



Otimização de Interligação de Religadores por Fibra Óptica para Automação de Redes Auto-recuperáveis de Energia Elétrica

Marcos Nogueira Lobo de Carvalho, Escola de Engenharia Elétrica, Mecânica e de Computação - UFG,
Maykon Renan Pereira da Silva, Escola de Engenharia Elétrica, Mecânica e de Computação - UFG,
Flávio Henrique Teles Vieira, Escola de Engenharia Elétrica, Mecânica e de Computação - UFG,
Flávio Geraldo Coelho Rocha, Escola de Engenharia Elétrica, Mecânica e de Computação - UFG

Resumo—Neste artigo, propõe-se a utilização de um algoritmo genético multiobjetivo *NSGA-II* (Algoritmo Genético de Classificação Não-Dominada) para dimensionar a rede óptica em malha (*Mesh*) que interliga os religadores de uma rede de energia inteligente (*Smart Grid*) que possui a capacidade de se auto-recuperar. Considerando somente distâncias (custo) de enlace, este dimensionamento pode ser efetuado a partir da solução para o Problema do Caixeiro Viajante. Assim, primeiramente apresenta-se neste trabalho uma comparação de desempenho dos seguintes algoritmos utilizados para solucionar o Problema do Caixeiro Viajante (*Travelling Salesman Problem - TSP*): Força Bruta, Entropia Cruzada, Algoritmo Genético e Algoritmo Genético de Classificação Não-Dominada (*NSGA-II*). Em seguida, propõe-se a utilização do *NSGA-II* para dimensionar a rede óptica em malha considerando fatores como custo e vazão.

Palavras-chave—Algoritmo Genético. Otimização Multiobjetiva. Smart Grid. Self-Healing. Telecomunicações.

Optimization of Interconnection of Reclosers via Fiber Optic to the Automation of Energy Network with Self-Healing

Abstract—In this article, it is proposed the use of a multiobjective genetic algorithm *NSGA-II* (Nondominated Sorting Genetic Algorithm) for dimensioning the mesh optical network that connects the reclosers of a smart grid which has the self-healing capacity. Considering only link distances (cost), this dimensioning can be done considering the solution to the Traveling Salesman Problem. Thus, we first present in this work a performance comparison of the following algorithms used to solve the Traveling Salesman Problem (*TSP*): Brute Force, Cross Entropy, Genetic Algorithm, and Nondominated Sorting Genetic Algorithm (*NSGA-II*). Next, we propose the *NSGA-II* to dimension the mesh optical network, considering factors such as cost and throughput.

Index Terms—Genetic Algorithm. Multi-objective Optimization. Smart Grid. Self-Healing. Telecommuni-

cations.

I. INTRODUÇÃO

COM o aumento da população e a grande difusão de máquinas que utilizam a energia elétrica, a demanda por esse tipo de energia tende a aumentar nos próximos anos. De acordo com o estudo divulgado pela Empresa de Pesquisa Energética (EPE) [1], o consumo brasileiro vai triplicar até 2050, chegando a 1.624 terawatt-hora (Twh). Em função disso, será necessário que o Brasil invista muito em seu parque gerador elétrico para suportar essa crescente demanda [2].

Outro grande desafio consiste na distribuição confiável e eficiente de toda essa energia proveniente das usinas. Dessa forma, cabe às concessionárias de distribuição de energia elétrica lidarem com esse desafio, por meio de soluções inovadoras e bem planejadas para garantir a conexão, o atendimento e a entrega efetiva de energia elétrica ao consumidor.

Estudos e projetos têm sido desenvolvidos nos últimos anos para otimizar a distribuição da energia elétrica. Dentre eles, destaca-se a automação das redes elétricas por meio de *Smart Grid*.

Smart Grid é uma nova arquitetura de distribuição de energia elétrica, mais segura e inteligente, que integra e possibilita ações a todos os usuários conectados a esse tipo de rede, utilizando recursos computacionais para isso [3].

Grande parcela dos projetos de *Smart Grid* consistem na utilização de modernos equipamentos capazes de monitorar o fornecimento de energia e a detecção de falhas de abastecimento antes que elas provoquem interrupções. Aliados a esses equipamentos encontram-se os *softwares* que automatizam diversos processos por meio da análise dos dados obtidos, identificação de um problema e acompanhamento até que o mesmo seja resolvido.

Esse trabalho descreve parte da implantação do projeto P&D 368 CELG-D-ANEEL-UFG, onde o objetivo é a implantação de uma rede de energia elétrica auto-recuperável em Goiânia. A rede possui um sistema que

Autor correspondente: Marcos Nogueira Lobo de Carvalho, marcosnlc4@gmail.com

pode se autorrecuperar rapidamente e em pouco tempo caso ocorra algum tipo de interrupção e/ou falha na distribuição de energia elétrica ao consumidor. Para tanto, se faz necessário uma taxa de transmissão de dados entre os religadores (parte inteligente da rede de energia elétrica) confiável e alta. Sendo assim, a utilização de fibra óptica se torna uma alternativa viável.

Neste artigo propõe-se a aplicação de algoritmos genéticos para o dimensionamento de rede óptica em malha para interligação de religadores em uma rede auto-recuperável, comparando o desempenho a outros algoritmos na literatura. Para tal, inicialmente procurou-se solucionar o Problema do Caixeiro Viajante (*Travelling Salesman Problem (TSP)*)[4] [5] para garantir o menor caminho na interligação entre os nós da rede óptica em malha e assim minimizar o custo na implantação da fibra óptica. Logo depois, implementou-se a modelagem por Teoria de Filas [6] [7] para estimar a vazão média e retardo fim-a-fim médio na rede óptica em malha obtida para três diferentes cenários (enlace rompido na rede, topologia da rede inalterada e adição de enlaces a rede). Por fim utilizou-se o *NSGA-II* [8] para dimensionar a rede óptica em malha considerando fatores como custo e vazão (obtido por Teoria de Filas). Ou seja, emprega-se o algoritmo *NSGA-II* para resolver o problema multiobjetivo de maximizar a vazão da rede e minimizar o custo de implantação da rede de fibra óptica.

Este artigo está organizado da seguinte maneira. Na seção II, é apresentado o problema de interligação de religadores de energia elétrica em uma linha de distribuição autor-recuperável, localizada na cidade de Goiânia-GO. Na seção III, propõe-se a aplicação do *TSP* para o dimensionamento da rede óptica responsável pelo controle da linha de distribuição. Nas seções IV, V e VI, são abordadas soluções do *TSP* utilizando Força Bruta, Entropia Cruzada e Algoritmo Genético, respectivamente. Já a seção VII aborda uma estimativa de vazão e atraso na rede óptica, utilizando Teoria de Filas. A seção VIII propõe-se a utilização do *NSGA-II* para dimensionar a rede óptica em malha. Por fim, na seção IX, realiza-se uma conclusão sobre o presente estudo.

II. PROBLEMA DE INTERLIGAÇÃO DE RELIGADORES

O Projeto analisado consiste na implantação de uma Rede Óptica em Malha (*Mesh*) na cidade de Goiânia - GO para automatizar os religadores de uma rede elétrica inteligente (*Smart Grid*). Essa rede possui a característica de se auto-recuperar (*self-healing*) de eventuais adversidades que possam interromper a distribuição de energia elétrica [9] [10].

Optou-se por interligar os religadores com fibra óptica do tipo monomodo em busca de garantir uma confiabilidade do canal de comunicação próxima a 100% e alta taxa de vazão que possibilite comando em tempo real. Outras alternativas poderiam ser empregadas como enlace da rede, como outras tecnologias cabeadas (cabo coaxial ou cabo par trançado) e até mesmo tecnologias sem fio (redes de celulares, radiofrequência - *WiMax*, *ZigBee*,

Bluetooth -, satélites, dentre outras). A utilização de fibra óptica foi uma decisão dos projetistas do estudo de caso analisado (P&D 368 CELG-D-ANEEL-UFMG), visando aproveitar a rede óptica (*backhaul*) já existente.

Ademais, a utilização de Redes de Fibra Óptica em Malha (ROM) possui o objetivo de estabelecer a comunicação entre os religadores que fazem parte do sistema de automação. A proposta de ROMs para Redes Elétricas Inteligentes requer um *backhaul* que leve as informações até a concessionária. No estudo de caso, essa conexão se dá através da Rede Metropolitana de Goiânia (MetroGyn¹). A Figura 1 ilustra a proposta de utilização da rede MetroGyn para escoar o tráfego das células ROM. O tráfego da rede de fibra óptica em malha é escoado até o Centro de Operações (COD) através de uma rede de fibra óptica *Gigabit Ethernet* com redundância em anel propiciando alta taxa de transmissão e alta confiabilidade ao sistema.

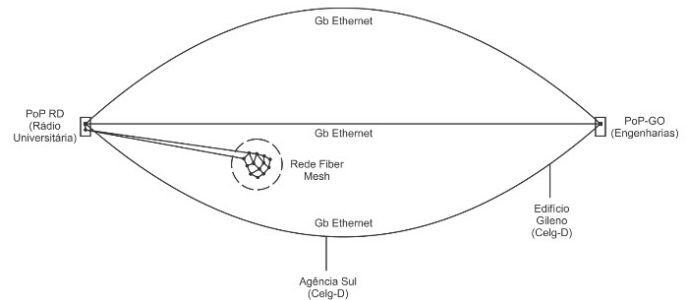


Fig. 1. Proposta de utilização da rede MetroGyn para escoar o tráfego das células ROM.

A Figura 2 apresenta os pontos de localização onde os religadores foram instalados a partir de uma imagem via satélite (obtida pelo *Google Maps*) da região de Goiânia - GO, referente ao projeto P&D 368 [9] [10]. O protocolo de roteamento utilizado nos nós (dispositivos *fibermesh*) da rede foi o *AODV (Ad hoc On demand Distance Vector Routing)* [11]. Esse protocolo é baseado no algoritmo vetor distância, onde o mesmo calcula a menor rota possível com base no custo de cada enlace.

III. DIMENSIONAMENTO DE REDE ÓPTICA EM MALHA ATRAVÉS DA SOLUÇÃO DO *TSP*

A história do Problema do Caixeiro Viajante (*TSP*) remete a um vendedor, situado numa determinada cidade, que procura obter o menor caminho possível que o permita visitar somente uma vez as demais cidades de um mapa, retornando à cidade de partida.

Numa análise matemática do problema, o *TSP* é conhecido como um clássico problema de otimização combinatória que consiste em partir de um vértice qualquer de

¹A Rede MetroGyn é uma rede de fibra óptica de alta velocidade destinada ao tráfego de dados, voz e vídeo. Ela faz parte da Redecomep, uma iniciativa do Ministério da Ciência, Tecnologia, Inovação e Comunicação (MCTIC), coordenada pela Rede Nacional de Ensino e Pesquisa (RNP), que tem o objetivo de implementar redes de alta velocidade nas regiões metropolitanas do país servidas pelos PoPs da RNP. Mais informações a respeito, pode ser encontrada no seguinte endereço: <https://metrogin.ufg.br/>

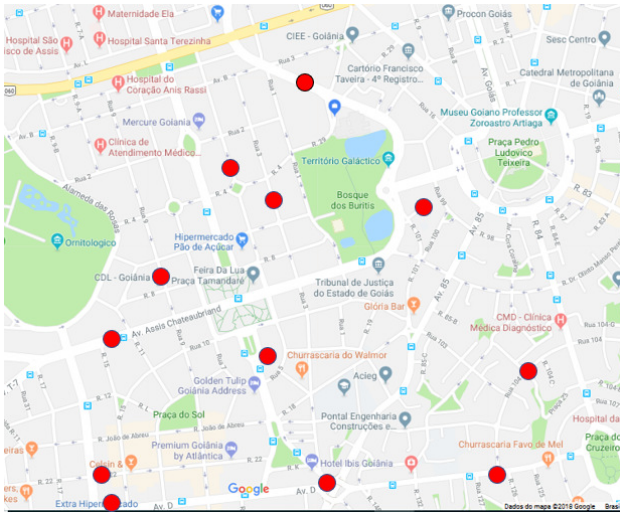


Fig. 2. Diagrama físico dos religadores de uma rede de energia elétrica instalados em uma região da cidade de Goiânia-GO.

um grafo, percorrer todos os outros vértices sem repeti-los, utilizando o menor caminho possível, e retornar para o vértice inicial [4] [5]. Segundo [4], é possível formular um algoritmo de programação linear para este problema, de forma que a solução ótima será dada pelo caminho de tamanho mínimo, onde as variáveis de decisão, a função objetivo e as restrições são dadas, respectivamente, por:

Variáveis de decisão:

$$x_{ij} = \begin{cases} 1, & \text{se a aresta } i \rightarrow j \text{ existe} \\ 0, & \text{caso contrário} \end{cases} \quad (1)$$

Minimizar:

$$\sum_i \sum_j d_{ij} x_{ij} \quad (2)$$

Sujeito a:

$$\sum_j x_{ij} = 1, \quad i = 1, \dots, N, \quad (3)$$

$$\sum_i x_{ij} = 1, \quad j = 1, \dots, N, \quad (4)$$

$$(x_{ij}) \in X,$$

onde N é o número de vértices, d_{ij} é a distância entre os vértices i e j e x_{ij} são as variáveis de decisão. A expressão $(x_{ij}) \in X$ denota o conjunto de restrições de sub-rotas de parada que restringem as soluções viáveis para aquelas que consistem num único caminho.

O projeto de uma rede de fibra óptica em malha (*Mesh*) capaz de fornecer dados para acionar religadores de energia pode ser relacionado com um típico problema do caixeiro viajante. Na maioria dos casos, a aplicação do *TSP* tende a formar uma configuração em anel. Essa topologia é realizada conectando os nós adjacentes da rede até que todos estejam conectados, garantindo que o menor caminho para se percorrer todos os nós (sem passar mais de uma vez por um mesmo nó) da rede seja obtido. Uma vez

que os nós da rede estejam interconectados, é necessário adicionar as redundâncias (rotas adicionais) para se obter uma rede óptica em malha que satisfaça os critérios do projeto. A Figura 3 apresenta um exemplo de tal configuração obtida no trabalho de [9] [10]. Nesse trabalho, ao se utilizar algoritmos para se interligar os religadores via solução do *TSP*, propomos acrescentar redundâncias, além do anel formado. Mais especificamente, propomos que cada nó deva estar conectado a pelo menos mais um nó (uma redundância) e no máximo mais quatro nós (quatro redundâncias). Apesar de demandar mais fibra óptica (aumento do custo), as redundâncias permitem obter uma configuração em malha para a rede óptica, possibilitando a formação de subanéis que proporcionam maior confiabilidade ao sistema, garantindo uma maior qualidade de serviço para a rede [9] [10].

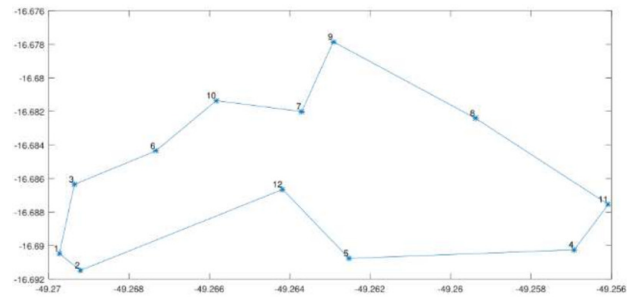


Fig. 3. Topologia em anel obtida com a aplicação do *TSP* [9] [10].

As informações sobre os algoritmos empregados no estudo, bem como os resultados das simulações aplicadas no estudo de caso são apresentados nas seções a seguir.

IV. SOLUÇÃO DO *TSP* COM FORÇA BRUTA

A solução do *TSP* por Força Bruta consiste basicamente em avaliar todos os caminhos possíveis na malha, armazenando todas as distâncias obtidas. Com todas as distâncias de todos os caminhos obtidas, basta escolher o caminho no qual se obteve a menor distância.

Apesar de garantir que sempre a melhor solução será obtida, esse algoritmo é extremamente ineficiente para malhas com muitos nós. Sua complexidade computacional é de $O((n-1)!)$, onde n é o número de nós da malha. Isso comprova o grande aumento de rotas ao se incrementar o número de nós [12].

No presente estudo, utilizou-se um programa implementado em MATLAB capaz de solucionar o *TSP* com Força Bruta [13]. Este código é executado ao se inserir um arquivo de texto contendo a tabela de distâncias dos nós da malha. Assim que a tabela é inserida, o algoritmo de Força Bruta é executado, retornando a menor distância, a maior distância e a distância média percorrida entre os religadores a serem interligados, bem como o número de caminhos verificados e o tempo de execução.

A. Resultados da Simulação do *TSP* com Força Bruta

Por meio dos resultados obtidos nas simulações com o algoritmo de Força Bruta apresentados na Figura 4,

fica evidente a ineficiência do algoritmo por Força Bruta com a elevação do número de nós da malha, visto que o tempo de execução para 12 nós ficou em torno de 8 horas. Apesar dessa ineficiência, a solução do *TSP* via Força Bruta garante sempre o melhor resultado possível ao fim da execução, uma vez que o mesmo é um algoritmo determinístico.

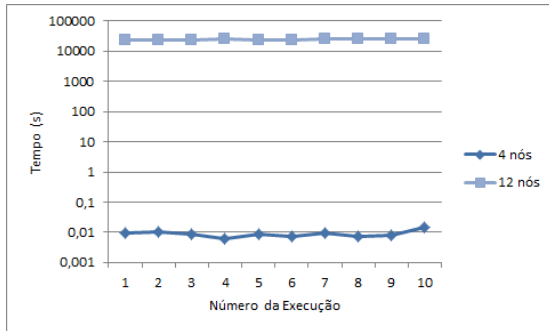


Fig. 4. Resultados do tempo de execução das simulações utilizando o algoritmo com Força Bruta.

V. SOLUÇÃO DO *TSP* COM ENTROPIA CRUZADA

DE acordo com [14], a Entropia Cruzada (*Cross Entropy (CE)*) é um tipo de algoritmo adaptativo que trabalha com a estimativa de probabilidades de eventos raros em redes complexas estocásticas que estejam relacionadas com a minimização da variância. Para se aplicar Entropia Cruzada em problemas de otimização combinatória difíceis, como o *TSP*, deve-se modificar o método original, convertendo o problema de otimização determinística para um problema relacionado à otimização estocástica. Para obter as amostras aleatórias de eventos de forma adaptativa, é necessário a utilização de técnicas de simulação de eventos raros [14].

O processo de geração de novas amostras possibilita a construção de novas seqüências de soluções que convergem (de forma probabilística) para a solução ótima ou mais aceitável. O método *CE* consiste num processo iterativo que pode ser dividido nas seguintes fases [10]:

- Geração de uma amostra de dados (vetores, trajetórias, etc.) de acordo com um determinado mecanismo;
- Atualização dos parâmetros do mecanismo aleatório de acordo com os dados para obter uma amostra melhor a cada iteração.

Para a formulação do problema de otimização combinatória estocástica, partiremos da Simulação de Eventos Raros. Para isso, vamos considerar o grafo ponderado de pesos aleatórios X_1, \dots, X_5 distribuídos exponencialmente com média μ_1, \dots, μ_5 apresentado na Figura 5. O problema da simulação de Eventos Raros consiste em encontrar a probabilidade da distância entre A e B ser maior ou igual do que γ [10].

Sejam $X = X_1, \dots, X_5$ e a função $S(X)$ que define a distância de A à B, a probabilidade de $S(X) \geq \gamma$ é dada

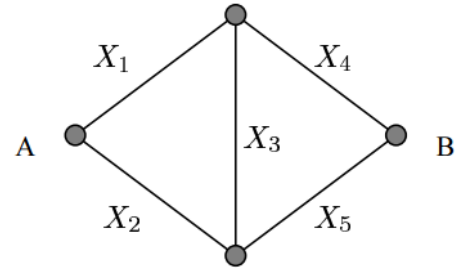


Fig. 5. Esquema do problema de simulação de eventos raros a partir de um grafo [14].

por [10]:

$$l = P(S(X) \geq \gamma) = El_{\{S(X) \geq \gamma\}} \quad (5)$$

A probabilidade l pode ser encontrada por simulação *Crude Monte Carlo (CMC)* realizando amostras aleatórias de X_1, \dots, X_n através de $f(x; \mu)$. Um estimador não enviesado de l é [10]:

$$\hat{l} = \frac{1}{N} \sum_{i=1}^N l_{\{S(X) \geq \gamma\}} W(X_i) \quad (6)$$

Na Equação 6, \hat{l} é o estimador de verossimilhança (*likelihood ratio*) (LR), $W(X)$ é chamada de razão de verossimilhança e é dada por [10]:

$$W(x) = \frac{f(x)}{g(x)} \quad (7)$$

Sejam X_1, \dots, X_n as amostras aleatórias de g . No caso particular onde $g = f$, temos que o estimador LR é o CMC.

O problema da otimização combinatória consiste em encontrar um vetor binário $y = (y_1, \dots, y_n)$ que para um dado vetor binário $x = (x_1, \dots, x_n)$ maximize a função $S(x)$ que representa o número de combinações entre x e y , dada por [10]:

$$S(x) = n - \sum_{j=1}^n |x_j - y_j| \quad (8)$$

Uma maneira simples de encontrar y é gerar vetores binários $X = (X_1, \dots, X_n)$, de forma que X_1, \dots, X_n sejam variáveis aleatórias independentes de Bernoulli com probabilidade de sucesso p_1, \dots, p_n . Se $p = y$, temos $S(X) = n$ e $X = y$. O algoritmo proposto para encontrar y chegará à solução com probabilidade de sucesso 1 para $(N \rightarrow \infty)$. O método CE consiste em aplicar o problema acima na estrutura de simulação de eventos raros e então criar uma seqüência de vetores de parâmetros $\hat{p}_0, \hat{p}_1, \dots$ e níveis $\hat{\gamma}_0, \hat{\gamma}_1, \dots$ de forma que $\hat{\gamma}_0, \hat{\gamma}_1, \dots$ convirja para $S(X) = n$ e $\hat{p}_0, \hat{p}_1, \dots$ convirja para y [10].

O Algoritmo 1 apresenta a seqüência de passos utilizados para se obter a otimização combinatória.

A interpretação do algoritmo sugere que para atualizar a j -ésima probabilidade de sucesso, conta-se quantos vetores

Algoritmo 1 ALGORITMO CE PARA OTIMIZAÇÃO COMBINATÓRIA [10].

1. Iniciar o vetor \hat{p}_0 e $t := 1$.
2. Gerar amostras X_1, \dots, X_n de vetores de Bernoulli com probabilidade de sucesso \hat{p}_{t-1} .
3. Calcular o valor de $S(X_i)$ para todo i e ordenar do menor para o maior, $S_1 \leq \dots \leq S_N$. $\hat{\gamma}_t$ recebe o valor do quantil $(1 - \rho)$ de S , $\hat{\gamma}_t = S[(1 - \rho)N]$.
4. Utilizar a mesma amostra para calcular o $\hat{p}_t = (\hat{p}_{(t,1)}, \dots, \hat{p}_{(t,n)})$ via $\hat{p}_{(t,j)} = \frac{\sum_{i=1}^n l_{\{S(X_i) \geq \hat{\gamma}_t\}} l_{\{X_{ij}=1\}}}{\sum_{i=1}^n l_{\{S(X_i) \geq \hat{\gamma}_t\}}}$, onde $j = \{1, \dots, n\}$, $X_i = (X_{i1}, \dots, X_{in})$.
5. Parar se o critério de parada for atingido. Caso contrário, $t := t + 1$ e reiterar do passo 2.

da última amostra X_1, \dots, X_N possuem o valor de $S(X_i) \geq \gamma_t$ e possuem a j -ésima posição igual a 1 e normaliza-se pelo número de vetores que tem o valor de $S_{X_i} \geq \gamma_t$ [10].

Partindo do grafo dirigido $G = [V, E]$, temos que χ é o conjunto de todas as rotas possíveis e $S(x)$ o somatório dos custos da rota $x \in \chi$. Cada rota pode ser representada por uma permutação de $(1, \dots, n)$. Assumindo a permutação de x para $x_1 = 1$ podemos formular o *TSP* como:

$$\min_{x \in \chi} S(x) = \min_{x \in \chi} \left(\sum_{i=1}^{n-1} c_{x_i, x_{i+1}} k_{x_i, x_{i+1}} + c_{x_n, 1} k_{x_n, 1} \right) \quad (9)$$

Para aplicar o algoritmo CE de otimização combinatória devemos alterar a formulação original de modo a calcular o mínimo. Para tal temos que especificar como gerar as rotas aleatórias e como atualizar os parâmetros a cada iteração.

Primeiramente, definimos o conjunto de rotas que partem e terminam em 1 como e podem visitar a mesma cidade mais de uma vez, conforme indicado pela Equação 10.

$$\tilde{\chi} = (x_1, \dots, x_n) : x_1 = 1, x_i \in [1, \dots, n], i = [2, \dots, n] \quad (10)$$

Dessa forma, a função de custo é definida por:

$$\tilde{S}(x) = \begin{cases} S(x), & x \in \chi \\ \infty, & \text{caso contrário} \end{cases} \quad (11)$$

A função objetivo do problema consiste em minimizar a função custo:

$$\min_{x \in \chi} \tilde{S}(x) \quad (12)$$

Após algumas manipulações matemáticas (o trabalho de [10] contém todos os passos dessas manipulações), conclui-se que o estimador de verossimilhança para o *TSP* é o seguinte:

$$\hat{p}_{i,j} = \frac{\sum_{k=1}^N l_{\{S(X_k) \leq \gamma\}} l_{\{X_k \in \chi_{ij}\}}}{\sum_{k=1}^N l_{\{S(X_k) \leq \gamma\}}} \quad (13)$$

onde χ_{ij} é o conjunto de todas as rotas nas quais as transições de i para j são realizadas.

A interpretação da atualização dos valores de p_{ij} consiste em obter a fração de vezes que as transições de i para j ocorrem levando em consideração as rotas que são menores ou iguais a γ . Em suma, gera-se uma rota através de um processo Markoviano com matriz de transição P atualizando os pesos de acordo com o estimador.

A resolução do problema do caixeiro viajante (*TSP*) garante ao site a aplicação da topologia em anel. Assim, garante-se ao menos uma rota alternativa por ponto. A complexidade computacional de um algoritmo que utiliza *CE* é da ordem de $O(n \ln n)$ [15].

Uma implementação de um programa em MATLAB capaz de solucionar o *TSP* via *CE* foi desenvolvida por [16]. As variáveis de entrada para o algoritmo de Entropia Cruzada são:

- N - Número de amostras para gerar cada rodada;
- ρ - fração das melhores amostras a se tomar;
- α - parâmetro de sensibilidade;
- C - matriz/tabela de distâncias; e
- τ - se 0, posicionamento dos nós. Se 1, transição dos nós.

O algoritmo de Entropia Cruzada fornece o menor caminho (ou mais aceitável) para solucionar o problema do caixeiro viajante.

A. Resultados da Simulação *TSP* com Entropia Cruzada

Primeiramente foram realizadas simulações com diferentes valores para o parâmetro de tolerância. Esse parâmetro representa a tolerância a uma resposta longe da ideal [9] [10]. Ou seja, trata-se de um parâmetro de tolerância a erros. Os resultados das simulações do *TSP* com Entropia Cruzada com diferentes valores para a tolerância indicaram que quanto maior o valor do parâmetro de tolerância, menor é o tempo de execução. Assim, ao fixar o parâmetro de tolerância em 0,005, foram obtidos os resultados descritos na Figura 6:

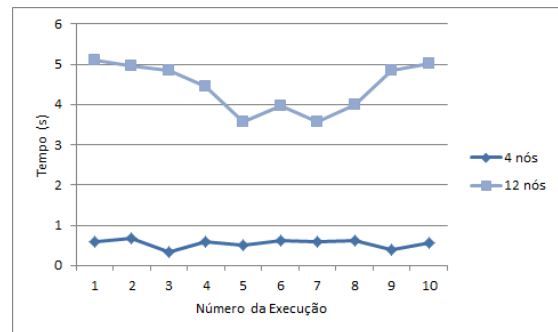


Fig. 6. Resultados de tempo de execução das simulações com Entropia Cruzada, com o parâmetro de tolerância fixo.

Os demais parâmetros utilizados, para realizar essas simulações foram os seguintes:

- $N = 1000$;
- $\rho = 0,05$;
- $\alpha = 0,8$;
- $\tau = 0$.

A Figura 7 ilustra o resultado da malha óptica (com as redundâncias aplicadas nos nós) após a otimização da rede através do algoritmo *TSP* com Entropia Cruzada, obtido a partir do trabalho de [9] [10].

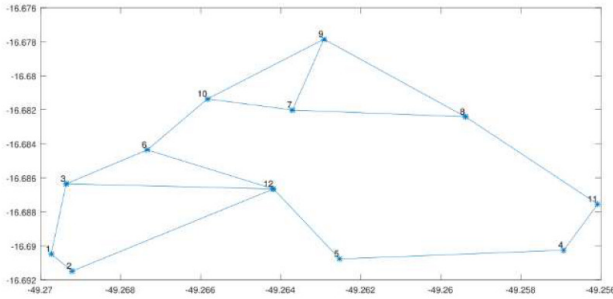


Fig. 7. Solução do *TSP* com Entropia Cruzada. Devido à inclusão das redundâncias, obtém-se uma configuração em malha ao invés de uma configuração em anel, como a obtida na solução nativa do *TSP*

VI. SOLUÇÃO DO *TSP* COM ALGORITMO GENÉTICO

COM o objetivo principal de buscar soluções aproximadas em problemas de busca ou otimização, os Algoritmos Genéticos (AGs) trabalham com uma busca probabilística baseada nos princípios de uma população natural, tais como seleção, cruzamento e mutação. Assim, os AGs são aleatórios-guiados, quando aplicados em sua forma mais pura, posto que o conhecimento do problema está embutido na função de avaliação e, eventualmente, nos operadores [17].

Para abstrair o problema, cada indivíduo de uma população é chamado de cromossomo ou genótipo, onde estes podem ser representados por uma estrutura de dados como vetor, lista, tabela *hash*, etc. Após ser inicializada, a população passa por um processo de avaliação, onde ocorre a verificação se o critério de término foi atingido. As apurações se a solução ótima global foi alcançada ou o número máximo de gerações foi atingida são utilizadas como critérios de término. Se o critério de término for atingido, o algoritmo retorna os melhores indivíduos alcançados. Caso contrário, deve-se gerar uma nova população por meio de uma seleção baseada na função de avaliação e em seguida, deve-se aplicar recombinação, mutação e realizar a avaliação dos indivíduos descendentes. Depois deve-se verificar se a nova geração da população obteve um critério de parada aceitável. Se não, repete-se todo o processo de geração de uma nova população. A Figura 8 ilustra o fluxograma de um típico AG.

As buscas heurísticas possuem tempo de execução de ordem polinomial. Dessa forma, um típico Algoritmo Genético possui uma complexidade computacional de $O(n^2)$ [18].

Uma implementação para a solução do *TSP* via Algoritmo Genético tradicional é proposta por [19]. Realizada com o MATLAB, essa implementação plota um conjunto de pontos no plano bidimensional, formando a malha do problema de acordo com a tabela de distâncias informada no código. Cada ponto possui vários caminhos com diferentes tamanhos para os demais pontos. Como o cromossomo

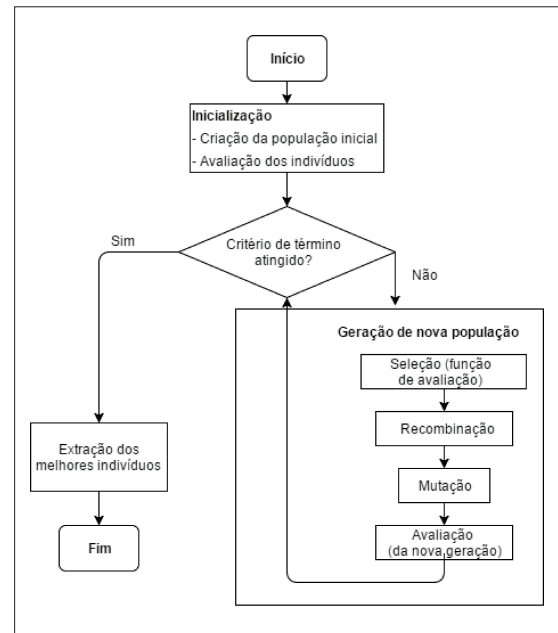


Fig. 8. Fluxograma de um típico Algoritmo Genético.

é uma possível solução do problema, nessa implementação, vetores de cidades (nós) são utilizados para representar os cromossomos, onde cada cidade (nó) representa um gene. As posições das cidades são informadas por meio de suas coordenadas cartesianas. Os parâmetros exigidos para executar o programa são os seguintes:

- *nn* - Número de nós;
- *ps* - Tamanho da população;
- *ng* - Número de gerações;
- Taxas de mutação;
 - *pm* - Probabilidade de mutação de troca de 2 cidades aleatórias no caminho (por gene, por geração);
 - *pm2* - Probabilidade de mutação de troca 2 pedaços de caminho (por gene, por geração); e
 - *pmf* - Probabilidade de mutação de troca aleatória de caminho.
- Tabela de distâncias entre os nós.

Para resolver o *TSP*, o Algoritmo Genético segue as seguintes etapas que são repetidas a cada geração:

- 1) Encontrar o tamanho dos caminhos;
- 2) Encontrar as probabilidades (aptidão);
- 3) Preparar para a recombinação de acordo com as probabilidades;
- 4) Recombinação e substituição dos pais pelos filhos; e
- 5) Aplicação de mutação.

A. Resultados da Simulação *TSP* com Algoritmo Genético

Os resultados das simulações com Algoritmo Genético são exibidos na Figura 9. Para essas simulações foram utilizados os seguintes parâmetros:

- *ps* = 3000 (Tamanho da População);
- *ng* = 200 (Número de Gerações);

- Taxas de mutação;
 - $pm = 0,01$ (Probabilidade de mutação na troca de duas cidades aleatórias pelo caminho: por gene, por geração);
 - $pm2 = 0,02$ (Probabilidade de mutação na troca de dois pedaços de caminho: por gene, por geração);
 - $pmf = 0,08$ (Probabilidade de mutação em um pedaço de caminho aleatório).

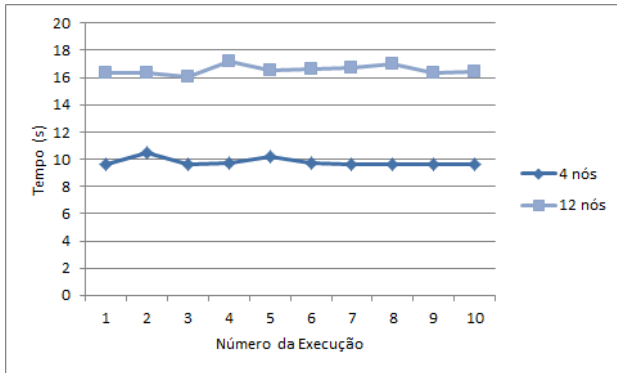


Fig. 9. Resultados do tempo de execução das simulações com Algoritmo Genético.

A Figura 9 apresenta o tempo de execução até a última geração de indivíduos. Assim, foi possível notar que o tempo de execução foi menor em certas simulações, onde o algoritmo convergiu mais rapidamente, ou seja, o algoritmo encontrou o menor caminho na rede antes de chegar na última geração.

A Figura 10 apresenta a comparação do tempo médio de execução dos três algoritmos de função objetivo única considerados. Os três algoritmos obtêm a malha em anel e em seguida adiciona as redundâncias a partir das restrições consideradas no trabalho. Foi proposto que cada nó deveria estar conectado a no mínimo um nó (uma redundância) e no máximo a quatro nós (quatro redundâncias).

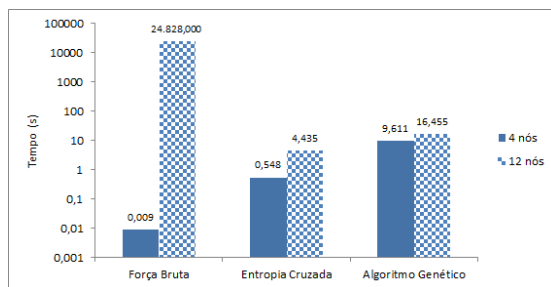


Fig. 10. Comparação do tempo médio dos algoritmos de único objetivo.

O comprimento total de fibra óptica encontrado pelos algoritmos ao se considerar apenas a solução do TSP foi o mesmo. Na prática, dadas as configurações das ruas e do cenário, obteve-se um comprimento total de 13,55 km. A Figura 11 apresenta o comprimento total obtido para a rede real (13,55 km). Os pontos em vermelho indicam

os religadores conectados aos dispositivos *fibermesh* que precisam ser interligados.

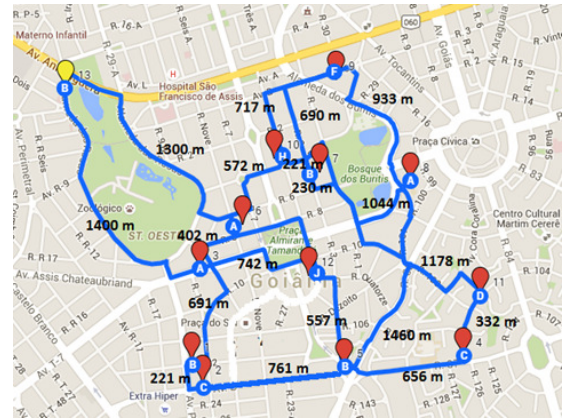


Fig. 11. Comprimento total da rede óptica real.

VII. ESTIMAÇÃO DE VAZÃO E ATRASO UTILIZANDO TEORIA DE FILAS

COMO se objetiva nesse trabalho inserir a vazão da rede como critério a ser maximizado pelo *NSGA II*, aborda-se nesta seção a estimação da vazão via Teoria de Filas.

A Teoria de Filas é uma das técnicas da Pesquisa Operacional, que trata de problemas de congestionamentos de sistemas, onde clientes solicitam algum tipo de serviço. Esse serviço é limitado por restrições intrínsecas do sistema, que, devido a isso, podem causar filas. A Figura 12 representa um esquema de fila, onde existem diversos indivíduos que disputam entre si o atendimento por um determinado serviço [20].

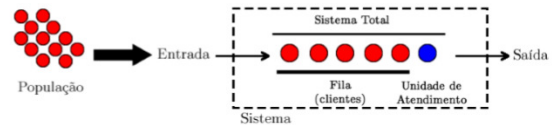


Fig. 12. Esquema de uma fila [20].

Na modelagem utilizada, considera-se uma rede estática composta por N nós em malha (*mesh*) e um *gateway* que atua como concentrador dos nós *mesh*. No modelo do sistema, um nó cuja distância em contagem de saltos até o *gateway* for igual a x é referido como nó *xhop*.

Aplica-se Teoria de Filas para melhorar o desempenho de um serviço ou sistema, reduzindo seus custos operacionais. Para otimizar o desempenho dos modelos de filas de espera, é necessário analisar os resultados gerados por fórmulas apropriadas a um determinado modelo [7].

Na modelagem matemática do tráfego da rede óptica, considera-se uma rede estática, conforme apresentada na Figura 3, composta por 12 nós em malha (*mesh*) e um *gateway* (nó 12) que atua como concentrador dos nós *mesh*. A capacidade do canal sem fios é W bits por segundo, isto é, para cada nó, a taxa de transmissão é de W bits por

segundo. Esta velocidade é constante e independente do número de nós da rede. Os pacotes são enviados através de múltiplos saltos para ou a partir do *gateway*. Para facilitar a explicação e sem perder a generalidade, considera-se o tráfego unidirecional, isto é, o tráfego que parte apenas dos nós de rede *mesh* para o *gateway*, também referido como tráfego a montante. Em [6], considera-se a condição saturada, ou seja, cada nó é sempre *backlogged* (há sempre fila no *buffer*) e sempre existem dados locais para serem transmitidos para o *gateway*.

Denota-se a proporção do número de nós *xhop* na rede por $h(x)$. Para se ter um valor de referência, H denota a distância máxima da contagem de saltos possível a partir do *gateway* da rede. Isto significa que os nós $Hhop$ são os nós mais distantes do *gateway*. Assim, não haverá nenhum tráfego retransmitido para o *gateway* vindo através dos nós $Hhop$ [6], [7].

Dada a proporção de nós *xhop* na rede por $h(x)$, pode-se obter o número esperado de nós *xhop* da rede, que é representado por $N(x)$, como [6], [7]:

$$N(x) = N \times h(x) \quad (14)$$

Uma vez que cada nó *xhop* é assumido para retransmitir os pacotes dos nós $(x+1)hop$, o número esperado de nós $(x+1)hop$ para o qual cada nó *xhop* tem que retransmitir seus pacotes, designado por $N_r(x)$, é dado pela seguinte equação [6], [7]:

$$N_r(x) = \begin{cases} \frac{N(x+1)}{N(x)} = \frac{h(x+1)}{h(x)}, & x = 1, 2, \dots, H-1 \\ 0, & x = H \end{cases} \quad (15)$$

Para analisar o comportamento das filas, é preciso saber as taxas de chegada de pacotes previstas para as filas. O parâmetro $\mu(x)$ representa a taxa de chegada de pacotes para um nó *xhop*, enquanto que, $\lambda(x)$ representa a taxa de retransmissão do nó *xhop*. A Equação 17 apresenta a relação entre $\mu(x)$ e $\lambda(x)$ [6], [7].

$$\lambda(x) = \begin{cases} N_r(x) \cdot \mu(x+1), & x = 1, 2, \dots, H-1 \\ 0, & x = H \end{cases} \quad (16)$$

Por fim, existe o parâmetro $\rho(x)$, que representa a intensidade de tráfego em um nó *xhop*, sendo definido por [6], [7]:

$$\rho(x) = \frac{\lambda(x)}{\mu(x)} = \begin{cases} \frac{N_r(x) \cdot \mu(x+1)}{\mu(x)}, & x = 1, 2, \dots, H-1 \\ 0, & x = H \end{cases} \quad (17)$$

A. Estimação da Vazão (Throughput)

Após a obtenção das distribuições de entrada e saída de cada nó *xhop* é possível determinar a vazão do tráfego de ponta a ponta. Primeiramente é necessário determinar a probabilidade de bloqueio de mensagens/pacote $T(x)$, é definida como o número médio de pacotes recebidos com sucesso pelo destino, ou seja, o *gateway*, por unidade de

tempo. Mais especificamente, a vazão do nó *xhop* é definida como a taxa de pacotes que deixam o nó *xhop* e não são bloqueados por qualquer um dos nós intermediários entre este e o *gateway*.

A probabilidade de bloqueio para o nó *xhop* é dada por $P_b(x)$. A partir das fórmulas de Teorias de Filas M/M/1/K (assume-se um sistema com 1 servidor e *buffer* de tamanho K), temos [6], [7]:

$$P_b(x) = \begin{cases} \frac{[1-\rho(x)] \cdot \rho^K}{1-\rho(x)^{K+1}}, & \rho(x) \neq 1 \\ \frac{1}{K+1}, & \rho(x) = 1 \end{cases} \quad (18)$$

onde $\rho(x)$ é dado pela Equação 17 e K representa o *buffer* permitido pelo nó *xhop*. Por conseguinte, a probabilidade de não bloqueio de pacote para o nó *xhop* é $[1 - P_b(x)]$. Para um caminho, a probabilidade fim-a-fim de não haver bloqueio é igual ao produto das probabilidades de não bloqueio em todos os nós intermediários. Portanto, para o nó *xhop*, a sua vazão $T(x)$ é igual ao produto da taxa de saída efetiva e a probabilidade de não bloqueio fim-a-fim, ou seja [6], [7]:

$$T(x) = \begin{cases} \mu(1), & x = 1 \\ \mu(x) \cdot \prod_{i=1}^{x-1} [1 - P_b(i)], & x = 2, \dots, H \end{cases} \quad (19)$$

O valor agregado das vazões por nó, é denotado pelo parâmetro T_{agg} . Por meio deste, é possível obter o rendimento do sistema. Já o rendimento médio por nó é obtido pelo parâmetro T_{ave} . Posto que o número de nós *xhop* é $N(x)$, temos [6], [7]:

$$T_{agg} = \sum_{x=1}^{x=H} [N(x) \cdot T(x)] \quad (20)$$

$$T_{ave} = \frac{T_{agg}}{N} \quad (21)$$

B. Estimação do Atraso (Delay)

O primeiro passo para se obter o atraso fim-a-fim consiste em investigar o número esperado de pacotes enfileirados no nó *xhop*. Por meio da equação que descreve o valor de $L_r(x)$, obtém-se o tamanho da fila no estado estacionário para o nó *xhop*. De acordo com as fórmulas de Teoria de Filas M/M/1/K, temos [6], [7]:

$$L_r(x) = \begin{cases} \frac{\rho(x)}{1-\rho(x)} - \frac{\rho(x)[K\rho(x)^{K+1}]}{1-\rho(x)^{K+1}}, & \rho(x) \neq 1 \\ \frac{K(K-1)}{2(K+1)}, & \rho(x) = 1 \end{cases} \quad (22)$$

O retardo fim-a-fim de um pacote é definido como o tempo entre o momento em que o primeiro bit deste pacote é enviado pela fonte e quando o pacote é completamente recebido pelo de destino, ou seja, o *gateway*. Assume-se que o retardo de propagação é negligenciável. Dessa forma, o atraso fim-a-fim de pacotes é dado pela soma dos tempos de transmissão e os atrasos na fila para todos os nós intermediários.

O parâmetro $D(x)$ é o atraso praticado por um pacote gerado pelo nó $xhop$. Para se obter $D(x)$, deve-se calcular o tempo de espera em cada salto. O tempo de espera de um pacote de um nó intermediário é igual ao tempo entre quando este pacote é colocado dentro da fila deste nó e quando este nó começa a enviar o primeiro bit do pacote para o próximo nó (ou para a *gateway*). Portanto, o tempo total para um pacote passar por um nó intermediário é igual ao tempo de espera na fila mais o tempo de transmissão. O tempo de espera para pacotes no nó $xhop$ é denotado por $W_r(x)$. De acordo com a fórmula de Little [21], temos [6], [7]:

$$W_r(x) = \frac{1}{\mu(x)} + \frac{L_r(x)}{\lambda(x)[1 - P_b(x)]}, \quad x = 1, 2, \dots, H - 1 \quad (23)$$

O $D(x)$ é obtido a partir da soma dos tempos de espera gastos nos nós com os tempos de transmissão para atravessar os x saltos. A Equação 24 calcula o valor de $D(x)$, onde t_c representa o tempo de transmissão [6], [7]:

$$D(x) = \begin{cases} t_c, & x = 1 \\ x \cdot t_c + \sum_{i=1}^{x-1} W_r(i), & x = 2, 3, \dots, H \end{cases} \quad (24)$$

O valor médio do atraso fim-a-fim é dado por D_{ave} e este é obtido a partir da média dos atrasos dos pacotes que são recebidos com sucesso pelo *gateway*. Uma vez que, dentro de uma unidade de tempo, $T(x)$ representa o número total de pacotes gerados pelo nó $xhop$ e recebido com sucesso pelo *gateway*, o número total de pacotes gerados por todos os nós $xhop$ e recebidos com sucesso pelo *gateway* é igual a $N(x)T(x)$. Portanto, o retardo total dos pacotes gerados e recebidos com sucesso por todos os nós $xhop$ é $N(x)T(x)D(x)$. Consequentemente, temos [6], [7]:

$$D_{ave} = \frac{\sum_{x=1}^{x=H} [N(x) \cdot T(x) \cdot D(x)]}{T_{agg}} \quad (25)$$

C. Resultados com Modelagem de Rede Mesh por Teoria de Filas

Nessa subseção, apresenta-se os resultados de estimação de vazão e atraso obtidos através da modelagem da rede por Teoria de Filas. Posteriormente, serão comparadas essas estimativas com os valores obtidos via simulação computacional das redes.

Foram estimados parâmetros como (vazão média e atraso médio fim-a-fim) para três diferentes cenários da rede óptica em malha (Figura 3) com o intuito de analisar o desempenho de rede:

- *Cenário 1*: Com rompimento de um enlace na rede (nó 8 com o nó 11);
- *Cenário 2*: Sem modificação (a topologia é mantida);
- *Cenário 3*: Adição de quatro enlaces à rede ([3,12], [6,12], [7,10] e [9,10]) Figura 7.

A Figura 13 apresenta a estimação da vazão média para os três cenários, variando o *buffer* e mantendo fixo a taxa de chegada de pacotes no nó ($\mu(x)$) em 1000 pacotes por

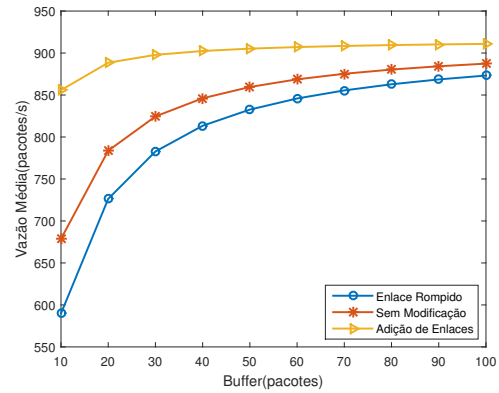


Fig. 13. Estimação da vazão média para diferentes tamanhos de *buffer*.

segundo e o tempo de transmissão (t_c) em 1 milissegundo. Observando as curvas do gráfico percebe-se que quanto maior o *buffer* maior a vazão média, pois *buffers* maiores armazenam mais pacotes na fila, reduzindo o descarte de pacotes. Percebe-se que a rede do cenário 3 apresentou os maiores valores para a vazão em relação aos outros dois tipos de rede quando se varia o *buffer*. Tal resultado era esperado, pois possuindo mais redundâncias (enlaces), diminui-se o congestionamento e as perdas de pacotes nos nós.

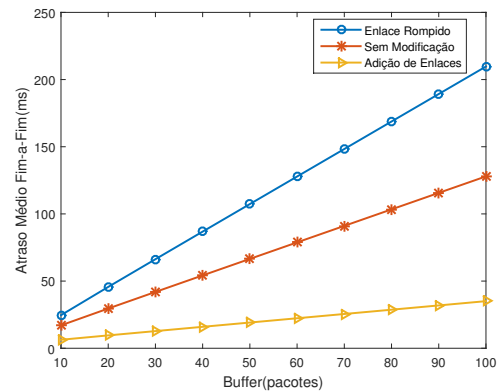


Fig. 14. Estimação do atraso fim-a-fim para diferentes tamanhos de *buffer*.

A Figura 14 apresenta a estimação do atraso fim-a-fim para os três cenários, mantendo o mesmo valor de taxa de chegada e tempo de transmissão da Figura 13. O atraso fim-a-fim aumenta com o aumento do *buffer* para os três cenários, isso ocorre porque quanto maior o tamanho do *buffer*, maior a fila, logo, maior o tempo de espera dos pacotes no nó, o que afeta diretamente o atraso médio fim-a-fim de cada pacote. Novamente, a rede descrita no cenário 3 obteve o melhor desempenho, pois apresentou os menores valores de retardo para diferentes valores de *buffer*.

A Figura 15 apresenta a estimação da vazão média para os três cenários, variando a taxa de chegada de pacotes e mantendo o *buffer* fixo em 50 pacotes e o tempo de

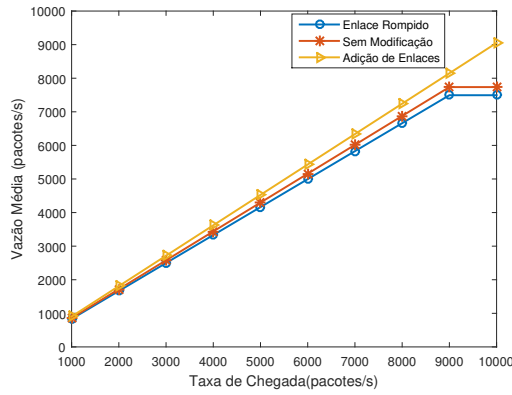


Fig. 15. Estimativa da vazão média para diferentes valores de taxa de chegada.

transmissão (t_c) em 1 milissegundo. Como mostram os gráficos, a vazão aumenta linearmente com o aumento da taxa de chegada. O cenário 3 apresentou o melhor desempenho para a vazão, o que mais uma vez era esperado por possuir um maior número de enlaces.

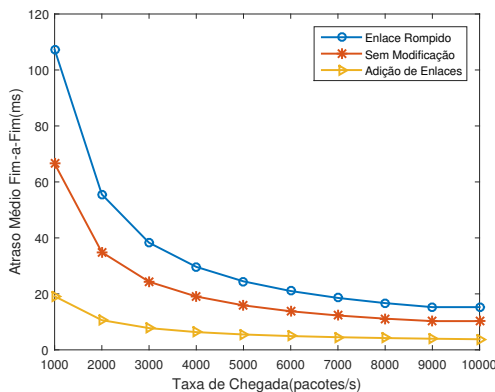


Fig. 16. Estimativa do atraso fim-a-fim para diferentes valores de taxa de chegada.

A Figura 16 apresenta a estimativa do atraso fim-a-fim para os três cenários, mantendo o mesmo valor de *buffer* (50 pacotes) e tempo de transmissão (1 milissegundo). O atraso diminuiu com o aumento da taxa de chegada para os três casos. Como a vazão média aumenta, o retardo tende a diminuir, visto que o aumento da vazão média na rede acarretará numa redução do congestionamento.

A Figura 17 apresenta a vazão de dados para a rede óptica em malha da Figura 3 durante 60 segundos, onde a geração de pacotes iniciou aos 10 segundos e terminou aos 50 segundos. A simulação foi feita no *Network Simulator 2* (NS2²) para comparar os resultados dos parâmetros (vazão média e retardo fim-a-fim médio) obtidos pela modelagem matemática com os resultados obtidos pelo simulador.

²O *Network Simulator 2* é um simulador de eventos discretos voltado para pesquisa na área de Redes. Mais informações a respeito do simulador podem ser encontradas no seguinte endereço: <https://www.isi.edu/nsnam/ns/>

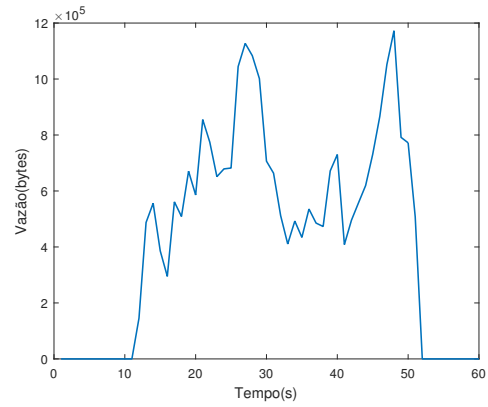


Fig. 17. Tráfego da vazão média pelo tempo.

Para realizar essa comparação, utilizaram-se os seguintes parâmetros no simulador de rede:

- Taxa de transmissão dos enlaces: 1 Gigabit por unidade de tempo (Esse valor foi escolhido devido às fibras do tipo monomodo possuírem taxas de transmissão na ordem de 1 Gigabit por segundo);
- Tamanho do pacote: 1024 Bytes;
- Tamanho do *buffer*: 50 pacotes;
- Tráfego exponencial com taxa de envio de pacotes: 1×10^6 pacotes por segundo;
- Protocolo de Roteamento: Vetor Distância; e
- Tipo de Fila: *Drop Tail* que representa o algoritmo (*First In, First Out*).

Já para a modelagem por Teoria de Filas utilizou-se os seguintes parâmetros:

- Taxa de chegada de pacotes no nó: 1×10^6 pacotes por segundo;
- Tamanho do *buffer*: 50 pacotes;
- Tempo de transmissão: 0,01 milissegundos.

A Tabela I apresenta os resultados para os parâmetros vazão média por nó e atraso fim-a-fim médio por pacote obtidos pela modelagem por teoria de filas e pelo NS2. Analisando a tabela percebe-se que os resultados obtidos pela modelagem matemática ficaram próximos dos obtidos pelo simulador de rede.

TABELA I
COMPARAÇÃO ENTRE OS RESULTADOS ESTIMADOS PELA MODELAGEM POR TEORIA DE FILAS E PELO SIMULADOR DE REDE.

Resultados	Vazão (Mb/s)	Atraso (ms)
Modelagem por Teoria de Filas	0,6768	0,0449
Simulador de Rede (NS2)	0,6550	0,0470

VIII. DIMENSIONAMENTO DE REDE ÓPTICA UTILIZANDO *NSGA-II*

QUANDO um problema de otimização possui múltiplos objetivos, obter apenas uma solução ótima não é suficiente. Nesse caso, é interessante obter um conjunto de soluções ótimas. Esse conjunto é chamado de soluções pareto-ótimas [8].

As soluções clássicas para a solução desse tipo de problema consistem em dividir os problemas de otimização multiobjetivos em problemas simples de único objetivo, de forma a tratar cada solução pareto-ótima de uma vez. Posteriormente, unem-se todas as soluções pareto-ótima obtidas individualmente e aplica-se esse método várias vezes até se obter a solução ótima.

O *NSGA-II* é capaz de solucionar problemas de otimização multiobjetivo com restrições com uma complexidade computacional da ordem de $O(MN^2)$, onde M é o número de objetivos e N é o tamanho da população. Para isso, o *NSGA-II* utiliza um rápido procedimento de seleção não-dominada com uma abordagem elitista de preservação da diversidade das soluções [8].

O primeiro passo desse algoritmo é, para cada solução, calcular duas entidades na seguinte ordem:

- 1) Contagem dominante, n_p : trata-se do número de soluções que dominam a solução p ; e
- 2) s_p : é o conjunto de soluções que a solução p domina.

Em seguida, deve-se gerar uma população aleatória P_0 de pais. Essa população é ordenada de acordo com a não-dominância, onde a cada solução é atribuído um fator de aptidão igual ao seu nível de não-dominância. Quanto menor o nível de não-dominância, melhor será a função de aptidão (ou seja 1 é o melhor nível, 2 é o próximo melhor nível e assim por diante). Nesse sentido, a função de aptidão deve ser minimizada.

A população inicial de descendentes Q_0 , de tamanho N , é criada a partir da seleção de torneio binário comum, sob influência dos fatores de mutação escolhidos. Para os demais descendentes é utilizado o elitismo por meio da comparação da população atual com as melhores soluções não-dominadas. O Algoritmo 2 exibe o pseudocódigo do *NSGA-II*, destacando a formação da t -ésima geração [8].

Algoritmo 2 FUNÇÃO PRINCIPAL DO *NSGA-II* [8].

```

 $R_t = P_t \cup Q_t$ 
 $F = \text{selecaoNaoDominadaRapida}(R_t)$ 
 $P_{t+1} = \emptyset$ 
 $i = 1$ 
 $|P_{t+1}| + |F_i| \leq N \text{ comparacaoPorAglomeracao}(F_i)$ 
 $P_{t+1} = P_{t+1} \cup F_i$ 
 $i = i + 1$ 
 $\text{ordena}(F_i, \prec_n)$ 
 $P_{t+1} = P_{t+1} \cup F_i[1 : (N - |P_{t+1}|)]$ 
 $Q_{t+1} = \text{fazerNovaPopulacao}(P_{t+1})$ 
 $t = t + 1$ 
    
```

No algoritmo de geração de populações do *NSGA-II*, uma população R_t , de tamanho $2N$ é formada pela união da população de pais (P_t) e de descendentes (Q_t). Em seguida, a população R_t é classificada de acordo com a não-dominância. Nesse caso, o elitismo é garantido, visto que todos os membros das populações anteriores e atuais dos pais e dos descendentes estão contidos em R_t .

As soluções pertencentes ao melhor conjunto não-dominado F_1 são as melhores soluções na população

combinada e devem, portanto, ser enfatizadas mais do que qualquer outra solução na população combinada. Se o tamanho da população F_1 for menor que N , deve-se escolher todos os membros do conjunto F_1 para a nova população P_{t+1} .

Os membros restantes de P_{t+1} são escolhidos a partir das frentes não-dominadas subsequentes na ordem de sua classificação. O mesmo procedimento é repetido para os demais conjuntos F_i até que não seja possível acomodar mais nenhum conjunto.

Digamos que o conjunto F_l seja o último conjunto não-dominado além do qual nenhum outro conjunto pode ser acomodado. Em geral, o número de soluções em todos os conjuntos de F_1 até F_l seria maior do que o tamanho da população.

Para escolher o número exato de membros (N), ordena-se as soluções da última frente F_l utilizando o operador de comparação por aglomeração (\prec_n) em ordem decendente e escolhe-se as melhores soluções necessárias para preencher todos os espaços da população.

A Figura 18 apresenta um esquema do processo realizado no *NSGA-II*. A nova população P_{t+1} de tamanho N é utilizada agora para seleção, recombinação e mutação para criar a nova população Q_{t+1} de tamanho N [8].

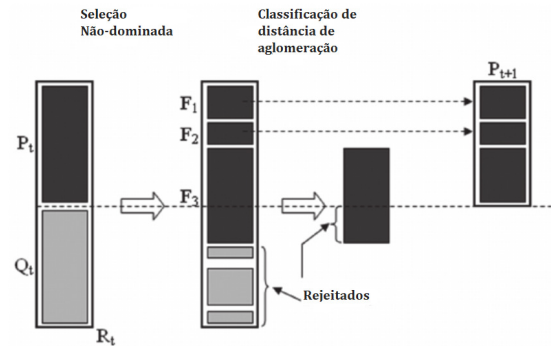


Fig. 18. Esquema do algoritmo de geração de populações do *NSGA-II* [8].

A. Resultados do Dimensionamento de Rede Óptica com *NSGA-II*

Para o estudo de caso, utilizou-se como variável de decisão uma relação de todos os religadores (nós) da rede, indicando a conexão de cada nó com os demais nós da rede. Para estabelecer essa relação, utilizou-se uma matriz de custos para representar as posições dos religadores. Nessa estrutura, foram indicados, a partir de valores binários, quais nós estavam conectados, onde 1 representa conexão e 0 indica que os nós estão soltos. Em cada geração do algoritmo, uma nova matriz de custos otimizada é gerada. A última geração representa a melhor configuração (menor custo e maior vazão - otimização das funções objetivo) da rede obtida pelo algoritmo.

As funções objetivo otimizadas no problema foram o custo e a vazão. Análogo ao objetivo otimizado nos algoritmos apresentados anteriormente, buscou-se minimizar

o custo, no sentido de obter sempre o menor caminho na rede. Já o segundo objetivo consistiu em maximizar a vazão de cada nó da rede, ou seja, a taxa efetiva de pacotes transmitidos por unidade de tempo. Esse objetivo foi adotado para garantir um bom desempenho na transmissão dos dados na rede. Mais informações a respeito de cálculos da vazão podem ser encontradas em [7] [6].

Foram consideradas as seguintes restrições, quando se aplicou o *NSGA-II* para solucionar o problema:

- O número total de conexões em cada nó deve ser maior ou igual a 1, ou seja cada nó deve estar conectado a pelo menos um outro nó da rede; e
- O número máximo de conexão para cada nó é 4, pelo fato de os religadores utilizados na rede possuírem apenas 4 conectores.

Para realizar as simulações, considerou-se 100 gerações e uma população de 50 indivíduos. Além disso, o projeto do estudo de caso utilizou os nós 3, 4, 6 e 10 para conectar diretamente ao *gateway*. Por motivos práticos, considerou-se na implementação da solução com o *NSGA-II* que apenas um nó estaria conectado ao *gateway* de cada vez. Essa implementação foi feita de tal forma a ser flexível quanto a escolha de um nó da rede para se conectar ao *gateway*.

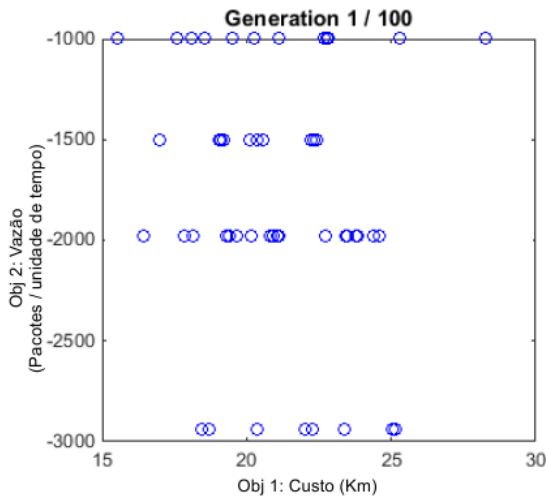


Fig. 19. Conjunto de soluções pareto-ótimas obtido na primeira geração da simulação considerando as restrições com o *gateway* conectado ao nó 3.

As Figuras 19 e 20 apresentam o conjunto de soluções pareto-ótimas da primeira e da última geração, respectivamente, considerando as restrições, um *buffer* de tamanho 50 (pacotes) e uma taxa de chegada de 1000 pacotes por segundo para o *gateway* conectado ao nó 3. Esse nó foi escolhido por ser um dos nós que estão conectados ao *gateway* no projeto do estudo de caso. O eixo das abcissas representa o custo (em quilômetros) a ser minimizado, enquanto que o eixo das ordenadas representa a vazão (em pacotes por unidade de tempo) a ser maximizada. Posto que a vazão consiste no objetivo a ser maximizado, utilizou-se nas simulações um vetor de valores negativos. Uma vez que o algoritmo *NSGA-II* procura encontrar os

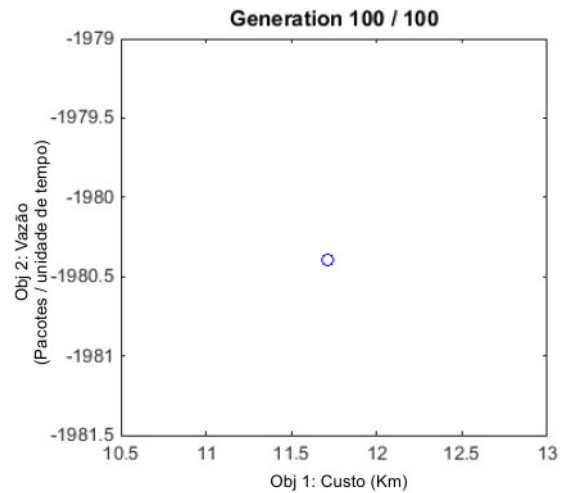


Fig. 20. Conjunto de soluções pareto-ótimas obtido na última geração na simulação considerando as restrições com o *gateway* conectado ao nó 3.

mínimos das funções objetivos, neste trabalho multiplicou-se a vazão por -1 para se obter maximização da vazão. Por esse motivo, os gráficos ficaram com valores negativos no eixo das ordenadas, apesar disso não representar vazão “negativa”.

A partir da comparação dos resultados apresentados nas Figuras 19 e 20, é possível verificar o avanço do algoritmo em obter um conjunto de soluções ótimas. Nesse caso, todas as soluções do algoritmo convergiram para o mesmo resultado, ou seja, na última geração é possível verificar que todas as 50 populações de indivíduos encontram-se no mesmo ponto do gráfico, enquanto que na primeira geração as populações de indivíduos estavam dispersas. Ao se considerar a maximização da vazão e a minimização da distância (custo) simultaneamente, o comprimento total de fibra óptica encontrado foi de, aproximadamente, 11,75km, conforme indicado pelo resultado obtido pelo gráfico apresentado na Figura 20.

A Figura 21 ilustra um grafo contendo a configuração da melhor rede obtida nessa simulação, apresentando as conexões entre cada um dos nós. Analisando-se o grafo obtido na Figura 21 nota-se que cada nó realiza no mínimo 1 conexão e no máximo 4 conexões, respeitando as restrições estabelecidas pelo problema. Note que a restrição de realizar no mínimo 1 conexão aumenta as chances de os nós 4 e 5 ficarem isolados, considerando um possível cenário de rompimento dos enlaces que chegam a esses nós. Nesse caso, seria mais seguro que os nós realizassem no mínimo duas ligações (redundâncias) para diminuir as chances desse problema ocorrer. Essa solução gerou uma vazão de, aproximadamente, 1980 pacotes por segundo e custo na ordem de 12 quilômetros.

Apesar de a simulação ter encontrado somente um conjunto de soluções ótimas, nem sempre isso ocorre. Em alguns casos, obteve-se mais de um conjunto de soluções ótimas, conforme pode ser observado na fronteira de Pareto obtida na simulação da Figura 22. Nesse exemplo de simulação, utilizou-se o nó 10 conectado ao *gateway*,

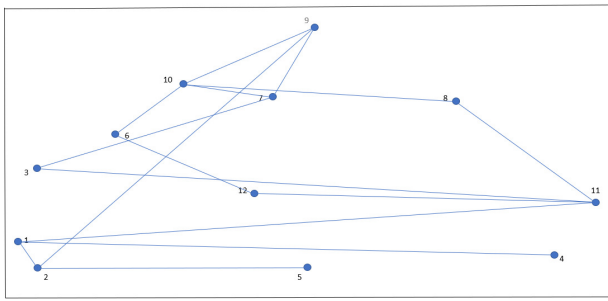


Fig. 21. Melhor configuração da rede obtida na simulação considerando as restrições com o gateway conectado ao nó 3.

considerando as restrições do problema.

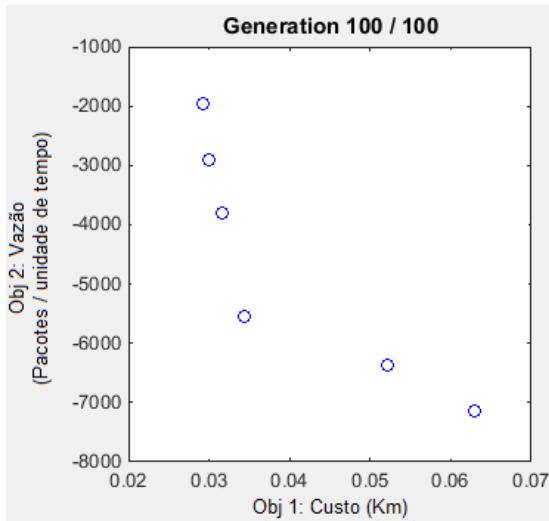


Fig. 22. Múltiplas soluções obtidas na última geração da simulação considerando as restrições com o nó 10 conectado ao gateway.

Pela Figura 22 pode-se observar que as soluções possuem um certo padrão, onde o custo e a vazão são inversamente proporcionais. Ou seja, a função objetivo da vazão aumenta para valores menores da função objetivo do custo e vice-versa. O algoritmo *NSGA-II* encontra uma configuração de rede que consegue aumentar a vazão sem aumentar tanto o custo. O algoritmo não avalia a melhor localização para as redundâncias em termos de confiabilidade, garantindo apenas uma solução que maximiza vazão a um menor custo possível.

Para confirmar os resultados obtidos, foram realizadas simulações também no simulador de rede *Network Simulator 2 (NS2)* para se comparar a vazão da rede obtida na simulação com o *NSGA-II* (considerando as restrições para o gateway conectado ao nó 3 - Figura 21) e a vazão obtida pelo simulador *NS2*.

Para realizar essa comparação, utilizou-se os seguintes parâmetros no simulador de rede:

- Taxa de transmissão dos enlaces: 1 Gigabit por unidade de tempo (Esse valor foi escolhido devido às fibras monomodos possuírem taxas de transmissão na ordem de 1 Gigabit por segundo);
- Tamanho do pacote: 2KB;

- Tamanho do *buffer*: 50 pacotes;
- Taxa de envio de pacotes: 1000 pacotes por segundo;
- Protocolo de Roteamento: Vetor Distância; e
- Tipo de Fila: *Drop Tail* que representa o algoritmo (*First In, First Out*);

O resultado obtido pelo simulador foi uma vazão média de aproximadamente 1.835 pacotes por segundo. De acordo com a Figura 20, o resultado obtido pelo *NSGA-II* foi de 1.980 pacotes por segundo. Analisando-se os dois resultados, pode-se considerar uma certa proximidade entre a simulação feita no *NSGA-II* e o *NS2*, já que não espera-se que o resultado final da vazão média da simulação teórica (com o *NSGA-II*) seja exatamente igual à vazão da simulação prática (com o *NS2*), mas que os valores mostrem uma certa proximidade que torne os resultados teóricos mais confiáveis.

Também utilizou-se o *NS2* para simular qual seria a vazão do anel apresentado na Figura 3, obtido com a aplicação direta do *TSP*. Com isso, verificou-se que a vazão total da rede para a configuração em anel foi de 22.000 pacotes por segundo, ou seja uma vazão média de aproximadamente 1.830 pacotes por segundo. Assim, a configuração em anel apresentada na simulação da Figura 3 possui uma vazão pouco menor do que a vazão obtida na simulação da rede com o *NSGA-II* apresentada na Figura 21. Esse resultado é mais uma justificativa para o uso da abordagem multiobjetiva.

IX. CONCLUSÃO

O presente trabalho teve o objetivo de analisar e otimizar a rede de comunicação em malha *Mesh* de fibra óptica de um sistema *self-healing* de energia elétrica, além de tentar maximizar a transmissão de dados (vazão) da mesma. Foram utilizados os algoritmos *TSP* para otimizar a distância de nós e enlaces e o *NSGA II* para prover solução para o problema multiobjetivo de comunicação dos religadores. Este estudo mostrou que o algoritmo que utiliza Entropia Cruzada foi o que obteve menor média de tempo de execução.

Embora o Algoritmo Genético tenha proporcionado um tempo médio de execução superior ao do Entropia Cruzada, valores de tempo de execução mais baixos podem ser obtidos pelo AG de acordo com o tamanho da população e parâmetros iniciais. Por fim, o algoritmo de Força Bruta foi o que obteve o pior desempenho com relação ao tempo de execução. Esse resultado era esperado, devido à sua alta complexidade computacional, já previamente conhecida.

Dessa forma, para o caso de um problema com apenas um objetivo (minimizar o custo de fibra óptica, ou seja, obter o caminho que exija o menor comprimento total de fibra) o algoritmo mais adequado para se utilizar é o método de Entropia Cruzada, uma vez que o mesmo encontrou os caminhos mais curtos possíveis (ou mais aceitáveis) de forma mais rápida.

A inclusão da otimização da vazão como mais um objetivo faz com que o problema torne-se multiobjetivo (maximizar a vazão e minimizar o custo). Para isso, se

faz necessário o uso do *NSGA-II*, um algoritmo genético multiobjetivo. O motivo de se maximizar a vazão se deve à busca de uma comunicação veloz entre os religadores, para que quando ocorra falhas na rede, a distribuição de energia não seja interrompida ou, em caso de interrupção, o tempo de recuperação da rede seja o menor possível. Isso é benéfico tanto para o fornecedor quanto para o consumidor, uma vez que a operadora de energia elétrica irá ter menos despesas, evitando ter de pagar multas para a Agência Nacional de Energia Elétrica (ANEEL), além de que os consumidores evitarão os transtornos ocasionados pelas quedas de energia elétrica. Também foi possível perceber que a inclusão da vazão sacrifica em parte o custo (e vice-versa), uma vez que os valores dessas variáveis mostraram um comportamento inversamente proporcional na simulação com mais de um conjunto de soluções ótimas.

De acordo com as simulações da rede de comunicação em conjunto com as equações de vazão de Teoria de Filas ([7]), percebe-se que a vazão está diretamente ligada a dois parâmetros: taxa de chegada de pacotes e o *buffer* de armazenamento. Com base nas simulações realizadas, percebe-se que essas variáveis impactam diretamente na transmissão de dados da rede. Por fim, observou-se que de fato utilizando o algoritmo genético multiobjetivo, as configurações dos nós dos enlaces da rede são atualizadas até se encontrar a melhor configuração possível para que a vazão seja máxima e o custo (tamanho dos enlaces) seja o menor possível.

Para trabalhos futuros, estamos avaliando a possibilidade de maximizar a confiabilidade da rede ao mínimo custo possível utilizando o algoritmo *NSGA-II*.

AGRADECIMENTOS

Os autores gostariam de agradecer à Celg Distribuição S.A. (CELG D) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio a esta pesquisa.

REFERÊNCIAS

- [1] *Empresa de Pesquisa Energética (EPE) divulga estudo sobre demanda do PNE 2050*, Disponível em: <https://goo.gl/gGkS4b> (Acesso em: 22 mar. 2018), 2014.
- [2] *Ministério do Planejamento PAC - Demanda por eletricidade no Brasil vai triplicar até 2050*, Disponível em: <http://www.pac.gov.br/noticia/13554306> (Acesso em: 22 mar. 2018), 2014.
- [3] *Companhia Energética de Minas Gerais (Cemig) - O Que São as Redes Inteligentes de Energia?*, Disponível em: http://www.cemig.com.br/pt-br/A_Cemig_e_o_Futuro/sustentabilidade/nossos_programas/Redes_Inteligentes/Paginas/as_redes_inteligentes.aspx (Acesso em: 22 mar. 2018), 2017.
- [4] J.-Y. Potvin, “Genetic algorithms for the traveling salesman problem”, *Annals of Operations Research*, n° 63, pp. 339–370, 1996, https://iccl.inf.tu-dresden.de/w/images/b/b7/GA_for_TSP.pdf.

- [5] A. Maredia, “History, Analysis, and Implementation of Traveling Salesman Problem (TSP) and Related Problems”, Department of Computer and Mathematical Sciences, University of Houston-Downtown, Houston, USA, 2010.
- [6] T. Liu e W. Liao, “Location-dependent throughput and delay in wireless mesh networks”, *IEEE Transactions on Vehicular Technology*, vol. 57, n° 2, pp. 1188–1198, mar. de 2008, <http://ieeexplore.ieee.org/document/4357197/>. DOI: 10.1109/TVT.2007.905389.
- [7] G. B. de Castro Souza, “Otimização do Posicionamento de Concentradores GPRS em Redes Elétricas Inteligentes Utilizando Programação Linear e Teoria de Filas”, Mestrado em Engenharia Elétrica e de Computação, Universidade Federal de Goiás, Goiânia, Brasil, 2014. endereço: <https://repositorio.bc.ufg.br/tede/handle/tede/3858>.
- [8] K. Deb, A. Pratap, S. Agarwal e T. Meyarivan, “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”, *IEEE Transactions on Evolutionary Computation*, vol. 6, n° 2, pp. 182–197, abr. de 2002, https://www.iitk.ac.in/kangal/Deb_NSGA-II.pdf. DOI: 1089-778X/02/\$17.00.
- [9] R. R. de Carvalho Vaz, F. H. T. Vieira, P. Queiroz, G. A. de Deus Junior e S. G. de Araújo, “Aplicação do Método de Entropia Cruzada para Projeto de Redes Ópticas em Malha via Resolução do TSP”, *Anais do XLVIII Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, Setembro de 2016, <http://www.din.uem.br/sbpo/sbpo2016/pdf/156742.pdf>.
- [10] R. R. de Carvalho Vaz e Flávio, “Metodologia de Posicionamento de Religadores e Dimensionamento de Rede de Fibra Óptica Para Automação de Sistemas de Energia”, Mestrado em Engenharia Elétrica e de Computação, Universidade Federal de Goiás, Goiânia, Brasil, 2017. endereço: <https://repositorio.bc.ufg.br/tede/handle/tede/7735>.
- [11] Á. O. Alves e F. D. M. de Oliveira, “Análise de Desempenho dos Protocolos de Roteamento de Redes de Sensores Sem Fio Para Monitoramento em Smart Grids”, *Revista de Sistemas e Computação*, vol. 6, n° 1, pp. 45–54, jan. de 2016, <http://www.revistas.unifacs.br/index.php/rsc/article/view/4332>, ISSN: 2237-2903.
- [12] A. Sahalot e S. Shrimali, “A Comparative Study of Brute Force Method, Nearest Neighbour and Greedy Algorithms to Solve the Travelling Salesman Problem”, *IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)*, vol. 2, n° 6, pp. 59–72, jun. de 2014, <http://www.impactjournals.us/download.php?fname=2-77-1401959097-8.%20Eng-A%20Comparative%20study%20of%20Brute%20force%20method-Antima%20Sahalot.pdf>, ISSN: 2321-8843.
- [13] J. Burkardt, *Brute Force Solution of Travelling Salesman Problem*, Available at: <https://people.sc.fsu.edu>.

- edu/~jburkardt/m_src/tsp_brute/tsp_brute.html (Accessed March 22 2018), 2013.
- [14] P.-T. de Boer, “A Tutorial on the Cross-Entropy Method”, *Annals of Operations Research*, pp. 19–67, dez. de 2005, <https://people.smp.uq.edu.au/DirkKroese/ps/aortut.pdf>.
- [15] R. Y. Rubinstein e D. P. Kroese, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*, 1^a ed. Springer; Softcover reprint of the original, 2004.
- [16] I. C. for Scientific Computing (IWR), *Travelling Salesman*, Available at: <http://www.maths.uq.edu.au/CEToolBox/node8.html> (Accessed March 22 2018), 2004.
- [17] F. T. Alves e P. E. D. Pinto, “Aplicação de Algoritmos Genéticos ao Problema do Percurso do Cavalo”, *Cadernos do IME - Série Informática*, vol. 22, pp. 09–25, Dezembro de 2006, <http://www.e-publicacoes.uerj.br/index.php/cadinf/article/view/6556/4674>, ISSN: 1413-9014. DOI: dx.doi.org/10.12957/cadinf.
- [18] J.-D. Wei, “Approaches to the Travelling Salesman Problem Using Evolutionary Computing Algorithms”, *InTech - Open Science Open Minds*, pp. 63–74, set. de 2008, https://www.intechopen.com/books/traveling_salesman_problem/approaches_to_the_travelling_salesman_problem_using_evolutionary_computing_algorithms, ISSN: 1063-6560. DOI: 10.5772/5584.
- [19] M. Vedenyov, *Travelling Salesman Problem with Genetic Algorithm*, Available at: <https://www.mathworks.com/matlabcentral/fileexchange/31818-travelling-salesman-problem-with-genetic-algorithm> (Accessed March 22 2018), 2011.
- [20] M. Sucena, *Módulo 5 - Teoria das Filas (Queueing Theory)*, Disponível em: http://www.sucena.eng.br/ST/ST5_Mod5TeoriadeFilas.pdf (Acesso em: 22 mar. 2018), 2014.
- [21] D. Gross e C. M. Harris, *Fundamentals of Queueing Theory*, 3^a ed. John Wiley Sons, Inc, 1998.