



**UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE
COMUNICAÇÃO E AUTOMAÇÃO**

TARCISIO DA SILVA BARRETO

**ANÁLISE DE TAXA MÉDIA DE BLOQUEIO EM
CONEXÕES POR ALGORITMOS DE CAMINHOS
MÍNIMOS: ALGORITMO DE YEN E ALGORITMO
GENÉTICO**

**MOSSORÓ – RN
2014**

TARCISIO DA SILVA BARRETO

**ANÁLISE DE TAXA MÉDIA DE BLOQUEIO EM
CONEXÕES POR ALGORITMOS DE CAMINHOS
MÍNIMOS: ALGORITMO DE YEN E ALGORITMO
GENÉTICO**

Dissertação de mestrado acadêmico apresentada ao Programa de Pós-Graduação em Sistemas de Comunicação e Automação – PPGSCA, como requisito para a obtenção do título de Mestre em Sistemas de Comunicação e Automação.

Orientador: Prof. D. Sc. José Patrocínio da Silva –
UFRN

**MOSSORÓ – RN
2014**

Dados Internacionais de Catalogação na Publicação (CIP)
Biblioteca Central Orlando Teixeira (BCOT)
Setor de Informação e Referência

B273a Barreto, Tarcisio da Silva

Análise de taxa média de bloqueio em conexões por algoritmos de caminhos mínimos: algoritmo de Yen e algoritmo genético/ Tarcisio da Silva Barreto -- Mossoró, 2015.
63f.: il.

Orientador: Prof. Dr. José Patrocínio da Silva

Dissertação (Mestrado em Sistemas de Comunicação e Automação) – Universidade Federal Rural do Semi-Árido. Pró-Reitoria de Pesquisa e Pós-Graduação.

1. Algoritmo de Yen. 2. Algoritmo Genético. 3. Inteligência computacional. 4. Menor caminho. 5. Taxa média de bloqueio.
I. Título.

RN/UFERSA/BCOT/028-15

CDD: 005.1

Bibliotecária: Vanessa Christiane Alves de Souza Borba
CRB-15/452


TARCISIO DA SILVA BARRETO

**ANÁLISE DE TAXA MÉDIA DE BLOQUEIO EM
CONEXÕES POR ALGORITMOS DE CAMINHOS
MÍNIMOS: ALGORITMO DE YEN E ALGORITMO
GENÉTICO**

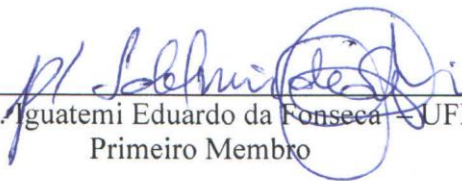
Dissertação de mestrado acadêmico apresentada ao Programa de Pós-Graduação em Sistemas de Comunicação e Automação – PPGSCA, como requisito para a obtenção do título de Mestre em Sistemas de Comunicação e Automação.

APROVADA EM: 15/12/2014.

BANCA EXAMINADORA



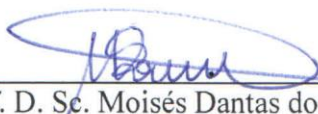
Prof. D. Sc. José Patrocínio da Silva – UFRN
Presidente



Prof. D. Sc. Igatemi Eduardo da Fonseca – UFERSA
Primeiro Membro



Prof. D. Sc. Elmer Rolando Llanos Villarreal – UFERSA
Segundo Membro



Prof. D. Sc. Moisés Dantas dos Santos – UFPB
Examinador Externo

Mossoró, 15 de dezembro de 2014.

DEDICATÓRIA

Aos meus pais Erivan Barreto e Daura, e
aos meus irmãos que tanto amo.

AGRADECIMENTOS

À Deus, a quem devo todo o meu louvor e minha vida, Ele que sempre me deu forças nos momentos mais difíceis, sempre me guiou, iluminou e me ajudou a vencer os desafios nessa nova fase da vida acadêmica.

Aos meus pais Erivan Barreto e Dona Daura, e a meus irmãos Vanda, Cleverlando, Cláudia, Vandeildo, Daurivan e Daurili, por estarem sempre de meu lado e torcendo por mim.

Ao meu orientador, Prof. D. Sc. José Patrocínio da Silva, pela disponibilidade sempre que solicitado, pela credibilidade depositada ao trabalho, pela paciência e, principalmente, pelos ensinamentos necessários à conclusão do mesmo.

Ao Prof. D. Sc. Idalmir de Souza Queiroz Júnior pela força, conselhos e ânimo que tem me dado durante esse tempo de mestrado.

À Maria de Fátima (Fatinha), pessoa maravilhosa que entrou em minha vida, que está sempre ao meu lado e que busca sempre me ajudar de toda e qualquer forma.

Aos “irmãos” que encontrei aqui no mestrado, em especial Ádller, Emmanuel e Jonathan.

À Vanessa Diniz, por ter incentivado à minha entrada no mestrado visando naquele momento o meu crescimento pessoal.

Ao casal José Diniz e Lenilda Diniz por ter me recebido de braços abertos e ter me tratado como um filho, me ajudando a vencer todas as barreiras encontradas nesse duro caminho.

Aos amigos e colegas do GEDEA, por meio dos quais me foi mostrado que através da união conseguimos alcançar todos os objetivos.

Ao amigo Carlos Evandro (UFRN) pelos ensinamentos e disponibilidade.

Aos demais professores e, em especial, à secretária Lívia do PPGSCA, os quais sempre se mostraram atenciosos e disponíveis e ajudar os mestrandos em suas dificuldades.

Ao apoio estrutural da UFERSA e UFRN, e suporte financeiro da CAPES e FAPERN.

EPÍGRAFE

“Por isso não tema, pois estou com você;
Não tenha medo, pois sou o seu Deus.
Eu o fortalecerei e o ajudarei; Eu o segurarei
com a minha mão direita vitoriosa.”
(Isaías 41:10)

RESUMO

Os estudos sobre bloqueio de conexões em redes de computadores vêm ganhando destaque em recentes pesquisas voltadas à comunicação computacional e tecnologia. Vários pesquisadores têm utilizado diversos métodos buscando identificar e minimizar ao máximo a taxa média de bloqueio que impedem que uma conexão seja estabelecida. Este trabalho apresenta uma análise de taxa média de bloqueio em conexões por algoritmos de caminhos mínimos. Têm sobre o desempenho de uma rede ótica transparente. Serão utilizados dois algoritmos para realizar a análise e as simulações, o Algoritmo Genético (AG) e o Algoritmo de Yen (AY). O Algoritmo Genético fundamentado por Inteligência Computacional (IC) e o Algoritmo de Yen baseado no princípio de encontrar e identificar os K menores caminhos. Simulações numéricas realizadas em diferentes cenários da rede mostram que, quanto maior o número de conexões, maior será a taxa média de bloqueio nas conexões. Através desse estudo será possível identificar qual algoritmo se comporta melhor para os casos específicos descritos nesse trabalho.

Palavras-Chave: Algoritmo Genético, Algoritmo de Yen, Inteligência Computacional, Menor caminho e Taxa média de bloqueio.

ABSTRACT

Studies on connections lock in computer networks have been gaining prominence in recent research focused on computational communication and technology. Several researchers have used various methods in order to identify and minimize the blocking rate that prevent a connection is established. This paper presents a blocking rate analysis in connections of shortest paths algorithms. They have on the performance of a transparent optical network. Two algorithms will be used to perform the analysis and simulations, the Genetic Algorithm (AG) and the algorithm Yen (AY). The Genetic Algorithm is based on Computational Intelligence (CI) and the Yen algorithm is based on the principle of finding and identifying the K shortest paths. Numerical simulations performed on different network scenarios show that the greater the number of connections, the higher the blocking rate in the connections. This study will help to identify which algorithm behaves better in the specific cases described in this work.

Keywords: Genetic Algorithm, Yen Algorithm, Computational Intelligence, Shortest Path, average block.

SUMÁRIO

LISTA DE TABELAS	xii
LISTA DE FIGURAS	xiii
LISTA DE SIGLAS, SÍMBOLOS E ABREVIATURAS	xiv
CAPÍTULO 1 - INTRODUÇÃO	17
1.1 MOTIVAÇÃO	18
1.2 OBJETIVOS	19
1.3 METODOLOGIA	20
1.4 ESTRUTURA DO TRABALHO.....	20
CAPÍTULO 2 - REVISÃO BIBLIOGRÁFICA	22
2.1 REDES ÓPTICAS	22
2.1.1 RWA Tradicional	23
2.1.2 <i>First-fit</i>	25
2.2 PROBLEMA DE MENOR CAMINHO	27
2.3 ALGORITMO DE DIJKSTRA.....	28
2.4 ALGORITMO DE YEN	32
2.5 INTELIGÊNCIA COMPUTACIONAL	35
2.5.1 Classificação da Inteligência Computacional	36
2.5.2 Algoritmos Genéticos	37
2.5.2.1. População inicial	40
2.5.2.2. Avaliação	41
2.5.2.3. Seleção	41
2.5.2.4. Operadores de cruzamento	42
2.6 TRABALHOS RELACIONADOS.....	42
CAPÍTULO 3 - CARACTERIZAÇÃO DO PROBLEMA E ANÁLISE PROPOSTA	44
3.1 CARACTERIZAÇÃO DO PROBLEMA.....	44

3.1.2 Análise proposta.....	45
3.1.3 Topologia da rede e gerador de conexões	46
3.2 SIMULAÇÕES REALIZADAS	47
CAPÍTULO 4 - RESULTADOS E DISCUSSÕES	49
CAPÍTULO 5 - CONCLUSÕES E TRABALHOS FUTUROS	54
REFERÊNCIAS BIBLIOGRÁFICAS	55
ANEXO A – PSEUDOCÓDIGOS.....	60
A1. GERADOR DE CONEXÕES ALEATÓRIAS.....	60
A2. ALGORITMO DE DIJKSTRA.....	61
A3. GENERALIZAÇÃO DO ALGORITMO DE YEN.....	62
A4. MENOR CAMINHO COM ALGORITMO GENÉTICO	62

LISTA DE TABELAS

Tabela 1 – Tabela de Iterações do Algoritmo de Dijkstra	31
Tabela 2 – Analogia do vocabulário do AG	39
Tabela 3 – Taxa média de bloqueio – AY	50
Tabela 4 – Taxa média de bloqueio – AG	51
Tabela 5 – Comparativo da Taxa média de bloqueio – AY e AG.....	52

LISTA DE FIGURAS

Figura 1 – Fluxograma do algoritmo RWA tradicional.....	25
Figura 2 – Execução do <i>First-Fit</i>	26
Figura 3 – Grafo de Representação das Cidades	29
Figura 4 – Iterações do Algoritmo de Dijkstra.	30
Figura 5 – Inteligência computacional – Taxonomia	36
Figura 6 – Classificação dos métodos de otimização	38
Figura 7 – Esboço básico de execução dos algoritmos genéticos.....	40
Figura 8 – Ambiente de Simulações	45
Figura 9 – Topologia de Rede NFSNET-2	46
Figura 10 – Exemplo de simulação realizada.	48
Figura 11 – Taxa média de bloqueio – Algoritmo de Yen	49
Figura 12 – Taxa média de bloqueio – Algoritmo Genético	50
Figura 13 – Comparativo – Algoritmo de Yen e Algoritmo Genético	52

LISTA DE SIGLAS, SÍMBOLOS E ABREVIATURAS

AG	Algoritmo Genético
AGs	Algoritmos Genéticos
AY	Algoritmo de Yen
EE	Estratégias Evolutivas
FF	<i>First-Fit</i>
GPS	Sistema de Posicionamento Global
IA	Inteligência Artificial
IC	Inteligência Computacional
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
ME	Matriz de Estados
N_{Simul}	Número de Simulações
P2P	Peer-to-peer (Ponto-a-Ponto)
PE	Programação Evolutiva
PPGSCA	Programa de Pós-Graduação em Sistemas de Comunicação e Automação
RWA	<i>Routing and Wavelength Assignment</i>
T_b	Taxa de Bloqueio
T_{mb}	Taxa Média de Bloqueio
TON	<i>Transparent Optical Networks</i> (Rede Óptica Transparente)
UFERSA	Universidade Federal Rural do Semi-Árido
UFRN	Universidade Federal do Rio Grande do Norte
WDM	<i>Wavelength Division Multiplexing</i> (Multiplexação por Divisão de Comprimento de Onda)
λ	Comprimento de Onda

CAPÍTULO 1 - INTRODUÇÃO

A demanda de novos usuários que utilizam as redes de comunicações cresce diariamente, seja para ingressar em redes sociais, efetuar chamadas telefônicas, receber informações de assuntos populares dentre outros. Essa demanda se dá em virtude ao crescimento acelerado de novos dispositivos que suportam esse tipo de serviço.

Devido à ascensão da internet, já é possível se comunicar praticamente com qualquer parte do mundo instantaneamente a baixíssimos custos, fazer negociações comerciais de forma segura, prática e rápida, e até mesmo prover uma relação mais próxima entre o governo e o cidadão. Na mesma proporção em que há a evolução da internet, há também o constante avanço em pesquisas nas áreas de telecomunicações e transferência de dados. Porém, juntamente com esses crescimentos, aumentam os congestionamentos de informações e a probabilidade de falhas de transmissão, sendo necessária a criação de novos métodos ou adaptação dos já existentes para reduzir a interferência nessas conexões.

As transmissões de sinais podem ser afetadas por vários motivos, dentre eles se têm as limitações físicas de fibras e componentes óticos, essas limitações são conhecidas como camadas de deficiências físicas. Dessa forma, é imprescindível a busca, implantação e evolução de novas tecnologias utilizadas em serviços que fazem requisição de imagem, voz e vídeo através da rede, e que as redes também evoluam e sejam capazes de dar suporte às conexões exigidas por cada aplicação. Em consequência dessas falhas, passou a ser estudado tanto o problema de bloqueios de conexões, como soluções baseadas em algoritmos de caminhos mínimos.

O algoritmo de Dijkstra é um dos algoritmos de busca mais utilizados para cálculo de caminho de custo mínimo entre vértices de um grafo. O Dijkstra soluciona o problema do caminho mínimo em grafos dirigidos ou não dirigidos com arcos ou arestas de peso não negativo.

Algoritmo desenvolvido para sempre encontrar um caminho mais curto a partir de um vértice origem do grafo, o Dijkstra foi utilizado como algoritmo base para outros algoritmos voltados para cálculos de caminhos mínimos. Mas, porém, esse algoritmo encontra apenas um caminho mais curto entre um nó origem e um destino. Por encontrar apenas um único caminho podem existir problemas como a taxa de bloqueio

em redes nas redes utilizadas, ou seja, caso a menor rota encontrada pelo Dijkstra já esteja sendo usada por outra conexão essa conexão será bloqueada. Devido a essa deficiência é indispensável à utilização de algoritmos que não encontrem apenas um caminho, mas um conjunto de possíveis caminhos ou rotas a serem percorridas em um grafo.

Pesquisas apontam que essa necessidade começou a ser suprida em meados dos anos 50 com o surgimento do primeiro algoritmo para determinar os K caminhos mais curtos. Mas, foi no início dos anos 70, que Yen propôs um algoritmo para determinar os K caminhos mais curtos entre um dado par de nós, algoritmo esse tido como o mais eficiente para este problema até os dias atuais.

Também em meados dos anos 70, Holland introduz uma técnica de busca de soluções de um determinado problema através de estratégia de sobrevivência do mais apto. Assumindo que os indivíduos mais aptos de qualquer população tendem a reproduzir e sobreviver à geração seguinte. Enfim, os algoritmos genéticos têm sido mostrados para resolver problemas lineares e não lineares, explorando todas as regiões do espaço do problema e conseqüentemente promovendo a descoberta de áreas promissoras através da mutação, cruzamento e demais operações de seleção possíveis de ser aplicadas aos indivíduos de uma população (MICHALEWICZ, 1996).

Trabalhos na literatura apontam a utilização dos algoritmos citados, Algoritmo de Yen (AY) e Algoritmo Genético (AG), como algoritmos utilizados em pesquisas voltadas para a solução do problema de caminhos mínimos. Desta forma com o conhecimento e disponibilidade dos algoritmos relacionados, um estudo comparativo entre o AY e o AG foi realizado. Este estudo investigou a taxa média de bloqueio (Tmb) de conexões em redes transparentes, visando identificar qual algoritmo teria uma menor taxa de bloqueio para os cenários simulados.

1.1 MOTIVAÇÃO

Dentro da área de comunicação e conexão de redes, existe um grande interesse tanto na análise e implementação de novas aplicações quanto na otimização das aplicações existentes, sempre buscando por respostas rápidas para solucionar os problemas pertinentes relacionados à comunicação de redes. Nesse contexto, é

importante salientar que este trabalho caracteriza-se, por sua aplicabilidade na análise e comparação dos algoritmos utilizados e das respostas através AY e do AG, com a possibilidade de identificar qual algoritmo teve um maior ganho com as simulações realizadas.

O processo de simulações de aplicações deste porte, tendo em vista o uso dessa importante área da comunicação de redes, representa uma etapa imprescindível para o desenvolvimento de novas tecnologias na região, podendo resultar a médio e longo prazo em aplicações experimentais de alto nível para o meio industrial, além de servir como base para futuras pesquisas na área de tecnologia. Por fim, espera-se, com este trabalho, abrir portas ou pelo menos ideias para novas pesquisas dentro do estudo da área de comunicação, conexões de redes e suas aplicações no PPGSCA/UFERSA.

1.2 OBJETIVOS

O objetivo geral deste trabalho é a realizar uma análise comparativa com os algoritmos AY e AG em um modelo de rede transparente de padrão americano para conhecer qual dos algoritmos é o mais indicado para tal aplicação.

Dentro destas perspectivas, os objetivos específicos deste trabalho são:

- Pesquisar e entender as características de redes ópticas e suas estruturas
- Realizar levantamento bibliográfico do Algoritmo de Yen e do Algoritmo Genético;
- Adaptar ambos os algoritmos para calcular a taxa média de bloqueio de conexões.
- Realizar simulações com vários cenários do número de conexões.
- Descrever os resultados obtidos
- Destacar qual algoritmo é o mais indicado para tal aplicação.

1.3 METODOLOGIA

Para o alcance dos objetivos específicos estabelecidos para este trabalho, as seguintes atividades foram realizadas:

- Entendimento das teorias e funcionalidades das redes ópticas
- Familiarização com as metodologias utilizadas na construção do Algoritmo de Yen voltado ao problema de caminho mínimo.
- Adaptar o AY para que seja calculada a taxa média de bloqueio de conexões.
- Conhecer novas aplicações utilizadas com Algoritmo Genético, baseada nos problemas de caminhos mínimos.
- Utilizar um gerador de conexões aleatórias para gerar conexões concorrentes.
- Validar teoricamente os resultados obtidos a partir das simulações do AY e do AG como proposto neste trabalho.

1.4 ESTRUTURA DO TRABALHO

O trabalho proposto será organizado da seguinte forma.

No Capítulo 1 será feita uma introdução, motivação e metodologia Além disso, serão expostos os objetivos gerais e específicos levantados como metas a serem alcançadas ao término do mesmo.

No Capítulo 2 será realizado o levantamento bibliográfico e apresentação dos conceitos e ações importantes dos principais algoritmos utilizados, suas constituições e os principais modos de operação. Nesse capítulo serão apresentados alguns fundamentos teóricos e definições de rede ópticas, problema de caminho mínimo, algoritmo de Dijkstra, Algoritmo de Yen, Inteligência computacional, com ênfase na utilização de Algoritmos Genéticos e trabalhos relacionados. Tendo como objetivo adquirir conhecimento suficiente a cerca do problema em questão, além disso, visa identificar o melhor algoritmo para o problema a partir de sucessivas simulações. Além disso, serão expostos alguns trabalhos relacionados que foram utilizados como base teórica para o desenvolvimento deste documento.

No Capítulo 3 é apresentada a caracterização do problema de taxa média de bloqueio em conexões e a análise proposta que foi realizada para a obtenção dos resultados. Nesse Capítulo também será apresentado o ambiente de simulação utilizado, a topologia da rede e gerador de conexões, e a descrição das simulações realizadas.

No Capítulo 4 serão descritos os resultados das simulações realizadas durante o processo de análise, visando identificar qual dos algoritmos é o mais indicado para o problema em questão.

No Capítulo 5 serão apresentadas as conclusões de acordo com as simulações feitas no Capítulo 3 e resultados apresentados no Capítulo 4. Além disso, serão apresentadas algumas sugestões de trabalhos futuros com a finalidade de dar continuidade a essa vertente de pesquisa.

Nos Anexos serão descritos os pseudocódigos e aspectos fundamentais e relevantes utilizados nesse trabalho.

CAPÍTULO 2 - REVISÃO BIBLIOGRÁFICA

Este capítulo abordará conceitos fundamentais ao desenvolvimento do trabalho. Na Seção 2.1 será realizado um levantamento introdutório sobre redes ópticas e seus conceitos. A Seção 2.2 apresenta a descrição do problema de caminho mínimo. Na Seção 2.3 será apresentado o primeiro algoritmo para cálculo de caminho mínimo descrito na literatura, o algoritmo de Dijkstra. Na Seção 2.4 será descrito e apresentado o primeiro algoritmo para determinação e enumeração dos K menores caminhos, o algoritmo de Yen. Na Seção 2.5 é introduzida e classificada a Inteligência Computacional com ênfase na a estrutura de execução dos AGs. Finalmente, na Seção 2.6 é feita uma discussão dos trabalhos mais recentes relacionados a caminhos mínimos e taxa media de bloqueio de conexões.

2.1 REDES ÓPTICAS

As redes de comunicação de dados estão em plena expansão ao redor de todo o mundo para atender a crescente demanda por largura de banda das redes de acesso. Novos estudos voltados a aplicações para transmissão de vídeo em alta resolução, transferência de grandes arquivos em redes P2P e ambientes de vídeo conferência impulsionam a demanda por taxas de transmissão cada vez maiores nas redes de acesso.

Outro fator que contribui para a ciência de expansão da capacidade das redes de comunicação são as novas gerações de redes celulares em *smartphones*. Tecnologias 3G e 4G viabilizam uma gama de aplicativos para dispositivos móveis que consomem uma enorme banda. Para atender esta crescente demanda por largura de banda, as redes precisam ser ampliadas e modernizadas.

Segundo Murthy e Gurusamy (2002), as redes ópticas WDM (*Wavelength Division Multiplexing*) constitui a grande maioria das redes da atualidade. Nestas redes de transporte, os dados são transmitidos através de diversos comprimentos de onda em uma única fibra óptica, resultando em uma melhor utilização da sua enorme capacidade de transmissão. Para o estabelecimento de uma comunicação entre um par de nós de uma rede óptica WDM é indispensável construir um caminho óptico fim-a-fim. Cada

caminho óptico utiliza uma rota e um comprimento de onda. O problema da escolha de uma rota e um comprimento de onda adequado para cada caminho óptico é conhecido como RWA (*Routing and Wavelength Assignment*) (ZANG *et al.*, 2000). Os algoritmos de RWA têm por objetivo minimizar a probabilidade de bloqueio de novas conexões em uma rede óptica transparente.

De acordo com Soares *et al.* (2008), o estabelecimento de uma conexão em um caminho óptico se confirma mediante a reserva de um comprimento de onda disponível. Em uma fibra óptica, com vários comprimentos de onda, o comprimento de onda do nó de origem pode diferir do comprimento de onda do nó de destino, ou não, ao longo dos enlaces da rota em função da capacidade de conversão de comprimento de onda dos comutadores ópticos. Na ausência de conversores de comprimento de onda, a propriedade de continuidade de comprimento de onda precisa ser respeitada ao longo de todos os enlaces do caminho óptico, caso isso não aconteça a conexão será bloqueada.

Vários trabalhos encontrados na literatura abordam o problema RWA considerando a camada óptica um meio ideal de transmissão e, portanto, qualquer escolha de rota e comprimento de onda é considerada válida do ponto de vista de desempenho da camada óptica (ZANG *et al.*, 2000).

2.1.1 RWA Tradicional

Segundo Maranhão *et al* (2010), o roteamento e atribuição de comprimento de onda também conhecido como RWA, define que cada caminho ótico utilizado em uma rede deve ser formado por um comprimento de onda e uma rota adequada. Tem por objetivo evitar ao máximo que novas conexões sejam rejeitadas reduzindo a probabilidade que uma Rede Ótica Transparente (*Transparent Optical Networks*) receba interferências.

Segundo Drummond e Fonseca (2009), caso o conjunto de caminhos óticos estabelecidos na rede não seja adequado para comportar a rota de um novo fluxo, um novo caminho ótico deve ser estabelecido, passando a ser necessária a solução do problema de RWA. Assim, o problema de agregação de tráfego consiste na definição de algoritmos para definir as rotas e os comprimentos de onda para chamadas com diferentes níveis em uma rede ótica.

De acordo com Azodolmolky *et al.* (2009), além das restrições da camada física, geralmente há uma restrição de comprimento de onda imposta ao problema RWA em redes óticas. Essa restrição significa que uma determinada ligação deve ser composta de comprimentos de onda idênticos nos links utilizados pelo caminho ótico. Essa exigência pode afetar tanto o desempenho da rede, como a complexidade do algoritmo RWA no qual a instalação de um novo caminho ótico está condicionada à disponibilidade do mesmo comprimento de onda.

Santos *et al.* (2012) descrevem que para existir o estabelecimento de uma conexão entre dois nós de uma rede ótica, é necessário definir os caminhos óticos por onde o tráfego será encaminhado e alocar os recursos necessários para o estabelecimento desta conexão (SOARES *et al.* 2006), (CHU *et al.*, 2004). O processo realizado para estabelecimento desta conexão é definido como problema de roteamento e alocação de comprimentos de onda (RAMASWAMI *et al.* 2009), (MUKHERJEE, 1997).

Zang e Jue (2000) e Christodoulopoulos, Tomkos e Varvarigos (2009), relatam que a maioria dos algoritmos RWA propostos na literatura possuem uma camada física ideal onde a transmissão do sinal é livre de erro.

Segundo Chaves *et al.* (2008), o RWA é o algoritmo responsável por encontrar uma rota com comprimento de onda disponível para atender uma requisição de conexão. Esse determinado algoritmo consiste em selecionar uma rota e um comprimento de onda através dos nós de uma rede.

Analisando os trabalhos existentes na literatura que abordam o problema de RWA, observou-se que o método RWA é dividido em duas partes distintas, uma para o cálculo do caminho de custo mínimo e outra para buscar um comprimento de onda disponível em uma determinada rota. O algoritmo de Dijkstra é o algoritmo responsável por retornar o menor caminho/rota, e o *First-Fit* (FF), é encarregado de verificar se há comprimento de onda disponível para que seja realizada a conexão.

Para evitar que o FF receba apenas um único caminho, alguns trabalhos utilizam outros algoritmos que retornam não apenas um, mas um conjunto dos K menores caminhos. Com esse conjunto dos K menores caminhos, o algoritmo FF verifica se em alguma deles existe um comprimento de onda disponível presente em todas as arestas desse caminho ótico. A Figura 1 representa a sequência de passos utilizados no RWA tradicional, onde é exposta cada etapa realizada pelo processo antes de admitir ou

rejeitar uma conexão. De forma objetiva, mostra que uma conexão somente é admitida se existir um comprimento de onda disponível.

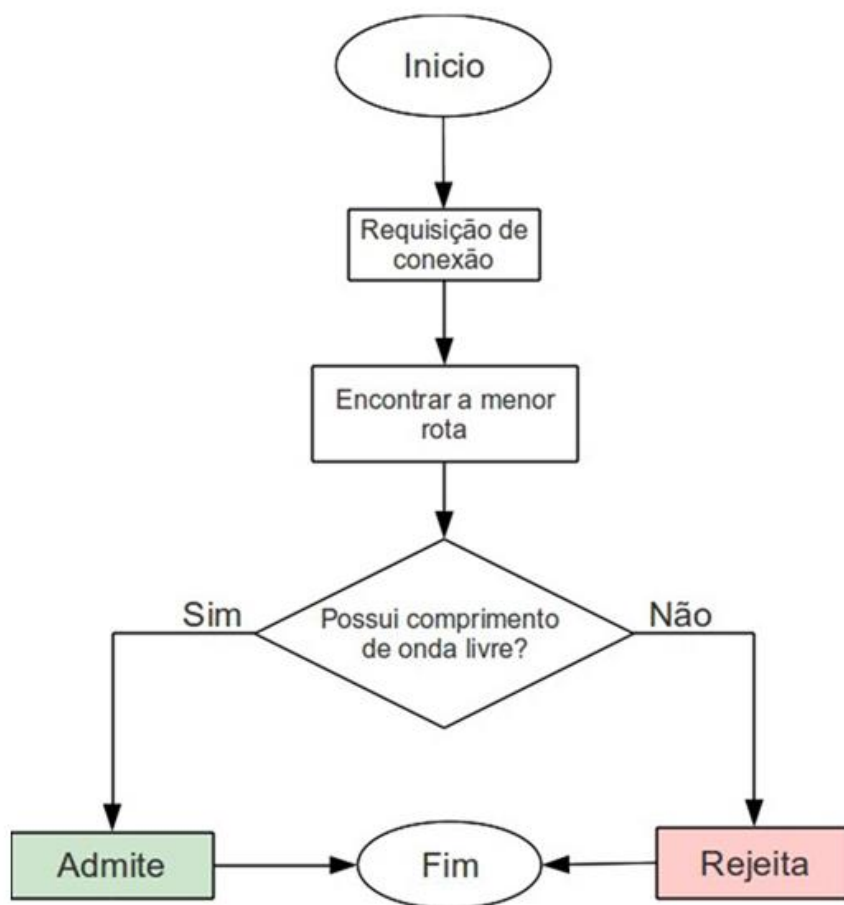


Figura 1 – Fluxograma do algoritmo RWA tradicional

Fonte: Oliveira (2009).

Segundo Oliveira (2012), esse procedimento passa por algumas etapas:

- Recebe uma requisição de conexão;
- Utiliza o algoritmo de Dijkstra para encontrar o menor caminho;
- Após o caminho encontrado, é utilizado o FF para encontrar um comprimento de onda disponível nessa rota.

2.1.2 *First-fit*

Algoritmo utilizado para identificação e seleção dos comprimentos de onda, o *First-Fit* (FF) recebe um determinado caminho, formado por um ponto inicial e um

ponto final, percorre todos os arcos identificando cada comprimento de onda que está disponível simultaneamente nos mesmos.

Após identificar os comprimentos de onda que estão disponíveis em todos os arcos do caminho solicitado, o algoritmo FF seleciona o comprimento de onda disponível com menor índice para satisfazer a conexão. Este algoritmo não necessita de informações completas dos enlaces da rede, precisa apenas conhecer o estado dos comprimentos de onda dos enlaces que formam o caminho da requisição a ser estabelecida (SOARES e GIOZZA, 2004).

Na Figura 2 mostra o exemplo de uma rota com seus devidos comprimentos de onda disponíveis na rede. Ao conhecer uma rota, o algoritmo percorre todo esse caminho identificando todos os comprimentos de ondas disponíveis que poderão ser utilizados nessa rota, seleciona os comprimentos de onda que satisfazem a conexão e dentre eles utiliza o de menor índice.

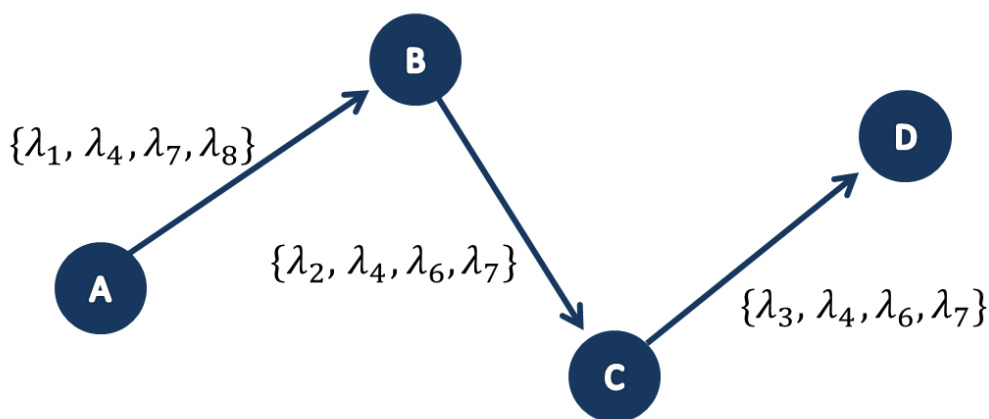


Figura 2 – Execução do *First-Fit*

Fonte: Autoria própria (2014).

Conforme mostra a Figura 2, foi adotada a rota A a D, então é feita a análise dos comprimentos de ondas disponíveis, nota-se que, em todos os enlaces foram obtidos os comprimentos de onda λ_4 e o λ_7 , assim o FF seleciona o λ_4 que possui o menor índice.

2.2 PROBLEMA DE MENOR CAMINHO

O problema de menores caminhos é um clássico de otimização em Redes que vem merecendo a atenção de muitos investigadores desde meados dos anos 50.

Estudos apontam que o surgimento do algoritmo mais antigo para determinar os K caminhos mais curtos, ainda nos anos 50, parece dever-se a Hoffman (1959). Anos mais tarde, foi proposto por Dreyfus (1969), um algoritmo para determinar ordenadamente não, apenas, os K caminhos entre um dado par de nós, mas os K caminhos de um dado nó para todos os restantes. Este algoritmo baseia-se no fato do k –ésimo caminho mais curto ser constituído por j –ésimos caminhos mais curtos, com $j \leq k$ e pode ser considerado como um melhoramento do algoritmo proposto por Hoffman (1959).

Mas nos anos 70, John Yen Professor da Universidade de Ciências e Tecnologia da Informação da Pensilvânia, apresenta o algoritmo proposto para encontrar os K caminhos (trajetos sem ciclos) mais curtos entre um dado par de nós em um grafo.

Segundo Perko (1986), desde os anos 70, o problema dos K menores caminhos tem sido abordados para se propor diferentes implementações do algoritmo de Yen, por exemplo, em um artigo de Katoh, Ibaraki e Mine (1982), onde se propõe um algoritmo para redes não orientadas, tentando-se tirar partido desta característica para obter uma maior eficiência.

Utilizando um grafo $G(V, A)$ para representar a rede em questão, podemos associar a cada aresta (i, j) de (V, A) um valor real c_{ij} , que designaremos por distância (ou custo) da aresta (i, j) , por analogia com a questão apresentada.

Como é natural, à distância (ou custo) de um trajeto entre duas estações será a soma das distâncias (ou custos) de todas as ligações entre estações intermédias desse trajeto. Assim, dados dois nós $u, v \in V$, consideremos a função denominada também função distância ou custo.

$$c : P_{uv} \rightarrow \mathbb{R} : p \rightarrow c(p) = \sum_{(i,j) \in p} C_{ij},$$

Para facilitar a notação da função, será descrita de forma simplificada $c(p) = \sum_{(i,j) \in p} C_{ij}$.

No problema de menor caminho, dados dois vértices (nós) u e v de (V,A) , é determinado um trajeto p^* de u para v , cuja distância (ou custo) seja mínima, isto é, tal que $c(p^*) \leq c(p)$, para todo o $p \in P_{uv}$.

De modo semelhante poderiam ser formulados problemas como o de encontrar o menor caminho, ou a mais rápida, de ir de um vértice para outro, se c_{ij} representar, respectivamente, o custo ou o tempo necessário para ir de cada vértice i a j utilizando a aresta (i, j) .

Este problema será utilizado nos problemas a seguir como no Algoritmo de Yen que encontra os K menores caminhos de um vértice para os demais de um grafo $G(V, A)$, pelo que adiante será apresentada a sua descrição mais minuciosa. Um estudo mais aprofundado pode ser conseguido recorrendo á vasta literatura existente e da qual podemos citar Bellman (1958), Dial *et al.* (1979), Dijkstra (1959), Gallo e Pallotino (1986), Moore (1959) e Pape (1974), dentre outros.

Sem perda de generalidade, quando nos referirmos a este problema iremos considerar c_{ij} como sendo um valor inteiro, apenas com o objetivo de tornar mais simples os programas elaborados.

Este trabalho inicia-se com uma breve abordagem ao problema de menor caminho, não só por ser um subproblema dos algoritmos que serão utilizados, mas também por ser um caso particular deste documento. Nesse trabalho serão utilizados dois algoritmos para analisar a probabilidade de bloqueio em redes óticas, o Algoritmo de Yen e o Algoritmo Genético.

O AY é baseado em outros algoritmos criados para encontrar a menor rota sendo necessário utilizar um algoritmo padrão para achar uma rota mais curta, como o algoritmo Dijkstra, por exemplo. Já o AG foi adaptado para o ser aplicado a esse problema, como pode ser visto o pseudocódigo na Seção A2 do Anexo A desse trabalho.

2.3 ALGORITMO DE DIJKSTRA

O algoritmo de Dijkstra é um dos algoritmos de busca mais utilizados para cálculo de caminho de custo mínimo entre vértices de um grafo. O mesmo também pode ser utilizado em Sistema de Posicionamento Global (GPS), traçando rotas de menor

caminho entre pontos ou cidades existentes em um mapa. O Dijkstra soluciona o problema do caminho mínimo em grafos dirigidos ou não dirigidos com arcos ou arestas de peso não negativo.

Segundo Gross (2006), esse algoritmo foi desenvolvido para encontrar um caminho mais curto a partir de um vértice (nó) origem do grafo, para cada um dos vértices do grafo, produzindo uma árvore de caminhos mais curtos. Porém ele pode ser utilizado para apenas encontrar um caminho mais curto entre um nó origem e um destino (para alocar rotas). Para a utilização do mesmo é necessário que os pesos dos enlaces não tenham valores negativos e que o grafo conectado não possua laços.

Dijkstra (1959) mostra que seu algoritmo soluciona problema de caminhos de custo mínimo em grafos. Analisando completamente todas as rotas e sempre calculando a menor entre elas, o algoritmo encontra um caminho ótimo de um vértice inicial para todos os outros vértices do grafo.

Segundo Atzingen *et al.* (2012), uma rede pode ser representada simbolicamente por um grafo $G(V, A)$, assumindo que V é o conjunto de vértices e A o conjunto de arestas os quais interligam estes vértices.

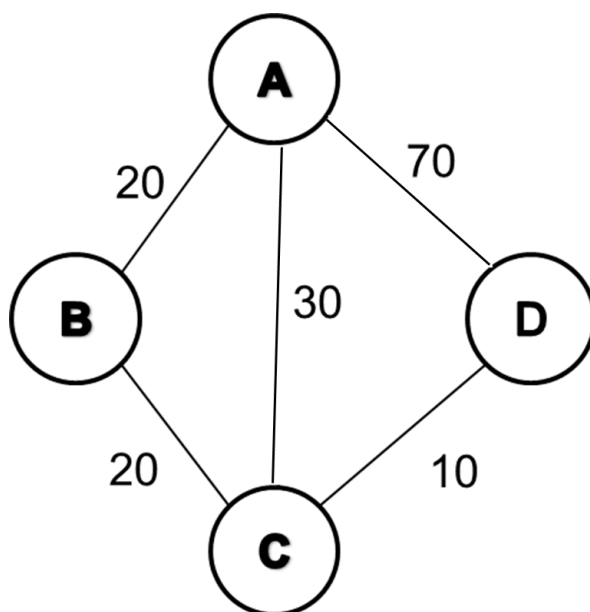


Figura 3 – Grafo de Representação das Cidades

Fonte: Autoria própria (2014).

Ao escolher o ponto inicial o algoritmo de Dijkstra calcula os menores caminhos levando em consideração um peso para cada conexão existente, esse peso assume valores não negativos como distância ou o tempo gasto. Então são calculadas todas as

distâncias entre todos os pontos presentes nessa rota. Na Figura 3 o grafo representa o conjunto de cidades $V = \{A, B, C, D\}$ com as respectivas estradas e distâncias que as ligam, nota-se que o grafo é não dirigido formado por um conjunto de vértices e arestas $G(V, A)$, onde os vértices são representados pelas cidades e as arestas representam as ligações/estradas existentes entre os pontos, cada uma com seu determinado peso/distância e não existindo um vértice com conexão com ele mesmo.

Após a análise do grafo, é obtida Matriz de Adjacência ($MAdj$) resultante das conexões e das distâncias entre as cidades presentes nessa rota.

$$MAdj = \begin{bmatrix} & A & B & C & D \\ A & 0 & 20 & 30 & 70 \\ B & 20 & 0 & 20 & 0 \\ C & 30 & 20 & 0 & 10 \\ D & 70 & 0 & 10 & 0 \end{bmatrix}$$

Para esclarecer esse exemplo foi escolhido o vértice A como raiz da busca e o vértice D o ponto final, mesmo que os vértices inicial e final tenham caminhos diretos, não quer dizer que terá a menor distância entre eles, tornando-se inicialmente inviável essa escolha de trajeto sendo necessário utilizar a dinâmica do Dijkstra para obter os caminhos mais curtos. A seguir na Figura 4 serão mostrados os caminhos utilizados pelo algoritmo e na Tabela 1 os resultados das iterações.

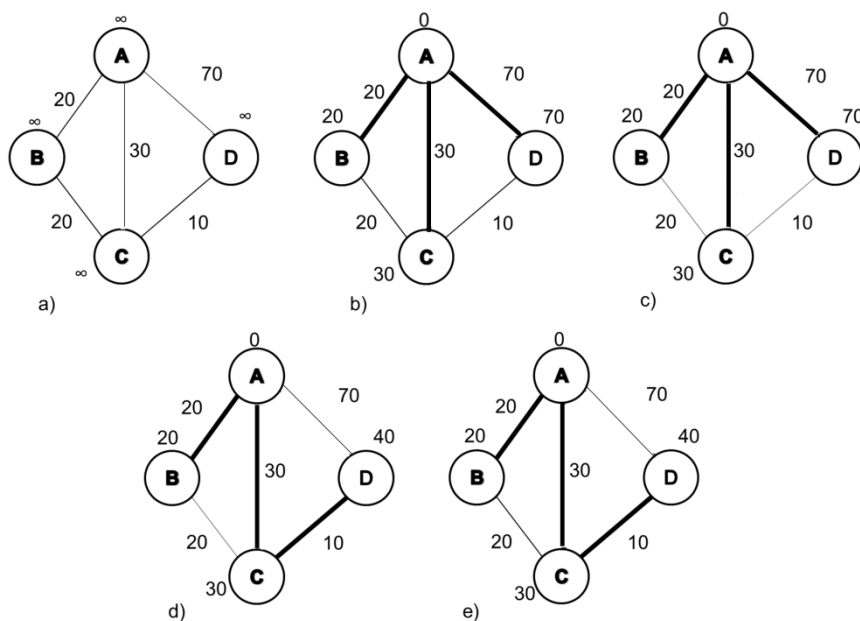


Figura 4 – Iterações do Algoritmo de Dijkstra.

Fonte: Autoria própria (2014).

Tabela 1 – Tabela de Iterações do Algoritmo de Dijkstra

Iteração	Vértice	Dist. [A]	Dist. [B]	Dist. [C]	Dist. [D]
1)	{ \emptyset }	∞	∞	∞	∞
2)	{A}	0	20	30	70
3)	{A, B}	0	20	30	70
4)	{A, B, C}	0	20	30	70
5)	{A, B, C, D}	0	20	30	40

Inicialmente todas as distâncias entre os vértices são preenchidas com valor infinito, assumindo assim um valor superior a qualquer outro encontrado em seu caminho, logo depois será atribuído o valor zero ao ponto inicial da busca. Então depois de estabelecida a origem da busca, o custo mínimo será calculado para todos os pontos presentes no grafo.

O algoritmo Dijkstra realiza as seguintes etapas:

1- Define um custo inicial de valor muito grande para todos os vértices do grafo na matriz de estados.

2- Define o nó origem s como nó que está atualmente sendo visitado (v) e marca como visitado na matriz de estados (ME), e armazena o valor zero como custo para o nó origem (não existe custo nem caminho entre o nó origem e ele mesmo).

3- Enquanto existir algum nó que ainda não foi definido como visitado e enquanto o nó destino não for marcado como visitado, o algoritmo realizará as seguintes etapas:

- Calcular novos custos dos k nós que têm ligação direta com o nó v . Utilizando as seguintes funções que acessam e manipulam valores da matriz de estado.
- Escolher um nó que tenha ligação com o nó v , com o menor custo e que ainda não tenha sido visitado, para ser o próximo nó v .
- Definir o novo nó v como visitado na matriz de estados.

Segundo Gomes (2009), o algoritmo de Dijkstra é um exemplo de algoritmo para cálculo de caminho mínimo, onde a rota entre dois pontos é definida como sendo o menor caminho entre eles, levando em consideração a distância ou peso.

O algoritmo de Dijkstra soluciona o problema de caminho mínimo, porém não garante que esse caminho seja o melhor, podendo haver congestionamento ou falha de equipamentos durante a transmissão (GOMES, 2009).

O Dijkstra possui uma limitação em seu algoritmo, o mesmo encontra somente o menor caminho em grafos com arestas de pesos não negativos. A limitação de encontrar apenas um caminho pode afetar o desempenho de certas aplicações que necessitam de alocação de caminhos o mais rápido possível.

Existem problemas como a taxa de bloqueio em redes de computadores, onde a menor rota para uma conexão já esteja bloqueada ou sendo usada por outra conexão. Ou seja, como o algoritmo de Dijkstra sempre encontra, apenas, o menor caminho de um ponto de origem para um ou mais pontos de destinos de grafos com arestas não negativas a solução pode não ser encontrada no menor caminho. Assim dá-se a necessidade da utilização de um algoritmo que não encontre apenas um caminho, mas um conjunto de possíveis caminhos ou rotas a serem percorridas em um grafo.

2.4 ALGORITMO DE YEN

O algoritmo de Yen foi criado para a obtenção dos K menores caminhos de forma ordenada de uma origem s para um destino t numa rede, que se baseia ainda na construção da árvore dos trajetos de s para t , ou de parte dela. No que se segue este algoritmo poderá ser designado abreviadamente por algoritmo AY.

No início da década de 70, Yen apresenta o algoritmo proposto para encontrar os K caminhos (trajetos sem ciclos) mais curtos entre um dado par de nós em um grafo, (YEN, 1971; YEN, 1975) algoritmo esse adotado como o mais eficiente para este problema até então. Nota-se que a determinação dos K caminhos mais curtos é um problema de maior complexidade, o que será devido ao fato de poder existir um k – *ésimo* caminho que não seja constituído apenas por j – *ésimos* caminhos mais curtos, com $j \leq k$.

Desenvolvido para encontrar os k menores caminhos em um grafo, o AY encontra as menores rotas entre um nó de origem e um nó de destino. Esse algoritmo pode ser utilizado em grafos direcionados ou não e que não tenha laços, contendo

somente pesos não negativos nas arestas (enlaces da rede), e possui uma complexidade $O(kn^3)$.

O algoritmo é baseado em outros algoritmos criados para encontrar a menor rota. É necessário utilizar um algoritmo padrão para achar uma rota mais curta, como o algoritmo Dijkstra, por exemplo. Logo após ser encontrado o menor caminho, as $k - 1$ menores rotas serão calculadas utilizando a abordagem do nó de desvio. Os nós de desvio podem ser todos os nós do caminho mais curto a ser percorrido, com exceção do último nó, sendo utilizados para provocar um desvio intencional na rota mais curta.

Segundo Fernandes (2010), obtido o caminho Dijkstra como a menor rota do caminho a ser percorrido, essa rota é adicionada a lista A. Para se encontrar as $k - 1$ menores rotas utilizando o algoritmo de Yen, as seguintes etapas devem ser seguidas:

1. Estabelecer o caminho Dijkstra como caminho a ser percorrido.
2. Enquanto não encontrar as $k - 1$ menores rotas, repetir as seguintes etapas:
 - I. O nó origem é especificado como nó de desvio (dv);
 - II. Remover o enlace entre o nó dv e o próximo nó no caminho Dijkstra, especificado na matriz de adjacências.
 - III. Utilizar o algoritmo Dijkstra entre os nós origem e destino.
 - IV. Adicionar o novo caminho mais curto encontrado em uma lista dos caminhos mais curtos encontrados.
 - V. O nó seguinte ao dv no caminho que está sendo percorrido, é utilizado como novo dv .
 - VI. Enquanto o nó dv não for o nó destino, realizar as seguintes etapas:
 - Armazenar a sub-rota entre os nós origem e dv em um vetor p_{sv} .
 - Remover o enlace entre o dv e o nó seguinte do caminho que está sendo percorrido.
 - Remover todos os enlaces da sub-rota do caminho que está sendo percorrido.
 - Utilizar o algoritmo Dijkstra entre o nó dv e o nó destino do caminho que está sendo percorrido.
 - Adicionar o novo caminho mais curto encontrado em uma lista dos caminhos mais curtos, após ser verificado se este novo caminho não é igual a nenhuma outro na lista.

- Adicionar os caminhos da lista dos caminhos mais curtos, na lista A, levando em consideração se é mesmo um caminho mais curto novo, e se ainda não foram adicionados k caminhos mais curtos à lista A.

Os caminhos encontrados são adicionados a uma lista de resultados (lista A), onde o caminho Dijkstra é o primeiro da lista. A condição de parada do algoritmo de Yen é quando não existir mais caminhos para serem percorridos ou quando terminar de percorrer um caminho, o número de k caminhos mais curtos encontrados for maior ou igual ao especificado, descartando os caminhos adicionais encontrados.

De um modo geral, pode-se dizer que a origem dessa ideia consiste em encontrar um determinado número de trajetos de um nó de origem para um nó de destino iniciando sempre pelo mais curto e, a partir de cada um dos encontrados, calcular outros. Cada um destes novos caminhos poderá ser um dos k menores caminhos, ou seja, são candidatos ao k –ésimo trajeto mais curto, com $k \in \{2, \dots, k\}$. O procedimento é repetido e com base no candidato de menor custo são calculados novos trajetos, candidatos ao trajeto mais curto seguinte.

Este algoritmo utiliza um conjunto, que designaremos por P , que contém os trajetos candidatos ao k –ésimo mais curto, com $1 \leq k \leq K$, ainda por analisar. De modo a obter a ordenação dos trajetos, em cada passo do algoritmo será retirado de P aquele que tiver custo mínimo. O trajeto retirado de P no k –ésimo passo é p_k , com $k \in \{1, \dots, K\}$.

O conjunto de trajetos P que vão sendo calculados e que não foram ainda analisados, enquanto que k indica o número de menores caminhos determinados até um dado momento, ou seja, o número de caminhos analisados.

De acordo com o raciocínio apresentado por Pascoal (1997), é possível obter uma generalização do Algoritmo de Yen, que enumera os K trajetos mais curtos entre dois nós de uma rede. Este raciocínio é descrito na generalização do algoritmo na Seção A4 de Anexos.

Uma vez que os trajetos são determinados de forma ordenada, o algoritmo irá parar a simulação assim que forem encontrados os K primeiros caminhos, ou seja, os K caminhos mais curtos. Se o número de caminhos de s para t em $G(V, A)$ for inferior a K o algoritmo termina assim que tiverem sido calculados todos os caminhos possíveis da rede.

2.5 INTELIGÊNCIA COMPUTACIONAL

Segundo Goldschmidt (2010), a Inteligência Computacional (IC) teve origem no ano de 1943, momento em que foi reconhecido o primeiro trabalho proposto por Warren McCulloch e Walter Pitts pertencente à área da Inteligência Artificial (IA), o qual propunha um simples modelo de neurônios artificiais de dois estados (ligado ou desligado); além de sugerirem que redes de neurônios artificiais seriam capazes de aprender. Este modelo ficou mundialmente conhecido como modelo de McCulloch e Pitts. Goldschmidt (2010) afirma ainda que no final da década de 80, Rumelhart e McClelland foram responsáveis pela publicação de vários trabalhos afins em uma coletânea denominada *Parallel Distributed Processing*. A partir de então, pesquisadores começaram a utilizar a expressão Inteligência Computacional como uma extensão à Inteligência Artificial.

Dessa forma, segundo Engelbrecht (2007), a definição de IA proposta pela IEEE Neural Networks em 1996 assegura que, trata-se do estudo de como fazer computadores tomar atitudes semelhantes às melhores que um ser humano poderia realizar em atividades específicas. Analogamente, Goldschmidt (2010) analisa as definições de IC e conclui ser uma ciência multidisciplinar que pesquisa o desenvolvimento e aplicação de técnicas computacionais que simulem o comportamento humano diante de várias situações.

Segundo Guimarães (2013), A partir da análise das definições, observa-se a grande semelhança entre os dois princípios e conclui-se que as diferenças são meramente práticas, tendo em vista que a pesquisa na IA gira em torno de técnicas que trabalham em nível simbólico (geralmente modelagem baseada em agentes utilizando lógica), enquanto que a IC gira em volta de técnicas e algoritmos subsimbólicos (através de redes neurais, algoritmos evolucionários, entre outros). Analogamente, a mente humana simbólica “executa” sobre uma estrutura subsimbólica, o cérebro.

Vale lembrar que, embora a comunidade tenha se seguido logicamente nestas duas linhas, as mesmas se complementam tecnicamente no que diz respeito a abordagens híbridas, tendência cada vez mais frequente em pesquisas do ramo acadêmico.

2.5.1 Classificação da Inteligência Computacional

Segundo Goldschmidt (2010), a IC apresenta divisões e subdivisões que abrange o estudo de técnicas e aplicações inspiradas na natureza. A Figura 5 representa uma taxonomia com a breve descrição dos principais paradigmas da Inteligência Computacional.

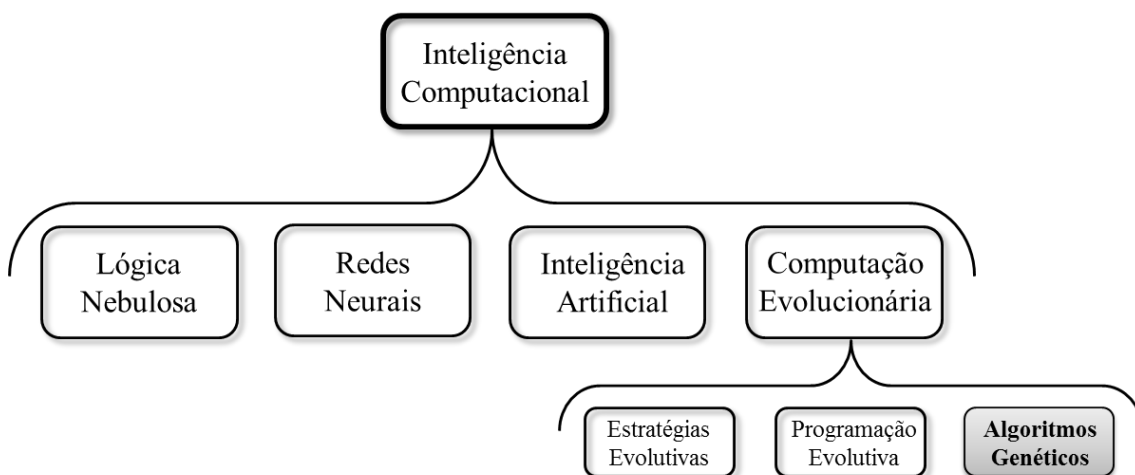


Figura 5 – Inteligência computacional – Taxonomia

Fonte: Adaptado de Goldschmidt (2010).

- Lógica Nebulosa (do inglês *Fuzzy Logic*):

Parte do princípio de modelagem aproximada do raciocínio humano estruturada em uma incerteza não estatística, ou seja, baseada em ambientes de subjetividade, imprecisão e/ou ambiguidade. Portanto, para ser implementada com eficácia, esta técnica exige do programador um conhecimento ou experiência prévia do comportamento do sistema em estudo.

- Redes Neurais:

Busca executar principalmente as tarefas de reconhecimento de padrões, percepção e controle motor, de forma semelhante às realizadas pelo cérebro. Em adição a estas características, o algoritmo baseado no sistema neural biológico, permite a capacidade de aprender e memorizar algumas ações; apresentando limitações tendo em vista o comportamento não linear e complexo do cérebro diante de situações específicas.

- **Inteligência Artificial:**

Retomando o conceito já esclarecido, esta linha de pesquisa restringe-se ao processamento simbólico do conhecimento, por meio da elaboração de programas que possam dotar computador de “inteligência”.

- **Computação Evolucionária:**

É uma técnica que abrange paradigmas baseados em conceitos biológicos da lei da evolução natural de Charles Darwin, em complemento à utilização do princípio da recombinação genética. Dentro do qual se observa o conceito de Algoritmos Genéticos (detalhado na Seção 2.5.2).

2.5.2 Algoritmos Genéticos

De acordo com Holland (1975), os Algoritmos Genéticos (AGs) são métodos computacionais inteligentes de busca e otimização, baseado em conceitos biológicos e na teoria evolutiva descrita por Charles Darwin; hipótese a qual justifica as mudanças na natureza e a geração de novas espécies como sendo advindas de pequenas variações herdadas pelos seres vivos juntamente com o processo de seleção natural.

Introduzidos por Holland (1975), esta técnica busca pelas melhores soluções de um determinado problema através da estratégia de sobrevivência do mais apto. Em geral, os indivíduos mais aptos de qualquer população tendem a reproduzir e sobreviver à geração seguinte, melhorando assim as sucessivas gerações. No entanto, os indivíduos inferiores podem, por acaso, sobreviver e também reproduzir. Enfim, os algoritmos genéticos têm sido mostrados para resolver problemas lineares e não lineares, explorando todas as regiões do espaço do problema e conseqüentemente promovendo a descoberta de áreas promissoras através da mutação, cruzamento e demais operações de seleção possíveis de ser aplicar aos indivíduos de uma população (MICHALEWICZ, 1996).

De acordo com a qualidade requerida de soluções, existem duas grandes famílias de algoritmos de busca para problemas de otimização probabilística: Métodos exatos e Métodos aproximados (TALBI, 2002).

Esta primeira grande família tem como objetivo encontrar a melhor solução para o problema em questão e provar a sua otimização. A segunda tem como objetivo encontrar uma boa qualidade ou uma solução quase ótima para o problema abordado em um tempo razoável, explorando uma parte selecionada do espaço de soluções em que as soluções de boa qualidade são esperados (MEHDI, 2011).

A Figura 6 apresenta como estão classificados esses algoritmos de otimização de acordo com Mehdi (2011). Os algoritmos genéticos pertencem ao grupo dos algoritmos meta-heurísticos baseados em população (em destaque), e, portanto tratam-se de métodos aproximados para solução de problemas.

É válido ressaltar que essa subjetividade, ou falta de precisão dos métodos heurísticos é tratada como uma particularidade comparada à inteligência humana, e não como uma deficiência ou ineficácia na solução do problemas.

De acordo com esta teoria, a combinação entre os genes dos indivíduos que sobrevivem pode produzir um novo indivíduo muito melhor adaptado às características de seu meio ambiente ao combinar características possivelmente positivas de cada um dos reprodutores (GABRIEL e DELBEM, 2008).

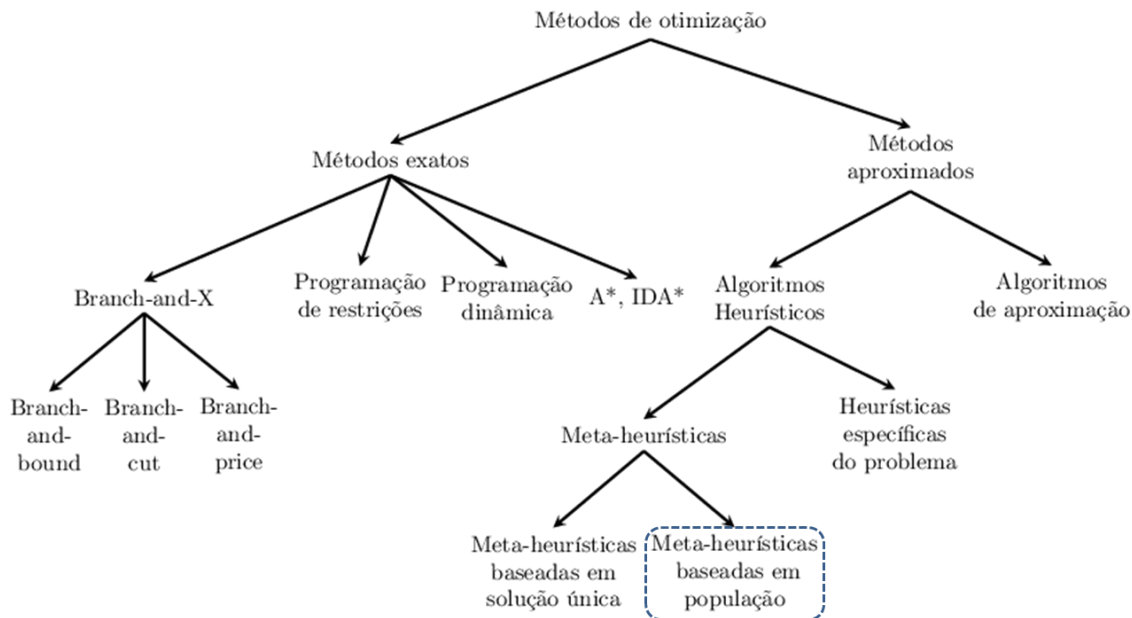


Figura 6 – Classificação dos métodos de otimização
Fonte: Adaptado de Talbi, 2002 *apud* Mehdi, 2011.

Por ser um método que espelha-se na Genética Natural de Charles Darwin, prevalecendo os indivíduos (soluções) mais adaptados ao ambiente (função *fitness*), o

AG herdou alguns termos da biologia. A Tabela 2 lista a analogia do vocabulário desses termos.

Tabela 2 – Analogia do vocabulário do AG

Genética Natural	Algoritmo Genético
População	Conjunto de soluções
Indivíduo	Solução de um problema
Cromossomo	Representação de uma solução
Gene	Parte da representação de uma solução
Ambiente	Função objetivo ou aptidão (<i>fitness</i>)
Cruzamento, Mutação	Operadores de busca

Fonte: Adaptado de Pacheco (1999).

Segundo Barcellos (2000), os AGs se encontram na classe dos Algoritmos Evolucionários juntamente com as Estratégias Evolutivas (EE) e a Programação Evolutiva (PE) conforme classificação de técnicas de busca a seguir:

- Técnicas Baseadas em Cálculo – As soluções de um problema de otimização devem atender a um conjunto de condições necessárias e suficientes.
 - Métodos Diretos – Usam informações da função como derivadas de primeira e/ou segunda ordem.
 - Métodos Indiretos – Buscam por um extremo local resolvendo um conjunto de Equações, comumente não lineares, quando se considera o gradiente da função objetivo nulo.
- Técnicas Enumerativas – Se utilizam da busca por uma solução procurando sequencialmente cada ponto do espaço de busca.
- Técnicas dirigidas por pesquisa aleatória – São baseadas em técnicas enumerativas, porém utilizam informações adicionais para proceder com a pesquisa.
 - *Simulated Annealing* – Baseia-se no processo de evolução da termodinâmica para proceder com a pesquisa para um estado de mínima energia.
 - Algoritmos Evolucionários – Baseados no princípio da seleção natural. (conforme citado esta classe se subdivide em: Estratégias Evolutivas e a Programação Evolutiva, Algoritmos de Genéticos).

Segundo Holland (1975), o algoritmo genético padrão proposto de forma geral, pode ser definido conforme a sequência das etapas destacadas na Figura 7.

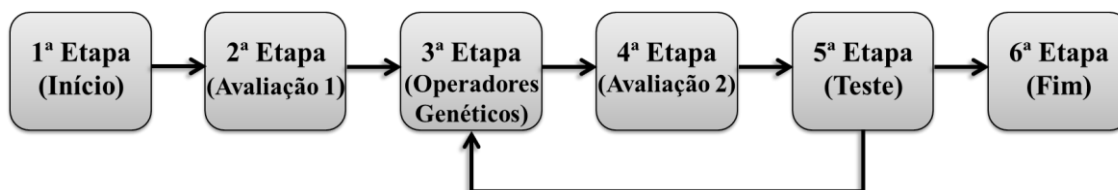


Figura 7 – Esboço básico de execução dos algoritmos genéticos

Fonte: Adaptado de Czarkowski (2002).

- **1ª Etapa (Início):** É gerada uma população aleatória de n cromossomos (soluções adequadas para o problema).
- **2ª Etapa (Avaliação):** A função *fitness* de cada indivíduo da população é avaliada.
- **3ª Etapa (Operadores Genéticos):** Os indivíduos são **selecionados** de acordo com a melhor função *fitness* (escolha dos pais) para o **cruzamento** entre os pais, objetivando formar a nova geração. Caso ocorra a **mutação**, cromossomas da nova geração nos *locus* (posição nos cromossomas) são alterados.
- **4ª Etapa (Avaliação):** A função *fitness* dos novos indivíduos da população é avaliada.
- **5ª Etapa (Teste):** Se a condição final desejada para o problema for atingida, o algoritmo exibe a melhor solução da população atual e finaliza na próxima etapa, se não, retorna à 3ª etapa.
- **6ª Etapa (Fim)**

2.5.2.1. População inicial

No AG, os indivíduos que representam as soluções para um problema específico podem ser representados por meio de cadeias binárias que consistem de zeros e uns, modelados segundo a maneira de cadeias de DNA de seres vivos orgânicos (BITTENCOURT, 2001).

A inicialização da população no AG consiste na criação dos primeiros indivíduos para o início da execução do algoritmo. Normalmente a geração desses

indivíduos acontece de forma aleatória, mas também poderá ser guiada por uma heurística para melhorar o número de indivíduos viáveis e/ou de boa aptidão. Cada heurística aplicada depende diretamente do problema em questão e poderá ser bastante útil para gerar um número alto de indivíduos que representam boas soluções. Assim, a criação de uma boa população significa dizer que o algoritmo tenderá a convergir para soluções eficazes, caso contrário, a população inicial poderá comprometer a convergência do AG.

2.5.2.2. Avaliação

A etapa de avaliação torna-se um ponto chave do algoritmo, pois através dela é que o cromossomo é pontuado. A função necessita representar muito bem o problema fazendo com que boas soluções possam levar o algoritmo a convergência para o ótimo resultado.

Segundo Sivanandam e Deepa, (2010), a avaliação é realizada através de uma função que melhor represente o problema e tem como objetivo fornecer uma medida de aptidão de cada indivíduo da população atual, que vai dirigir o processo de busca.

A função de aptidão (do inglês *fitness*) é o parâmetro que indica a qualidade de um cromossomo, o qual permite a obtenção de melhores repostas para o sistema. A função de aptidão necessita ser minuciosamente elaborada de forma a representar fielmente as condições do problema, objetivando soluções que possam levar o AG para o resultado ótimo.

2.5.2.3. Seleção

De acordo com Dantas (2013), o processo de seleção do algoritmo genético seleciona indivíduos para reprodução de acordo com a aptidão. Indivíduos mais aptos (com maior *fitness*) têm maior probabilidade de serem selecionados para a reprodução.

Segundo Pereira (2012), nesse processo os indivíduos com melhor aptidão, ou seja, os melhores indivíduos possuem maior probabilidade de serem escolhidos para a reprodução. Em vários trabalhos, são descritos cinco principais mecanismos de seleção: proporcional, por torneios, com truncamento por normalização linear e por

normalização exponencial (Pacheco, 1999). Um mecanismo de seleção possui a característica de intensificar a variação na aptidão média da população, permitindo que a população evolua sem perder a diversidade do material genético o que poderia levar o algoritmo a um mínimo local.

2.5.2.4. Operadores de cruzamento

O cruzamento é responsável por fornecer uma nova combinação para os indivíduos da população e é utilizado após a fase de seleção.

Os indivíduos selecionados serão recombinados por meio de um valor de probabilidade. A maneira como o material genético dos pais será combinado para formar um novo descendente é particular de cada problema.

De maneira geral essa combinação acontece pelo corte dos cromossomos pais em um ou mais pontos. A formação do indivíduo é, então, realizada pela mescla das partes dos cromossomos dos genitores provenientes do corte. O operador de mutação é aplicado também mediante uma probabilidade. No geral o valor de probabilidade é baixo para evitar uma grande diversificação da população.

2.6 TRABALHOS RELACIONADOS

Esta seção discute os principais trabalhos diretamente relacionados com aplicações de técnicas ao problema de busca de caminhos mínimos em grafos e bloqueio de conexões.

O recente artigo desenvolvido por Fernandes (2010), sob o qual o presente trabalho se encontra baseado, utiliza o Algoritmo de Yen para encontrar/alocar os menores caminhos e comprimentos de onda em redes limitadas. O trabalho de Fernandes (2010) consiste na técnica de encontrar as K menores rotas/caminhos e realizar a alocação de comprimento de onda em redes óticas. O trabalho em questão busca sempre encontrar uma rota disponível para estabelecer a conexão, porém, quando a conexão não é concretizada essa conexão é contabilizada para o cálculo da taxa média de conexões bloqueadas.

A dissertação de mestrado de Pascoal (1997) da Universidade de Coimbra faz uma abordagem sobre alguns algoritmos para cálculo de caminhos mínimos em redes óticas. O trabalho intitulado de Algoritmos para a Enumeração do K trajectos mais curtos, descreve algoritmos voltados praticamente para o cálculo de caminhos mínimos.

Tomando por base, esse trabalho também está relacionado com a dissertação de mestrado de Pereira (2012) que elaborou seu documento relacionado a planejamento de caminhos com algoritmo genético. O mesmo é utilizado para determinar o menor caminho entre os pontos de origem e destino de um grafo.

Com base nas publicações discutidas, será abordada no Capítulo 3 a caracterização do problema e a análise proposta deste trabalho.

CAPÍTULO 3 - CARACTERIZAÇÃO DO PROBLEMA E ANÁLISE PROPOSTA

Na área de telecomunicações, existem sempre a necessidade de conexão, comunicação e transferência de dados em taxas de transferências cada vez maiores, levando ao uso de técnicas para maximizar o uso das fibras óticas. O problema da taxa de bloqueio de conexão é baseado no fundamento de que quanto mais congestionada uma rede, mais conexões são bloqueadas, passando a não ser possível a conexão de dados. Com isso dar-se a necessidade de utilizar algumas métricas para evitar o bloqueio dessas conexões. Essas métricas podem ser desde a identificação de um mecanismo que minimizem a taxa de bloqueio como a utilização de novas estratégias ou algoritmos para que o problema seja solucionado.

3.1 CARACTERIZAÇÃO DO PROBLEMA

O problema consiste na utilização de dois algoritmos, o Algoritmo de Yen e o Algoritmo Genético, para simular e analisar um tráfego em uma rede ótica com conexões aleatórias e concorrentes, ou seja, uma conexão pode ser solicitada antes mesmo que a conexão anterior seja finalizada. Para que essas conexões aleatórias fossem simuladas foi necessário à implementação de um gerador de conexões para ambos os algoritmos. Para que esse gerador de conexões seja executado, é necessário que o mesmo conheça algumas informações da rede em que irá criar as rotas, recebendo o total de vértices da rede, o número de conexões que serão realizadas e a média de tempo de cada conexão.

Esse gerador irá retornar vetores de origens, destinos, durações e inícios de cada conexão. Os vetores de origem e destino serão preenchidos randomicamente recebendo valores aleatórios, essas informações serão utilizadas no AY e no AG para obtenção da lista de menores caminhos.

Se uma conexão for solicitada e não for realizada, essa conexão será contabilizada no cálculo da taxa de média de bloqueio de conexão.

3.1.2 Análise proposta

Nesta seção serão apresentados os cenários da análise proposta e serão descritas as simulações que foram realizadas nos algoritmos AY e AG. As simulações foram implementadas na linguagem C++ no software Qt Creator 3.0.0 como mostra a Figura 8, em um processador Intel® Core™ i5-3337U (1.80GHz) e 4GB de memória RAM.

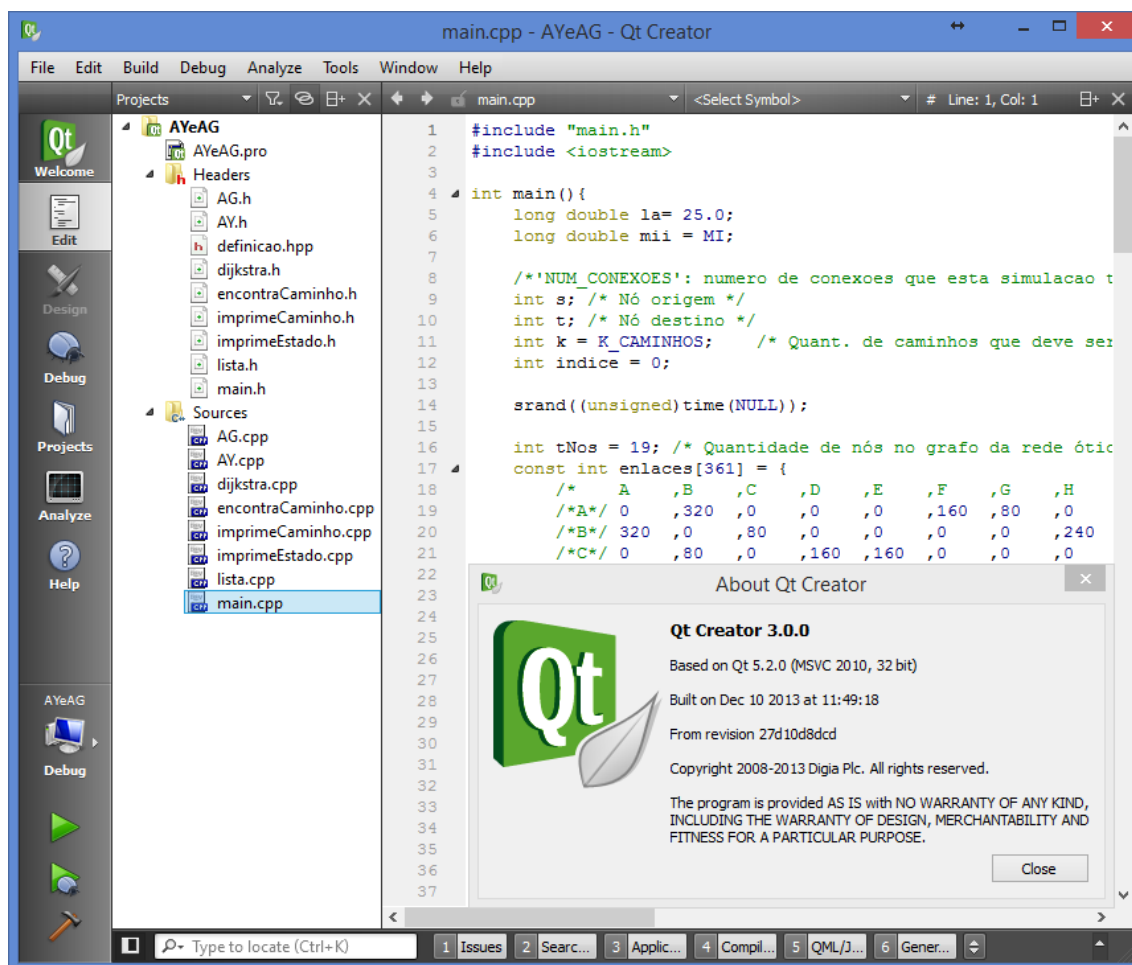


Figura 8 – Ambiente de Simulações
Fonte: Autoria própria (2014).

Buscando identificar um algoritmo que tenha uma menor taxa de bloqueio de conexões, torna-se necessário que os dois algoritmos AY e AG sejam analisados e comparados, essa análise será imprescindível para saber qual dos algoritmos comporta-se melhor para o caso específico estudado nesse trabalho.

3.1.3 Topologia da rede e gerador de conexões

As simulações foram realizadas a partir dos algoritmos AY e AG, com base na aplicação de caminhos mínimos e foram simulados na topologia de rede NSFNET-2 com 19 nós e 28 arestas bidirecionais (modelo real da rede ótica americana), rede ótica transparente e possui todos os enlaces bidirecionais como mostra a Figura 9.

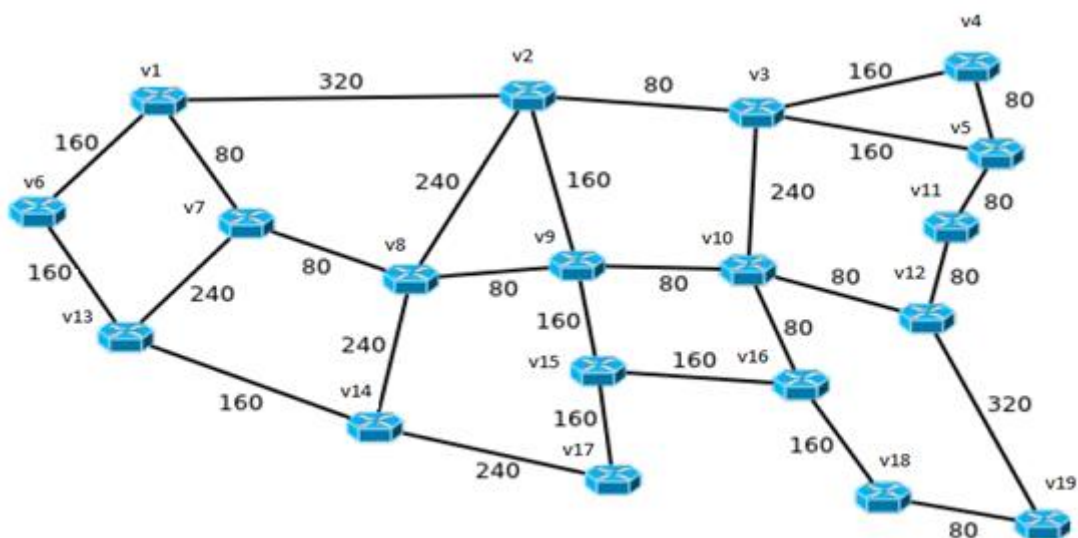


Figura 9 – Topologia de Rede NFSNET-2
Fonte: Adaptada de Oliveira (2012).

Em ambos os algoritmos (AG e AY) foram utilizados um gerador de conexões aleatórias. Com a utilização do gerador de requisições foi simulado um cenário dinâmico, no qual foram geradas 100, 1.000, 10.000, 100.000 e 1.000.000 de requisições aleatórias que possuem um padrão de tráfego uniforme entre os nós da rede, isto é, possuem a mesma probabilidade de serem selecionados, e seguem uma distribuição Poissoniana de momento de chegada da conexão, tendo duração com distribuição exponencial (média = 1s), como pode ser visto no pseudocódigo do gerador de requisição de conexões aleatórias na Seção A1 no Anexo A.

3.2 SIMULAÇÕES REALIZADAS

As simulações foram realizadas em ambos os algoritmos, respeitando o número de conexões, parâmetros de entrada de dados como origem, destino e média de conexão. Após a obtenção dos parâmetros aplicados no AY e no AG os dois algoritmos realizam a busca dos menores caminhos para realizar a conexão. Caso a conexão seja estabelecida, essa conexão ficará ativa por um tempo médio que é implementado e calculado no gerador de conexões.

Cada requisição que consegue iniciar e terminar a conexão recebe o status de Finalizado e cada requisição que não consegue realizar a conexão recebe o status de Bloqueado. Na Figura 10 é exibida uma simulação realizada com 1000 conexões. Nela também são exibidas as informações do nó de origem, o nó de destino, o caminho mínimo encontrado e percorrido e, a probabilidade de bloqueio, a quem chamamos nesse trabalho de taxa de bloqueio.

Foram realizadas diversas etapas para cada intervalo de conexões. Devido ao grande esforço computacional, para os eventos que utilizaram 100, 1.000 e 10.000 conexões aleatórias, foram realizadas 100 simulações. Para os eventos que utilizaram 100.000 conexões aleatórias, foram realizadas 50 simulações. E para os eventos que utilizaram 1.000.000 de conexões, apenas 10 simulações foram realizadas.

De acordo com os resultados adquiridos com as diversas simulações, foram calculadas as taxas de bloqueios de conexão para cada ponto do intervalo de conexões como podem ser vistos no Capítulo 4.

Logo após o cálculo de todas as taxas de bloqueio, é realizada a média de todas as conexões realizadas. Ou seja, foram somadas todas as taxas e divididos pelo total de conexões efetuadas

$$T_{mb} = \left(\frac{\sum T_b}{n_{simul}} \right)$$

Onde T_{mb} é a taxa média de bloqueio, T_b é a taxa de bloqueio de cada simulação e, n_{simul} é o número de simulações realizadas.

Vale salientar que a alta taxa de bloqueio (T_b) em cada simulação os algoritmos também sofrem influência de que em todas as simulações foram retornados três menores caminhos e apenas com um comprimento de onda por enlace. Ou seja, para cada conexão do vértice inicial s para o vértice final t foram retornados três caminhos

ópticos, se em algum desses caminhos ópticos possuírem pelo menos um comprimento de onda disponível que satisfaça a conexão a mesma poderá ser iniciada e finalizada, mas se não houver um comprimento de onda disponível para que seja realizada a conexão, a mesma será bloqueada.

```

Total de Conexoes: 1000
Total de bloqueios: 739
Prob. de bloqueios: 73.90%
conexao[1]: FINALIZADO s:v11 t:v8 [v11, v12, v10, v9, v8]
conexao[2]: FINALIZADO s:v9 t:v11 [v9, v2, v3, v5, v11]
conexao[3]: BLOQUEADO s:v2 t:v12 [v2, v9, v10, v3]
conexao[4]: BLOQUEADO s:v12 t:v8 [v12, v10, v9, v2]
conexao[5]: FINALIZADO s:v14 t:v18 [v14, v17, v15, v16, v18]
conexao[6]: BLOQUEADO s:v19 t:v16 [v19, v12, v10]
conexao[7]: BLOQUEADO s:v13 t:v15 [v13, v7, v8, v9, v10]
conexao[8]: BLOQUEADO s:v13 t:v9 [v13, v7, v8, v2]
conexao[9]: BLOQUEADO s:v7 t:v10 [v7, v8, v9, v15]
conexao[10]: FINALIZADO s:v18 t:v9 [v18, v16, v15, v9]
conexao[11]: BLOQUEADO s:v19 t:v3 [v19, v18, v16, v10, v9]
conexao[12]: BLOQUEADO s:v3 t:v6 [v3, v2, v1, v7]
conexao[13]: BLOQUEADO s:v4 t:v17 [v4, v3, v2, v9, v10, v16]
conexao[14]: BLOQUEADO s:v13 t:v3 [v13, v7, v8, v2, v9]
conexao[15]: BLOQUEADO s:v14 t:v16 [v14, v8, v9, v10, v12]
conexao[16]: BLOQUEADO s:v16 t:v9 [v16, v15, v17]
conexao[17]: FINALIZADO s:v8 t:v7 [v8, v7]
conexao[18]: BLOQUEADO s:v5 t:v18 [v5, v11, v12, v10, v9, v15]
conexao[19]: BLOQUEADO s:v3 t:v6 [v3, v2, v1, v7]
conexao[20]: BLOQUEADO s:v18 t:v4 [v18, v16, v10, v12, v11, v5, v3]
conexao[21]: BLOQUEADO s:v4 t:v6 [v4, v3, v2, v1, v7]
conexao[22]: BLOQUEADO s:v7 t:v18 [v7, v8, v9, v10, v16, v15]
conexao[23]: BLOQUEADO s:v4 t:v8 [v4, v3, v2, v9]
conexao[24]: BLOQUEADO s:v15 t:v3 [v15, v16, v10, v9]
conexao[25]: FINALIZADO s:v4 t:v16 [v4, v5, v11, v12, v10, v16]
conexao[26]: FINALIZADO s:v15 t:v13 [v15, v17, v14, v13]
conexao[27]: BLOQUEADO s:v10 t:v17 [v10, v9, v15, v16]
conexao[28]: BLOQUEADO s:v13 t:v5 [v13, v7, v8, v9, v10, v3, v5, v5]
conexao[29]: BLOQUEADO s:v7 t:v10 [v7, v8, v9, v15]
conexao[30]: BLOQUEADO s:v10 t:v19 [v10, v12, v11, v5]
conexao[31]: BLOQUEADO s:v6 t:v16 [v6, v1, v7, v8, v9, v10, v12]
conexao[32]: BLOQUEADO s:v15 t:v8 [v15, v16, v10]
conexao[33]: FINALIZADO s:v1 t:v14 [v1, v6, v13, v14]
conexao[34]: BLOQUEADO s:v13 t:v3 [v13, v7, v8, v2, v9]

```

Figura 10 – Exemplo de simulação realizada.

Fonte: Autoria própria, (2014).

Os resultados obtidos com as simulações realizadas nesse Capítulo serão apresentados no Capítulo 4, onde serão apresentadas e analisadas cada resposta fornecida pelos algoritmos utilizados.

CAPÍTULO 4 - RESULTADOS E DISCUSSÕES

Nesse tópico serão apresentados os resultados da análise dos testes realizados nas simulações aplicando o Algoritmo de Yen e o Algoritmo Genético. Para validar as técnicas utilizadas serão apresentadas as comparações com os algoritmos utilizados.

A partir da aplicação dos dois algoritmos baseados em caminhos mínimos, foram feitas as simulações descritas na Seção 3.4 do Capítulo 3. Buscando realizar um comparativo mais preciso e adquirir resultados confiáveis, as simulações foram rigorosamente realizadas com todos os padrões de rede e o mesmo total de conexões para os dois algoritmos, como pode ser visto nas Figuras 11 e 12, respectivamente.

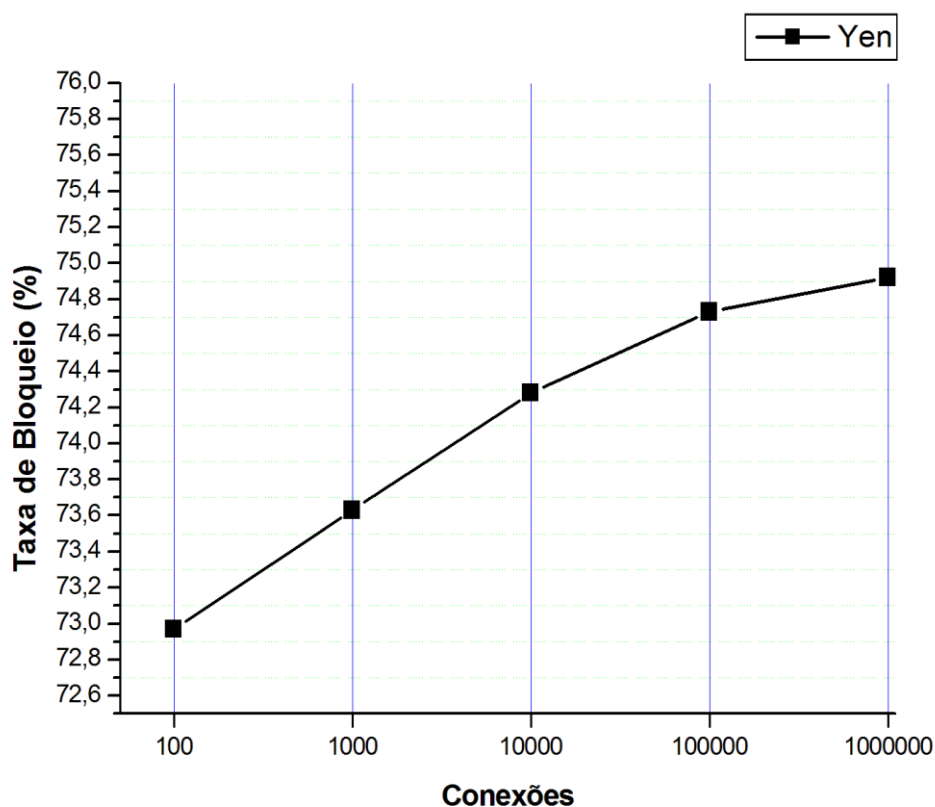


Figura 11 – Taxa média de bloqueio – Algoritmo de Yen

Fonte: Autoria própria, (2014).

De acordo com o gráfico da Figura 11, em simulações realizadas no Algoritmo de Yen, podemos destacar a taxa média de bloqueio (Tmb) de conexão para as determinadas requisições. Os resultados mostram que quanto maior o número de

conexões, maior também a Tmb , obtendo, assim, uma Tmb de 72,97% para 100 conexões e de 74,92% para 1.000.000 de conexões, conforme Tabela 3.

Tabela 3 – Taxa média de bloqueio – AY

Número de Conexões	Taxa média de bloqueio (%)
100	72,97
1000	73,63
10000	74,28
100000	74,73
1000000	74,92

Fonte: Autoria própria, (2014).

Os dados da Tabela 3 mostram a detalhadamente os resultados obtidos para o restante das conexões utilizadas por padrão.

Assim como na Figura 11 apresentou os resultados obtidos com o AY, a Figura 12 também apresenta os resultados obtidos nas simulações realizadas com o AG.

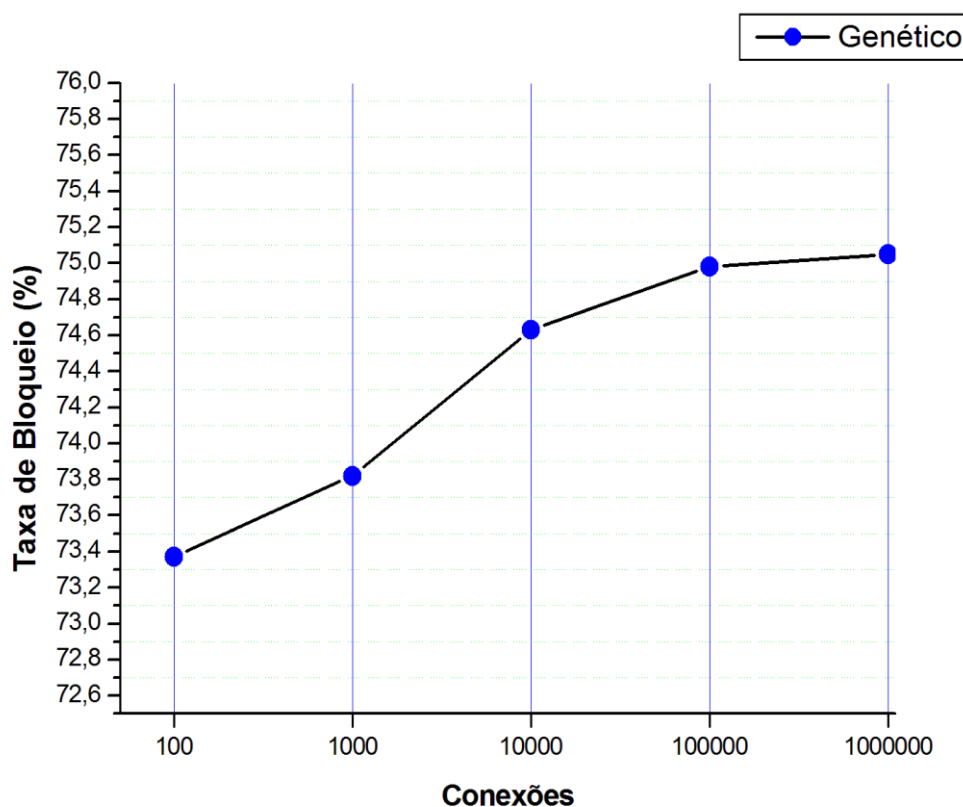


Figura 12 – Taxa média de bloqueio – Algoritmo Genético

Fonte: Autoria própria, 2014.

Do mesmo modo ocorrido no gráfico da Figura 11, o gráfico da Figura 12 em simulações realizadas no Algoritmo de Genético, apontam que a *Tmb* de conexão para as determinadas requisições crescem juntamente com o total de conexões realizadas. Resultando em uma *Tmb* de 73,37% para 100 conexões e de 75,05% para 1.000.000 de conexões, conforme Tabela.

Tabela 4 – Taxa média de bloqueio – AG

Número de Conexões	Taxa média de bloqueio (%)
100	73,37
1000	73,82
10000	74,63
100000	74,98
1000000	75,05

Fonte: Autoria própria, (2014).

Do mesmo modo que a Tabela 3 mostra detalhadamente os resultados obtidos para o gráfico da Figura 11, a Tabela 4 mostra, também, detalhadamente os resultados do gráfico da Figura 12, para o restante das conexões utilizadas por padrão.

Assim, após a análise dos gráficos e dos dados obtidos nas simulações, torna-se necessário realizar uma análise da taxa média de bloqueio visualizando o mesmo cenário. Na Figura 13 é apresentado o resultado comparativo de forma unificada.

A partir dos resultados obtidos nas simulações e demonstrados no gráfico da Figura 13, nota-se que o AY adapta-se melhor que o AG para esse determinado tipo de problema, como pode ser comprovado na Tabela 5.

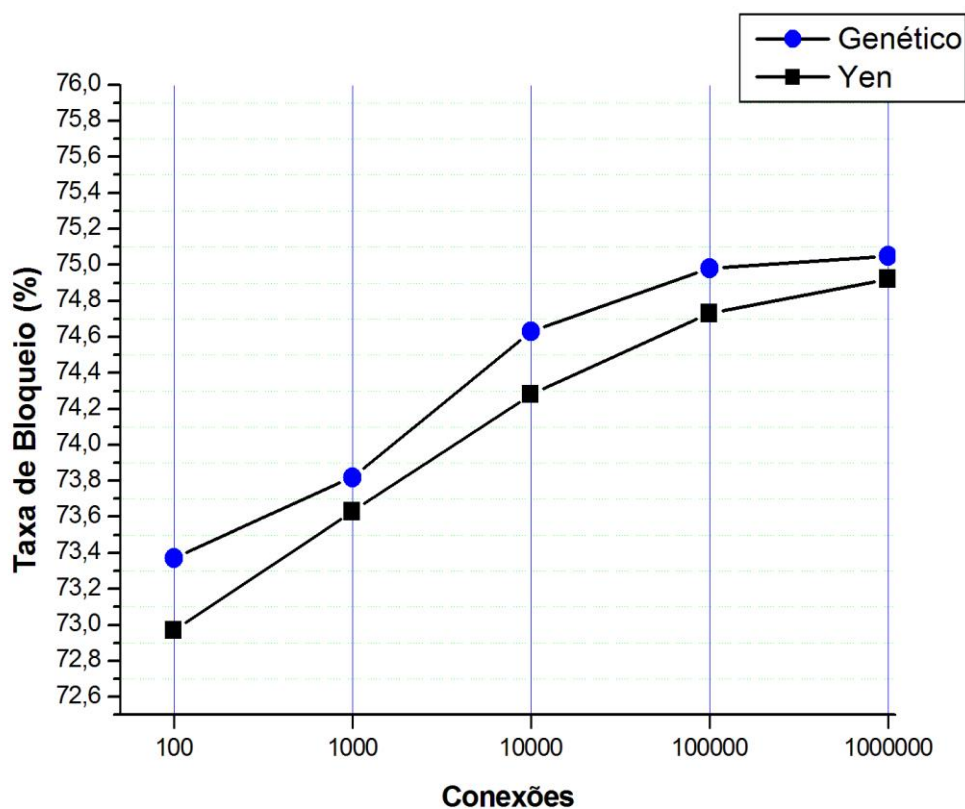


Figura 13 – Comparativo – Algoritmo de Yen e Algoritmo Genético

Fonte: Autoria própria, 2014.

Tabela 5 – Comparativo da Taxa média de bloqueio – AY e AG

Número de Conexões	Taxa média de bloqueio (%)	
	Algoritmo de Yen	Algoritmo Genético
100	72,97	73,37
1000	73,63	73,82
10000	74,28	74,63
100000	74,73	74,98
1000000	74,92	75,05

Fonte: Autoria própria, (2014).

É notável o crescimento da taxa média de bloqueio em ambos os algoritmos (AY e AG) quando é elevado o número de conexões.

Dentro dos parâmetros atualizados para a simulação e comparação de resultados e, dentro do intervalo do número de conexões, percebe-se que, mesmo mantendo-se

com a taxa média de bloqueio inferior em todos os pontos do intervalo, o AY teve um crescimento de 1,95% quando comparado com o AG que teve o crescimento de 1,68%.

CAPÍTULO 5 - CONCLUSÕES E TRABALHOS FUTUROS

Nesse trabalho foi realizada uma análise de taxa média de bloqueio em conexões por algoritmos de caminhos mínimos: Algoritmo Genético e Algoritmo de Yen, buscando uma forma de encontrar qual destes tem uma menor taxa média de bloqueio de conexões. Ambos os algoritmos utilizaram algoritmos de caminhos mínimos para encontrar as rotas e conseqüentemente realizar as conexões.

A principal contribuição desse trabalho foi realizar a análise comparativa entre os algoritmos utilizados. Varias simulações foram realizadas com o AY e o AG com o objetivo de identificar qual algoritmo seria mais aplicável ao problema de taxa média de bloqueio por conexões em uma rede ótica transparente.

De acordo com os resultados descritos no Capítulo 4, os dois algoritmos (AY e AG) tem um crescimento da taxa média de bloqueio em conformidade com o aumento do número de conexões realizadas. Ou seja, sempre que os algoritmos foram simulados com um maior número de conexões, também foi retornada uma maior taxa média de bloqueio, como pode ser visto nas Figuras 11 e 12 e comparado na Figura 13.

As simulações para o caso determinado da topologia de rede NSFNET-2 com 19 nós e 28 arestas bidirecionais (modelo real da rede ótica americana), rede ótica transparente que possui todos os enlaces bidirecionais, mostra que no intervalo de conexões utilizadas o AY teve uma taxa média de bloqueio inferior à taxa média de bloqueio do AG como descrito na Tabela 5. Porém, mesmo o AY resultando em uma taxa média de bloqueio inferior à taxa de bloqueio do AG, o algoritmo obteve um crescimento de 1,95% enquanto, por sua vez, o AG obteve uma taxa de crescimento de 1,68%.

Levando em consideração os resultados apresentados, como trabalhos futuros propõe-se a aplicação de novas simulações com um maior número de conexões e com um maior número de comprimentos de onda por enlace. Propõe-se, também, que sejam utilizadas novas abordagens, tanto para analisar a taxa média de bloqueio de conexões, como também minimizar esse problema.

REFERÊNCIAS BIBLIOGRÁFICAS

- ADHIKARI, N. P.; CHOUBEY, M. e SINGH, R. **Dc Motor Control Using Ziegler Nichols and Genetic Algorithm Technique**. **International Journal of Electrical, Electronics and Computer Engineering**. Jabalpur. v. 1. p. 33-36. 15 Abril 2012. ISSN 2277-2626. Department of Electrical Engineering, Gyan Ganga College of Technology.
- ATZINGEN, V. *et al.* **Análise comparativa de algoritmos eficientes para o problema de caminho mínimo**. Universidade de São Paulo (USP). São Paulo. Escola Politécnica. p. 3. 2012.
- AZODOLMOLKY, S. *et al.* **A Survey on Physical Layer Impairments Aware Routing and Wavelength Assignment Algorithms in Optical Networks**. **Computer Networks**, Elsevier, V. 53, N. 7, P. 926-944, 2009.
- BARCELOS, J. C. H. **Algoritmos genéticos adaptativos: Um estudo comparativo**. Escola Politécnica. Universidade de São Paulo (USP). São Paulo. p. 131. 2000.
- BELLMAN, R.E. **On a routing problem**, **Quarterly Applied Mathematics**. (1958), 425-447.
- BITTENCOURT, G. **Inteligência Artificial - Ferramentas e Teorias**. 2a. ed. Florianópolis: Editora da UFSC. 2001.
- CHAVES, D. A.; MARTINS-FILHO, J. F.; FILHO, C. J. B. **Ferramenta Computacional para Simulação de Redes Ópticas Transparentes**. **Anais do MOMAG**. 2008. p. 908-913.
- CHRISTODOULOPOULOS, K.; TOMKOS, I.; VARVARIGOS, E. **A Comparison of Offline IA-RWA Approaches**. Athens Information Technology Center. Peania, Atenas, Grécia.
- CHU, X.; LIU, J.; ZHANG, Z. **Analysis of Sparse-Partial Wavelength Conversion in Wavelength-Routed WDM Networks**. In: IEEE. Infocom 2004. Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies. [S.L.], 2004. V. 2, P. 1363-1371.

- CZARKOWSKI, D. **Identification and Optimization PID parameters using MATLAB**. Cork Institute of Technology. Gdynia Maritime University. Poland. p. 22. 2002.
- DANTAS, E. R. M. **Desenvolvimento de uma Toolbox para aplicação de inteligência computacional em sistemas de controle clássico**. Universidade Federal Rural do Rio Grande do Norte (UFERSA). Mossoró. p. 33. 2013.
- DIAL, R; GLOVER, G; KARNEY, D. e KLINGMAN, D. **A computational analysis of alternative algorithms and labelling techniques for finding shortest path trees**. Networks 9, (1979), 215-348.
- DIJKSTRA, E. W. **A note on two problems in connexion with graphs**. Numerische mathematik. Springer. v1. n1. p 269-271. 1959.
- DREYFUS, S. E. **An appraisal of some shortest-path algorithms**. *Operations Research* 17. (1969). 395-412
- DRUMMOND, A. C.; FONSECA, N. L. DA. **Um Algoritmo Rápido Eficiente e Justo para Agregação Dinâmica de Tráfego em Redes WDM**. 27º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. 2009. 17-30.
- ENGELBRECHT, A. P. **Computational Intelligence: An Introduction**. 2ª. ed. Pretoria: John Wiley e Sons. 2007.
- FERNANDES, C. E. M. **Algoritmos de alocação de rota e comprimento de onda em redes óticas limitadas por PMD e XPM/SPM**. Universidade Estadual do Rio Grande do Norte (UERN)/ Universidade Federal Rural do Rio Grande do Norte (UFERSA). Mossoró. p. 76. 2010.
- GABRIEL, P. H. R.; DELBEM, A. C. B. **Fundamentos de Algoritmos Evolutivos. Notas Didáticas do ICMC-USP**. São Carlos/SP: ICMC-USP. v. 75. 2008. 35 p. ISBN 0103-2585.
- GALLO G. E PALLOTINO, S. **Shortest path methods: a unifying approach**. Mathematical Programming Study 26, (1986), 38-64.
- GOLDSCHMIDT, R. R. **Uma Introdução à Inteligência Computacional: fundamentos, ferramentas e aplicações**. 1ª. ed. Rio de Janeiro: IST-Rio, 2010.
- GOMES, A. F. **Extratécnicas de roteamento para provimento de QoS em redes óticas limitadas por dispersão de modo de polarização**. Universidade

- Estadual do Rio Grande do Norte (UERN)/ Universidade Federal Rural do Rio Grande do Norte (UFERSA). Mossoró. p. 36. 2009.
- GROSS, J. L.; YELLEN, J. **Graph theory and its applications**. 2nd ed. Boca Raton. FL: Chapman and Hall. 2006. 779 p. ISBN 9781584885054.
- GUIMARÃES, A. O. **Algoritmo Genético aplicado no controle de posição do rotor de um motor de corrente contínua com rejeição a distúrbios por ação feedforward**. Universidade Federal Rural do Rio Grande do Norte (UFERSA). Mossoró. p. 40. 2013.
- HOFFMAN, R. e PAVLEY, R. R. A method for the solution of the Nth best path problem. **Journal of the Association for Computing Machinery** 6. (1959). 506-514.
- HOLLAND, J. H. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**. New York: Ann Arbor. 1975.
- KATOH, N.; IBARAKI, T. e MINE, H. **An efficient algorithm for K shortest simple paths**. Networks 12. (1982). 411-427.
- MARANHÃO, J.; SOARES, A.; WALDMAN, H. **Alocação De Comprimento De Onda em Redes Ópticas Considerando as Degradações de Camada Física**. XXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. 2010. 829-840.
- MEHDI, M. **Parallel Hybrid Optimization Methods for Permutation Based Problems**. Universidade de Luxemburgo. Luxemburgo. 2011.
- MICHALEWICZ, Z. **Genetic Algorithms+Data Structures=Evolution Programs**. 3^a. ed. New York: Springer-Verlag. 1996.
- MOORE, E. F. **The shortest path through a maze, In Proc. of the Int. Symp. On the Theory of Switching**, (1959), Harvard University Press, 285-292.
- MURTHY, C. S. R.; GURUSAMY, M. **WDM Optical Networks - Concepts, Design and Algorithms**. Prentice Hall PTR. 2002.
- OLIVEIRA, V. A. P. **Algoritmos e um Sistema Distribuído para Simulação de Redes Ópticas com Variação do Ganho dos Amplificadores**. Universidade

- Estadual do Rio Grande do Norte (UERN)/ Universidade Federal Rural do Rio Grande do Norte (UFERSA). Mossoró. p. 62. 2012.
- PACHECO, M. A. **Algoritmos Genéticos: Princípios e Aplicações**. 1ª. ed. Rio de Janeiro: ICA-Rio. 1999. ICA: Laboratório de Inteligência Computacional Aplicada - www.ica.ele.puc-rio.br.
- PAPE, U. **Implementation and efficiency of Moore algorithms for the shortest root problem**. *Mathematical Programming*. 7, (1974), 212-222.
- PASCOAL, M. M. B. **Algoritmos para a enumeração dos K trajectos mais curtos**. Departamento de Matemática. Universidade de Coimbra. p. 39. 1997.
- PEREIRA, J. P. P. **Aplicação de algoritmo genético ao problema de planejamento de caminhos com a abordagem de decomposição em células convexas para o caso aproximado**. Universidade Federal Rural do Rio Grande do Norte (UFERSA). Mossoró. p. 17. 2012.
- PERKO, A. **Implementation of algorithms for K shortest loopless paths**, *Networks* 16. (1986). 149-160.
- RAMASWAMI, R; SIVARAJAN, K; SASAKI, G. **Optical networks: a practical perspective**. 3ed. Morgan Kaufmann, 2009. ISBN 978-0-12-3792-2.
- SANTOS, A. F. *et al.* **Roteamento e Alocação De Espectro Em Redes Ópticas: O Conceito Slice**. XXX Simpósio Brasileiro de Telecomunicações (SBRT'12), 2012.
- SIVANANDAM, S. N.; DEEPA, S. N. **Introduction to Genetic Algorithms**. Heidelberg: Springer. 2010. ISBN 978-3-540-73189-4.
- SOARES, A. C. B.; MARANHÃO, J.; GIOZZA, W. **First Load Priority: A Wavelength Converter Placement Scheme For Optical Networks**. *Revista Brasileira De Redes De Computadores e Sistemas Distribuídos*. 2008. Pages 23-36.
- SOARES, A. C.; GIOZZA, W. F. **Avaliação De Desempenho De Algoritmos Para Alocação Dinâmica De Comprimentos De Onda Em Redes Ópticas Transparentes**. 2004.

- SOARES, A. *et al.* **Wavelength Converter Placement Scheme For Optical Network With Sparse-Partial Wavelength Conversion Capability.** In: 13th International Conference On Telecommunications. [S.L.: S.N.], 2006.
- TALBI, E. G. **A Taxonomy of Hybrid Metaheuristics.** Journal of Heuristics. 8. n. 5. 01 set. 2002. 541-564
- YEN, J. Y. **Finding the K shortest loopless paths in a network,** Management Science 17. (1971). 712-716.
- YEN, J. Y.; **Shortest path network problems.** Mathematical Systems in Economics. Heft 18. Hain. (1975). Meisennheim am Glan.
- ZANG, H., JUE, J. AND MUKHERJEE, B.. **A Review Of Routing And Wavelength Assignment Approaches For Wavelength-Routed Optical WDM Networks.** Optical Network Magazine. 2000. Pages 47-59.

ANEXO A – PSEUDOCÓDIGOS

A1. GERADOR DE CONEXÕES ALEATÓRIAS

função (Orig, Dest, Duracao, Inicio) = gerador(Nvertices, Nconex, MedT, rede)

% Vetor de Origens e Destinos iniciando vazios

Orig= \emptyset ;

Dest= \emptyset ;

% Definindo Pontos de Origem e Destino e cada conexão

% Primeiro laço de 1 até o número de conexões.

para i de 1 até Nconex **faça**

% Recebendo os valores de origem e destino randomicamente.

origem = ceil(n*rand);

destino = ceil(n*rand);

% Preenchendo os vetores de origens e destinos.

Orig = [Orig;origem];

Dest = [Dest;destino];

fim

% Definindo Inicio e Duração de cada conexão

lambda = 1; %Distribuição Poissoniana de momento de chegada da conexão

expmean = MedT; % Distribuição exponencial de duração da conexão

Inicio = 1 + poissrnd(lambda, 1, Nconex);

Duracao = exprnd(expmean, 1, Nconex);

para i de 2:Nconex **faça**

Inicio(i) = Inicio(i-1) + Inicio(i);

fim

fim

% Fim do Gerador de Conexões

A2. ALGORITMO DE DIJKSTRA

```

função [Pred]= dijkstra(Matriz, n, m, Orig, Dest)

    Pred = zeros(n,1); % Criando os vetores
    dt = zeros(n,1);
    S = zeros(n,1);

    para i de 1 até n faça % n numero total de vértices
        dt(i) = inf; % Colocando todas as distancias infinitas;
        S(i) = true; % Colocando todos como verdadeiros;
    fim

    dt(Orig) = 0; % Distância do vértice inicial para ele mesmo é zero
    Pred(Orig) = Orig; % Predecessor de vértice inicial recebe ele mesmo

    % Loop infinito para visitar todos os vértices enquanto não estiver vazio

    enquanto existir vértices faça
        w=-1;
        para v de 1 até n faça
            se (S(v) == true) então
                se (w==-1) então
                    w=v;
                senão_se (dt(v) < dt(w)) então
                    w=v;
            fim
        fim
    fim

    % Varredura para verificar se todos os vértices já foram visitados

    se (w == -1)
        break
    fim
    S(w)=false;
    para k de 1 até n faça
        se (Matriz(w,k) > 0) então
            se (dt(k) > (dt(w)+Matriz(w,k))) então
                dt(k)=dt(w)+Matriz(w,k);
                Pred(k)=w;
            fim
        fim
    fim
    fim
fim
% Fim do Dijkstra

```

A3. GENERALIZAÇÃO DO ALGORITMO DE YEN

$T_t \leftarrow$ Árvore de caminhos mínimos para todos os nós para t em (V, A) ;
 $p \leftarrow T_t(s)$ (Menor caminho de s para t em (V, A));
 $P \leftarrow \{p\}$;
 $k \leftarrow 1$;
Enquanto $(k \leq K)$ e $(P \neq \emptyset)$ **Fazer**
 $p_k \leftarrow$ trajeto de P tal que $c(p_k) \leq c(p)$, qualquer que seja $p \in P$;
 $P \leftarrow P - \{p_k\}$;
 $\alpha_k \leftarrow$ Ordem do nó de desvio de p_k ;
Para (todo o $i \in \{\alpha_k, \dots, l_k - 1\}$) **fazer**
 $X \leftarrow \{(v_i^k, j) \langle \rangle T_t(j) : (v_i^k) \text{ não pertence à árvore construída}\}$
Se $(X \neq \emptyset)$ **Então**
 $q_i^k \leftarrow$ trajeto mais curto em X ;
 $P_i^k \leftarrow \text{sub}_k(s, v_i^k) \langle \rangle q_i^k$;
 $P \leftarrow P \cup \{p_i^k\}$;
FimSe
FimPara
 $k \leftarrow k + 1$;
FimEnquanto

A4. MENOR CAMINHO COM ALGORITMO GENÉTICO

Entrada: Grafo, Origem, Destino

Saída: Menor Caminho

Início

para $i \leftarrow 1$ até N **faça**

cromossomos[i][1] \leftarrow Nó que contém o ponto inicial

enquanto $j <$ quantidade de nós do grafo e cromossomos[i] \neq Nó de destino **faça**

Sorteia um vizinho de cromossomo[i];

Cromossomo[i][j] \leftarrow vizinho sorteado

fim

fim

para $i \leftarrow 1$ até N **faça**

Utilizar a função de ajuste para remover nós repetidos do cromossomo[i];

Calcular a aptidão do cromossomo[i];

fim

Remover cromossomos repetidos;

enquanto critério de parada não satisfeito **faça**

se o número de pais aptos < 2 **então**

completar população gerando indivíduos até o limite N ;

utilizar a função de ajuste para remover os nós repetidos e/ou laços

fim

senão

calcular a probabilidade de cruzamento

se probabilidade de cruzamento \leq taxa de cruzamento **então**

Calcular custo dos cromossomos utilizando função de valorização;

Selecionar os indivíduos mais aptos utilizando método da roleta;

Verificar os pontos para realizar o cruzamento;

Cruzar aleatoriamente os indivíduos mais aptos até atingir N indivíduos

Calcular a probabilidade de mutação

se probabilidade de mutação \leq taxa de mutação **então**

Aplicar aleatoriamente a mutação;

fim

fim

se não atingiu N indivíduos **então**

completar população gerando indivíduos aleatoriamente até o limite N;

fim

fim

fim

fim