -31.74% , em sua maior amplitude.

Na comparação entre o Roteamento KDPPs + AG e Roteamento NMPCD + AG, nota-se que o segundo, apresentou em média 5.55% menos probabilidade de bloqueio em relação ao primeiro, e sua maior diferença foi de -10.81% . Já na comparação entre o Roteamento Fixo + FLF e o Roteamento NMPCD + AG, o segundo apresenta em média 26.08% menos probabilidade de bloqueio em comparação com primeiro, e na máxima diferença apresentou 35.13% menor probabilidade de bloqueio.

Ressalta-se que, o uso do algoritmo genético requer um tempo maior de computação, tornando essas implementações dessas soluções para o RSA viáveis somente para dimensionamento de redes *off-line*, conforme proposta nesse trabalho. Nos experimentos realizados da rede USA100, a maior rede utilizada por exemplo, enquanto o tempo de computação do Roteamento Fixo + FLF foi em torno de 1 a 2 segundos, com o uso do algoritmo genético, o tempo

de computação foi em média de 36 minutos, para cada lista de demandas experimentadas.

Com esses resultados, é possível observar a otimização obtida com o uso do algoritmo genético para solução do problema RSA. As duas implementações propostas, obtiveram uma redução considerável na probabilidade de bloqueio em relação a uma implementação amplamente utilizada na literatura. Outro ponto apresentado é em relação ao método de geração do conjunto de pares de caminhos disjuntos, candidatos para atender a uma determinada demanda. Em ambos experimentos, o tamanho máximo do conjunto de candidatos por demanda foi de 25, a implementação utilizando o roteamento KDPPs [Phung et al., 2005] apresentou maior probabilidade de bloqueio que a implementação do roteamento NMPCD, proposta por esse trabalho.

7 CONSIDERAÇÕES FINAIS

Esse trabalho abordou o problema RSA para redes ópticas elásticas com proteção dedicada, para falhas de um enlace. Foram apresentados soluções da literatura para o problema, adicionalmente foi proposto o uso de algoritmos genéticos para otimizar as soluções para o problema. Esses algoritmos escolhem, entre um conjunto de possíveis soluções, uma solução que minimiza a probabilidade de bloqueio por demanda ou por *slots*, as duas métricas adotadas.

Outro ponto abordado, foi a geração do conjunto de caminhos candidatos de cada demanda. Foi implementada uma solução da literatura, o roteamento KDPPs [Phung et al., 2005], e proposto um novo método para roteamento, o NMPCD apresentado na Seção 5.2.1. Essas duas implementações, geram os candidatos para que o algoritmo genético possa escolher o par de caminhos disjuntos em cada demanda que minimize as probabilidades de bloqueio.

Os resultados foram obtidos através de experimentos, em múltiplos cenários, em redes com diferentes topologias e tamanhos. Constatou-se que o uso do algoritmo genético proposto nesse trabalho, reduziu a probabilidade de bloqueio por demanda e por *slot*. A redução das probabilidades de bloqueio, foi de 22.94% e 26.08% em relação a implementação sem o uso de algoritmo genético para as implementações com KDPPs e NMPCD respectivamente. Outro resultado dos experimentos aponta para a geração dos caminhos candidatos, usados na entrada do algoritmo genético. O algoritmo de roteamento NMPCD, obteve uma probabilidade de bloqueio 5.55% inferior em relação ao roteamento KDPPs.

Portanto, o estudo aponta que há uma vantagem quanto ao uso de algoritmo genético para dimensionamento de redes ópticas elásticas, levando em consideração a probabilidade de bloqueio. Combinando ao roteamento NMPCD, essa vantagem é ampliada. A redução da probabilidade de bloqueio traz vantagens quanto ao uso dos recursos da rede, atendendo ao maior número de demandas possível, sem a necessidade de ampliar a estrutura da rede.

Para trabalhos futuros, sugere-se o estudo do impacto dessa otimização sobre outras métricas, como o custo de implementação, custo de trabalho da rede, tempo de atraso e salto médio provocado por escolhas de caminhos maiores do algoritmo genético, etc. Também pode-se ampliar o número de variáveis alternadas pelo algoritmo genético, como por exemplo, a escolha da modulação.

REFERÊNCIAS

- [1] Spectral grids for wdm applications: Dwdm frequency grid. Recommendation itu-t g.694.1, International Telecommunication Union, Outubro 2012.
- [2] Architecture of optical transport networks. Recommendation itu-t g.872, International Telecommunication Union, Outubro 2012.
- [3] E. Bouillet, G. Ellinas, J. Labourdette, and R. Ramamurthy. *Path Routing in Mesh Optical Networks*. Wiley, 2007. ISBN 9780470032978.
- [4] B. C. Chatterjee, N. Sarma, and E. Oki. Routing and spectrum allocation in elastic optical networks: A tutorial. *IEEE Communications Surveys Tutorials*, 17(3):1776–1800, thirdquarter 2015. ISSN 1553-877X. doi: 10.1109/COMST.2015.2431731.
- [5] K. Christodoulopoulos, I. Tomkos, and E. A. Varvarigos. Elastic bandwidth allocation in flexible ofdm-based optical networks. *Journal of Lightwave Technology*, 29(9):1354–1366, May 2011. ISSN 0733-8724. doi: 10.1109/JLT.2011.2125777.
- [6] D. Eppstein. Finding the k shortest paths. In *Proc. 35th Symp. Foundations of Computer Science*, pages 154–165. IEEE, November 1994.
- [7] O. Gerstel, M. Jinno, A. Lord, and S. J. B. Yoo. Elastic optical networking: a new dawn for the optical layer? *IEEE Communications Magazine*, 50(2):s12–s20, February 2012. ISSN 0163-6804. doi: 10.1109/MCOM.2012.6146481.
- [8] R. R. Goldschmidt. *Uma Introdução à Inteligência Computacional: fundamentos, ferramentas e aplicações*. IST-Rio, 2010. URL <http://www.boente.eti.br/boente2012/fuzzy/ebook/ebook-fuzzy-goldschmidt.pdf>.
- [9] 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*, 47(11):66–73, November 2009. ISSN 0163-6804. doi: 10.1109/MCOM.2009.5307468.
- [10] M. Klinkowski. A genetic algorithm for solving rsa problem in elastic optical networks with dedicated path protection. In *SOFTCOMP*. Springer Publishing Company, Incorporated, 2012. ISBN 3642330177, 9783642330179.

- [11] R. M. Morais, C. Pavan, A. N. Pinto, and C. Requejo. Genetic algorithm for the topological design of survivable optical transport networks. *IEEE/OSA Journal of Optical Communications and Networking*, 3(1):17–26, January 2011. doi: 10.1364/JOCN.3.000017.
- [12] C. Pavan. *Dimensioning of Multilayer Optical Networks*. PhD thesis, Universidade de Aveiro, Aveiro, Portugal, March 2011.
- [13] C. Pavan. A set of real-world optical transport network topologies., Julho 2016. URL <http://www.av.it.pt/anp/on/refnet2.html>.
- [14] Q. V. Phung, D. Habibi, H. N. Nguyen, and K. Lo. K pairs of disjoint paths algorithm for protection in wdm optical networks. In *2005 Asia-Pacific Conference on Communications*, pages 183–187, Oct 2005. doi: 10.1109/APCC.2005.1554044.
- [15] S. Ramamurthy and B. Mukherjee. Survivable wdm mesh networks. part i-protection. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 744–751 vol.2, Mar 1999. doi: 10.1109/INFCOM.1999.751461.
- [16] A. F. D. SANTOS. *Algoritmos para Roteamento e Alocação de Espectro em Redes Ópticas Elásticas*. Tese de doutorado, UNIVERSIDADE FEDERAL DA BAHIA – UFBA, Salvador- BA, Brasil, January 2015.
- [17] J. W. Suurballe. Disjoint paths in a network. *Networks*, 4(2):125–145, 1974. ISSN 1097-0037. doi: 10.1002/net.3230040204. URL <http://dx.doi.org/10.1002/net.3230040204>.
- [18] J. W. Suurballe and R. E. Tarjan. A quick method for finding shortest pairs of disjoint paths. *Networks*, 14(2):325–336, 1984. ISSN 1097-0037. doi: 10.1002/net.3230140209. URL <http://dx.doi.org/10.1002/net.3230140209>.
- [19] K. Walkowiak, M. Klinkowski, B. Rabięga, and R. Goścień. Routing and spectrum allocation algorithms for elastic optical networks with dedicated path protection. *Optical Switching and Networking*, 13:63 – 75, 2014. ISSN 1573-4277. doi: <http://dx.doi.org/10.1016/j.osn.2014.02.002>. URL <http://www.sciencedirect.com/science/article/pii/S1573427714000150>.

- [20] P. Wright, A. Lord, and L. Velasco. The network capacity benefits of flexgrid. In *Optical Network Design and Modeling (ONDM), 2013 17th International Conference on*, pages 7–12, April 2013.
- [21] X. Yu, Y. Zhao, J. Zhang, L. Gao, J. Zhang, and X. Wang. Spectrum defragmentation implementation based on software defined networking (sdn) in flexi-grid optical networks. In *Computing, Networking and Communications (ICNC), 2014 International Conference on*, pages 502–505, Fevereiro 2014. doi: 10.1109/ICCNC.2014.6785386.
- [22] G. Zhang, M. D. Leenheer, A. Morea, and B. Mukherjee. A survey on ofdm-based elastic core optical networking. *IEEE Communications Surveys Tutorials*, 15(1):65–87, First 2013. ISSN 1553-877X. doi: 10.1109/SURV.2012.010912.00123.
- [23] Y. Zhao, J. Zhang, J. Zhang, and W. Gu. Analysis of blocking probability for ofdm-based bandwidth-variable optical networks. In *Communications and Networking in China (CHINACOM), 2012 7th International ICST Conference on*, pages 288–292, Aug 2012. doi: 10.1109/ChinaCom.2012.6417492.