XVIII Seminário Nacional de Distribuição de Energia Elétrica

SENDI 2008 - 06 a 10 de outubro

Olinda - Pernambuco - Brasil

Algoritmo Evolutivo Eficiente para Restabelecimento de Energia em Sistemas de Distribuição de Energia Elétrica

Augusto Cesar dos Santos
ETF-TO/EESC-USP
augusto@etfto.gov.br

Alexandre C. B. Delbem
ICMC-USP
acbd@icmc.usp.br

Newton G. Bretas
EESC-USP
ngbretas@sel.eesc.usp.br

Palavras-chave: Algoritmos Evolutivos, Estrutura de Dados, Representação Nó-Profundidade, Restabelecimento de Energia, Sistemas de Distribuição.

RESUMO

Interrupção no fornecimento de energia elétrica em sistemas de distribuição é um problema constante das concessionárias. Ocorre tanto para manutenção corretiva, quanto para manutenção preventiva. O tempo indisponível do serviço influencia diretamente na qualidade da energia fornecida, portanto é de interesse da concessionária e do consumidor que o mesmo seja re-estabelecido o quanto antes. A meta é obter um plano de restabelecimento para suprir áreas que ficaram sem energia após uma falta ter sido identificada e isolada. Algumas restrições devem ser atendidas: rede radial, limite de queda de tensão, carregamento da rede e das subestações. Além disso, buscam-se configurações que reduza o total de perdas resistivas e o total de chaveamentos. Assim, o problema possui múltiplos objetivos sendo alguns conflitantes, aumentando seu grau de complexidade. Este trabalho propõe um Algoritmo Evolutivo que produz somente configurações factíveis e utiliza uma estrutura de dados eficiente para manipular grafos. A mesma produz ótimos planos de restabelecimento para sistemas de distribuição de grande porte. Com essa estrutura de dados houve uma redução no tempo de processamento, mesmo utilizando computadores convencionais, pois necessita de pouca memória RAM. Testes serão realizados no sistema de distribuição reduzido de São Carlos-SP com 1473 barras, 249 chaves.

1. INTRODUÇÃO

Devido à ocorrência de uma falta em alguma parte do sistema de distribuição, dispositivos de segurança atuam impedindo que energia chegue até a área afetada. Esse procedimento deixa os consumidores que se encontram a jusante da área em falta sem energia também. Para evitar que áreas que não estejam com problemas fiquem sem energia, deve-se: identificar o local da falta, isolar o setor em falta abrindo todas as chaves que o conecta ao circuito e, restabelecer energia aos setores a jusante do setor em falta. Esses setores devem ser transferidos para outro alimentador sem violar algumas restrições: rede radial, queda de tensão, carregamento da rede e, carregamento das subestações. Além disso, buscam-se configurações que reduza o total de perdas resistivas e operações de chaveamento.

Para Sistemas de Distribuição Radiais (SDRs) de grande porte, a quantidade de chaves pode tornar o problema computacionalmente intratável. A fim de resolver este problema,

vários Algoritmos Evolutivos (AEs) têm sido propostos 1, 2, 3. Os resultados obtidos têm encorajado pesquisas nesta área, principalmente quando comparados com Programação Matemática (PM) 4 e técnicas de Inteligência Artificial 5 em sua forma convencional de representação. AEs em sua representação convencional tem tido um desempenho relativamente bom para redes de pequeno e médio porte. Porém, quando aplicado a sistemas de grande porte, o tempo de processamento é muito elevado.

Em grandes redes, a codificação utilizada na representação de uma árvore de grafo para AEs é um fator crítico. Em um AE convencional, uma configuração da rede é representada por um único cromossomo (*string*) formando uma cadeia de bits (genes) contendo o estado de todas as chaves de seccionamento do sistema. Na tentativa de gerar novas configurações, o estado de duas ou mais chaves é alterado aleatoriamente. Isso produz muitas configurações infactíveis[1], e sua convergência, quando ocorre, é muito lenta.

Para superar esses problemas e aumentar o desempenho dos AEs, principalmente em redes de grande porte, este artigo adapta a codificação que tem apresentado melhor desempenho para projeto de redes. A representação é um *array* de pares (n_x,d_x), onde n_x é um nó do grafo e d_x sua profundidade na árvore. Juntamente com a Representação Nó-Profundidade (RNP) 6, um AE e dois operadores computacionalmente eficientes são desenvolvidos. Eles são capazes de modificar a topologia da rede sem gerar soluções infactíveis.

Com essa nova codificação é possível representar florestas e aumentar o desempenho do fluxo de carga. Isso possibilitou resolver problemas de reconfiguração em SDRs de tamanho real em um tempo relativamente pequeno. O desempenho obtido nos testes demonstra que o método pode produzir soluções eficientes, podendo ser utilizado em problemas que requerem soluções em tempo real.

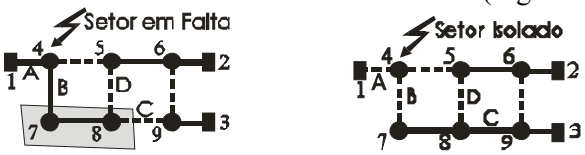
2. PROBLEMA DE RESTABELECIMENTO DE ENERGIA

2.1 Restabelecimento de Energia

Na ocorrência de uma falha, o setor em falta deve ser isolado do SDR e a área a jusante do mesmo conectada a outro alimentador. A Figura 1 mostra um SDR com três alimentadores. Retângulos indicam uma barra em uma subestação (SE), círculos representam setores[2], linha cheia representa chaves fechadas e, linha pontilhada, chaves abertas.

Figura 1. Sistema de Distribuição com três Alimentadores.

A Figura 2 mostra um exemplo de restabelecimento, supondo o setor 4 em falta. Esse setor foi isolado do SDR abrindo as chaves A e B. Os setores 7 e 8, a princípio sem energia (área destacada na Figura 2a), foram conectados ao alimentador 3 fechando a chave C (Figura 2b).



(a) SDR em falta. (b) Nova Configuração

Figura 2. Reconfiguração de um SDR.

[1] Configurações factíveis nesse contexto são as configurações radiais em que todos os consumidores são atendidos após o setor em falta ter sido isolado, ou seja, não apresentam *loops* ou interligação de um alimentador com outro.

[2] Setor é um grupo de barras e linhas não separadas por chaves.

Após a falta ter sido isolada e a área fora de serviço re-conectada ao sistema, as seguintes restrições devem ser atendidas 7:

- A estrutura de rede radial deve permanecer após o serviço de restabelecimento;
- Deve-se restabelecer energia às áreas a jusante do setor em falta;
- O total de cargas em cada alimentador não deve exceder a capacidade limite da subestação;
- A corrente elétrica em cada ramo deve respeitar seus limites permissíveis;
- A queda de tensão nas barras do sistema deve atender os limites impostos pela concessionária.

Além do restabelecimento de energia, mais dois objetivos geralmente são considerados:

Objetivo 1. Minimizar o total de perdas resistivas;

Objetivo 2. Minimizar o número de chaveamentos.

2.2 Formulação Matemática

Para obter a formulação geral de um problema de reconfiguração de SDRs, consideram-se juntos todos os objetivos e restrições.

O problema pode ser formulado como segue:

$$\begin{aligned} &Min. \quad E(F) \\ &s.a. \quad \begin{cases} H(F) = 0 \\ I(F) \leq 0 \\ F \text{ ser uma floresta}, \end{cases} \end{aligned} \tag{1}$$

onde,

- F – grafo correspondente a configuração do sistema, onde cada árvore corresponde a um alimentador do mesmo ligado a uma subestação;
- $E(F)$ – função objetivo;
- $H(F)$ – restrições de igualdade representando as equações de fluxo de carga;
- $I(F)$ – restrições de desigualdade representando as restrições operacionais do sistema.
- A função $E(F)$ contém, em geral, um ou mais dos seguintes componentes:
- $\phi(F)$ – quantidade de cargas fora de serviço para uma topologia radial de rede (uma floresta F);
- $\varphi(F)$ – perdas resistivas no sistema para F;
- $\psi(F,F_0)$ – número de operações de chaveamento para obter uma dada configuração F, a partir de uma configuração original F_0.

As restrições de igualdade correspondem às equações de fluxo de carga. Um sistema linear do tipo $Ax= b$ pode representá-las, onde:

- A – matriz incidência de F;
- x – vetor corrente de linha;
- b – vetor com as correntes (constantes) nas barras ($b_i \leq 0$), ou injeções de correntes nas subestações ($b_i > 0$).

As restrições operacionais de $I(F)$ para esses problemas geralmente incluem:

- um limitante superior de corrente $\overline{x}_j$ para cada corrente de linha x_j. A maior taxa $x_j / \overline{x}_j$ é denominada carregamento da rede;
- a máxima injeção de corrente $\overline{b}_i$ possível para cada subestação i, onde a maior taxa $b_i / \overline{b}_i$ é denominada carregamento da subestação;
- um limitante inferior para tensão. Seja v_i a tensão na barra i e v_b a tensão base no sistema; a menor relação v_i / v_b é denominada menor tensão relativa. O limite inferior para tensão relativa é $\underline{v}$[3].

[3] O vetor de tensão nos nós $\underline{v}$ é dado por $Yv = b$, onde Y é a matriz de admitância nodal ($Y = AY_xA^T$, com Y_x sendo a matriz de admitância diagonal).

Para SDRs, é adequado utilizar o modelo de corrente constante e ordenação das barras segundo o modelo pai-filho (MPF), da subestação para as barras terminais para o cálculo do fluxo de carga. Dessa forma, calculam-se os fluxos de corrente das barras terminais para a subestação e, então as tensões da subestação para as barras terminais. Detalhes do fluxo de carga podem ser vistos em 8, 9.

A função objetivo para problemas envolvendo SDRs geralmente é não linear, descontínua e com vários ótimos locais, dificultando a utilização de PM. Por outro lado, quando AEs são empregados para resolução desse tipo de problema, algumas modificações são realizadas na formulação apresentada em (1). A fim de penalizar as configurações da rede que violarem as restrições operacionais *I(F)*, são inseridos fatores de penalidades 10. Dessa forma, o problema pode ser reformulado como segue:

$$\begin{aligned} &Min. \quad E(F) + |\Omega I(F)| \\ &s.a. \quad \begin{cases} H(F) = 0 \\ F \text{ ser uma floresta}, \end{cases} \end{aligned} \tag{2}$$

onde Ω é uma matriz diagonal como segue:

$$w_{11} = \begin{cases} w_x, & se, \text{pelo menos para um } j, x_j > \overline{x}_j \\ 0, & \text{caso contrário}; \end{cases}$$

$$w_{22} = \begin{cases} w_s, & se, \text{pelo menos para um } i, b_i > \overline{b}_i \\ 0, & \text{caso contrário}; \end{cases}$$

$$w_{33} = \begin{cases} w_v, & se, \text{pelo menos para um } i, v_i < \underline{v} \\ 0, & \text{caso contrário}; \end{cases}$$

os pesos w_x, w_s e w_v são valores positivos e $|\cdot\ |$ é a norma infinito usual 11.

A formulação do problema anterior pode ser simplificada adaptando a RNP e seus operadores para SDRs. O AE proposto utilizando a RNP produz somente configurações factíveis. Assim, o problema descrito em (2) pode ser reescrito como segue:

$$\begin{aligned} &Min. \quad E(F) + |\Omega I(F)| \\ &s.a. \quad \begin{cases} H(F) = 0 \\ F \text{ ser dado pelos operadores da RNP}. \end{cases} \end{aligned} \tag{3}$$

A RNP de um SDR possui naturalmente as barras de cada árvore (alimentador) ordenadas segundo o MPF. Com isso, evita-se o uso de um algoritmo de busca 12 para obter tal modelo. O fluxo de carga pelo MPF com RNP é mais eficiente que fluxos de carga convencionais para SDRs. Além disso, o uso do MPF garante que as restrições de igualdade (*H(F)*) em (3) são satisfeitas. O problema para SDR pode ser reescrito com a RNP:

$$\begin{aligned} &Min. \quad E(F) + |\Omega I(F)| \\ &s.a. \quad \begin{cases} F \text{ ser dado pelos operadores da RNP} \\ \text{Utilizar MPF com RNP}. \end{cases} \end{aligned} \tag{4}$$

O tempo de processamento é reduzido quando se utiliza o MPF com RNP. Isso ocorre devido a produção exclusiva de configurações factíveis para SDRs e, pela avaliação relativamente rápida de cada configuração por um fluxo de carga mais eficiente. Assim, não há a necessidade de utilizar uma rotina específica para verificar nem corrigir infactibilidades.

A utilização da RNP e seus operadores juntamente com o fluxo de carga pelo MPF com RNP tornaram a modelagem matemática do problema mais simples. Compare (4) com (1).

3. REPRESENTAÇÃO NÓ-PROFUNDIDADE

Um SDR pode ser representado por grafos e deve ser factível. Portanto, um grafo para representar SDRs deve ser conexo e acíclico 13.

Um sistema de distribuição pode ser visto como um conjunto de alimentadores. Cada alimentador é composto de um ou mais setores.

A Figura 3 apresenta uma floresta com duas árvores. Os nós 1 e 2 são os nós raízes das árvores 1e 2 respectivamente. O Grafo da Figura 3 pode ser visto como um SDR com 2 alimentadores onde os nós são setores e as arestas são chaves seccionadoras. As arestas em linha cheia representam as chaves fechadas e, as arestas em linha pontilhada representam as chaves abertas.

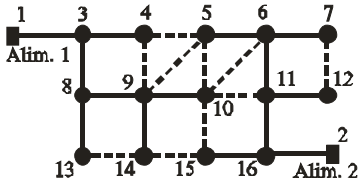


Figura 3. SDR com dois alimentadores. Corresponde a um Grafo com duas árvores.

A codificação proposta é baseada no conceito de nó e profundidade de nó em uma árvore de grafo. A representação consiste basicamente de uma lista linear contendo os nós e suas respectivas profundidades na árvore. A ordem em que os pares (nó, profundidade) são dispostos na lista é importante. Uma ordem apropriada para inserção dos pares pode ser obtida por uma busca em profundidade 12 em um processo off-line. A Figura 4 apresenta a RNP para as árvores do grafo da Figura 3.

$$T_1 \begin{bmatrix} \text{Nó} \\ \text{Profundidade} \end{bmatrix} \begin{bmatrix} 1 & 3 & 4 & 8 & 9 & 10 & 14 & 13 \\ 0 & 1 & 2 & 2 & 3 & 4 & 4 & 3 \end{bmatrix}$$

$$T_2 \begin{bmatrix} \text{Nó} \\ \text{Profundidade} \end{bmatrix} \begin{bmatrix} 2 & 16 & 15 & 11 & 12 & 6 & 5 & 7 \\ 0 & 1 & 2 & 2 & 3 & 3 & 4 & 4 \end{bmatrix}$$

Figura 4. RNP para as árvores da Figura 3.

A codificação para uma floresta é composta pela união das codificações de todas as árvores da mesma. Assim, a estrutura de dados da floresta pode ser facilmente implementada utilizando um *array* de ponteiros, onde cada ponteiro indica a RNP de uma árvore da floresta.

4. OPERADORES

Esta seção apresenta dois operadores para gerar novas florestas utilizando a RNP. Ambos operadores geram uma floresta *F'* de um grafo *G* quando eles são aplicados a outra floresta *F* de *G*.

Os resultados produzidos pela aplicação de ambos operadores são similares. Dado uma floresta com duas árvores ou mais, a aplicação do operador 1 (ou 2) a essa floresta é equivalente a transferir uma sub-árvore de uma árvore T_{de} para uma outra árvore T_{para} da floresta. A diferença básica entre os operadores 1 e 2 é que ao aplicar o operador 1, a raiz da sub-árvore cortada de T_{de} será também a raiz dessa sub-árvore na nova árvore T_{para}. Ao aplicar o operador 2, a sub-árvore cortada terá um novo nó raiz que poderá ser qualquer nó da sub-árvore cortada diferente da raiz original.

Como resultados, o operador 1 pode produzir simples e pequenas mudanças nas árvores da floresta; enquanto o operador 2 pode gerar grandes e mais complexas alterações.

O operador 1 requer um conjunto de dois nós determinados previamente: o nó de poda (n_p) que indica a raiz da sub-árvore a ser transferida e, o nó adjacente (n_a), que é um nó em uma árvore diferente da árvore cortada (T_{de}) e também adjacente a n_p em *G*. Os nós adjacentes a n_p podem ser armazenados em uma lista de adjacência[4] 13.

O operador 2 requer um conjunto de três nós: o nó de poda (n_p), o nó adjacente (n_a) e o novo nó raiz (n_r) da sub-árvore a ser transferida. Uma maneira eficiente de determinar o conjunto de nós requerido pelos operadores pode ser visto em 14. A seguir apresenta-se o funcionamento dos operadores considerando que o conjunto de nós requerido foi determinado previamente.

[4] A lista de adjacência considera todas as conexões possíveis de cada nó, isto é, conexões referentes tanto a chaves abertas quanto fechadas. Desta forma, não altera de uma configuração para outra.

4.1 Operador 1

Considerando que a RNP foi desenvolvida utilizando *arrays* e, que os nós n_p e na foram previamente determinados, podemos assumir que os índices de n_p (i_p) e n_a (i_a) nos *arrays* T_{de} e T_{para} respectivamente também são conhecidos.

Do grafo da Figura 3 foram escolhidas as árvores T_1 e T_2 como as árvores T_{de} e T_{para} respectivamente, para aplicação do operador 1. Os nós n_p e n_a são, respectivamente, 9 e 5. O operador 1 pode ser descrito pelos seguintes passos (ver Figura 5):

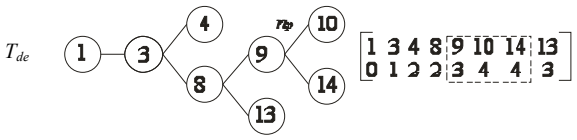
Determine as posições ($i_p - i_u$) dos índices em T_{de} correspondente a sub-árvore que contém o nó n_p como raiz. Como i_p é conhecido, é necessário encontrar apenas i_u, que corresponde ao índice do último nó na sub-árvore que tem o nó n_p como raiz. O conjunto ($i_p - i_u$) corresponde ao nó n_p em i_p e aos consecutivos nós n_x no array T_{de} de forma que $i_x > i_p$ e $p_x > p_p$ (as linhas pontilhadas na Figura 5a), onde p_x é a profundidade do nó n_x e p_p é a profundidade do nó n_p;

Copie os dados do conjunto ($i_p - i_u$) de T_{de} em um *array* temporário T_{tmp} (sub-árvore que está sendo transferida), ver Figura 5c. A profundidade de cada nó n_x do conjunto i_p–i_u é atualizado da seguinte maneira: $p_x = p_x - p_p + p_a + 1$;

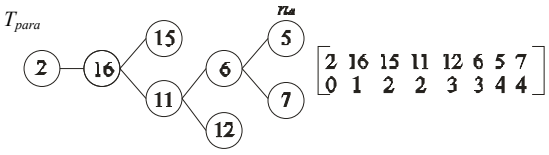
Crie um *array* T'_{para} contendo os dados de T_{para} e T_{tmp}, inserindo T_{tmp} a partir da posição i_a, ver Figura 5e;

Construa um *array* T'_{de} compreendendo os nós de T_{de} sem os nós de T_{tmp}, ver Figura 5d;

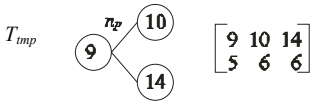
Copie a estrutura de dados da floresta F para F' trocando os ponteiros dos *arrays* T_{de} e T_{para} para os *arrays* T'_{de} e T'_{para}, respectivamente.



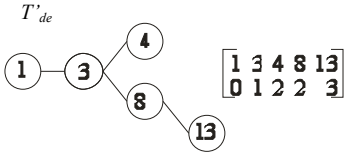
(a) T_{de}, e sua RNP.



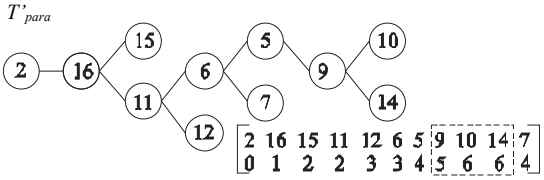
(b) T_{para} e sua RNP.



(c) T_{tmp} e sua RNP.



(d) T'_{de} e sua RNP.



(e) T'_{para} e sua RNP.

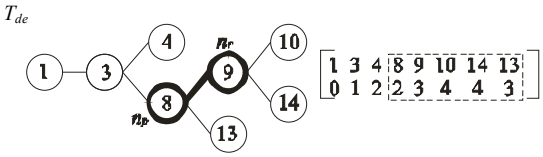
Figura 5. Exemplo de aplicação do operador 1.

4.2 Operador 2

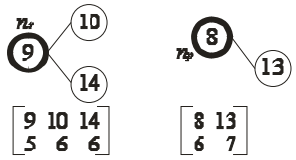
As diferenças entre o operador 1 e operador 2 estão nos passos 2 e 3 do procedimento do operador 1 (ver a sub-seção 4.1), isto é, a formação da sub-árvore cortada e o armazenamento da mesma no *array* temporário T_{tmp} são diferentes.

Para o operador 2 o procedimento de copiar a árvore cortada pode ser separado em dois passos: o primeiro passo é similar ao passo 2 do operador 1, diferindo apenas na troca de i_p por i_r. O segundo passo considera os nós no caminho de n_r a n_p (isto é, $r_0, r_1, r_2, ..., r_n$, onde $r_0 = n_r$ e $r_n = n_p$) como raízes das sub-árvores (linhas destacadas na Figura 6a). O algoritmo para o segundo passo deverá copiar a sub-árvore com a raiz r_i (i = 1, ..., n) sem a sub-árvore com raiz r_{i-1} (Figura 6b) e armazenar a sub-árvore resultante em um *array* temporário T_{tmp}, ver Figura 6c. O operador 2 utiliza T_{tmp} para construir T'_{para}.

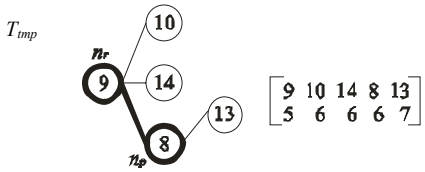
As árvores T_{de} e T_{para} utilizadas na aplicação do operador 2 são as mesmas utilizadas na Sub-seção 4.1. Os nós n_p e n_r são, respectivamente 8 e 9.



(a) T_{de} e sua RNP.



(b) Sub-árvores enraizadas nos nós do caminho de n_r a n_p.



(c) RNP para a sub-árvore cortada.

Figura 6. Aplicação do operador 2. Em (a), as linhas em destaque realçam os nós na cadeia de n_r a n_p. As profundidades em (b) e (c) são relativas ao nó 5 como adjacente.

5. ALGORITMO EVOLUTIVO PROPOSTO

O AE proposto busca estratégias para restabelecimento de energia para SDRs de grande porte após a ocorrência de uma falta. Como a área fora de serviço (setores a jusante do setor em falta) é uma sub-árvore de *G*, aplica-se o operador 2 para conecta-la a algum alimentador, gerando o primeiro indivíduo da população. Esse indivíduo pode violar restrições de operação do sistema. Assim, buscam-se outras configurações que melhore a qualidade do mesmo. O AE inicia sua busca por melhores soluções com apenas uma configuração factível e, a partir dela, produz outras melhores.

Dependendo do mecanismo de seleção adotado, os AEs podem ser atraídos para ótimos locais. Para resolver este problema, o AE proposto utiliza uma população para cada objetivo do problema e outra envolvendo todos os objetivos multiplicados por fatores de ponderação. As populações consideradas são: (i) perdas ôhmicas; (ii) queda de tensão; (iii) carregamento da rede; (iv) carregamento das subestações; (v) função ponderada.

Alguns parâmetros devem ser estabelecidos:

- Tamanho de cada população t_{pi}.

- Número de gerações g_{max}. Neste trabalho g_{max} é usado também como regra de parada.

Soluções geradas pelo AE podem ser armazenadas nas populações ou descartadas, dependendo do grau de adaptação do indivíduo a cada objetivo do problema. Inicialmente, existe apenas um indivíduo em cada população $_i$ (corresponde à configuração original F_0). Novos indivíduos são gerados e adicionados às populações até que atinja t_{pi}. O processo de gerar e selecionar novos indivíduos continua até atingir g_{max}. A seguir é apresentado um pseudocódigo para o AE proposto:

Algoritmo V.1. Pseudocódigo para o AE proposto

ALGORITMO proposto AE(F_0)

// inicia o contador de gerações

g := 0;

// gera populações iniciais $P_i(g_0)$ a partir da floresta original F_0

$P(g)$:= POP_INIC($P_i(g_0,F_0)$);

// avalia os indivíduos das populações iniciais

AVALIE ($P_i(g)$);

// teste do critério de parada (g_{max})

ENQUANTO critério de parada não atingido **FAÇA**

    // aleatoriamente selecione uma população (P_i).

    P_i := ALEATORIAMENTE-SELECIONE(pop.);

    // aleatoriamente selecione um indivíduo (F_s) de P_i.

    F_s := ALEATORIAMENTE-SELECIONE($P_i(g)$);

    // decide se aplica operador 1 ou operador 2

    OP :=DECIDE_OPERADOR(op1, op2);

    // aplicar OP para produzir um novo indivíduo F_g de F_s

    F_g := OP(F_s);

    // avaliação do novo indivíduo F_g

    AVALIE(F_g);

    // selecione sobreviventes entre $P(g)$ e F_g

    $P(g + 1)$:= ALTER_POP($P(g),F_g$);

    //incrementa o contador de gerações

    g : = g + 1;

FIM

Para a seleção de novos indivíduos, cada configuração é avaliada. Se tiver melhor fitness que qualquer outro indivíduo da população *i*, o mesmo é adicionado à população, descartando o indivíduo com o pior *fitness*.

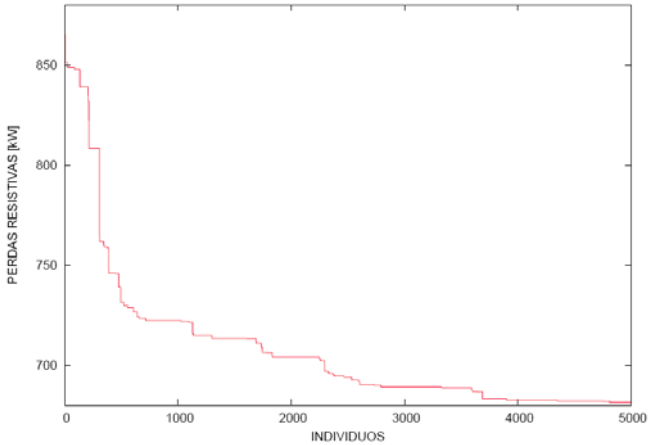
6. RESULTADOS EXPERIMENTAIS

Nesta seção, o método proposto é aplicado a um sistema de distribuição de grande porte e resultados experimentais são mostrados. O método foi programado utilizando um computador pessoal com processador core 2 duo 1,86 GHz, 2GRAM de memória e sistema operacional Linux, versão Ubuntu 7.10 e compilador C gcc.

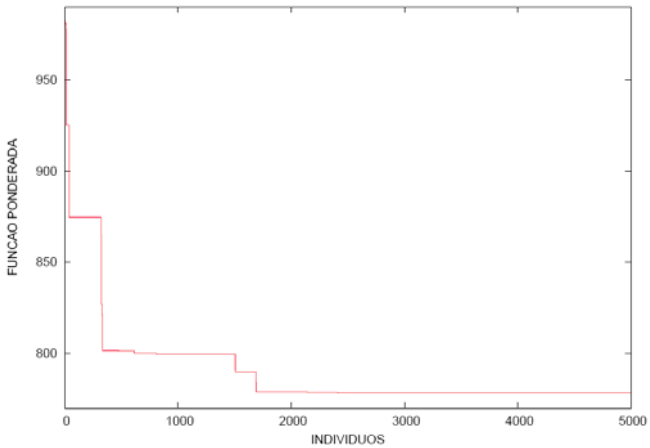
Testes foram realizados em um SDR da CPFL-Paulista com 1473 barras, 204 setores, 249 chaves, 3 subestações e 23 alimentadores, sistema utilizado em 15. Foram utilizadas 5 populações para minimização de: (i) perdas resistivas; (ii) máxima queda de tensão; (iii) máximo carregamento do SDR; (iv) máximo carregamento das subestações; (v) função ponderada. A função ponderada utilizada consta de: $f(x) = \delta_1 x_1 + \delta_2 x_2 + \delta_3 x_3 + \delta_4 x_4 + \delta_5 x_5$,onde x_1, x_2, x_3, x_4 e x_5 são respectivamente perdas resistivas em kW, quantidade de operações de chaveamento, máximo carregamento da rede em p.u.,

máximo carregamento das subestações em p.u. e, máxima queda de tensão em p.u.; δ_i é o peso de cada objetivo x_i. Os parâmetros utilizados nos testes foram: g_{max}=5000, tamanho de cada população t_{pi} = 5. Os primeiros 1000 indivíduos são gerados pelo operador 1. Os restantes são gerados pelo operador 2.

Os testes foram realizados com o setor 122 em falta. Esse setor está conectado diretamente a uma subestação, deixando o maior alimentador do SDR fora de serviço. A Figura 7 mostra o desempenho do algoritmo em duas populações. Os gráficos mostram o *fitness* do melhor indivíduo em cada geração.



(a) População de Perdas Resistivas.



(b) População da Função Ponderada.

Figura 7. Desempenho do algoritmo.

A Tabela 1 mostra características do melhor indivíduo em cada população ao final do processo. Isto é, cada linha da Tabela 1 apresenta as características do melhor indivíduo para uma população específica. A Tabela 2 mostra algumas estatísticas para 30 execuções quando o setor 122 é isolado. A função ponderada foi utilizada com os seguintes pesos:

$$\delta_1 = 1;$$

$$\delta_2 = 4;$$

$$\delta_3 = \begin{cases} 100, & se\ pelo\ menos\ para\ um\ j,\ x_j > 1pu \\ 0, & caso\ contrário; \end{cases}$$

$$\delta_4 = \begin{cases} 100, & se\ pelo\ menos\ para\ um\ i,\ b_i > 1pu \\ 0, & caso\ contrário; \end{cases}$$

$$\delta_5 = \begin{cases} 100, & se\ pelo\ menos\ para\ um\ i,\ v_i < 0.9pu \\ 0, & caso\ contrário. \end{cases}$$

Tabela 1. Melhor indivíduo em cada população

População	Perdas Totais [kW]	Queda Tensão [%.]	Carregamento SDR [%.]	Carregamento SE [%]	Manobras
Ponderada	**710,32**	1,67	63,3	94,1	17
Perdas	**681,35**	1,42	76,1	95,9	41
Q. Tensão	682,08	**1,42**	76,1	95,9	41
Car. SDR	720,83	1,59	**58,8**	99,3	43
Car. SE	726,13	1,57	65,4	**92,8**	35

Tabela 2. Múltiplos Objetivos – Valores médios.

	Original	Otimizado
Perdas Totais na Rede	865 kW	707 kW
Queda de Tensão Máxima	3,13%	1,74%
Carregamento Máximo da Rede	86,2%	72%
Número de Manobras	0	08 pares
Tempo médio de processamento	0	0,13s

7. CONCLUSÕES

Restabelecimento de energia para SDRs é um problema de otimização combinatorial altamente complexo.Vários métodos tem sido propostos para lidar com este tipo de problema, porém não apresentam desempenho satisfatório quando utilizados em redes de distribuição reais, pois requerem alto custo computacional. A codificação do AE tem uma importância crítica no desempenho do algoritmo para esse tipo de sistema.

Para resolver esse problema, é proposta uma codificação para AEs baseada em nós e na profundidade de nós na árvore de grafo, juntamente com dois operadores que geram apenas configurações factíveis. A estrutura de dados desenvolvida acelerou os cálculos de fluxo de carga e aumentou o desempenho da técnica, possibilitando trabalhar com sistemas reais de grande porte.

O AE desenvolvido foi aplicado ao sistema simplificado da cidade de São Carlos-SP (CPFL-Paulista). Os resultados obtidos demonstram que a técnica proposta é capaz de lidar com restabelecimento de energia para grandes redes de distribuição, encontra uma rápida solução onde a energia é restabelecida á área fora de serviço, as restrições de operação do sistema são atendidas, as perdas totais no sistema são reduzidas. Além disso, devido ao ótimo desempenho computacional, pode ser aplicada a sistemas que requerem soluções em tempo real.

8. AGRADECIMENTOS

Os autores agradecem cordialmente ao CNPq e a FAPESP o suporte financeiro.

9. REFERÊNCIAS BIBLIOGRÁFICAS

1 DELBEM, A. C. B., CARVALHO, A. C. P. L. F., BRETAS, N. G. Energy restoration in distribution systems using search with fuzzy heuristics. Int. J. Eng. Intell. Syst. Elct. Eng. Commun., vol. 6, 1998, pp. 201-205.

2 AUGUGLIARO, A., DUSONCHET, L., RIVA SANSEVERINO, E. Multiobjective service restoration in distribution networks using an evolutionary approach and fuzzy sets. International Journal of Electrical Power & Energy Systems, vol. 22, 2000, pp. 103-110.

3 SU, C. T., LEE, C. S. Network reconfiguration of distribution systems using improved mixed-integer hybrid differential evolution. IEEE Transactions on Power Delivery, vol. 18, July, 2003, pp. 1022-1027.

4 NARA, K., MISHIMA, Y., GJYO, A., ITO, T., KANEDA, H. Loss minimum re-configuration of distribution system by tabu search. Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES, vol. 1, 2002, pp. 232-236.

5 MERLIN, BACK, H. Search for a minimal-loss operating spanning tree configuration in an urban power distribution system. Power System Computation Conference, 5, vol. 1, 1975, pp. 1-18.

6 DELBEM, A. C. B., CARVALHO, A. C. P. L. F., POLICASTRO, C. A., PINTO, A. K. O., HONDA, K., GARCIA, A. C. Node-Depth Encoding for Evolutionary Algorithms Applied to Network Design. 6th Genetic and Evolutionary Computation Conference – GECCO 2004, 2004, pp. 678-687.

7 INAGAKI, J., NAKAJIMA, J., HASEYAMA, M. A multiobjective service restoration method for power distribution systems. Proceedings IEEE International Symposium on Circuits and Systems, ISCAS 2006, may, 2006, pp. 1784-1787.

8 DELBEM, A. C. B., CARVALHO, A. C. P. L. F., BRETAS, N. G. Load Flow for Radial Distribution System using the Graph Chain Representation. In Proc. IEEE/PES T&D Latin America, CD-ROOM, 2002.

9 CHENG, C. S., SHIRMOHAMMADI, D. A three-phase power flow method for real-time distribution system analysis. IEEE Transactions on Power Systems, vol. 10, May, 1995, pp. 671-679.

10 GOLDBERG, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 1989.

11 GRADSHTEYN, I. S., RYZHIK, I. M. Tables of Integrals, Series, and Products. San Diego, CA: Academic Press, 6th ed, 2000.

12 CORMEM, T. H., LEISERSON, C. E., RIVEST, R. L., STEIN, C. Introduction to Algorithms. Cambridge, MA: MIT Press/McGraw Hill, 1990.

13 AHUJA, R. K., MAGNANTI, T. L., ORLIN, J. B. Network Flows: Theory, Algorithms, and Applications. Englewood Cliffs: Printce Hall, 1993.

14 SANTOS, A. C., DELBEM, A. C. B., BRETAS, N. G. Reconfiguração de Sistemas de Distribuição de Energia Elétrica utilizando Algoritmo Evolutivo com Representação Nó-Profundidade. Cong. Latino Americano de Ger. e Transm. de Energia Elétrica - CLAGTEE, Chile, 2007.

15 DELBEM, A. C. B., CARVALHO, A. C. P. L. F., BRETAS, N. G. Main chain representation for evolutionary algorithms applied to distribution system reconfiguration. IEEE Transactions on Power Systems, vol. 20, Feb., 2005, pp. 425-436.