

Análise de Desempenho de Metaheurísticas Aplicadas ao Problema de Restauração de Redes de Distribuição

Vitor dos Santos Batista¹, Ubiratan Holanda Bezerra, Andrey Ramos Vieira

¹Centro de Excelência em Eficiência Energética da Amazônia (CEAMAZON)
Instituto de Tecnologia (ITEC)
Universidade Federal do Pará (UFPA)
CEP: 66.077-830 – Belém – PA – Brazil

vitorordsbatista@gmail.com, bira@ufpa.br, andreyrv@hotmail.com

Abstract. *One of the characteristics of the Smart Grid is the restoration of the electrical network, using metaheuristics to find the best temporary configuration after a fault. Based on the probabilistic characteristic of metaheuristics, this article aims to perform a comparative analysis between four metaheuristics applied to the restoration problem, using the node-depth representation as a data structure to represent the electric network.*

Resumo. *Uma das características das Smart Grid é a restauração da rede elétrica, para isso são utilizadas metaheurísticas para encontrar a melhor configuração temporária após ocorrer uma falta. Impulsionado pela característica probabilística das metaheurísticas, este artigo visa realizar uma análise comparativa entre quatro metaheurísticas aplicadas ao problema de restauração, utilizando a representação nó-profundidade como estrutura de dados para representar a rede elétrica.*

1. Introdução

Devido o grande crescimento¹ populacional dos centros urbanos ocorrido nas últimas décadas, surgiram diversos desafios relacionados a distribuição de energia elétrica. Diante deles, o conceito de Redes Elétricas Inteligentes (SG - Smart Grids) foi estabelecido, propondo a integração de diversas tecnologias de informação e comunicação com as redes de energia elétrica [Cabello 2012].

Uma das características das SG é o self-healing, onde a própria rede elétrica é capaz de modificar a sua topologia ao ocorrer uma determinada falta de caráter permanente, limitado pelas restrições existentes na rede elétrica [Kwong et al. 2002]. Nestes casos, as meta-heurísticas são bastante utilizadas para encontrar uma nova topologia temporária para a rede elétrica.

Diversos artigos artigos utilizam as metaheurísticas para a modificação da topologia da rede elétrica. Em [Delbem et al. 1998] e [Augugliaro et al. 2000] é utilizada a lógica Fuzzy para a restauração de SD. Em [Santos 2009] é apresentado o AEMT (Algoritmo Evolutivo Multi-Objetivo por Tabela) utilizando a RNP (Representação Nó-Profundidade) como estrutura de dados e dois operadores de mutação no processo de restauração, o PAO (do inglês, *Preserve Ancesor Operator*) e o CAO (do inglês *Change*

¹Bolsista do CNPq - Brasil

Ancestor Operator). Esses operadores computacionais, juntamente com a RNP, são capazes de realizar as modificações na topologia da rede elétrica sem gerarem soluções não-radiais.

Existem diversas heurísticas foram baseadas no AEMT e aplicadas ao problema de restauração, como o AEMT-SND [Sanches et al. 2011], o AEMT-SP [Sanches et al. 2013] e o AEMT-H [Sanches 2012], contudo a exploração de outras meta-heurísticas foi pouco realizada. Um exemplo está em [Mansour 2009] que realiza uma modificação no NSGA-II para trabalhar com a RNP. Devido a característica probabilística das meta-heurísticas, é necessário realizar comparações com as mais diferentes meta-heurísticas para encontrar uma que possa resolver um determinado problema de forma eficiente e no menor tempo possível [Gendreau et al. 2010] [Talbi 2009]. Diante disso, este artigo visa realizar a comparação do AEMT com algumas abordagens meta-heurísticas com outras características, como métodos de busca local, populacionais e algoritmos evolutivos. Os algoritmos selecionados estão em suas formas mais tradicionais, pois o intuito é avaliar os diferentes tipos de meta-heurísticas para o problema de restauração utilizando a RNP.

O restante do artigo está organizado da seguinte forma: Na seção 2 é aprofundado o problema de restauração de redes elétricas e apresentado a formulação matemática deste. Na seção 3 são expostas as meta-heurísticas avaliadas e seus algoritmos. Na seção 4 são apresentados os dois sistemas testes utilizado na avaliação e os resultados das execuções e discutidos os resultados. Na seção 5 é apresentada a conclusão e a exposição de possíveis trabalhos futuros.

2. O Problema de Restauração de Redes Elétricas

O problema de restauração consiste em reestabelecer o fornecimento de energia para o maior número de consumidores possíveis do sistema de distribuição a jusante a falta até que ela seja corrigida. Para isso é necessário reconfigurar o sistema elétrico através da abertura e fechamento de chaves seccionadoras espalhadas pela rede elétrica. Essa reconfiguração deverá ser realizada com a menor sequência de chaveamentos possíveis, cumprindo as restrições de carregamento de linhas, potência demandada da subestação e níveis de tensão estabelecidos. Os dados referentes as restrições são extraídos após ser executado o fluxo de carga nas topologias encontrada. Sendo assim, o problema de restauração de SD pode ser modelado como um problema multi-objetivo, cujos objetivos são minimizar o número de chaveamentos e maximizar o número de consumidores energizados [Marques 2013].

Com a utilização da RNP e dos operadores computacionais, desenvolvidos por [Santos 2009], é garantido o reestabelecimento de todos os consumidores desligados, sendo necessário apenas diminuir o número de chaveamentos. Com isso o problema de restauração de SD utilizando RNP e aplicando as devidas penalidades caso alguma restrição não seja cumprida pode ser modelado conforme 1.

Onde f é o número de manobras, T é a rede elétrica representada na RNP, X é o maior carregamento da rede, B é o maior carregamento da subestação, V é a maior queda de tensão na rede e G , H e I são as penalidades das restrições.

$$\begin{aligned}
& \text{Min } f(T) + X(T) * G + B(T) * H + V(T) * I \\
& G = \begin{cases} 100 & \text{se } X(T) > 1 \\ 0 & \text{se } X(T) \leq 1 \end{cases} \\
& H = \begin{cases} 100 & \text{se } B(T) > 1 \\ 0 & \text{se } B(T) \leq 1 \end{cases} \\
& I = \begin{cases} 100 & \text{se } V(T) > 1.05 \\ 100 & \text{se } V(T) < 0.95 \\ 0 & \text{se } 0.95 \leq V(T) \leq 1.05 \end{cases}
\end{aligned} \tag{1}$$

3. Meta-heurísticas Avaliadas

Nesta seção, serão apresentados os algoritmos que serão analisados, sendo eles o AEMT, Evolutionary Strategies (M+L), ABC (Artificial Bee Colony) e Busca Tabu. Todos os algoritmos foram programados pelos autores utilizando seus respectivos artigos como base e os parâmetros utilizados foram selecionados com base em testes preliminares.

3.1. AEMT

O AEMT utiliza subpopulações de soluções distribuídas em tabelas no processo de otimização [Santos 2009]. Ele foi desenvolvido com o intuito de solucionar problemas multi-objetivos e com restrições. Cada tabela ou subpopulação do AEMT armazena as melhores soluções segundo cada objetivo e/ou restrição (se mensurável), sendo que uma das tabelas deverá conter as melhores soluções com base na função de agregação, onde serão agregados todos os objetivos e restrições, bem como a punição por violação destas [Santos 2009].

O Algoritmo 1 apresenta o pseudocódigo do AEMT. Primeiramente as tabelas são geradas a partir da aplicação da função de vizinhança (neste caso o PAO ou o CAO) de uma solução inicial. Em seguida um indivíduo é selecionado aleatoriamente de qualquer uma das tabelas e é aplicado a função de vizinhança nele para gerar uma solução candidata, as tabelas são ordenadas segundo cada objetivo e restrição e a solução candidata é comparada com as piores soluções de cada tabela. Caso seja melhor que uma delas, a solução candidata a substitui. Com isso, uma solução candidata poderá estar presente em mais de uma tabela.

São utilizadas cinco tabelas no AEMT, sendo uma para a função de agregação, uma para o número de manobras e outras três para as restrições do carregamento das linhas, carregamento das subestações e níveis de tensões.

3.2. ES-(M+L)

O ES é um dos primeiros algoritmos baseados na teoria da evolução, Ele foi criado por Rechenberg na década de 1960. Seus formatos mais usados são ES - $(\mu + \lambda)$ e ES (μ, λ) , onde o símbolo μ denota o número de pais de todas as gerações, e λ refere-se à quantidade de descendentes de gerados por μ [Back et al. 1991].

Algoritmo 1: Algoritmo Evolutivo Multiobjetivo por Tabelas

```
1 Gera a primeira solução factível;
2 Cria a população inicial de soluções;
3 Distribui as soluções da população inicial nas tabelas;
4 while Condição de parada não é satisfeita do
5   | Seleciona uma solução aleatória de uma tabela aleatória;
6   | Realiza a mutação na solução escolhida e gera uma nova solução;
7   for Tabelas do
8     | if Solução gerada for melhor que a pior solução da tabela then
9       | | Substitui a pior solução com a solução gerada;
10    | end
11  end
12 end
13 Retorna a melhor solução final
```

Os símbolos "+" e "," representa o tipo de seleção aplicado aos indivíduos gerados. De acordo com o [Beyer and Schwefel 2002], na seleção de tipo ",", μ pais geram λ descendentes, sendo $\lambda > \mu$ e a partir do λ filhos, os melhores μ serão selecionados para a próxima geração. Na seleção do tipo "+" λ filhos são gerados, contudo $\lambda < \mu$ e seleção dos melhores indivíduos será realizada com a união de filhos e pais de cada geração. Segundo [Beyer and Schwefel 2002] cada uma das formas de seleção de estratégias evolutivas tem sua aplicação característica. Tipo de seleção "," é mais indicado para aplicações com espaços de pesquisa ilimitados, e o tipo de seleção "+" é recomendado para espaços de busca discretos e finitos.

O Algoritmo 2 representa o fluxograma do ES- $(\mu + \lambda)$. O ES foi escolhido como representante das meta-heurísticas populacionais baseadas na teoria da evolução. Ele foi escolhido no lugar do Algoritmo Genético pois é possível utilizá-lo apenas com a mutação (neste caso o PAO e o CAO), sem a etapa de cruzamento, o que é atualmente impossível de se realizar utilizando a RNP sem gerar soluções infactíveis.

Algoritmo 2: Estratégias Evolutivas $\mu + \lambda$

```
1 Gera uma população inicial de  $\mu$  soluções;
2 while Condição de parada não é satisfeita do
3   | Realiza a mutação nos  $\mu$  pais, gerando  $\lambda$  filhos ;           /*  $\mu + \lambda$  */
4   | Uni os  $\mu$  pais com os  $\lambda$  filhos numa única população;
5   | Ordena a população;
6   | Seleciona as  $\mu$  melhores soluções;
7 end
8 Retorna a melhor solução final
```

3.3. Busca Tabu

A Busca Tabu foi proposta por Glover em 1986 [Glover 1989]. Ela é uma metaheurística de busca local onde as alterações que proporcionam novas soluções são armazenadas

numa lista, e caso uma nova solução apresente estas alterações, ela será descartada. Como mostra o Algoritmo 3. A lista tabu possui um número máximo de alterações que poderão ser armazenadas, quando ela ficar totalmente preenchida, as alterações mais antigas são removidas da lista e dão lugar a novas alterações. Neste trabalho, a lista tabu será composta das chaves que foram modificadas na utilização do PAO/CAO.

Algoritmo 3: Busca Tabu

```
1 Gera a primeira solução factível;
2 MelhorSolução = primeira solução factível;
3 SoluçãoAtual = primeira solução factível;
4 while Condição de parada não é satisfeita do
5   | Gera a vizinhança;
6   | Ordena a vizinhança;
7   | if Melhor vizinho for a melhor solução encontrada then
8   |   | MelhorSolução = Vizinho;
9   |   end
10  | for Vizinho do
11  |   | if Vizinho não está na lista tabu then
12  |   |   | SoluçãoAtual = Vizinho;
13  |   |   | Atualiza a lista tabu;
14  |   |   | break;
15  |   |   end
16  |   end
17 end
18 Retorna a melhor solução final
```

3.4. ABC

O ABC (*Artificial Bee Colony*) é um algoritmo de inteligência de enxames desenvolvido por [Karaboga 2005]. O algoritmo ABC é apresentado no Algoritmo 4. Ele é constituído basicamente uma sequência de gerações de vizinhança e seleções, alternando entre gulosa e por torneio ou roleta. Por último, é realizada uma etapa de exploração, onde é verificada se as soluções presentes em O estão a um determiando número preestabelecido de iterações sem melhora, ou seja, se ela não for substituído por um vizinho nas seleções gulosas esta solução é descartada e outra solução é gerada aleatoriamente

Um dos principais pontos do ABC está na sua fácil implementação e dos poucos parâmetros presentes, basicamente só será necessário definir o número de soluções presentes em O e o número de iterações que uma solução poderá ser armazenada no algoritmo [Shukran et al. 2011].

4. Resultados Computacionais

4.1. Sistemas Teste

Para a comparação entre os algoritmos, serão realizados testes no Sistema de Distribuição da Taiwan Power Company (TPC) modificado, com 94 barras [Su and Lee 2003], e no

Algoritmo 4: ABC

```
1 Gera a primeira solução factível;  
2 MelhorSolução = primeira solução factível;  
3 O = Inicia a população de soluções;  
4 while Condição de parada não é satisfeita do  
5   O' = Função de vizinhança aplicada em O;  
6   B = Seleção gulosa entre O e O';  
7   V = Seleção por torneio ou roleta em B;  
8   V' = Função de vizinhança aplicada em V;  
9   F = Seleção gulosa entre V e V';  
10  Armazena a melhor solução encontrada;  
11  if Alguma solução em O não apresentou melhora then  
12    | Substitui por uma solução aleatória;  
13  end  
14 end  
15 Retorna a melhor solução final
```

sistema real de 143 barras [Mantovani et al. 2000], como mostra as Figuras 1 e 2, respectivamente. As faltas serão simuladas realizando interrupções nas barras 18 e 36 para o sistema de 94 barras e nas barras 22 e 107 no caso de 143 barras.

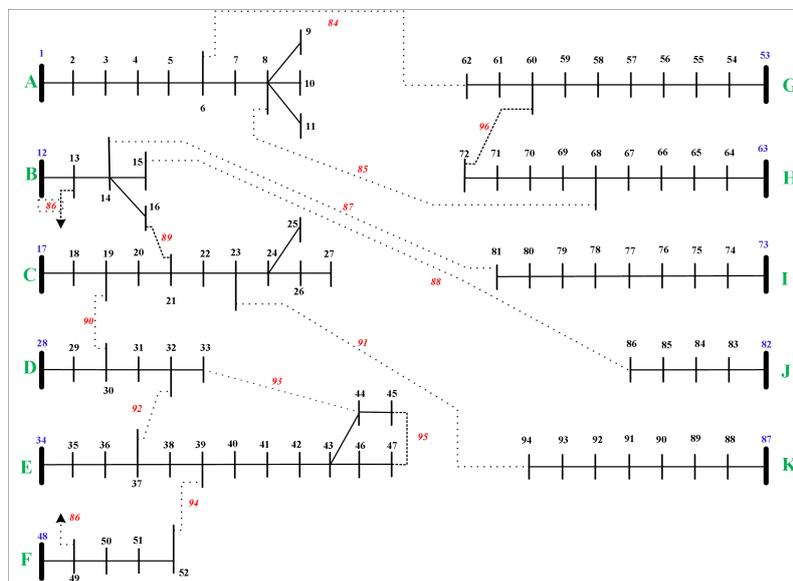


Figure 1. Sistema de 94 barras

Os testes foram executados num computador com processador Intel Core i5-3210-M de 3.1 GHz e com 6 GB de memória RAM. Os algoritmos foram programados utilizando a linguagem Python 2.7.14. Para cada algoritmo, foram realizadas 30 execuções e retirada a média de cada execução.

Os parâmetros utilizados na avaliação dos algoritmos são a função de agregação, a qual engloba o objetivo principal (número de manobras) e todas as restrições e suas penalidades, e o tempo de execução. Com relação ao tempo de execução, o processo

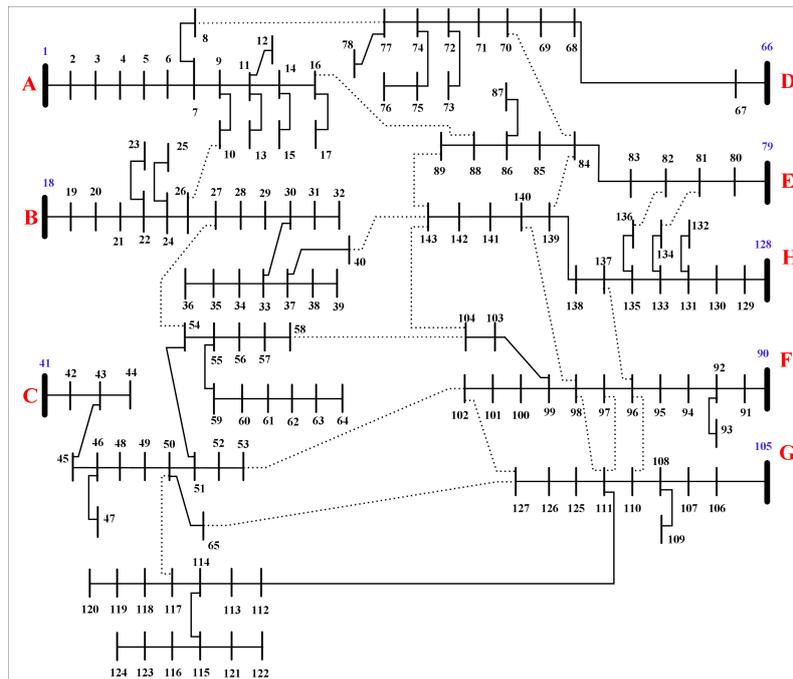


Figure 2. Sistema de 143 barras

com maior custo computacional é o cálculo do fluxo de carga, sendo esta uma etapa indispensável por retornar diversos parâmetros da rede elétrica, como o carregamento das linhas e o nível de tensão nas barras, logo o tempo de execução fica atrelado ao número de vezes em que um algoritmo executa o cálculo do fluxo de carga. Por isso para avaliar o tempo computacional optou-se por atribuir um número fixo de vezes em que os algoritmos executarão o fluxo de carga através da limitação do número de soluções manipuladas, sendo assim os algoritmos mais rápidos irão convergir para um ótimo local com um número menor de soluções manipuladas.

4.2. Análise de Desempenho

Nas Figuras 3 e 4 são exibidos os resultados das execuções dos algoritmos para os sistemas de 94 e 143 barras. A Figura 3 apresenta a evolução dos algoritmos de acordo com o número de soluções avaliadas, sendo exibida a média das 30 execuções e a Figura 4 mostra o diagrama de caixa para os resultados finais de cada algoritmo.

No primeiro caso, apresentado na Figura 3(a), observa-se que o AEMT apresentou um resultado superior aos outros algoritmos, convergindo para um ótimo com menos de 3000 soluções avaliadas. Também observa-se que o ABC não apresentou uma convergência, indicando que se este algoritmo avaliar mais soluções talvez ele também consiga convergir para um ótimo local. Analisando as soluções finais deste caso, apresentada na Figura 4 (a), o AEMT continua apresentando uma superioridade em relação aos outros três algoritmos, contudo o ES e o ABC apresentarão soluções com resultados bem próximos as soluções do AEMT.

No Segundo caso, apresentado na Figura 3(b), o AEMT continuou demonstrando um desempenho melhor que os outros três algoritmos e o ABC continuou não convergindo, contudo o ES conseguiu apresentar uma evolução melhor que o ABC, con-

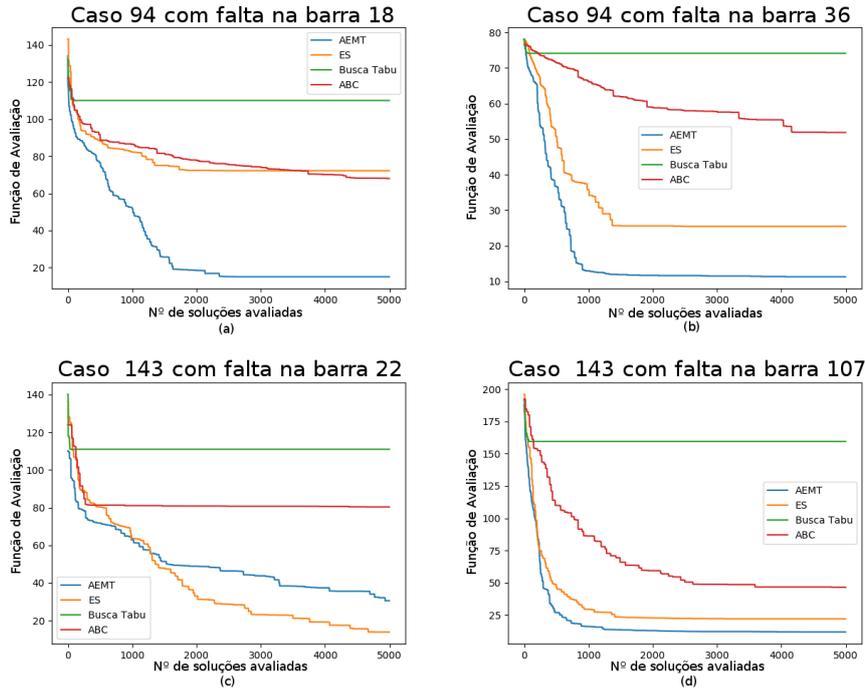


Figure 3. Convergência das execuções

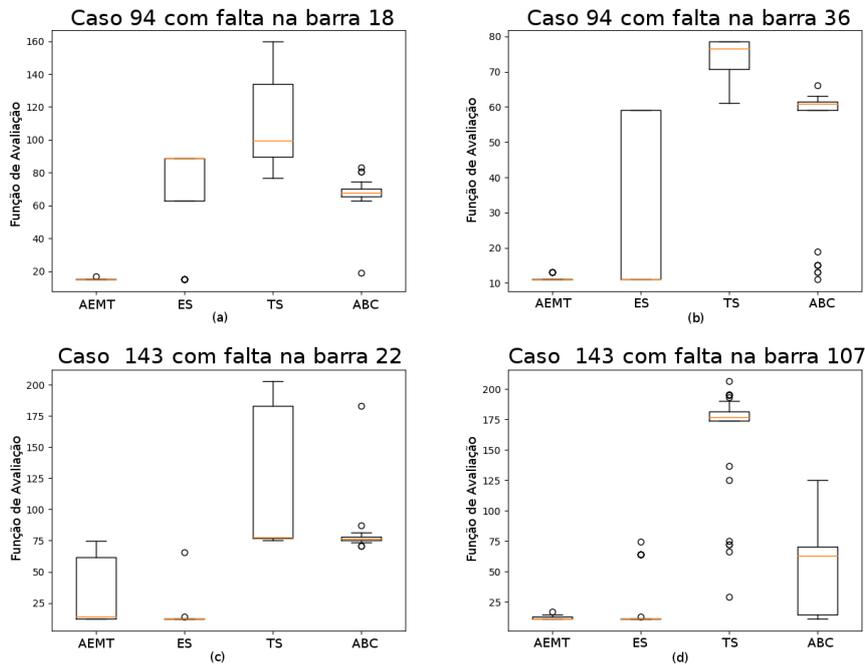


Figure 4. Diagramas de caixa

vergingo antes dele e demonstrando resultados próximos ao AEMT. Analisando as soluções finais deste caso, apresentada na Figura 4 (b), o AEMT continua apresentando uma superioridade em relação aos outros três algoritmos, porém desta vez o ES e o ABC apresentarão algumas soluções com valores iguais aos do AEMT, com destaque para o ES que apresentou um número maior de soluções com valores próximos ao do AEMT.

No Terceiro caso, apresentado na Figura 3(c), o ES apresentou um melhor desempenho que os outros três. Desta vez o ABC convergiu prematuramente e tanto o ES quanto o AEMT não convergiram indicando, assim como o ABC nos casos anteriores, que um aumento no número de soluções possa levar esses dois algoritmos a uma convergência para um ótimo local. Na análise das soluções finais, apresentada na Figura 4 (c), fica evidente a superioridade do ES neste caso em comparação ao AEMT, onde grande parte das soluções encontradas nas execuções possuem valores iguais, possivelmente convergindo para um ótimo local.

No último caso, apresentado na Figura 3(d), o AEMT volta a apresentar resultados superiores aos outros algoritmos, convergindo para um ótimo com menos de 2000 soluções avaliadas e o ABC continuou apresentando as mesmas características dos dois primeiros casos. Com relação à análise das soluções finais, o AEMT e o ES apresentaram resultados bem próximos, seguidos pelo ABC e pelo TS.

De modo geral o AEMT foi o algoritmo que apresentou os melhores resultados, tanto em relação à função objetivo quando ao tempo de execução, pois ele foi o algoritmo que nos casos (a), (b) e (d) convergiu para um ótimo local com o menor número de soluções avaliadas. Isto provavelmente ocorre pois o AEMT apresenta uma exploração mais abrangente do espaço de busca, isto também explica o motivo do TS apresentar os piores resultados, pois ele se trata de um algoritmo de busca local. Mesmo o ES não apresentando resultados satisfatórios, com algumas modificações adicionando uma maior exploração no espaço de busca, possivelmente ele apresente resultados iguais ou melhores que o AEMT. Também pode-se observar que, dependendo do local da falta, a complexidade do algoritmo pode aumentar significativamente, como ocorreu no terceiro caso, onde não houve um algoritmo, dos avaliados, que convergiu para um ótimo local.

5. Conclusão

Este trabalho apresentou uma análise comparativa entre 4 meta-heurísticas aplicadas ao problema de restauração de redes elétricas. Mostrou-se que é possível ter bons resultados utilizando outras meta-heurísticas aplicadas ao problema. As análises neste artigo também mostraram que a complexidade do problema está diretamente relacionado ao local da falta nas redes elétricas, sendo necessário utilizar algoritmos que possam lidar com tal complexidade.

Futuramente pretende-se realizar modificações nas ES para obter melhores resultados, e também incluir outras restrições presentes da rede elétrica, como a análise da proteção dos dispositivos presentes nela e a priorização de carga na rede elétrica.

References

- Augugliaro, A., Dusonchet, L., and Sanseverino, E. R. (2000). Multiobjective service restoration in distribution networks using an evolutionary approach and fuzzy sets. *International Journal of Electrical Power & Energy Systems*, 22(2):103–110.
- Back, T., Hoffmeister, F., and Schwefel, H.-P. (1991). A survey of evolution strategies. In *Proceedings of the fourth international conference on genetic algorithms*, volume 2. Morgan Kaufmann Publishers San Mateo, CA.
- Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strategies—a comprehensive introduction. *Natural computing*, 1(1):3–52.

- Cabello, A. F. (2012). Redes elétricas inteligentes no brasil: a necessidade de uma avaliação adequada de custos e benefícios.
- Delbem, A., Bretas, N., and De Carvalho, A. (1998). Energy restoration in distribution systems using search with fuzzy heuristics. *INT J ENG INTELL SYST ELECTIC ENG COMMUN*, 6(4):201–205.
- Gendreau, M., Potvin, J.-Y., et al. (2010). *Handbook of metaheuristics*, volume 2. Springer.
- Glover, F. (1989). Tabu search—part i. *ORSA Journal on computing*, 1(3):190–206.
- Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
- Kwong, S., Chan, T.-M., Man, K.-F., and Chong, H. (2002). The use of multiple objective genetic algorithm in self-healing network. *Applied Soft Computing*, 2(2):104–128.
- Mansour, M. R. (2009). *Algoritmo para obtenção de planos de restabelecimento para sistemas de distribuição de grande porte*. PhD thesis, Universidade de São Paulo.
- Mantovani, J. R., Casari, F., and Romero, R. A. (2000). Reconfiguração de sistemas de distribuição radiais utilizando o critério de queda de tensão. *Controle and Automacao*, pages 150–159.
- Marques, L. T. (2013). *Restabelecimento de energia por reconfiguração de redes em sistemas de distribuição de grande porte com priorização de chaves, consumidores e definição de sequência de chaveamento*. PhD thesis, Universidade de São Paulo.
- Sanches, D., JR Mazucato, S., Castoldi, M., Delbem, A., and London, J. (2013). Algoritmo evolutivo multi-objetivo baseado em tabelas de subpopulações de soluções não dominadas e força de pareto para tratar do problema de restabelecimento de energia em sistemas de distribuição. pages 1–6.
- Sanches, D., Mansour, M., London, J., Delbem, A., and Santos, A. (2011). Integrating relevant aspects of moeas to solve loss reduction problem in large-scale distribution systems. In *2011 IEEE Trondheim PowerTech*, pages 1–6. IEEE.
- Sanches, D. S. (2012). *Algoritmos evolutivos multi-objetivo para reconfiguração de redes em sistemas de distribuição de energia elétrica*. PhD thesis, Universidade de São Paulo.
- Santos, A. C. d. (2009). *Algoritmo evolutivo computacionalmente eficiente para reconfiguração de sistemas de distribuição*. PhD thesis, Universidade de São Paulo.
- Shukran, M. A. M., Chung, Y. Y., Yeh, W.-C., Wahid, N., and Zaidi, A. M. A. (2011). Artificial bee colony based data mining algorithms for classification tasks. *Modern Applied Science*, 5(4):217.
- Su, C.-T. and Lee, C.-S. (2003). Network reconfiguration of distribution systems using improved mixed-integer hybrid differential evolution. *IEEE Transactions on Power Delivery*, 18(3):1022–1027.
- Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons.