

LUCIANA MACEDO DE OLIVEIRA

**Reconfiguração de sistemas de
distribuição de energia elétrica para
minimização de perdas ativas totais
via otimização por enxame de
partículas (PSO)**

Trabalho de Conclusão de Curso apresentado
à Escola de Engenharia de São Carlos, da
Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase
em Sistemas de Energia e Automação

ORIENTADOR: Prof. Dr. Eduardo Nobuhiro Asada

São Carlos
2012

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

Oliveira, Luciana Macedo
048r Reconfiguração de sistemas de distribuição de
energia elétrica para minimização de perdas ativas
totais via otimização por enxame de partículas (PSO) /
Luciana Macedo Oliveira; orientador Eduardo Nobuhiro
Asada. São Carlos, 2012.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Sistemas de Energia e Automação) -- Escola de
Engenharia de São Carlos da Universidade de São Paulo,
2012.

1. Reconfiguração. 2. Sistemas de Distribuição. 3.
Minimização de Perdas Ativas. 4. Metaheurística. 5.
Otimização por Enxame de Partículas. I. Título.

FOLHA DE APROVAÇÃO

Nome: Luciana Macedo de Oliveira

Título: “Reconfiguração de Sistemas de Distribuição de Energia Elétrica para Minimização de Perdas Ativas via Otimização por Enxame de Partículas (PSO)”

*Trabalho de Conclusão de Curso defendido e aprovado
em 28 / 11 / 2012,*

com NOTA 10 (Dez, Zero), pela Comissão Julgadora:

*Prof. Dr. Eduardo Nobuhiro Asada (Orientador)
SEL/EESC/USP*

*Prof. Titular Geraldo Roberto Martins da Costa
SEL/EESC/USP*

*Prof. Dr. Rogério Andrade Flauzino
SEL/EESC/USP*

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Homero Schiabel

Dedicatória:

Aos meus pais, verdadeiros heróis, que
inúmeras vezes se sacrificaram para ver
o sonho de suas filhas serem
alcançados.

AGRADECIMENTOS

Aos meus pais, José Antônio e Livia Maria, às minhas irmãs Marina e Roberta, aos meus sobrinhos Murilo e Mariana, por estarem sempre presente em minha vida, me apoiando em todos os momentos.

Aos Professores Dr. Eduardo Nobuhiro Asada e Dr. Geraldo Roberto Martins da Costa pela excelente orientação no desenvolvimento deste trabalho e dedicação em repassar seu vasto conhecimento na área de sistemas elétricos de potência.

Aos amigos do LASEE, Fernando, Fillipe, Eduardo e Wellington, pelo apoio e companheirismo ao longo destes anos de Iniciação Científica e Trabalho de Conclusão de Curso.

Aos amigos da USP, pela presença e apoio em todos estes anos de graduação.

Aos amigos do Anexo STI, Roberto Ragonesi, Alex, Leandro, Francisco, Talles, Amanda, Pedro, Carol, Yuri, Dani, Ricardo, André e Itamar pela oportunidade de trabalho e compreensão nos momentos difíceis da graduação.

Aos amigos da CPFL, pelo conhecimento e experiência profissional que me proporcionaram.

À todos aqueles que, de alguma forma, contribuíram de forma positiva para a conclusão desta etapa da minha vida.

RESUMO

OLIVEIRA, L. M., Reconfiguração de Sistemas de Distribuição de Energia elétrica para Minimização de Perdas Ativas Totais via Otimização por Enxame de Partículas (PSO). 2012. Trabalho de conclusão de curso - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2012.

O objetivo deste trabalho é apresentar uma nova abordagem para a obtenção de configurações radiais para o Sistema de Distribuição de Energia Elétrica (SDEE) com o intuito de minimizar o valor das perdas ativas totais, sem violar as restrições operacionais. Propõe-se a utilização de uma metaheurística (PSO - *Particle Swarm Optimization*) como método de solução para o problema analisado. Para o cálculo das perdas ativas totais, este trabalho também apresenta uma abordagem para a resolução do problema de Fluxo de Potência (FP) em sistemas de distribuição radial.

Palavras chave: Reconfiguração. Sistemas de distribuição. Minimização de perdas ativas. Metaheurística. Otimização por enxame de partículas.

ABSTRACT

OLIVEIRA, L. M., Reconfiguration of Electric Power Distribution Systems for Minimization of the Total Active Power Losses using Particle Swarm Optimization (PSO). 2012. Final Paper – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2012.

The objective of this paper is to present a new approach for obtaining radial configurations for Distribution Networks (DN) in order to minimize the total amount of active losses without violating the operating constraints. It is proposed a metaheuristic method (PSO - Particle Swarm Optimization) as a solution to the problem analyzed. To calculate the total active losses, this paper also presents an approach for solving the problem of Power Flow (PF) in radial distribution systems.

Keywords: Reconfiguration. Distribution systems. Active losses minimization. Metaheuristic. Particle swarm optimization.

SUMÁRIO

1. INTRODUÇÃO	8
1.1. Metodologia	8
1.2. Organização do Trabalho	8
2. REVISÃO BIBLIOGRÁFICA	9
2.1. Introdução	9
2.2. Metodologias	9
2.2.1. Heurísticas	9
2.2.2. Metaheurísticas	11
2.2.2.1. Inteligência Artificial (IA).....	11
2.2.3. Otimização Clássica	12
2.3. Considerações Finais	12
3. FLUXO DE POTÊNCIA	14
3.1. Introdução	14
3.2. O Método Backward/Forward Sweep (BFS)	16
3.3. Resultados	19
3.3.1. Sistema de 10 Barras	19
3.3.2. Sistema de 34 Barras	20
3.3.3. Sistema de 70 Barras	21
3.4. Considerações Finais	23
4. OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS	25
4.1. Introdução	25
4.2. PSO Clássico	25
4.3. Algoritmo PSO Binário.....	29
5. ALGORITMO PROPOSTO PARA RECONFIGURAÇÃO DE SISTEMAS DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA	34
5.1. Introdução	34
5.2. Reconfiguração de Sistemas de Distribuição via PSO	35
5.2.1. Carregamento de Dados e Montagem da <i>Matriz A</i>	39
5.2.2. Algoritmo PSO Binário para RSDEE.....	41
5.2.3. Metodologia para Avaliação da Conectividade da Rede	50
5.3. Considerações Finais	54
6. RESULTADOS	56
7. CONCLUSÃO	59
REFERÊNCIAS BIBLIOGRÁFICAS	60
APÊNCIDE A	62

1. INTRODUÇÃO

A configuração dos Sistemas de Distribuição de Energia Elétrica (SDEE) é originalmente malhada, porém o sistema opera com topologia radial. A reconfiguração consiste na determinação de uma topologia radial conexa da rede, mediante a permutação dos estados (aberto ou fechado) das chaves manobráveis acopladas aos sistemas de distribuição. Este procedimento é normalmente utilizado para o balanceamento de cargas entre alimentadores, melhoria dos níveis de tensão, isolamento de faltas ou para minimização das perdas ativas do sistema.

1.1. Metodologia

O problema da Reconfiguração de Sistemas de Distribuição de Energia Elétrica (RSDEE) é um problema de programação não-linear inteira mista, caracterizado pela explosão combinatória devido à grande dimensão do espaço de soluções. Sendo assim, sua resolução por técnicas de otimização clássicas apresenta dificuldades devido ao grande número de soluções, dando espaço às técnicas heurísticas e metaheurísticas (PEREIRA, 2010). Portanto, será utilizado neste trabalho, um algoritmo baseado na metaheurística PSO (*Particle Swarm Optimization*) para a solução do problema analisado.

As perdas ativas totais dos sistemas radiais serão obtidas através do método *Back-forward Sweep*, um algoritmo de varredura para cálculo de fluxo de potência em sistemas de distribuição radial.

A implementação destas soluções foi feita a partir de programação em linguagem C, utilizando o *software* Microsoft Visual Studio®.

1.2. Organização do Trabalho

A primeira parte deste trabalho consistiu na elaboração das rotinas computacionais para a busca de soluções do problema de RSDEE e cálculo de fluxo de carga. Concluído o programa, foram realizadas simulações a partir de sistemas testes, cujos resultados são encontrados em dissertações e trabalhos acadêmicos publicados.

2. REVISÃO BIBLIOGRÁFICA

2.1. Introdução

A Reconfiguração de Sistemas de Distribuição de Energia Elétrica (RSDEE) consiste em uma ferramenta de planejamento e controle, em tempo real, da operação dos Sistemas Elétricos de Potência (SEP). RSDEE visa a obtenção de topologias de rede que acarretam em menores perdas ativas ao sistema, melhorias no perfil de tensão nas barras de carga (consumidores) mediante o balanceamento de carga entre os alimentadores, aumento dos níveis de confiabilidade do sistema e eliminação ou isolamento de faltas de energia a fim de restabelecer seu fornecimento em áreas prejudicadas. A RSDEE, visando a redução de perdas ativas nos alimentadores, consiste em uma solução eficaz e de baixo custo quando comparado a outras alternativas de mesmo propósito.

O problema da reconfiguração possui natureza combinatória, sendo dependente do número de chaves existentes no sistema. São utilizadas diversas técnicas para sua solução, diferindo-se entre elas pela estratégia matemática/computacional utilizada no tratamento do problema.

O número de configurações possíveis de um sistema é dado por 2^{nc} , onde nc é o número de chaves do sistema. Sendo assim, para sistemas portadores de grande número de chaves (sistemas reais), ocorre uma explosão combinatória, dificultando ou até mesmo impossibilitando sua solução por técnicas de otimização clássicas. Neste contexto, as técnicas metaheurísticas ganham espaço na literatura, apresentando resultados e eficiência computacional satisfatórios.

2.2. Metodologias

2.2.1. Heurísticas

Heurística consiste em uma técnica de otimização baseada em uma regra ou um conjunto de regras, cujo objetivo é a busca de boas soluções, em um tempo computacional razoável. Heurísticas nem sempre garantem a otimalidade da solução,

pois consiste em uma técnica aproximativa. As heurísticas são classificadas em três tipos: construtivas, de melhoria (busca local) e metaheurísticas. Neste tópico são apresentados os conceitos das heurísticas tipo construtivas e de melhoria sendo, posteriormente, dedicado um tópico para a explanação das metaheurísticas.

Heurísticas construtivas constroem a solução de um problema de forma incremental. Adicionando componentes individuais (nós, arcos, rotas, variáveis) - *Método Guloso* - procura o melhor componente. Passo a passo, um componente é escolhido e depois inserido na solução até gerar uma solução completa. O componente escolhido é, em geral, o melhor candidato de acordo com algum critério preestabelecido.

Heurística de melhoria ou busca local consiste em, partindo de uma solução inicial factível, "navegar" iterativamente pelo espaço de busca movendo-se a cada passo, da solução atual para uma solução vizinha (adjacente). É um método baseado na tentativa e erro que visa encontrar ótimos locais de maneira simples, não garantindo a otimalidade local da solução encontrada, embora seja possível.

Um dos primeiros trabalhos encontrados na literatura (MERLIN e BACK, 1975) apresenta dois métodos distintos para resolver o problema de reconfiguração. Uma delas consiste na utilização de um algoritmo heurístico construtivo, cuja metodologia consiste em partir de um sistema radial, fechar todas as chaves existentes, tornando-o malhado e, a partir de um cálculo de fluxo de potência do sistema, faz-se a abertura sucessiva das chaves de interconexão com menor fluxo de potência.

Em 1988 foi proposto um método heurístico por Civanlar et. al (1988), denominado *branch-exchange*, que parte de uma rede radial como solução inicial do problema. Neste método, é mantida a radialidade do sistema, uma vez que se alterna a abertura de uma chave com o fechamento de outra mediante mecanismo de filtragem proposto pelos autores, onde são eliminados os chaveamentos que não promovem redução de perdas no sistema (OLIVEIRA, 2011).

Em 1989, o algoritmo proposto por Merlin e Back (1975) foi modificado por Shirmohammadi e Hong (1989). Foram realizadas modificações tais como: inserção dos limites de tensão nos barramentos, limite de correntes nas linhas e cálculo do fluxo de carga para as energias reativas (PEREIRA, 2010).

Também, em 1989, deu-se origem à uma nova metodologia (BARAN e WU, 1989) obtida a partir da modificação do método de Civanlar et. al (1988). Nesta

modificação, seus autores aprimoraram o mecanismo de troca de ramos e, formularam também, métodos de resolução de fluxo de carga para redes radiais.

Em 1992 o algoritmo de Shirmohammadi e Hong (1989) foi modificado, dando origem a um novo método (GOSWAMI e BASU, 1992). Nesta metodologia, parte-se de um sistema radial, onde apenas uma chave é fechada, formando um único laço no sistema. A partir do cálculo de fluxo de potência, determina-se qual ramo será retirado do sistema, sendo aquele que apresenta menor fluxo de potência. Foi mostrado que esta metodologia apresenta resultados melhores que a de Shirmohammadi e Hong (1988) (GUIMARÃES, 2005).

2.2.2. Metaheurísticas

Metaheurística é uma combinação de métodos heurísticos que podem ser aplicados em problemas gerais de alta complexidade e espaço de soluções elevado. São usualmente aplicadas para buscas locais, porém suas características de busca inteligente, em muitos casos, permitem explorar novas regiões do espaço, podendo alcançar ótimos globais de maneira eficiente.

São encontradas na literatura diversas abordagens para o problema de RSDEE, utilizando-se metaheurísticas das mais variadas classificações. Dentre as quais, podemos citar:

- *Simulated Annealing* (CHIANG and JEAN-JUMEAU, 1990);
- *Greedy Randomized Adaptive Search Procedure* (GRASP) (OLIVEIRA, 2011);
- Busca Tabu (GUIMARÃES, 2005).

2.2.2.1. Inteligência Artificial (IA)

Inteligência Artificial (IA) é uma metaheurística de busca. Dentro desta classificação, tem-se uma grande variedade de métodos de otimização, os quais são baseados em comportamentos de seres vivos (evolução, adaptação, aprendizado, raciocínio, etc.). Dentre os métodos de IA encontrados na literatura para RSDEE, podemos citar:

- Redes Neurais Artificiais (SALAZAR et al., 2006);
- Algoritmo Genético (QUEIROZ et al., 2006);
- Algoritmo Evolutivo (PEREIRA, 2010).

2.2.3. Otimização Clássica

Métodos de otimização clássica são fundamentados no cálculo diferencial, onde, a partir de um ponto inicial, faz-se o cálculo do gradiente para se determinar a direção de busca e encontrar a próxima solução a ser avaliada.

A segunda proposta apresentada por Merlin e Back (1975) utiliza uma otimização clássica discreta, cuja metodologia consiste em partir de uma rede com topologia malhada (todas as chaves fechadas) e, utilizando a técnica *Branch and Bound*, realiza-se a abertura sucessiva de chaves até alcançar uma configuração radial com perdas ativas totais mínimas.

Branch and Bound consiste em um algoritmo, aplicado especialmente em problemas de otimização discreta e combinatória, que utiliza limites inferiores e superiores da função objetivo para filtragem de soluções indesejáveis no espaço de busca, até que se encontre a melhor/ótima.

2.3. Considerações Finais

Objetivando uma breve explanação da evolução nos estudos de reconfiguração, foram apresentados neste capítulo, alguns dos principais métodos utilizados na tratativa do problema em questão. Há diversas outras metodologias aplicadas à RSDEE, sendo em sua maioria, modificações e/ou aperfeiçoamentos de métodos já implementados.

A evolução das técnicas aplicadas ao problema de reconfiguração se deu, principalmente, ao aumento da dimensão e da complexidade dos sistemas elétricos, que estão ainda em contínua expansão. Neste contexto, foi sendo necessária a criação de métodos computacionalmente eficientes aliados à redução do tempo para obtenção das soluções, uma vez que se faz necessária a reconfiguração dos sistemas em tempo real, para atendimento às necessidades de operação.

Nota-se que cada metodologia possui sua particularidade, seja na maneira inicial de tratamento do problema ou no método de busca empregado para obtenção de soluções. Porém, a mudança de topologia da rede, mediante abertura ou fechamento de chaves de interconexão, com o objetivo, neste caso, de reduzir perdas ativas nos alimentadores, deve sempre respeitar as restrições operativas do sistema.

Neste trabalho, é apresentada uma metodologia para solução do problema de reconfiguração mediante aplicação de metaheurística baseada na evolução de seres vivos, que se enquadra na subdivisão "*computação evolutiva*". Otimização por Enxame de Partículas ou PSO (Particle Swarm Optimization) consiste em um algoritmo evolutivo baseado na inteligência de enxames, proposto inicialmente para problemas contínuos e que, contudo, foi utilizado neste trabalho uma versão binária para o tratamento do problema de RSDEE com o objetivo de minimização de perdas.

3. FLUXO DE POTÊNCIA

3.1. Introdução

O Fluxo de Potência (FP) ou fluxo de carga consiste em uma ferramenta básica para análises de Sistemas Elétricos de Potência (SEP). O cálculo do FP possui grande importância neste contexto, sendo muito utilizado por engenheiros atuantes em sistemas de energia elétrica, pois possibilita determinar o estado das redes elétricas sob o ponto de vista do planejamento, bem como a obtenção de melhor desempenho da operação dos sistemas em funcionamento.

Para o cálculo do FP é feita a modelagem dos sistemas de maneira estática, uma vez que as variações ocorridas nos sistemas são lentas o suficiente para que os efeitos transitórios sejam desprezados. Desta forma, é possível representar a rede através de equações e inequações algébricas não-lineares, não sendo necessário seu tratamento por equações diferenciais (MONTICELLI, 1983). O cálculo do FP tem como princípio básico a conservação da energia no sistema analisado, ou seja, assegura que a potência injetada no sistema seja igual às potências consumidas nas barras de carga somadas às perdas nos alimentadores, sendo para isto, utilizadas as Leis de Kirchhoff em sua resolução.

Em geral, dado um sistema elétrico, representado por subestação, alimentadores e barras de carga, por meio do cálculo do FP, podem ser obtidos valores de potência ativa e reativa, fator de potência, fasores de corrente e tensão elétrica em cada barra de carga, assim como corrente elétrica e perdas ativas nos alimentadores.

A formulação básica do problema pode ser feita de várias maneiras, diferenciando-se, entre elas, apenas pela definição de quais variáveis entram como dados e quais entram como incógnitas. A seguir, estão listadas na Tabela 3.1 as variáveis básicas existentes no problema e, na Tabela 3.2, o tipo de formulação utilizada neste trabalho para o cálculo do FP.

Tabela 3.1: Variáveis do Fluxo de Potência

Variáveis	Descrição
V_k	Módulo do fasor de tensão na barra de carga k.
Θ_k	Ângulo do fasor de tensão na barra de carga k.
P_k	Potência ativa consumida pela barra de carga k.
Q_k	Potência reativa consumida pela barra de carga k.

Tabela 3.2: Formulação do Fluxo de Potência

Tipo de Formulação Utilizada	Dados	Incógnitas
Barra PQ	P_k e Q_k	V_k e Θ_k

O valor V_{ref} caracteriza a tensão na saída da subestação (fonte de corrente do sistema), geralmente definida como *barra 1* (barra de referência ou *slack*), cuja tensão é controlada e há somente injeção de potência para o circuito, isto é, não há consumo de potência ativa e reativa neste ponto.

Há diversas metodologias empregadas para o cálculo do FP, desenvolvidas por diversos autores, sendo elas versões clássicas ou mesmo adaptações destas, em alguns casos. A aplicabilidade dos diversos métodos existentes está, muitas vezes, relacionada com a topologia do sistema, isto é, há métodos que, quando aplicados a sistemas de distribuição de energia, apresentam resultados satisfatórios, mas que, em contrapartida, quando aplicados a sistemas de transmissão, podem ser ineficazes. Dentro deste contexto, destaca-se o método *Backward/Forward Sweep* (BFS), muito utilizado no cálculo do FP em Sistemas Radiais de Distribuição de Energia Elétrica (SRDEE) devido à eficácia na obtenção de resultados e alta velocidade de convergência.

Neste capítulo será apresentado o método BFS, que foi implementado computacionalmente neste trabalho para o cálculo do fluxo de potência monofásico das topologias radiais encontradas no processo de reconfiguração.

3.2. O Método Backward/Forward Sweep (BFS)

Será utilizado *Backward/Forward Sweep* para cálculo do fluxo de potência, um método que trata exclusivamente de SRDEE, proposto por D. Shirmohammadi et al. (1988), que constitui de duas etapas de varredura do sistema para o cálculo do FP. Utilizando a formulação do tipo *Barra PQ* é possível realizar o cálculo do fluxo de potência, uma vez que se têm os valores de potência ativa e reativa de cada barra de carga, impedância das linhas que alimentam as cargas e a tensão de referência.

A seguir, é mostrado o esquemático do trecho de um dado um SRDEE de n barras e l linhas.

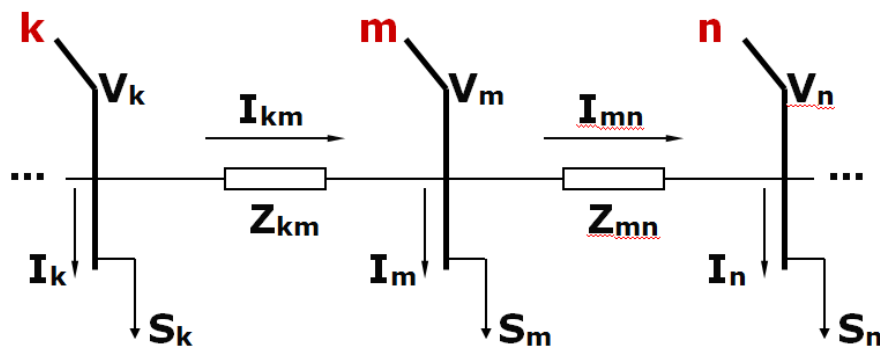


Figura 3.1 – Trecho de SRDEE

Tabela 3.3 – Legenda da Figura 3.1

Representação	Descrição
k, m, n	Barras de carga (ou nós).
km, mn	Linhas que interligam as barras k e m ; m e n , respectivamente.
V_k, V_m, V_n	Tensão nas barras k, m e n , respectivamente.
I_k, I_m, I_n	Corrente nas barras k, m e n , respectivamente.
S_k, S_m, S_n	Potência complexa das cargas k, m e n , respectivamente.
Z_{km}, Z_{mn}	Impedância das linhas km e mn , respectivamente.
I_{km}, I_{mn}	Corrente que flui nas linhas km e mn , respectivamente.

Sendo:

$$S_k = P_k + jQ_k \text{ (VA)} \quad (3.1)$$

$$Z_{km} = R_{km} + jX_{km} \text{ (}\Omega\text{)} \quad (3.2)$$

Os valores das resistências e reatâncias de cada linha, bem como as potências ativas e reativas de cada barra de carga, são dados do problema.

Inicialmente, adota-se a tensão de referência (V_{ref}) em todas as barras de carga, isto é, faz-se $V_k = V_{ref}$ para todos os nós do sistema. Este procedimento consiste em atribuir um valor inicial de tensão em todos os nós, para posterior ajuste através do princípio da conservação de energia, realizada a cada iteração do método. Sendo assim, tem-se inicialmente um total nulo de perdas já que, neste momento, não há quedas de tensão.

Após realizadas as considerações iniciais supracitadas (iteração $i=0$), a primeira etapa iterativa do método BFS consiste na aplicação dos itens 1 e 2, como segue:

1. Calcular a corrente em cada barra de carga, a partir dos valores de potência complexa S_k e pela tensão V_k .

$$I_k = \left(\frac{S_k}{V_k} \right)^* \quad (3.3)$$

2. Calcular a corrente que flui em cada linha no sentido inverso do fluxo de potência (*Backward*), isto é, iniciando pelos extremos do alimentador, calcula-se a corrente em cada linha, através da aplicação da 1ª Lei de Kirchhoff sequencialmente em cada nó, em direção à subestação:

$$I_{mn} = I_{km} - I_m \quad (3.4)$$

A segunda etapa do método consiste na aplicação dos itens 3 e 4, como segue:

3. Calcula-se as perdas totais nos alimentadores desta iteração ($i+1$):

$$(Perdas\ Totais)_i = \sum R_{km} \cdot (I_{km})^2 \quad (3.5)$$

4. A partir de um valor de tolerância pré-determinado (ϵ), estabelecer critério de convergência, tal que:

Se,

$$(Perdas\ Totais)_{i+1} - (Perdas\ Totais)_i \leq \epsilon \rightarrow \text{converge!} \quad (3.6)$$

Senão, ir ao próximo passo:

5. Para atualização da iteração atual, faz-se $i = i + 1$. Atualizar/calcular as tensões em cada barra de carga através da 2ª Lei de Kirchhoff, começando pela barra de referência (subestação), no sentido direto do fluxo de potência (*Forward*), considerando das quedas de tensão nos alimentadores obtidas através das correntes de linha calculadas na etapa anterior.

$$V_k = Z_{km} \cdot I_{km} + V_m \quad (3.7)$$

Voltar ao item 1 e realizar nova iteração até que o critério de convergência do item 4 seja obedecido.

6. Uma vez obedecido o critério de convergência (*iteração i*), chega-se ao fim do processo. Assim, toma-se os valores de tensão, corrente e perdas totais da *iteração i* como resultado do problema.

Cabe ressaltar que, na escolha do valor de tolerância, deve-se levar em consideração a precisão desejada para os resultados, isto é, quanto menor o valor de ϵ escolhido, maior será o número de iterações e, conseqüentemente, melhor será a qualidade das soluções encontradas pelo algoritmo.

- **Exemplo - Sistema 4 Barras**

Como exemplo de aplicação, foi realizado o cálculo de fluxo de carga no sistema teste de 4 barras (CARVALHO, 2005), a partir de um algoritmo implementado em linguagem C, baseado no método *Backward/Forward Sweep*.

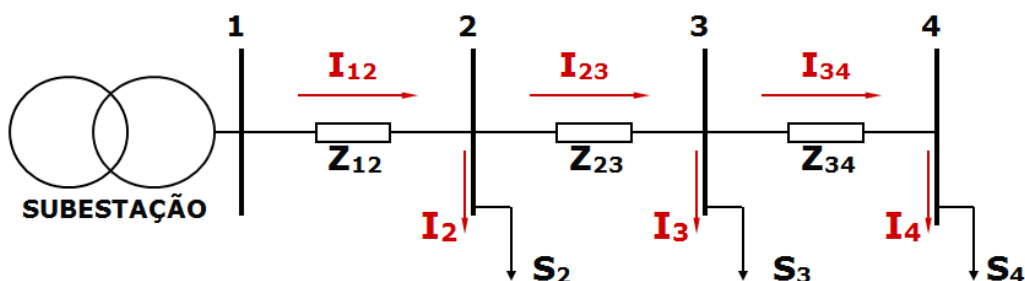


Figura 3.2 – Sistema de 4 Barras

❖ Dados

$$V_1 = 1,05 + j0,0 \text{ (p.u.)} \quad \epsilon = 0,00001$$

Tabela 3.4 – Dados do Sistema Teste de 4 Barras

Impedâncias (p.u.)	Cargas (p.u.)
$Z_{12} = 0,0236 + j0,0233$	$S_2 = 1,28 + j1,28$
$Z_{23} = 0,0030 + j0,0002$	$S_3 = 0,32 + j0,16$
$Z_{34} = 0,0051 + j0,0005$	$S_4 = 1,60 + j0,80$

❖ Resultados

Tabela 3.5 – Resultados do Sistema Teste de 4 Barras

Tensão		
Barras	Módulo (p.u.)	Ângulo (°)
1	1,050	0,00
2	0,909	-1,33
3	0,908	-1,33
4	0,898	-1,10
Perdas Totais (p.u.)		Numero de Iterações
0,462		4

3.3. Resultados

O algoritmo apresentou resultados satisfatórios em diversos outros testes realizados em sistemas de 10, 34 e 70 barras, apresentados a seguir. O valor de tolerância utilizado em todos os testes foi de 0,00001 e a subestação é caracterizada pela barra 1 em todos os sistemas.

3.3.1. Sistema de 10 Barras

Sistema retirado de J.J. Grainger e S.H. Lee (1982), cujos dados são apresentados no apêndice A. A seguir é apresentada a tabela de resultados do algoritmo implementado.

Tabela 3.6 - Resultados do Fluxo de Potência do Sistema de 10 barras

SISTEMA 10 BARRAS		
Fasores das Tensões das Barras		
Barra	Módulo (p.u.)	Fase (°)
1	1,000	0,00
2	0,993	-0,52
3	0,987	-1,27
4	0,963	-2,33
5	0,948	-2,65
6	0,917	-3,72
7	0,907	-4,14
8	0,889	-4,62
9	0,859	-5,40
10	0,838	-5,99
Número de Iterações: 6		
Perdas Ativas Totais: 783,757 kW		

3.3.2. Sistema de 34 Barras

Sistema retirado de M. Chis, M.M.A. Salama e S. Jayaram (1997), cujos dados são apresentados no apêndice A. A seguir é apresentada a tabela de resultados do algoritmo implementado.

Tabela 3.7- Resultados do Fluxo de Potência do Sistema de 34 barras

SISTEMA 34 BARRAS		
Fasores das Tensões nas Barras		
Barra	Módulo (kV)	Fase (°)
1	110,000	0,00
2	109,354	0,05
3	108,791	0,10
4	108,024	0,22
5	107,364	0,32
6	106,742	0,41
7	106,321	0,50
8	106,089	0,56

9	105,817	0,62
10	105,686	0,65
11	105,635	0,66
12	105,620	0,66
13	108,754	0,11
14	108,721	0,12
15	108,712	0,12
16	108,711	0,12
17	106,252	0,49
18	105,843	0,55
19	105,393	0,63
20	105,031	0,70
21	104,716	0,76
22	104,356	0,84
23	104,061	0,90
24	103,783	0,96
25	103,649	0,99
26	103,598	1,01
27	103,583	1,01
28	106,284	0,51
29	106,259	0,52
30	106,247	0,52
31	105,648	0,65
32	105,610	0,66
33	105,591	0,67
34	105,585	0,67
Número de Iterações: 4		
Perdas Ativas Totais: 222,289 kW		

3.3.3. Sistema de 70 Barras

Sistema retirado de M. E. Baran e F. F. Wu (1989), cujos dados são apresentados no apêndice A. A seguir é apresentada a tabela de resultados do algoritmo implementado.

Tabela 3.8 - Resultados do Fluxo de Potência do Sistema de 70 barras

SISTEMA 70 BARRAS		
Fasores das Tensões das Barras		
Barra	Módulo (kV)	Fase (°)
1	126,600	0,00
2	126,596	0,00
3	126,592	0,00
4	126,591	0,00
5	126,579	-0,01
6	126,475	-0,02
7	125,344	0,05
8	124,168	0,12
9	123,887	0,14
10	123,744	0,15
11	123,111	0,23
12	122,971	0,25
13	122,571	0,30
14	122,200	0,35
15	121,833	0,40
16	121,470	0,44
17	121,403	0,45
18	121,291	0,46
19	121,290	0,47
20	121,231	0,47
21	121,193	0,48
22	121,132	0,49
23	121,131	0,49
24	121,122	0,49
25	121,102	0,49
26	121,081	0,50
27	121,072	0,50
28	121,070	0,50
29	126,591	0,00
30	126,582	-0,01
31	126,566	0,00
32	126,564	0,00
33	126,550	0,00
34	126,518	0,00
35	126,475	0,01
36	126,467	0,01
37	126,589	0,00
38	126,567	-0,01
39	126,547	-0,01
40	126,542	-0,01

41	126,542	-0,01
42	126,453	-0,02
43	126,416	-0,03
44	126,411	-0,03
45	126,410	-0,03
46	126,398	-0,03
47	126,398	-0,03
48	126,573	-0,01
49	126,415	-0,05
50	125,928	-0,19
51	125,859	-0,21
52	123,883	0,14
53	123,881	0,14
54	123,391	0,17
55	122,980	0,19
56	122,414	0,23
57	121,861	0,27
58	119,016	0,66
59	117,616	0,86
60	117,074	0,95
61	116,438	1,05
62	115,501	1,12
63	115,465	1,12
64	115,416	1,13
65	115,175	1,14
66	115,102	1,15
67	122,964	0,25
68	122,964	0,25
69	122,529	0,31
70	122,529	0,31
Número de Iterações: 5		
Perdas Ativas Totais: 225,017 kW		

3.4. Considerações Finais

O algoritmo implementado para o cálculo do fluxo de potência, baseado no método *Back-forward Sweep*, trata-se de uma função a parte do problema de reconfiguração. Sendo, inicialmente, testado separadamente, para verificação de sua eficiência em resultados para sistemas radiais.

A aplicação computacional do método *Back-forward Sweep* apresentou rápida convergência e qualidade de resultados satisfatória em todos os sistemas testados. Sendo, portanto, considerada uma ferramenta eficaz para a resolução do problema de fluxo de carga para posterior avaliação da configuração ótima de sistemas de distribuição radial, foco principal deste trabalho.

4. OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

4.1. Introdução

Abordagens por técnicas de Inteligência Artificial ganharam espaço nos últimos anos para a solução de diversos problemas de otimização, como por exemplo, redes neurais artificiais, *simulated annealing*, algoritmos genéticos e programação evolutiva.

Dentro da classificação "*programação evolutiva*", existem algoritmos baseados no conceito de inteligência coletiva (*Swarm Intelligence*), isto é, imitam aspectos comportamentais coletivos de seres vivos, e têm sido amplamente utilizados nos últimos anos em diversas aplicações. Otimização por colônia de formigas (ACO) e otimização por enxame de partículas (PSO) são exemplos de algoritmos evolutivos baseados no conceito de inteligência coletiva, facilmente encontrados na literatura, devido a abrangência em aplicações. Não obstante, esses algoritmos têm sido cada vez mais aplicados a problemas de otimização dos sistemas elétricos de potência.

Neste trabalho foi utilizada uma versão binária do algoritmo PSO para solução do problema de reconfiguração. Será portanto, abordada neste capítulo, a formulação clássica desta metaheurística, bem como algumas de suas modificações ao longo dos anos. Será dedicado também um tópico com explanação detalhada da versão binária do PSO utilizada neste trabalho.

4.2. PSO Clássico

Otimização por Enxame de Partículas, do inglês *Particle Swarm Optimization* (PSO), é uma técnica de computação evolucionária inspirada na simulação de um sistema social. Tal técnica foi originalmente desenvolvida em 1995 por James Kennedy, um psicólogo social, em conjunto com o engenheiro eletricitista Russell Eberhart.

O algoritmo PSO, em sua formulação original (KENNEDY e EBERHART, 1995), simula a migração e agregação de um bando de aves procurando por comida e/ou local seguro. Fazendo analogia, o bando de aves representa o conjunto de soluções, sendo cada ave uma solução e, o local seguro e/ou com alimento representa

a função objetivo do problema de otimização. As possíveis soluções, também chamadas partículas, percorrem o espaço do problema, seguindo as melhores posições encontradas até o momento pelas próprias partículas e por todo o bando à procura do alvo (objetivo). Estas melhores posições, as quais as partículas procuram seguir, podem ser classificadas em: a melhor posição encontrada por ela mesma até o momento, chamada *pbest*; e a melhor posição encontrada por toda a população levando em consideração todas as partículas, chamada *gbest*. Ao final da execução, a melhor ou as melhores soluções (fitness), de acordo com uma função objetivo, são apresentadas como resultado.

O equacionamento do PSO clássico, para N partículas, é dado por:

$$v_{id}^{k+1} = w \cdot v_{id}^k + c_1 \cdot rand_1 \cdot (pbest_{id} - x_{id}^k) + c_2 \cdot rand_2 \cdot (gbest - x_{id}^k) \quad (4.1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (4.2)$$

Onde,

id: número da partícula. Sendo $id = (1, 2, \dots, N)$;

k: iteração corrente;

k+1: iteração nova.

A seguir, é mostrado na figura 4.1, diagrama do espaço de busca do PSO clássico.

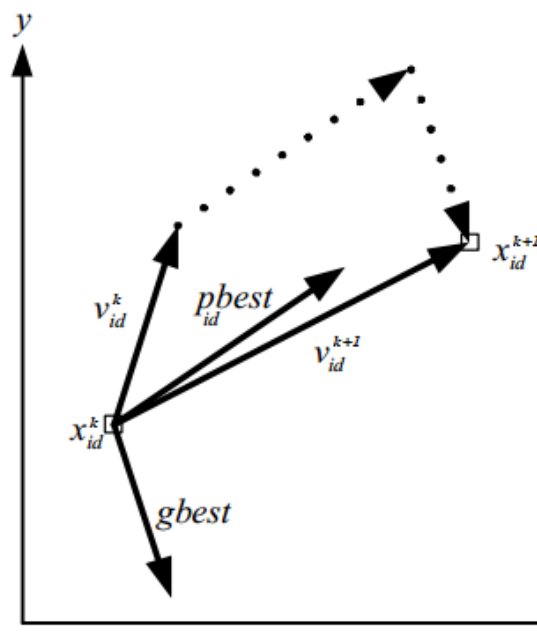


Figura 4.1 - Diagrama do Espaço de Busca do PSO Clássico

Onde v_{id}^k representa a velocidade atual da partícula x_{id}^k . A velocidade adquirida pela partícula à cada iteração (v_{id}^{k+1}), é determinada pela composição de sua velocidade atual, com sua posição em relação à sua melhor posição e à melhor posição do bando. A velocidade da partícula indica a direção que esta deve tomar, a fim de se aproximar ponderadamente do ótimo local ou global. Logo, a posição da partícula em cada iteração x_{id}^{k+1} é a soma de sua posição anterior com sua velocidade atualizada, v_{id}^{k+1} .

Os valores $rand_1$ e $rand_2$ são números aleatórios uniformemente distribuídos no intervalo [0, 1]. A inércia da partícula, representada por w , introduz a preferência da partícula em continuar se movendo na mesma direção que seguia na iteração anterior. Um valor de inércia alto facilita uma exploração mais global, enquanto um valor baixo facilita uma exploração mais local na busca pelo ótimo.

Os valores indicados por c_1 e c_2 são constantes positivas inteiras que correspondem a componentes cognitivas e sociais do bando, isto é, são parâmetros de confiança que indicam o quanto uma partícula confia em si (c_1) e quanto ela confia no bando (c_2). Os parâmetros de confiança e de inércia devem ser ajustados de acordo com o problema aplicado, pois são utilizadas para a atualização do vetor velocidade.

O método PSO clássico considera variáveis reais contínuas durante o processo de otimização. Entretanto, existem problemas cujo modelo envolve variáveis binárias e soluções combinatórias, em domínio discreto. O problema da RSDEE é um exemplo de problema combinatório em que se faz necessária a modelagem no domínio discreto e/ou binário. A modelagem binária para o problema da reconfiguração consiste em: para cada chave do sistema associa-se uma variável binária em que o valor 0 (zero) indica que a chave está aberta e o valor 1 (um) indica que a chave está fechada, por exemplo. As soluções consistem na combinação dos estados das chaves, sendo necessárias diversas modificações no tratamento das mesmas, no que diz respeito às atualizações a cada iteração, bem como nos mecanismos de busca realizados pelo algoritmo.

James Kennedy e Russell Eberhart (1997) propuseram uma versão discreta binária do PSO (DBPSO) capaz de operar variáveis binárias em domínios discretos. As modificações propostas no DBPSO preservam a estrutura do algoritmo PSO original e, além disso, possibilitam a aplicação do PSO discreto em uma nova classe de problemas. Nesta versão, cada partícula x_{id} se move num espaço cujo valor é restrito a 0 (zero) e 1 (um) em cada dimensão, onde a velocidade v_{id} representa um

valor uniformemente distribuído no intervalo $[0, 1]$, que posteriormente, será utilizada no cálculo da probabilidade do bit x_{id} tomar valor 1 (função sigmóide), para atualização do valor da partícula na iteração corrente.

O equacionamento do DBPSO (KENNEDY e EBERHART, 1997), para N partículas, é dado por:

$$v_{id}^{k+1} = w \cdot v_{id}^k + c_1 \cdot rand_1 \cdot (pbest_{id} - x_{id}^k) + c_2 \cdot rand_2 \cdot (gbest - x_{id}^k) \quad (4.3)$$

$$sigmoide(v_{id}^k) = \frac{1}{1 + e^{-v_{id}^k}} \quad (4.4)$$

$$x_{id}^{k+1} = \begin{cases} 1, & \text{se } rand < sigmoide(v_{id}^k) \\ 0, & \text{caso contrário} \end{cases} \quad (4.5)$$

Onde,

id : número da partícula. Sendo $id = (1, 2, \dots, N)$;

k : iteração corrente;

$k+1$: iteração nova.

Onde, $rand$ é um número aleatório uniformemente distribuído no intervalo $[0, 1]$. Os coeficientes w , c_1 , c_2 , $rand_1$ e $rand_2$ permanecem com os mesmos conceitos que na formulação clássica. Sendo diferenciados, apenas, pela limitação em seus valores no intervalo $[0, 1]$.

Na versão contínua do algoritmo PSO, a partícula é codificada como um conjunto de variáveis reais que representam a sua localização num espaço multidimensional e a velocidade é o operador responsável pela movimentação da partícula. Cabe ressaltar também que todas as dimensões normalmente são independentes umas das outras, de forma que a aplicação da velocidade à posição da partícula é realizada independentemente em cada dimensão. Entretanto, desta forma o algoritmo PSO clássico não é aplicável a problemas combinatórios, pois assim, o algoritmo violaria regras de restrição gerando soluções inválidas para o problema.

4.3. Algoritmo PSO Binário

A partir do algoritmo binário (KENNEDY e EBERHART, 1997), surgiram outras pesquisas dedicadas a modificar o PSO clássico com o intuito de lidar com os mais variados problemas de otimização combinatórios. Neste trabalho foi utilizada uma versão binária modificada do PSO, proposta Fu-Yuan Hsu e Men-Shen Tsai (2007), para reconfiguração de alimentadores de sistemas de distribuição de energia elétrica visando a redução de perdas ativas.

O problema de reconfiguração de alimentação é um problema de otimização combinatória, caracterizado pela permutação dos estados 0 (zero) e 1 (um) das possíveis chaves existentes no sistema. Isto é, cada solução do problema consiste em uma combinação dos estados de todas as chaves do sistema, por exemplo:

$$x_{id} = [1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1]$$

O vetor x_{id} representa uma solução em que, das 7 chaves existentes neste sistema, as chaves 3, 5 e 6 (cujas posições são dadas pela dimensão do vetor) se encontram abertas (valor 0), enquanto as demais se encontram fechadas (valor 1).

Neste trabalho, a função objetivo da RSDEE é minimizar as perdas totais do sistema. O método proposto neste trabalho garante, de maneira independente ao algoritmo PSO, que a estrutura radial seja sempre mantida e que os limites operacionais de tensão e corrente sejam admissíveis. Portanto, o problema de minimização de perdas pode ser formulado da seguinte forma:

$$Pt = \sum_{i=0}^n (r_i \cdot I_i^2) \quad (4.6)$$

Onde Pt são as perdas ativas totais nos alimentadores, n é o número total de linhas, I_i a magnitude da corrente elétrica e r_i a resistência elétrica em cada uma das linhas do sistema de distribuição.

Nesta versão binária do PSO, é definido um operador de deslocamento, cuja função é realizar a permuta entre uma solução binária corrente e a próxima a ser avaliada, ou seja, o vetor velocidade definido no PSO clássico, será utilizado nesta aplicação como um operador de deslocamento de bits.

O operador de deslocamento será definido como um vetor de três dimensões, onde cada uma dessas dimensões carrega uma informação:

- 1) Qual o bit da solução x_{id} será deslocado;
- 2) Em qual direção será realizado o deslocamento;
- 3) Quantos passos serão dados no deslocamento.

Considere um exemplo simplificado de aplicação do operador a seguir:

- Operador de deslocamento calculado: $v_{id}^{k+1} = [(3, -1, 2)]$
- Solução corrente: $x_{id}^k = [1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1]$

Tem-se que a atualização da partícula é dada da seguinte forma:

$$x_{id}^{k+1} = x_{id}^k \langle + \rangle v_{id}^{k+1} \quad (4.7)$$

Onde a operação ' $\langle + \rangle$ ' significa aplicação do operador v_{id}^{k+1} à solução corrente x_{id}^k , para permutação e obtenção da próxima solução. Sendo assim, temos a nova partícula:

$$x_{id}^k = [1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1] \xrightarrow{v_{id}^{k+1} = [(3, -1, 2)]} x_{id}^{k+1} = [0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1]$$

O operador de deslocamento pode conter mais de uma operação de comutação, ou seja, pode-se ter um conjunto de operações consecutivas a serem executadas numa mesma operação, devido ao agrupamento de operadores nas iterações. Este agrupamento se dá no cálculo de v_{id}^{k+1} , utilizado neste algoritmo, e que será explicado detalhadamente na sequência.

Um agrupamento de operadores, pode ser representado da seguinte forma:

$$v_{id}^{k+1} = [(3, -1, 2); (4, +1, 1)]$$

A operação de agrupamento de operadores é dada por ' \oplus ', conforme exemplo:

$$v_{id}^{k+1} = [(3, -1, 2)] \oplus [(4, +1, 1)],$$

$$\text{logo, } v_{id}^{k+1} = [(3, -1, 2); (4, +1, 1)]$$

$$x_{id}^k = [1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1] \xrightarrow{v_{id}^{k+1} = [(3, -1, 2); (4, +1, 1)]} x_{id}^{k+1} = [0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1]$$

Da mesma forma, o operador de deslocamento pode ser obtido pela operação entre duas soluções distintas. A seguir é mostrada a operação de obtenção do operador, pela comparação entre duas soluções, procedimento inverso ao demonstrado no exemplo anterior.

$$x_{id}^{k+1} \ominus x_{id}^k = [(3, -1, 2); (4, +1, 1)]$$

Uma vez definidos os conceitos de operador de deslocamento, agrupamento dos mesmos e sua atuação no algoritmo, é possível redefinir as fórmulas de atualização iterativa das partículas, formuladas nesta versão do PSO, para solução do problema de reconfiguração.

Sendo assim, o equacionamento do PSO binário (Fu-Yuan Hsu e Men-Shen Tsai , 2007), para N partículas, é dado por:

$$v_{id}^{k+1} = (w \otimes v_{id}^k) \oplus (rand() \langle x \rangle (pbest_{id} \ominus x_{id}^k)) \oplus (rand() \langle x \rangle (gbest \ominus x_{id}^k)) \quad (4.8)$$

$$x_{id}^{k+1} = x_{id}^k \langle + \rangle v_{id}^{k+1}$$

Onde,

id: número da partícula. Sendo $id = (1, 2, \dots, N)$;

k: iteração corrente;

k+1: iteração nova.

É possível notar a preservação do equacionamento clássico do PSO, salvo modificações operacionais realizadas para adaptação do algoritmo ao espaço discreto binário de soluções.

O coeficiente *w* ainda preserva seu conceito de inércia, sendo utilizado também para ajuste das áreas de busca. Sua atuação, nesta abordagem do PSO, consiste no incremento ou decremento do passo a ser realizado no operador de deslocamento a cada iteração. Sendo assim, o símbolo ' \otimes ' aplica o valor de *w* calculado à dimensão do operador que compreende o valor do passo do deslocamento.

A composição do coeficiente *rand()* com o símbolo ' $\langle x \rangle$ ' executam uma operação, cuja ação é escolher randomicamente um operador de deslocamento entre os operadores agrupados em $(pbest_{id} \ominus x_{id}^k)$ e em $(gbest \ominus x_{id}^k)$.

Para mostrar o funcionamento iterativo do algoritmo, segue um breve exemplo de aplicação, mostrando passo a passo cada operação realizada.

- **Exemplo**

Considerando a iteração corrente (k), seus valores são:

$$x_{id}^k = [1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1]$$

$$v_{id}^k = [(3, +1, 1); (7, +1, 1)]$$

$$pbest_{id} = [1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1]$$

$$gbest = [1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1]$$

$$w = 3$$

Pode-se agora determinar os operadores:

$$pbest_{id} \ominus x_{id}^k = [(3, -1, 1); (7, +1, 1)]$$

$$gbest \ominus x_{id}^k = [(3, +1, 1); (7, -1, 1)]$$

$$w \otimes v_{id}^k = [(3, +1, 3); (7, +1, 3)]$$

Supondo-se a escolha randômica dos operadores, tem-se:

$$rand() \langle x \rangle (pbest_{id} \ominus x_{id}^k) = [(3, -1, 1)]$$

$$rand() \langle x \rangle (gbest \ominus x_{id}^k) = [(7, -1, 1)]$$

Por fim, faz-se o agrupamento dos operadores calculados:

$$v_{id}^{k+1} = (w \otimes v_{id}^k) \oplus (rand() \langle x \rangle (pbest_{id} \ominus x_{id}^k)) \oplus (rand() \langle x \rangle (gbest \ominus x_{id}^k))$$

$$v_{id}^{k+1} = [(3, +1, 3); (7, +1, 3)] \oplus [(3, -1, 1)] \oplus [(7, -1, 1)]$$

$$v_{id}^{k+1} = [(3, +1, 3); (7, +1, 3); (3, -1, 1); (7, -1, 1)]$$

Nota-se que há operadores que podem ser simplificados, uma vez que operam no mesmo bit. Isto é, os operadores [(7, +1, 3)] e [(7, -1, 1)] podem ser reduzidos a apenas um operador, dado por [(7, +1, 2)]. Da mesma forma, os operadores [(3, +1, 3)] e [(3, -1, 1)] podem ser reduzidos ao operador [(3, +1, 2)].

Portanto, tem-se uma nova velocidade, ou operador de deslocamento, determinado para atualização da partícula:

$$v_{id}^{k+1} = [(7, +1, 2); (3, +1, 2)]$$

Assim, o novo valor da partícula, na iteração $(k+1)$ é dado por:

$$x_{id}^{k+1} = x_{id}^k (+) v_{id}^{k+1} = [1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0]$$

Neste momento, a nova solução pode ser então avaliada de acordo com sua factibilidade sendo, posteriormente, calculado o valor total das perdas ativas, dando continuidade ao processo iterativo do algoritmo.

O procedimento geral de funcionamento do algoritmo consiste em:

1. Determinar o tamanho da população de partículas (N), número máximo de iterações e passo máximo dos operadores.
2. Gerar N soluções iniciais factíveis aleatórias. Criar um operador inicial aleatório para gerar novas partículas.
3. Calcular valor da função objetivo de cada partícula.
4. Avaliar os valores de *pbest* e *gbest*.
5. Atualizar o operador de deslocamento de acordo com equação 4.8, e gerar partícula a partir do novo operador calculado, de acordo com equação 4.7.
6. Avaliar se o critério de parada foi obedecido. Senão, voltar ao passo 3.

A aplicação deste algoritmo ao problema da reconfiguração será detalhada no Capítulo 5 deste trabalho, onde serão apresentados os procedimentos realizados pelo algoritmo a partir de exemplo numérico, simulado em programa computacional cuja implementação foi realizada em linguagem C.

5. ALGORITMO PROPOSTO PARA RECONFIGURAÇÃO DE SISTEMAS DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA

5.1. Introdução

A reconfiguração consiste na determinação de uma topologia radial conexa à rede, mediante a permutação dos estados (aberto ou fechado) das chaves manobráveis acopladas aos sistemas de distribuição. Este procedimento é normalmente utilizado para o balanceamento de cargas entre alimentadores, melhoria dos níveis de tensão, isolamento de faltas ou para minimização das perdas ativas totais do sistema. As restrições, como a sobrecarga da capacidade do alimentador, limitações de tensão e radialidade devem ser respeitadas após a reconfiguração.

Para a reconfiguração dos sistemas testados neste trabalho, utiliza-se o princípio de que todas as linhas do sistema são consideradas chaves normalmente fechadas. Sendo assim, a metodologia proposta aqui consiste em, com todas as chaves inicialmente fechadas, gerar novas topologias radiais mediante a abertura consecutiva de chaves, até que se encontre a topologia que apresenta menor perda.

Deste modo, a formulação do problema de reconfiguração utilizada neste trabalho é dada por:

$$\text{Minimizar } Pt = \sum_{i=0}^n (r_i \cdot I_i^2) \quad (5.1)$$

$$\text{Sujeito à } \left\{ \begin{array}{l} \text{Limite de Magnitude das tensões nodais: } V_{\min} \leq V_j \leq V_{\max} \\ \text{Limite de Magnitude das correntes nas linhas: } |I_j| \leq I_{\max} \\ \text{Radialidade} \end{array} \right.$$

Onde V_j é a magnitude da tensão na barra j , V_{\max} e V_{\min} , as magnitudes máxima e mínima permitidas para a tensão, respectivamente.

É realizada, neste capítulo, uma apresentação detalhada do algoritmo proposto para a resolução do problema de reconfiguração.

5.2. Reconfiguração de Sistemas de Distribuição via PSO

O algoritmo proposto neste trabalho é dividido da seguinte forma:

1. Carregamento dos dados do sistema malhado a ser reconfigurado;
2. Definição da *Matriz A*, utilizada para avaliação da conectividade e radialidade da rede;
3. Função geradora de soluções (partículas), responsável por gerar aleatoriamente N soluções iniciais factíveis a serem utilizadas para dar início ao algoritmo PSO.
4. Processo iterativo do algoritmo PSO binário proposto, conforme apresentado no Capítulo 4 deste trabalho. Em paralelo, rotina de avaliação de conectividade da rede e cálculo de fluxo de carga das topologias factíveis encontradas;
5. Impressão, na tela de comando, do resultado encontrado ao final das iterações do algoritmo PSO.

Na etapa de carregamento de dados é feita alocação dinâmica de todos os valores necessários para a reconfiguração e cálculo da rede. Os dados armazenados são:

- Número de barras;
- Número de linhas;
- Valor de Referência (Tensão na Subestação);
- Valores de potência ativa e reativa de cada uma das barras de carga;
- Valores de resistência, reatância, barras de origem e destino de cada linha.

A partir dos dados do sistema é realizada a montagem da Matriz A, que será utilizada em diversos momentos do algoritmo, com o objetivo de informar e avaliar a conectividade da rede. Esta matriz relaciona a conectividade de cada barra às demais presentes no sistema mediante enumeração de suas posições, sendo as barras mapeadas por suas linhas e colunas. Mais detalhes da Matriz A serão dados em exemplo numérico apresentado neste capítulo.

A partir de estudos sobre a teoria dos grafos, é possível concluir que uma condição necessária, porém não suficiente, para que uma topologia seja radial é que:

$$(n_{\text{linhas}})_{SR} = n_{\text{barras}} - 1 \quad (5.2)$$

Onde,

n_{linhas} : número de linhas (chaves) do sistema radial

n_{barras} : número de barras do sistema

Sendo assim, o ponto inicial da reconfiguração é definir o número de chaves que serão abertas, a partir de tal informação. Em um sistema malhado o número de chaves pode ser igual ou superior ao número de barras. Logo, a determinação do número de linhas (chaves) a serem abertas é feita da seguinte forma:

$$n_{\text{chaves}} = (n_{\text{linhas}})_{\text{SM}} - (n_{\text{linhas}})_{\text{SR}} \quad (5.3)$$

Onde,

n_{chaves} : número de chaves a serem abertas

$(n_{\text{linhas}})_{\text{SM}}$: número de linhas (chaves) do sistema malhado

$(n_{\text{linhas}})_{\text{SR}}$: número de linhas do sistema radial dado pela equação (5.2)

A partir da determinação do número de chaves a serem abertas, é possível gerar soluções aleatórias para dar início ao processo de otimização. Porém, é necessário que estas soluções sejam factíveis para que possam ser incluídas no processo iterativo. O processo de geração de soluções aleatórias iniciais é muito simples, pois se trata somente da utilização do comando '*rand()*' em linguagem C para produção de valores positivos inteiros, cuja limitação inferior é 1 (um) e a superior é o número total de linhas do sistema malhado. Desta forma, a escolha inicial das chaves se dará de forma discreta, cujos valores indicam quais chaves serão abertas, sendo convertida posteriormente para a forma binária, de modo que possa ser aplicado ao algoritmo PSO binário. A dificuldade de tal rotina se dá na avaliação dos valores produzidos, podendo eles ser repetidos, distintos com solução infactível ou distintos com solução factível. Será realizado o detalhamento da rotina de avaliação de conectividade da rede, conforme exemplo numérico apresentado na sequência.

Após a geração das partículas iniciais, é feita uma escolha, também inicialmente aleatória, do operador de deslocamento, cujo conceito é análogo à velocidade no PSO clássico, isto é, responsável por produzir novas partículas. A cada iteração é aplicado um operador de deslocamento de bits calculado a partir de informações de iterações passadas, porém a aplicação deste operador não garante que as soluções geradas sejam factíveis. Portanto, também nesta etapa do algoritmo, as soluções devem passar por avaliação de conectividade para posterior cálculo do fluxo de carga. No

entanto, quando um operador gera uma solução inactível, tem-se outro desafio: como gerar outra solução na mesma iteração, com o mesmo operador, já que a encontrada não atende à restrição de radialidade.

Portanto, é proposta neste trabalho, uma modificação para que o PSO possa buscar soluções, em uma mesma iteração, até que encontre uma que seja factível. Não foi avaliado se tal modificação compromete ou não a eficiência computacional do método de maneira significativa, no entanto, apresenta a vantagem de limitar o espaço de busca à somente soluções factíveis.

Sendo assim, há uma modificação na proposta original do PSO binário em que este trabalho se baseia. Pois, em tal proposta, não são feitas considerações a cerca de inactibilidades das soluções geradas.

No intuito de unir e detalhar todas as informações ditas até aqui e para que se possa obter entendimento passo a passo da reconfiguração proposta neste trabalho, segue exemplo numérico.

- **Exemplo de Aplicação - Sistema 5 barras**

A figura a seguir representa um sistema de distribuição malhado fictício (PEREIRA, 2010). Os dados deste sistema se encontram no apêndice A.

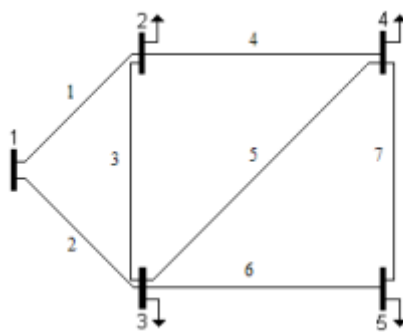


Figura 5.1 - Sistema de Distribuição de 5 barras¹

Considera-se a barra 1 como sendo a subestação e cada uma das 7 linhas é considerada chave manobrável do sistema. Sabe-se que ao todo existem 128 (2^7) topologias possíveis, sendo apenas 21 delas factíveis (PEREIRA, 2010).

¹Figura retirada de (PEREIRA, 2010)

Neste momento, é possível determinar o número de chaves que serão abertas para a reconfiguração. De acordo com as Equações (5.2) e (5.3), temos:

$$\begin{cases} n_{\text{barras}} = 5 \\ (n_{\text{linhas}})_{\text{SM}} = 7 \end{cases}$$

$$(n_{\text{linhas}})_{\text{SR}} = n_{\text{barras}} - 1 \quad \Leftrightarrow \quad (n_{\text{linhas}})_{\text{SR}} = 4$$

$$n_{\text{chaves}} = (n_{\text{linhas}})_{\text{SM}} - (n_{\text{linhas}})_{\text{SR}} \quad \Leftrightarrow \quad n_{\text{chaves}} = 3$$

Portanto, deverão ser abertas 3 chaves para que se possa ter soluções radiais, porém esta condição não garante suas respectivas factibilidades.

A seguir é apresentada uma tabela que relaciona as 21 topologias factíveis deste sistema com seus respectivos valores de perdas ativas totais, sendo estes calculados por meio do algoritmo de fluxo de carga de varredura elaborado neste trabalho.

Tabela 5.1 - Perdas ativas do Sistema de Distribuição de 5 Barras

Topologias	Chaves abertas	Perdas Ativas Totais (p.u.)
1	3, 5 e 7	0,0995
2	2, 5 e 7	0,1622
3	1, 5 e 7	0,0528
4	3, 4 e 7	0,0362
5	2, 4 e 7	0,1492
6	1, 4 e 7	0,0365
7	2, 3 e 7	0,1804
8	1, 3 e 7	0,0554
9	3, 5 e 6	0,1604
10	2, 5 e 6	0,1820
11	1, 5 e 6	0,0727
12	3, 4 e 6	0,0383
13	2, 4 e 6	0,1516
14	1, 4 e 6	0,0386
15	2, 3 e 6	0,1803
16	1, 3 e 6	0,0589
17	3, 4 e 5	0,0592
18	2, 4 e 5	0,1662
19	1, 4 e 5	0,0596
20	2, 3 e 5	0,1939
21	1, 3 e 5	0,1321

Na sequência, será apresentado o passo a passo do algoritmo implementado para a reconfiguração do sistema em questão. No algoritmo foi utilizado um critério de convergência para o PSO, dado por número máximo de iterações, estipulado inicialmente. Neste exemplo, será utilizado número máximo de iterações igual a 2.

5.2.1. Carregamento de Dados e Montagem da *Matriz A*

A partir do carregamento de dados do sistema, é possível obter o mapeamento das linhas atribuindo numeração a cada uma delas e relacionando suas barras terminais e de origem, conforme apresentado na Tabela 5.2, a seguir.

Tabela 5.2 - Barras terminais e finais

Linha	Barra inicial	Barra final
1	1	2
2	1	3
3	2	3
4	2	4
5	3	4
6	3	5
7	4	5

Sendo assim, é possível determinar a conectividade de cada uma das barras em relação às demais e, conseqüentemente, realizar a montagem da Matriz A para aplicação à diversas etapas do algoritmo de reconfiguração.

As linhas e colunas da matriz A são enumeradas pelas barras do sistema, sendo portanto, neste caso, uma matriz quadrada cuja dimensão é 5x5. Às barras que possuem conectividade entre si, é atribuído o valor -1 na posição da matriz que as relaciona para indicação da conexão, às barras que não possuem conexão entre si, é atribuído valor 0 e, por fim, na diagonal principal de cada barra de origem, será atribuída a soma do número de barras finais conectadas à ela. As barras de origem estão ordenadas ao longo das linhas da matriz, enquanto que as barras finais estão ordenadas ao longo das colunas da matriz. Segue a Tabela 5.3 que relaciona a cada barra, as barras finais a que estão conectadas e, em seguida, procedimento de montagem e resultado da Matriz A do sistema de 5 Barras.

Tabela 5.3 - Barras de origem e barras finais

Barra de origem	Barras finais
1	2 e 3
2	1, 3 e 4
3	1, 2, 4 e 5
4	2 e 5
5	3 e 4

Inicialmente, todos os elementos da matriz A são zerados:

$$A_{ij} = A_{ji} = 0$$

Onde,

nb: número de barras

$$i = [1, 2, \dots, nb]$$

$$j = [1, 2, \dots, nb]$$

Posteriormente, são atualizados os valores, conforme Tabela 5.3.

$$A_{12} = A_{21} = -1 \quad A_{13} = A_{31} = -1$$

$$A_{23} = A_{32} = -1 \quad A_{24} = A_{42} = -1$$

$$A_{34} = A_{43} = -1 \quad A_{35} = A_{53} = -1$$

$$A_{45} = A_{54} = -1$$

Por fim, são calculados os valores da diagonal principal.

$$A_{11} = | A_{12} + A_{13} + A_{14} + A_{15} | = 2$$

$$A_{22} = | A_{21} + A_{23} + A_{24} + A_{25} | = 3$$

$$A_{33} = | A_{31} + A_{32} + A_{34} + A_{35} | = 4$$

$$A_{44} = | A_{41} + A_{42} + A_{43} + A_{45} | = 3$$

$$A_{55} = | A_{51} + A_{52} + A_{53} + A_{54} | = 2$$

Logo, a matriz A é dada por:

$$A = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ -1 & -1 & 4 & -1 & -1 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

Por facilidades de implementação computacional, optou-se pelo sinal negativo dos valores dos elementos indicadores de conectividade.

5.2.2. Algoritmo PSO Binário para RSDEE

O ciclo de ações, bem como as funções executadas em cada etapa do algoritmo são representados no fluxograma a seguir.

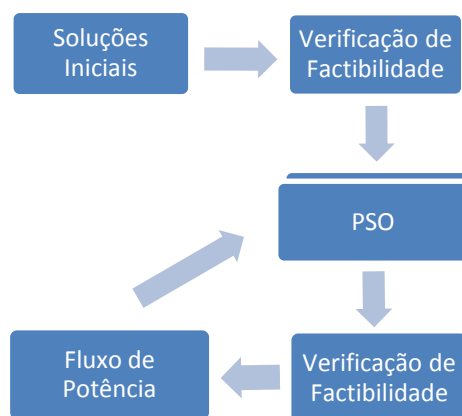


Figura 5.3 - Fluxograma representativo do algoritmo para RSDEE

Será mostrado na sequência o processo iterativo do algoritmo PSO utilizado.

Com base ainda no exemplo da Figura 5.1, onde tem-se um total de 7 chaves, cada partícula terá dimensão igual a 7, sendo cada uma delas avaliada em 0 ou 1, indicando seu estado aberto ou fechado, respectivamente. Sabe-se, também, que o número de chaves a serem abertas, corresponde ao número de zeros, ou seja, todas as partículas apresentarão 3 valores em zero, e 4 valores em 1.

A função "randômica", anteriormente comentada, apresenta duas soluções iniciais factíveis (SS_1 e SS_2) para este exemplo numérico. Conforme comentado, cada

solução gerada, passa pela rotina de avaliação de conectividade da topologia, que será detalhada posteriormente.

❖ Iteração 0

a) Partículas Iniciais

✓ Partícula 1

- $(SS_1)^0 = [0 1 1 1 0 1 0]$
- Chaves abertas: $(Ch_1)^0 = [1 5 7]$
- Perdas Ativas Totais: 0,0528 p.u.

✓ Partícula 2

- $(SS_2)^0 = [0 1 0 1 0 1 1]$
- Chaves abertas: $(Ch_2)^0 = [3 1 5]$
- Perdas Ativas Totais: 0,1321 p.u.

Devido à formulação do algoritmo, as soluções são tratadas na forma binária (Vetor SS_n). Porém, serão sempre acompanhadas por sua respectivas representações discretas (Ch_n) com o objetivo de informar quais as chaves estão abertas de maneira direta, tanto para fins didáticos, como também por sua utilização em diversas etapas do algoritmo.

Nesta etapa, é feita a escolha livre do primeiro operador de deslocamento a ser aplicado às soluções iniciais geradas para a produção de novas partículas (iteração 1). O operador, conforme descrito no capítulo 4 deste trabalho, consiste de 3 valores: bit de deslocamento, direção de deslocamento (+1 ou -1) e passo de deslocamento.

Por escolha, o bit a ser deslocado e a direção de deslocamento, foram padronizadas em -1 e +1 nestes operadores iniciais. O passo de deslocamento (ou inércia), para cada solução, será gerado aleatoriamente com limitação no intervalo [1, 6], ou seja, pode-se permutar um bit em, no mínimo uma posição e no máximo em seis posições.

Para a primeira iteração, deseja-se deslocar somente bits cujo valor é zero. Para garantir que o bit a ser deslocado seja de valor 0, o valor que consiste o bit de deslocamento no operador inicial (escolhido valor 1), representa, na realidade, a

dimensão no vetor solução na versão discreta. Logo, é feita uma transformação tal que o bit de deslocamento recebe a primeira chave aberta do vetor Ch_n . Sendo assim, a escolha do operador é aleatória, porém garante a condição de que o bit é zero (chave aberta).

❖ Iteração 1

a) Operadores de Deslocamento

✓ Partícula 1

- $(SS_1)^0 = [0\ 1\ 1\ 1\ 0\ 1\ 0]$
- Chaves abertas: $(Ch_1)^0 = [1\ 5\ 7]$
- Operador Inicial: $(Sh_1)^1 = [(sh1, sh2, sh3)]$

```
//Onde,
sh1 = 1 // é a dimensão do vetor  $(Ch_1)^0$ 

//Neste caso, o valor a ser alocado no operador é, na realidade, o número 1.
//Sendo assim, é feita a conversão:
sh1 = valor armazenado na dimensão-sh1 do vetor  $(Ch_1)^0$ 

//Logo,
sh1 = 1
sh2 = +1 //deslocamento para a direita no vetor  $SS_n$ 
sh3 = rand()%6+1 //Supondo: sh3 = 2

//Por fim, temos:
 $Sh_1^1 = [(1, +1, 2)]$  //Operador Inicial
```

Da mesma forma, para a partícula 2:

✓ Partícula 2

- $(SS_2)^0 = [0\ 1\ 0\ 1\ 0\ 1\ 1]$
- Chaves abertas: $(Ch_2)^0 = [3\ 1\ 5]$
- Operador Inicial: $(Sh_2)^1 = [(sh1, sh2, sh3)]$

```
//Onde,
sh1 = 3
sh2 = +1
sh3 = rand()%6+1 //Supondo: sh3 = 3

//Por fim, temos:
Sh21 = [(3, +1, 3)] //Operador Inicial
```

Neste momento, estão determinadas todas as condições iniciais suficientes para dar início ao processo iterativo do algoritmo PSO. Nesta etapa, basta-se aplicar os operadores gerados para obtenção de novas soluções. Cabe ressaltar que neste exemplo numérico serão apresentadas somente soluções factíveis.

b) Novas Partículas

A partir da equação 4.7 são calculadas novas partículas. Porém é feita adaptação das equações ao problema de reconfiguração:

$$SS_i^{k+1} = SS_i^k \langle + \rangle Sh_i^{k+1} \quad (5.4)$$

✓ Partícula 1:

- Solução corrente: $(SS_1)^0 = [0\ 1\ 1\ 1\ 0\ 1\ 0]$
- Operador: $(Sh_1)^1 = [(1, +1, 2)]$

```
SS11 = SS10 ⟨+⟩ Sh11
SS11 = [0 1 1 1 0 1 0] ⟨+⟩ [(1, +1, 2)]
SS11 = [1 1 0 1 0 1 0]

Perdas Ativas Totais = 0,0995 p.u.
```

✓ Partícula 2:

- Solução corrente: $(SS_2)^0 = [0\ 1\ 0\ 1\ 0\ 1\ 1]$
- Operador: $(Sh_2)^1 = [(3, +1, 3)]$

$$SS_2^1 = SS_2^0 \langle + \rangle Sh_2^1$$

$$SS_2^1 = [0\ 1\ 0\ 1\ 0\ 1\ 1] \langle + \rangle [(3, +1, 3)]$$

$$SS_2^1 = [0\ 1\ 1\ 1\ 0\ 0\ 1]$$

Perdas Ativas Totais = 0,0727 p.u.

c) Avaliação dos Valores $Pbest$ e $Gbest$

$Pbest$ e $Gbest$ são as partículas que apresentam melhores valores locais e globais, respectivamente.

Os valores de $Pbest$ são obtidos a partir da comparação dos valores de perdas ativas apresentadas por cada partícula em cada iteração, como segue:

Tabela 5.4 - Valores $Pbest$ - Iteração 1

Partícula	Iteração	Perdas ativas	$Pbest_1$
1	0	0,0528 p.u.	$(SS_1)^0 = [0\ 1\ 1\ 1\ 0\ 1\ 0]$
	1	0,0995 p.u.	
2	0	0,1321 p.u.	$(SS_2)^1 = [0\ 1\ 1\ 1\ 0\ 0\ 1]$
	1	0,0727 p.u.	

O valor de $Gbest$, cujo conceito envolve melhor valor global, é obtido mediante a comparação entre os melhores valores de perdas ativas das partículas ($Pbest$). Logo:

Tabela 5.5 - $Gbest$ - Iteração 1

Partícula	$Pbest$	Perdas Ativas	$Gbest_1$
1	$(SS_1)^0$	0,0528 p.u.	$(SS_1)^0 = [0\ 1\ 1\ 1\ 0\ 1\ 0]$
2	$(SS_2)^1$	0,0727 p.u.	

❖ **Iteração 2**

a) Operadores de Deslocamento

Nesta etapa, é feito o cálculo do operador de deslocamento, conforme equação 4.8. Traduzindo para o problema de reconfiguração, temos:

$$Sh_i^{k+1} = (w \otimes Sh_i^k) \oplus (rand() \langle x \rangle (Pbest_i \ominus SS_i^k)) \oplus (rand() \langle x \rangle (Gbest \ominus SS_i^k)) \quad (5.5)$$

Onde,

i : partícula, sendo $i = (1, 2, \dots, N)$

k : iteração corrente

$k+1$: nova iteração

O cálculo da inércia (w), neste trabalho, foi dado de forma aleatória. Esta escolha se deve ao fato de que no espaço discreto binário de soluções um passo pequeno ou um passo maior não podem ser mensurados como melhor ou pior. Tal conclusão foi possível pela tentativa de aplicação da inércia crescente com o número de iterações (conforme proposto no PSO binário para reconfiguração), comparado com a aplicação da inércia aleatória a cada iteração. Constatou-se, então, que na maioria dos casos o algoritmo com inércia randômica apresentou melhor porcentagem de convergência no valor ótimo.

Outra observação acerca do passo a ser dado no deslocamento diz respeito à um valor de passo em que o bit "excede" o espaço de dimensões, ou seja, caso o bit 4 sofra deslocamento igual a 5, teoricamente iria para a posição 9 do vetor de soluções. Porém, a posição máxima é limitada a 7. Para resolução deste problema, é feita rotina tal que, se o deslocamento for maior que 7, será subtraído 7 do mesmo, reiniciando a contagem a partir do 1.

Para obtenção dos novos operadores, aplica-se equação 5.5:

✓ Partícula 1:

- $Pbest_1 = (SS_1)^0 = [0\ 1\ 1\ 1\ 0\ 1\ 0]$
- $Gbest = (SS_1)^0 = [0\ 1\ 1\ 1\ 0\ 1\ 0]$
- Solução Corrente: $(SS_1)^1 = [1\ 1\ 0\ 1\ 0\ 1\ 0]$
- Operador Corrente: $(Sh_1)^1 = [(1, +1, 2)]$

$$(Gbest \ominus SS_1^1) = [(3, -1, 3)]$$

$$(Pbest_1 \ominus SS_1^1) = [(3, -1, 3)]$$

//Como só há um único operador resultante em cada operação, temos:

$$(rand() \langle x \rangle (Gbest \ominus SS_1^1)) = [(3, -1, 3)]$$

$$(rand() \langle x \rangle (Pbest_1 \ominus SS_1^1)) = [(3, -1, 3)]$$

//O cálculo da inércia será dado por:

$$w = rand() \% 6 + 1; \text{ // Supondo } w = 3$$

$$(w \otimes Sh_1^1) = [(1, +1, 3)] \text{ //Atualiza o passo do operador } Sh_1^1 \text{ com o valor de } w$$

//Logo,

$$Sh_1^2 = (w \otimes Sh_1^1) \oplus (rand() \langle x \rangle (Pbest_1 \ominus SS_1^1)) \oplus (rand() \langle x \rangle (Gbest \ominus SS_1^1))$$

$$Sh_1^2 = [(1, +1, 3)] \oplus [(3, -1, 3)] \oplus [(3, -1, 3)]$$

$$Sh_1^2 = [(1, +1, 3); (3, -1, 3); (3, -1, 3)]$$

//Simplificando os operadores de mesmo bit a ser deslocado, temos:

$$Sh_1^2 = [(1, +1, 3); (3, -1, 6)]$$

Da mesma forma, para a partícula 2.

✓ Partícula 2

- $Pbest_2 = (SS_2)^1 = [0\ 1\ 1\ 1\ 0\ 0\ 1]$

- $Gbest = (SS_1)^0 = [0\ 1\ 1\ 1\ 0\ 1\ 0]$

- Solução Corrente: $(SS_2)^1 = [0\ 1\ 1\ 1\ 0\ 0\ 1]$

- Operador Corrente: $(Sh_2)^1 = [(3, +1, 3)]$

Cálculos para nova iteração.

$$(Gbest \ominus SS_2^1) = [(6, +1, 1)]$$

$$(Pbest_2 \ominus SS_2^1) = [(0, 0, 0)]$$

//Como só há um único operador resultante em cada operação, temos:

$$(rand() \langle x \rangle (Gbest \ominus SS_2^1)) = [(6, +1, 1)]$$

$$(rand() \langle x \rangle (Pbest_2 \ominus SS_2^1)) = [(0, 0, 0)]$$

//O cálculo da inércia será dado por:

$$w = rand() \% 6 + 1; // Supondo w = 2$$

$$(w \otimes Sh_2^1) = [(3, +1, 2)] //Atualiza o passo do operador Sh_2^1 com o valor de w$$

//Por fim,

$$Sh_2^2 = (w \otimes Sh_2^1) \oplus (rand() \langle x \rangle (Pbest_2 \ominus SS_2^1)) \oplus (rand() \langle x \rangle (Gbest \ominus SS_2^1))$$

$$Sh_2^2 = [(3, +1, 2)] \oplus [(0, 0, 0)] \oplus [(6, +1, 1)]$$

$$Sh_2^2 = [(3, +1, 2); (6, +1, 1)]$$

b) Novas Partículas

✓ Partícula 1:

- Solução corrente: $(SS_1)^1 = [1\ 1\ 0\ 1\ 0\ 1\ 0]$

- Operador: $(Sh_1)^2 = [(1, +1, 3); (3, -1, 6)]$

$$SS_1^2 = SS_1^1 \langle + \rangle Sh_1^2$$

$$SS_1^2 = [1\ 1\ 0\ 1\ 0\ 1\ 0] \langle + \rangle [(1, +1, 3); (3, -1, 6)]$$

$$SS_1^2 = [1\ 0\ 1\ 1\ 0\ 1\ 0]$$

$$Ch_1^2 = [2\ 5\ 7]$$

Perdas Ativas Totais = 0,1662 p.u.

✓ Partícula 2:

- Solução corrente: $SS_2^1 = [0\ 1\ 1\ 1\ 0\ 0\ 1]$

- Operador: $Sh_2^2 = [(3, +1, 2); (6, +1, 1)]$

$$SS_2^2 = SS_2^1 (+) Sh_2^2$$

$$SS_2^2 = [0\ 1\ 1\ 1\ 0\ 0\ 1] (+) [(3, +1, 2); (6, +1, 1)]$$

$$SS_2^2 = [0\ 1\ 0\ 1\ 1\ 1\ 0]$$

$$Ch_2^2 = [1\ 3\ 7]$$

Perdas Ativas Totais: 0,0554 p.u.

c) Avaliação dos Valores P_{best} e G_{best}

Tabela 5.6 - Valores P_{best} - Iteração 2

Partícula	Iteração	Perdas Ativas	P_{best}_1
1	0	0,0528 p.u.	$(SS_1)^0 = [0\ 1\ 1\ 1\ 0\ 1\ 0]$
	2	0,1662 p.u.	
2	1	0,0727 p.u.	$(SS_2)^2 = [0\ 1\ 0\ 1\ 1\ 1\ 0]$
	2	0,0554 p.u.	

Tabela 5.7 - G_{best} - Iteração 2

Partícula	P_{best}	Perdas Ativas	G_{best}
1	$(SS_1)^0$	0,0528 p.u.	$(SS_1)^0 = [0\ 1\ 1\ 1\ 0\ 1\ 0]$
2	$(SS_2)^2$	0,0554 p.u.	

Conforme critério de parada estipulado para este exemplo, chegou-se à iteração 2 e, portanto ao fim do algoritmo.

Tabela 5.8 - Resultado Exemplo

Última iteração	G_{best}	Fitness
2	$(SS_1)^0 = [0\ 1\ 1\ 1\ 0\ 1\ 0]$	0,0528 p.u.

O valor de perda mínimo (fitness) encontrado pelo algoritmo foi de 0,0528 p.u., cuja configuração é dada pelas chaves 1, 5 e 7 abertas. Entretanto, este não é o valor mínimo de perdas dentre as 21 topologias deste exemplo. A configuração que apresenta perdas mínimas é, segundo Tabela 5.1, a topologia nº 4, dada pelas chaves 3, 4 e 7 abertas, e seu valor é 0,0362 p.u.

Sabe-se que, quanto maior o número de iterações, maior será a abrangência na varredura do algoritmo PSO pelo espaço de soluções em busca do ótimo e, conseqüentemente, melhor será a qualidade da solução apresentada.

5.2.3. Metodologia para Avaliação da Conectividade da Rede

Cada solução gerada produz uma topologia da rede que passa por uma rotina de avaliação de conectividade. No momento em que é gerada uma nova solução, é feita a atualização da Matriz A de acordo com sua topologia, isto é, para cada solução tem-se uma configuração que caracteriza sua conectividade.

Suponha que foram geradas as seguintes soluções ineficazes:

a) Ineficácia: Barra ilhada

Chaves abertas: $Ch_a = [4 \ 5 \ 7]$

Barra 4 ilhada (sem alimentação).

b) Ineficácia: Ramo ilhado

Chaves abertas: $Ch_b = [4 \ 5 \ 6]$

Barras 4 e 5 conectadas (linha 7), porém isoladas do restante do sistema (sem alimentação).

Serão detalhadas a seguir cada uma destas soluções de acordo com o tipo de ineficácia apresentada.

✓ Verificação de ineficácia - Barra Ilhada

Chaves abertas: $Ch_a = [4 \ 5 \ 7]$

Tabela 5.9 - Topologia - Chaves 4, 5 e 7 abertas

Linha	Barra inicial	Barra final
4	2	4
5	3	4
7	4	5

$$Aa_{24} = Aa_{42} = 0$$

$$Aa_{34} = Aa_{43} = 0$$

$$Aa_{45} = Aa_{54} = 0$$

$$Aa_{11} = | Aa_{12} + Aa_{13} + Aa_{14} + Aa_{15} | = 2$$

$$Aa_{22} = | Aa_{21} + Aa_{23} + Aa_{24} + Aa_{25} | = 2$$

$$Aa_{33} = | Aa_{31} + Aa_{32} + Aa_{34} + Aa_{35} | = 3$$

$$Aa_{44} = | Aa_{41} + Aa_{42} + Aa_{43} + Aa_{45} | = 0$$

$$Aa_{55} = | Aa_{51} + Aa_{52} + Aa_{53} + Aa_{54} | = 1$$

Logo, tem-se:

$$Aa = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & \mathbf{2} & -1 & \mathbf{0} & 0 \\ -1 & -1 & \mathbf{3} & \mathbf{0} & -1 \\ 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & -1 & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

Observe que o elemento da diagonal principal, Aa_{44} , é nulo. Isto significa que este valor representa o número total de barras a que a barra 4 está conectada e, portanto, conclui-se que a barra 4 está ilhada.

Sendo assim, uma das condições a serem avaliadas para a aprovação da solução é fazer a verificação dos "zeros" na diagonal principal da matriz de conectividade, ou seja, caso sejam detectados valores nulos, a solução é dispensada.

✓ Verificação de infactibilidade - Ramo Ilhado

Chaves abertas: $Ch_b = [4 \ 5 \ 6]$

Tabela 5.10 - Topologia - Chaves 4, 5 e 7 abertas

Linha	Barra inicial	Barra final
4	2	4
5	3	4
6	3	5

$$Ab_{24} = Ab_{42} = 0$$

$$Ab_{34} = Ab_{43} = 0$$

$$Ab_{35} = Ab_{53} = 0$$

$$Ab_{11} = | Ab_{12} + Ab_{13} + Ab_{14} + Ab_{15} | = 2$$

$$Ab_{22} = | Ab_{21} + Ab_{23} + Ab_{24} + Ab_{25} | = 2$$

$$Ab_{33} = | Ab_{31} + Ab_{32} + Ab_{34} + Ab_{35} | = 2$$

$$Ab_{44} = | Ab_{41} + Ab_{42} + Ab_{43} + Ab_{45} | = 1$$

$$Ab_{55} = | Ab_{51} + Ab_{52} + Ab_{53} + Ab_{54} | = 1$$

Logo, tem-se:

$$Ab = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

Neste caso, temos duas barras isoladas do restante do sistema, conectadas por uma linha. Sendo assim, esta infactibilidade não seria detectada facilmente como no caso anterior.

Nesta situação, ocorre que as duas barras, ditas como extremas, possuem conectividade entre si. Barra extrema ou extremo, é considerado, neste trabalho, como sendo uma barra que possui somente uma linha/barra conectada a ela. Desta forma, sempre que houver uma conectividade direta entre dois extremos, significa que há um ilhamento do ramo representado por eles e sua linha de conexão.

Cabe ressaltar que, caso um dos extremos for a subestação (barra 1, por convenção), não há condição de ilhamento, pois ela alimenta o sistema. Portanto, a barra 1 não está incluída na busca por extremos.

A verificação desta infactibilidade consiste, então, na detecção de todos os extremos do sistema mediante a verificação de valores "1" na diagonal principal da matriz de conectividade, e posterior verificação de conectividade entre eles. Portanto, caso dois extremos possuam conectividade entre si, a solução é descartada.

Esta verificação serve apenas para ramos simples, isto é, ramos que não possuem outras ramificações, que são os possíveis casos deste exemplo devido ao seu pequeno porte. Logo, somente estas duas rotinas descritas acima são suficientes para garantia da factibilidade das soluções deste sistema de 5 barras. Para sistemas maiores e, portanto, mais complexos, é necessária análise mais elaborada. Sendo

assim, a seguir, será detalhada a verificação de ilhamento para um conjunto de ramos (árvore), cuja avaliação é um pouco mais complexa.

✓ Verificação de infactibilidade - Árvore Ilhada

Supondo que, em um sistema hipotético, foi gerada a seguinte configuração:

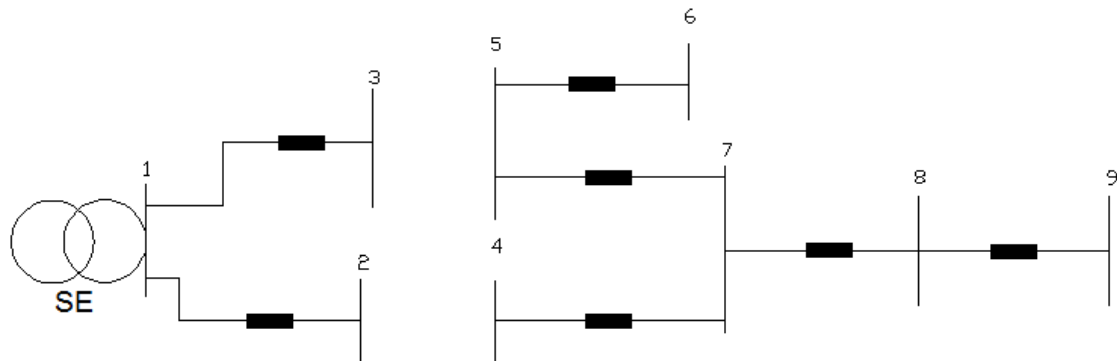


Figura 5.2 - Árvore Ilhada

A árvore ilhada corresponde à parte do sistema que compreende as barras 4, 5, 6, 7, 8 e 9.

O procedimento de verificação desta infactibilidade consiste em:

- i. Partindo de cada extremo, percorrer caminhos na rede até que se encontre uma barra cujo número de barras ligadas à ela é maior ou igual a 3, ou seja, uma barra que possui ramificações. Percorrer caminhos significa que, ao encontrar uma barra com número de barras ligadas à ela igual a 2, é o único caminho possível, logo não há ramificações. Caso a barra encontrada apresente número de barras ligadas à ela igual a 1, caise no mesmo caso analisado anteriormente (ramo ilhado), logo é detectada a infactibilidade imediatamente.
- ii. Quando todos os caminhos forem traçados, partir das barras cujo número de barras ligadas a ela é maior ou igual a 3 (barras de ramificação), de modo a verificar se, partindo dela, há caminhos que chegam à outras barras de ramificação, extremo ou subestação (barra 1). Caso se encontre uma barra de ramificação, o processo é realizado novamente. Caso se encontre um extremo, significa que a árvore está ilhada. Caso se encontre a subestação, significa que a árvore está alimentada.

Este processo é realizado até que não reste nenhum caminho a ser percorrido. Portanto, a cada caminho é feita uma atualização da matriz conectividade, de modo a indicar somente os caminhos restantes a serem percorridos.

Resumindo, a solução será factível quando caminhos iniciados dos extremos não encontram extremos e sim, barras com ramificações ou a subestação. Da mesma forma, dando continuidade, deseja-se também que caminhos iniciados das barras de ramificação não encontrem extremos e sim, outras barras de ramificação ou a subestação, e assim por diante.

Ao final da execução desta rotina, é retornado valor ao algoritmo PSO para indicação da factibilidade da solução. Conforme comentado no início deste capítulo, é feita modificação no procedimento do PSO para casos em que a solução é infactível. Esta modificação consiste em recalcular aleatoriamente o valor da inércia (w) do operador da iteração corrente e recalcular uma nova solução, partindo da mesma solução de origem (iteração anterior).

Portanto, só serão enviadas à função para cálculo de fluxo de potência, as configurações factíveis.

5.3. Considerações Finais

A metodologia utilizada apresenta vantagens por tratar somente de soluções factíveis, garantindo que será encontrada uma topologia da rede que atende aos requisitos mínimos de operação, isto é, garante que o sistema é radial e que todas as barras de carga (consumidores) estão alimentados pela fonte de injeção de potência do sistema, a subestação.

As adaptações feitas no algoritmo base foram feitas no intuito de solucionar as diversas dificuldades encontradas em relação ao tratamento de soluções. O algoritmo PSO binário, é de fácil implementação computacional, evidenciando sua eficácia também em termos de eficiência computacional. A complexidade da implementação se deu, principalmente, na filtragem das soluções factíveis, que demandou maior esforço computacional para execução do programa.

Foi testado, além do sistema de 5 barras apresentado neste capítulo, um sistema de 14 barras, que será apresentado no capítulo que contempla os resultados deste trabalho.

6. RESULTADOS

Neste capítulo são apresentados e comentados os resultados da reconfiguração para os sistemas de 5 e 14 barras.

- Sistema 5 Barras

Este sistema foi ilustrado no capítulo 5 e seus dados apresentados no apêndice A. Foram realizadas 20 execuções para cada valor de iteração máxima escolhido para verificação de desempenho do algoritmo.

O algoritmo obteve o resultado esperado de perdas iguais a 0,0362 p.u., em cerca de 80% dos testes, para um número de iterações igual a 25. Para valores entre 10 e 15 iterações, o algoritmo apresentou baixa qualidade nos resultados, isto é, nas diversas execuções do programa, grande parte das soluções ótimas apresentadas não era de 0,0362 pu.

Logo, conforme aumento do número de iterações, observou-se que uma porcentagem mínima das execuções apresentou valores diferentes do esperado. Para o número de iterações igual a 35, cerca de 96% das execuções testadas convergiram para o ótimo.

- Sistema 14 Barras

A Figura 5.4 ilustra um sistema de 16 barras (Cinvalar et. al., 1988), que apresenta 3 subestações (alimentadores). Porém será simplificado neste trabalho a um sistema de 14 barras, reduzindo os três alimentadores a somente uma barra, conforme ilustrado na Figura 5.5.

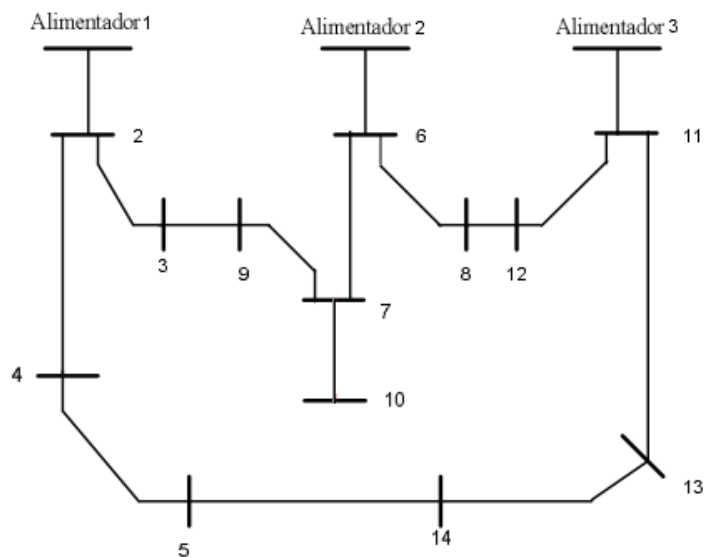


Figura 5.4 - Sistema 16 Barras Original²

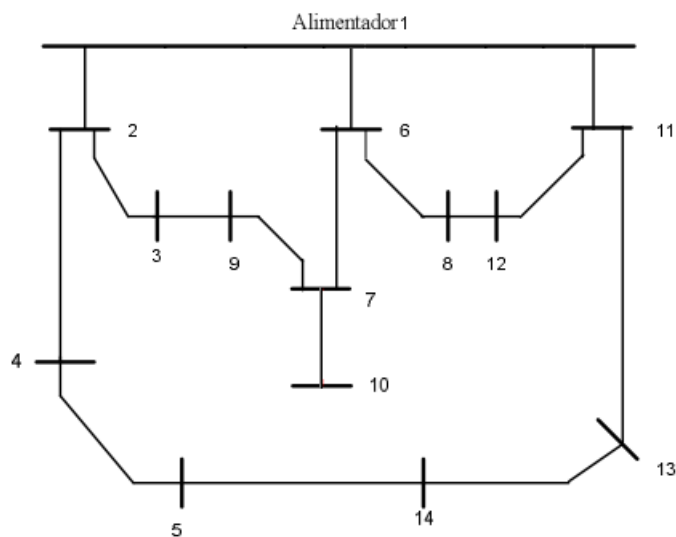


Figura 5.5 - Sistema 14 Barras²

Este sistema de 14 barras apresenta um total de 16 chaves. Sendo portanto, necessária a abertura de 3 delas para o início do algoritmo.

A partir deste sistema foi possível fazer a implementação da rotina de verificação de infactibilidades para o caso de árvores ilhadas. Foi possível assegurar bons resultados para um número de iterações superior a 120. Para um número de iterações igual a 130, obteve-se 92% dos casos convergindo ao ótimo, porém observou-se um aumento significativo no tempo de convergência.

²Figura retirada de (OLIVEIRA, 2011)

A seguir, tem-se um comparativo entre os resultados de alguns métodos aplicados à este sistema para reconfiguração visando minimização de perdas, com o proposto.

Tabela 6.1 - Resultados RSDEE Literatura

Métodos	Perdas Ativas Totais (kW)	Chaves abertas
PSO - Binário	466,69	7, 8 e 16
ACS - R	466,13	7, 8 e 16
Cinvalar	486,88	7, 14 e 16
Guimarães	466,13	7, 8 e 16

A avaliação da qualidade da solução deste método é dependente do número de iterações, pois se trata de um método cujas características de busca são parcialmente aleatórias. E, mesmo assim, não se tem uma relação exata do número de iterações com a porcentagem de acertos, pois não é possível direcionar o espaço de busca neste problema. Sendo assim, para um número elevado de simulações dificilmente será possível obter 100% de acerto nesta proposta de algoritmo, embora possam ser feitas melhorias em relação ao método no sentido de aumentar sua porcentagem de acerto, apresentando um novo desafio para desenvolvimento de novos trabalhos nesta área de estudo.

7. CONCLUSÃO

O objetivo deste trabalho foi realizar a reconfiguração de sistemas de distribuição de energia elétrica, para minimização de perdas ativas, através da implementação de um algoritmo binário de otimização por enxame de partículas (BPSO). PSO, originalmente proposto para aplicação em problemas de natureza contínua, sofreu modificações ao longo dos anos, possibilitando também, sua aplicação a problemas de natureza discreta e binária. Neste contexto, foi feita então, implementação de um algoritmo PSO binário para a reconfiguração de sistemas de distribuição.

A metodologia proposta neste trabalho considerou que todos as linhas do sistema possuem uma chave de interconexão, aumentando assim, o espaço de possíveis soluções e simplificando a implementação computacional. As soluções consistem em combinações binárias, onde cada bit representa o estado (aberto ou fechado) de cada uma das chaves do sistema.

Foram testados dois sistemas encontrados na literatura especializada, 5 e 16 barras, cujos resultados foram satisfatórios, porém com percentagem de erro associada, devido à aleatoriedade dos mecanismos de busca do método.

O método possui vantagens e desvantagens. A estratégia para busca de soluções (operador de deslocamento) apresenta uma vantagem pela facilidade de implementação computacional, sendo possível obter novas soluções de maneira rápida, porém não assegura a radialidade e conectividade da rede. A vantagem de se trabalhar somente com soluções factíveis, promove aumento da eficácia dos resultados obtidos, apesar de torná-lo mais lento.

O objetivo deste trabalho é, não só apresentar uma nova abordagem para a solução do problema de reconfiguração, mas trata-se também de uma ferramenta de estudos para melhorias desta proposta, bem como incentivo ao surgimento de novas abordagens para a reconfiguração e redução de perdas dos sistemas de distribuição.

REFERÊNCIAS BIBLIOGRÁFICAS

CARREÑO, E. M.; MOREIRA, N.; ROMERO, R. "Distribution network reconfiguration using an efficient evolutionary algorithm." *IEEE PES General Meeting, 2007*: 24 -28.

CARVALHO, M. R. "Estudo de Técnicas Eficientes para a Resolução do Problema de Fluxo de Potência para Sistemas de Distribuição Radial." Tese (Mestre em Engenharia Elétrica), Escola de Engenharia de São Paulo, USP - Universidade de São Paulo, São Carlos, 2005.

CHIANG, H. D.; JEAN-JUMEAU, R. "Optimal Network Reconfigurations in Distribution Systems: Part 2: Solution Algorithms and Numerical Results." *IEEE Trans. Power Delivery*, July 1990b: 1568-1574.

CHIANG, H. D.; JEAN-JUMEAU, R. "Optimal Network Reconfigurations in Distribution Systems: Part 1: A New Formulation and A Solution Methodology." *IEEE Trans. Power Delivery*, October 1990a: 1902-1909.

CIVANLAR, S.; GRAINGER, J. J.; LEE, S. S. H. "Distribution Feeder Reconfiguration for Loss Reduction," *IEEE Transaction on Power Delivery*, vol. 3, n. 3, pp. 1217–1223, 1988.

EBERHART, R. C.; KENNEDY, J. "Particle Swarm Optimization." *Proceedings of the International Conference on Neural Networks*, Perth, Australia, 1995, pp.1942-1948.

EBERHART, R.C.; KENNEDY, J. "A Discrete Binary Version of the Particle Swarm Algorithm." *In Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, vol. 5, 1997, pp.4104-4108.

FU-YUAN HSU and MEN-SHEN TSAI. "A New Binary Coding Particle Swarm Optimization for Feeder Reconfiguration." *International Conference on Intelligent Systems Applications to Power Systems - ISAP*, 2007.

GUIMARÃES, M. A. N. "Reconfiguração de Sistemas de Distribuição de Energia Elétrica utilizando algoritmos de Busca Tabu." Dissertação (Mestre em Engenharia Elétrica), Faculdade de Engenharia Elétrica e de Computação, UNICAMP - Universidade Estadual de Campinas, Campinas -SP, 2005, 108.

MERLIN, A. & BACK, G. "Search for Minimum – Loss Operational Spanning Tree Configuration for Urban Power Distribution System." *Proceeding 5th Power System Conference*, Cambridge, UK, September, 1975, pp. 1-18.

MONTICELLI, A. J. Fluxo de carga em redes de energia elétrica. São Paulo: Edgard Blucher Ltda., 1983.

OLIVEIRA, M. B. C. "Reconfiguração de alimentadores em sistemas de distribuição usando a metaheurística GRASP." Dissertação (Mestrado) - Faculdade de Engenharia de Ilha Solteira, Universidade Estadual Paulista, Ilha Solteira, 2011.

PEREIRA, F. S. "Reconfiguração Ótima de Sistemas de Distribuição de Energia Elétrica Baseado no Comportamento de Colônias de Formigas." Tese (Doutor em Engenharia Elétrica), Escola de Engenharia de São Paulo, USP - Universidade de São Paulo, São Carlos, 2010.

QUEIROZ, L. M. O.; FILHO, C. L. "Redução de Perdas por Reconfiguração de Redes Radiais de Distribuição de Energia Elétrica sob Demandas Variadas por um Algoritmo Genético Adaptativo." *Anais do XVI Congresso Brasileiro de Automática - CBA 2006*, Salvador - BA - Brasil, p. 2903-2908, 3-6 out. 2006.

SALAZAR, H.; GALLERO, R.; ROMERO, R. "Artificial Neural Networks and Clustering Techniques Applied in the Reconfiguration of Distribution Systems." *IEEE TRANSACTIONS ON POWER DELIVERY*, July 2006: 1735 - 1742.

APÊNCIDE A

DADOS DOS SISTEMAS DE DISTRIBUIÇÃO

A1 - Sistema de 5 barras

Tensão Subestação: $1,05 + j0,0$ (p.u.)

- Dados de Barras

Barra	Potência Ativa (p.u.)	Potência Reativa (p.u.)
1	0	0
2	1.28	1.28
3	0.32	0.16
4	1.6	0.8
5	0.74	0.37

- Dados de Linhas

Barra	Linhas		Resistência (p.u.)	Reatância (p.u.)
	Barra Origem	Barra Final		
1	1	2	0.0066	0.0033
2	1	3	0.0016	0.0006
3	2	3	0.0003	0.0002
4	2	4	0.0051	0.0005
5	3	4	0.0005	0.0005
6	3	5	0.0027	0.0012
7	4	5	0.0033	0.0015

A2 - Sistema de 10 barras

Tensão Subestação: 23kV

- Dados de Barras

Barra	Potência Ativa (W)	Potência Reativa (VAr)
1	0	0
2	1840000	460000
3	980000	340000
4	1790000	446000
5	1598000	1840000
6	1610000	600000
7	780000	110000
8	1150000	60000
9	980000	130000
10	1640000	200000

- Dados de Linhas

Barra	Linhas		Resistência (p.u.)	Reatância (p.u.)
	Barra Origem	Barra Final		
1	1	2	0.1233	0.4127
2	2	3	0.0140	0.6051
3	3	4	0.7463	1.2050
4	4	5	0.6984	0.6084
5	5	6	1.9831	1.7276
6	6	7	0.9053	0.7886
7	7	8	2.0552	1.1640
8	8	9	4.7953	2.7160
9	9	10	5.3434	3.0264

A3 - Sistema de 14 barras

Tensão Subestação: 23kV

- Dados de Barras

Barra	Potência Ativa (MW)	Potência Reativa (MVar)	Capacitor shunt (MVar)
1	0.000	0.000	0.000
2	2.000	1.600	0.000
3	3.000	1.500	1.100
4	2.000	0.800	1.200
5	1.500	1.200	0.000
6	4.000	2.700	0.000
7	5.000	3.000	1.200
8	1.000	0.900	0.000
9	0.600	0.100	0.600
10	4.500	2.000	3.700
11	1.000	0.900	0.000
12	1.000	0.700	1.800
13	1.000	0.900	0.000
14	2.100	1.000	1.800

- Dados de Linhas

Barra	Linhas		Resistência (p.u.)	Reatância (p.u.)
	Barra Origem	Barra Final		
1	1	2	0.0750	0.1000
2	2	3	0.0800	0.1100
3	2	4	0.0900	0.1800
4	4	5	0.0400	0.0400
5	1	6	0.1100	0.1100
6	6	7	0.0800	0.1100
7	6	8	0.1100	0.1100
8	7	9	0.1100	0.1100
9	7	10	0.0800	0.1100
10	1	11	0.1100	0.1100
11	11	12	0.0900	0.1200
12	11	13	0.0800	0.1100
13	13	14	0.0400	0.0400
14	3	9	0.0400	0.0400
15	8	12	0.0400	0.0400
16	5	14	0.0900	0.1200

A4 - Sistema de 34 barras

Tensão Subestação: 11kV

- Dados de Barras

Barra	Potência Ativa (W)	Potência Reativa (VAr)
1	0	0
2	230000	142500
3	0	0
4	230000	142500
5	230000	142500
6	0	0
7	0	0
8	230000	142500
9	230000	142500
10	0	0
11	230000	142500
12	137000	84000
13	72000	45000
14	72000	45000
15	72000	45000
16	13500	7500
17	230000	142500
18	230000	142500
19	230000	142500
20	230000	142500
21	230000	142500
22	230000	142500
23	230000	142500
24	230000	142500
25	230000	142500
26	230000	142500
27	137000	85000
28	75000	48000
29	75000	48000
30	75000	48000
31	57000	37500
32	57000	37500
33	57000	37500
34	57000	37500

- Dados de Linhas

Barra	Linhas		Resistência (Ω)	Reatância (Ω)
	Barra Origem	Barra Final		
1	1	2	0.1170	0.0480
2	2	3	0.1073	0.0440
3	3	4	0.1645	0.0457
4	4	5	0.1495	0.0415
5	5	6	0.1495	0.0415
6	6	7	0.3144	0.0540
7	7	8	0.2096	0.0360
8	8	9	0.3144	0.0540
9	9	10	0.2096	0.0360
10	10	11	0.1310	0.0225
11	11	12	0.1048	0.0180
12	3	13	0.1572	0.0270
13	13	14	0.2096	0.0360
14	14	15	0.1048	0.0180
15	15	16	0.0524	0.0090
16	6	17	0.1794	0.0498
17	17	18	0.1645	0.0457
18	18	19	0.2079	0.0473
19	19	20	0.1890	0.0430
20	20	21	0.1890	0.0430
21	21	22	0.2620	0.0450
22	22	23	0.2620	0.0450
23	23	24	0.3144	0.0540
24	24	25	0.2096	0.0360
25	25	26	0.1310	0.0225
26	26	27	0.1048	0.0180
27	7	28	0.1572	0.0270
28	28	29	0.1572	0.0270
29	29	30	0.1572	0.0270
30	10	31	0.1572	0.0270
31	31	32	0.2096	0.0360
32	32	33	0.1572	0.0270
33	33	34	0.1048	0.0180

A5 - Sistema de 70 barras

Tensão Subestação: 12,66kV

- Dados de Barras

Barra	Potência Ativa (W)	Potência Reativa (VAr)
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	2600	2200
8	40400	30000
9	75000	54000
10	30000	22000
11	28000	19000
12	145000	104000
13	145000	104000
14	8000	5500
15	8000	5500
16	0	0
17	45500	30000
18	60000	35000
19	60000	35000
20	0	0
21	1000	600
22	114000	81000
23	5300	3500
24	0	0
25	28000	20000
26	0	0
27	14000	10000
28	14000	10000
29	26000	18600
30	26000	18600
31	0	0
32	0	0
33	0	0
34	14000	10000
35	19500	14000
36	6000	4000
37	26000	18550
38	26000	18550
39	0	0
40	24000	17000

41	24000	17000
42	1200	1000
43	0	0
44	6000	4300
45	0	0
46	39220	26300
47	39220	26300
48	0	0
49	79000	56400
50	384700	274500
51	384700	274500
52	40500	28300
53	3600	2700
54	4350	3500
55	26400	19000
56	24000	17200
57	0	0
58	0	0
59	0	0
60	100000	72000
61	0	0
62	1244000	888000
63	32000	23000
64	0	0
65	227000	162000
66	59000	42000
67	18000	13000
68	18000	13000
69	28000	20000
70	28000	20000

- Dados de Linhas

Barra	Linhas		Resistência (Ω)	Reatância (Ω)
	Barra Origem	Barra Final		
1	1	2	0.0005	0.0012
2	2	3	0.0005	0.0012
3	3	4	0.0001	0.0001
4	4	5	0.0015	0.0036
5	5	6	0.0251	0.0294
6	6	7	0.3660	0.1864
7	7	8	0.3811	0.1941
8	8	9	0.0922	0.0470
9	9	10	0.0493	0.0251
10	10	11	0.8190	0.2707

11	11	12	0.1872	0.0619
12	12	13	0.7114	0.2351
13	13	14	10.300	0.3400
14	14	15	10.440	0.3450
15	15	16	10.580	0.3496
16	16	17	0.1966	0.0650
17	17	18	0.3744	0.1238
18	18	19	0.0047	0.0016
19	19	20	0.3276	0.1083
20	20	21	0.2106	0.0696
21	21	22	0.3416	0.1129
22	22	23	0.0140	0.0046
23	23	24	0.1591	0.0526
24	24	25	0.3463	0.1145
25	25	26	0.7488	0.2475
26	26	27	0.3089	0.1021
27	27	28	0.1732	0.0572
28	3	29	0.0044	0.0108
29	29	30	0.0640	0.1565
30	30	31	0.3978	0.1315
31	31	32	0.0702	0.0232
32	32	33	0.3510	0.1160
33	33	34	0.8390	0.2816
34	34	35	17.080	0.5646
35	35	36	14.740	0.4873
36	4	37	0.0044	0.0108
37	37	38	0.0640	0.1565
38	38	39	0.1053	0.1230
39	39	40	0.0304	0.0305
40	40	41	0.0018	0.0021
41	41	42	0.7283	0.8509
42	42	43	0.3100	0.3623
43	43	44	0.0410	0.0478
44	44	45	0.0092	0.0116
45	45	46	0.1089	0.1373
46	46	47	0.0009	0.0012
47	5	48	0.0034	0.0084
48	48	49	0.0851	0.2083
49	49	50	0.2898	0.7091
50	50	51	0.0822	0.2011
51	9	52	0.0928	0.0473
52	52	53	0.3319	0.1114
53	10	54	0.1740	0.0886
54	54	55	0.2030	0.1034
55	55	56	0.2842	0.1447
56	56	57	0.2813	0.1433
57	57	58	15.900	0.5337

58	58	59	0.7837	0.2630
59	59	60	0.3042	0.1006
60	60	61	0.3861	0.1172
61	61	62	0.5075	0.2585
62	62	63	0.0974	0.0496
63	63	64	0.1450	0.0738
64	64	65	0.7105	0.3619
65	65	66	10.410	0.5302
66	12	67	0.2012	0.0611
67	67	68	0.0047	0.0014
68	13	69	0.7394	0.2444
69	69	70	0.0047	0.0016