
REPRESENTAÇÃO POR CADEIAS DE GRAFO PARA AG APLICADOS AO RESTABELECIMENTO DE ENERGIA ÓTIMO EM SISTEMAS DE DISTRIBUIÇÃO RADIAIS

N. G. Bretas A. C. B. Delbem A. de Carvalho

Escola de Engenharia de São Carlos - Universidade de São Paulo
Av. Dr. Carlos Botelho 1450, 13560-250, São Carlos, SP, Brasil
ngbretas@sel.eesc.sc.usp.br

Resumo: Os métodos existentes de restabelecimento de energia para uma área desenergizada em sistemas de distribuição o fazem somente para circuitos com características específicas. Além disso, eles não são capazes de lidar com grandes áreas desenergizadas. Este artigo propõe um método que usa Algoritmos Genéticos (AG) para restabelecimento de energia em sistemas de distribuição radiais. Para isto, uma nova representação dos sistemas de distribuição através de Cadeias de Grafo é utilizada. O AG trabalha com função objetivo não linear e não contínua (inerente a este tipo de problema). A nova representação evita a geração de configurações não factíveis, reduzindo consideravelmente, por sua vez, a característica combinatorial intrínseca do problema. Como consequência, a não geração de configurações não factíveis reduz significativamente o uso de memória RAM e tempo de processamento. O algoritmo proposto é testado em um sistema de distribuição razoavelmente grande e complexo.

Abstract: The available approaches to restore energy for an outage zone in distribution systems perform the restoration only for circuits with specific characteristics. Additionally, they are not able to deal with large outage zones. This paper proposes a method using Genetic Algorithms (GA) for energy restoration in radial distribution systems. This GA has a new representation for distribution systems (using Graph Chains). The GA cope with non-linear and non-continuous objective function (inherent to this kind of problem). The new representation avoids the generation of unfeasible configurations, overcoming the intrinsic combinatorial characteristic of the problem. As a consequence, the non-generation of unfeasible configurations saves RAM memory and processing time. The proposed approach is tested in a fairly complex distribution system.

1 INTRODUÇÃO

Um plano de restabelecimento de energia elaborado e rápido é necessário para lidar com faltas permanentes em sistemas de distribuição. Durante a operação normal do sistema, muitas

vezes é necessário isolar zonas para execução de trabalhos de manutenção na rede. Dentre os principais objetivos em um plano de restabelecimento de energia podem-se destacar: i) número reduzido de operações de chaveamento, ii) pequeno número de consumidores sem energia (ou nenhum) e iii) ausência de elementos sobrecarregados. Portanto, o restabelecimento de serviços aos consumidores é um problema com múltiplos objetivos, alguns deles conflitantes. Outros objetivos podem ser considerados.

O crescimento acelerado dos sistemas de distribuição em países em desenvolvimento, como o Brasil, tem gerado configurações de redes muito complexas. Por exemplo, tais sistemas não possuem em geral seções normalmente abertas (Devi, Gupta and Anandalingam, 1995) ou estruturas nas quais cada lateral de um alimentador pode sempre ser suprida por outros dois alimentadores principais (Hsu and Kuo, 1994), (Chen and Hsu, 1989).

Os Algoritmos Genéticos (AG) têm-se mostrado adequados ao problema de reconfiguração de sistemas de distribuição (Chen and Hsu, 1989), (Delbem and Bretas, 1998), (Miranda, Ranito and Proença, 1994), (Nara *et al.*, 1992), (Fukuyama and Ueki, 1995), (Yeh, Venkata and Sumic, 1996), (Pereira and Murari, 1998), (Lin, Su and Tsay, 1998) pelo fato de serem capazes de lidar com problemas de busca conjuntamente com o problema de otimização de funções não lineares (funções descontínuas). Contudo, deve-se lembrar que o problema de reconfiguração de sistemas é um problema de roteamento, isto é, um problema de caminho ótimo (Siqing *et al.* 1998), (Zbigniew, 1994). Para tal problema, os métodos existentes restringem-se a fazer otimizações locais a partir do uso de heurísticas. Também não há algoritmos que resolvam problemas de roteamento de dimensão grande e as soluções para dimensões maiores se restringem a aplicações não em tempo-real (Zbigniew, 1994).

Neste trabalho propõe-se um AG capaz de lidar com reconfiguração de sistemas de distribuição grandes e complexos e apropriado para aplicações em tempo-real. A seção 1.1 traz uma rápida revisão bibliográfica sobre restabelecimento de energia em sistemas de distribuição. Nas demais seções, o método de Algoritmo Genético (AG) com a Representação por Cadeias de Grafo (RCG) é introduzido e analisado.

Artigo Submetido em 25/10/99

1a. Revisão em 19/04/00; 2a. Revisão em 22/11/00

Aceito sob recomendação do Ed. Consultor Prof. Dr. José Luiz Rezende Pereira

1.1 Revisão Bibliográfica

As configurações dos sistemas de distribuição são normalmente representadas por um conjunto de estados de chaves. Essas configurações podem ser descritas pelo vetor:

$$V = [v_1 v_2 \dots v_n]^T \quad (1)$$

onde v_i indica se a chave está aberta ($v_i = 0$) ou fechada ($v_i = 1$). As propostas baseadas nesta representação (Morelato and Monticelli, 1989), (Hsu and Kuo, 1994), (Aoki *et al.*, 1989), (Castro, Bunch and Terry, 1980), (Chen and Hsu, 1989) podem gerar até 2^n combinações para encontrar uma possível configuração que forneça energia para a área desenergizada. Isso implica em um tempo de processamento muito grande, mesmo quando técnicas heurísticas são usadas.

Delbem *et al.* (Delbem, Bretas, and Carvalho, 1997), (Delbem and Bretas, 1998) propuseram um algoritmo de busca que não usa o espaço dos estados das chaves, e sim o espaço dos estados dos setores. Setores são trechos da rede entre chaves, veja Fig. 1, onde quadrados cheios são chaves fechadas e quadrados vazios são chaves abertas, números representam setores, os S_i 's são os nomes das chaves e SE significa subestação. Como o espaço dos estados dos setores é menor que o espaço dos estados das chaves, o tempo de processamento é significativamente reduzido (Delbem, Bretas, and Carvalho, 1997).

Outras propostas objetivam um restabelecimento de energia ótimo (Morelato and Monticelli, 1989), (Hsu and Kuo, 1994), (Wang, Chiang and Darling, 1996), (Liu, Lee and Venkata, 1988). Não obstante, todas essas propostas ou geram vetores V factíveis baseados em características locais do sistema e características da área desenergizada, ou geram um grande número de vetores V (todo o espaço dos estados). Como a geração e análise de um grande número de vetores V não é prática, a primeira alternativa tem sido a única maneira de se atingir um restabelecimento de energia ótimo.

É importante lembrar que o procedimento para gerar vetores V factíveis através de heurísticas é uma busca em profundidade que pode otimizar localmente o plano de restabelecimento de energia, mas que em geral não atinge a solução ótima (Winston, 1984). Além disso, a função objetivo para problemas de reconfiguração é uma função não linear e não contínua. Por conseguinte, parte dos vetores V factíveis não podem ser gerados pelos algoritmos baseados em heurísticas.

Diante desses problemas, alguns algoritmos para restabelecimento de energia usando AG foram propostos (Delbem, Bretas and Carvalho, 1997), (Fukuyama and Ueki, 1995), (Siqing *et al.* 1998). Isto pelo fato de serem capazes de gerar novas configurações do sistema não dependentes de características peculiares de cada sistema. Esses métodos são capazes de lidar com funções objetivo não lineares e não contínuas inerentes à reconfiguração de sistemas de distribuição. Assim utilizando de AG, um plano de restabelecimento de energia ótimo pode ser obtido. Entretanto, tais propostas são inviáveis se o sistema de distribuição for "grande" (neste artigo, um sistema será denominado como grande caso este tenha mais de 100 chaves após ter sido feito o agrupamento de setores em blocos contendo mais de um setor; tais blocos também são chamados setores).

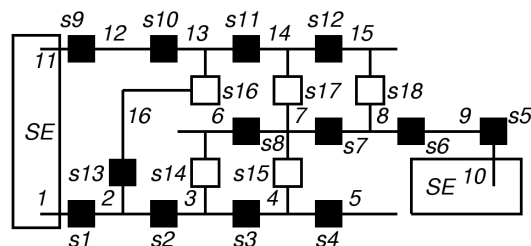


Fig. 1: Exemplo de sistema de distribuição.

1.2 Dificuldades do Uso de AG em Problemas de Reconfiguração de Sistemas Grandes

Como é mostrado em (Delbem, Bretas and Carvalho, 1997), os AG podem obter um restabelecimento de energia ótimo. Porém, para sistemas grandes, o algoritmo é lento (geram-se muitas configurações não factíveis pelos processos de "crossover" e mutação) e a exigência de memória RAM é muito grande.

Essas dificuldades são causadas em parte pela forma de representação do sistema. Uma das melhores formas de representar computacionalmente um sistema que pode ser interpretado como um grafo é a forma matricial, por exemplo, através de uma matriz incidência de nós, veja (Wilson, 1972), (Zbigniew, 1994). Com essa representação, n^2 posições de memória de computador são requeridas para se estocar uma configuração do sistema. Assim a memória requerida cresce quadraticamente com o tamanho do sistema.

Além disso, o processo de reconfiguração requer alterações nas matrizes que representam as configurações. Para tal, é preciso verificar parte das conexões existentes representadas em cada matriz. Essa verificação pode ser feita das seguintes formas:

1. executando uma busca na matriz incidência de nós;
2. resolvendo o sistema $Ax=b$, onde A é uma matriz incidência (Wilson, 1972) de nós $n \times n$, $b=[0 \dots 0 \ b_i \ 0 \dots 0]^T$ com $b_i \neq 0$. Assim, cada $x_j \neq 0$ indica que há um caminho de grafo (veja seção 7.2) que liga o nó i ao nó j ;
3. executando uma busca na lista de adjacências de nós.

Tanto (1) quanto (2) ou (3) requerem tempos de processamento grandes. Como, em geral, propostas baseadas em AG precisam gerar muitas configurações diferentes, essas formas de verificar as conexões podem consumir tempos abusivos de computação. Mesmo que métodos de esparsidade sejam utilizados para as formas (1) e (2), ainda assim ter-se-á de realizar buscas ou resoluções de equações lineares para se verificar conexões a cada nova configuração gerada.

Para superar essas barreiras, uma forma nova de representar a configuração do sistema no computador é proposta neste artigo.

Deve-se observar que as propostas de AG (veja o Apêndice 7.1) para problemas representados por grafos que tem melhor desempenho não usam, em geral, AG na sua forma convencional (Zbigniew, 1994). Deve-se observar também que AG para problemas de roteamento que possuem um melhor desempenho não possuem a representação padrão dos cromossomos nem os operadores na forma convencional (Zbigniew, 1994).

1.3 Representação de Sistemas de Distribuição através de Cadeias de Grafo (RCG)

Um grafo (veja a seção 7.2) de um sistema de distribuição pode ser obtido considerando suas chaves como os arcos do grafo e os trechos do sistema de distribuição entre chaves como os nós do grafo, isto é, setores (Delbem, Bretas, and Carvalho, 1997). O grafo de um sistema de distribuição é um grafo desconexo. Os alimentadores são os componentes do grafo do sistema de distribuição. Cada alimentador é composto por cadeias não-fechadas. Por exemplo, a matriz de cadeias de grafo do sistema da Fig. 1 é mostrada na Tabela 1 (onde as linhas representam cadeias de grafo; os setores 1, 10 e 11 estão conectados diretamente na subestação, assim os respectivos alimentadores são chamados de alimentador 1, alimentador 10 e alimentador 11).

Número da cadeia	Cadeia
1	1 2 3 4 5
2	1 2 16
3	10 9 8 7 6
4	11 12 13 14 15

Tabela 1: Cadeias de grafo do sistema da Fig. 1.

Uma mudança na configuração de um sistema feita pelo procedimento desenvolvido tem dois passos:

abra uma chave conectada ao setor_{*i*} (no exemplo da Fig. 1, $i \in \{2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 15, 16\}$) no alimentador_{*f*} (no exemplo da Fig. 1, $f \in \{1, 10, 11\}$) deixando desenergizada parte do alimentador_{*f*};

- i) feche uma chave que conecta o setor_{*j*} (um setor na parte do alimentador_{*f*} que está fora de serviço) ao alimentador_{*g*}, $g \in \{1, 10, 11 \mid g \neq f\}$.

A operação i) é semelhante à descrita no **Corolário 1** e a operação ii) é semelhante à operação mostrada no **Teorema 1** (veja o Apêndice 7.2). Em outras palavras, dado um grafo desconexo formado por componentes sem cadeias fechadas, após mudanças como as descritas no **Corolário 1** e **Teorema 1**, mantém-se um grafo desconexo formado por componentes sem cadeias fechadas, isto é, o universo de grafos desconexos sem cadeias fechadas é preservado.

Portanto, toda vez que muda-se a configuração do sistema com os procedimentos i) e ii), ambos os teoremas são respeitados. Assim, após uma mudança em uma configuração factível com n_{al} alimentadores sem cadeias fechadas, a nova configuração do sistema é mantida com n_{al} alimentadores sem cadeias fechadas, isto é, a nova configuração do sistema também é factível.

1.4 As Vantagens das Cadeias de Grafo

1.4.1 Não Geração de Configurações Não Factíveis

A primeira vantagem importante da representação por cadeias de grafo é a não geração de configurações não factíveis. Para comparar a representação por cadeias de grafo com a representação pelo vetor V , considere primeiramente o sistema de distribuição da Fig. 1. Este sistema tem 18 chaves e 16 setores. Para um número fixo de operações de chaves, por exemplo um par de operações de chaves, há $C_{18,2}=153$ possíveis vetores V . Por outro lado, há 28 estruturas possíveis

quando são considerados: a representação por cadeias de grafo; as mudanças de configuração de acordo com aquelas definidas na seção 1.3; e um par de operações de chaves (veja a Tabela 2).

O número aproximado de configurações factíveis (n_{FAC}) que podem ser geradas com um par de manobras pode ser calculado por (Delbem and Bretas, 2001)

$$n_{FAC} \approx \frac{\bar{x}n_{al}}{2} (\bar{y}^2 + \bar{y} - 2 + \sigma_e^2) \quad (2)$$

onde $\bar{y} = \frac{Y}{n_{al}}$, $\bar{x} = \frac{2X}{Y}$, n_{al} é o número de alimentadores do sistema, σ_e é o desvio-padrão do número de chaves fechadas nos alimentadores, Y é o número de chaves fechadas no sistema, X é o número de chaves abertas no sistema e n é o número de setores do sistema. A equação (2) é tão mais próxima do valor exato de n_{FAC} quanto mais próximos de \bar{x} estiverem os números de chaves abertas ligadas a cada um dos setores. O parâmetro σ_e pode ser estimado por (Delbem and Bretas, 2001)

$$\hat{\sigma}_e = \frac{\sigma_c}{\sqrt{\bar{c}}} \quad (3)$$

onde $\hat{\sigma}_e$ é uma estimativa de σ_e , σ_c é o desvio-padrão das cargas dos alimentadores e \bar{c} é o valor médio das cargas dos setores do sistema. A estimativa $\hat{\sigma}_e$ é tão melhor quanto mais parecidas estiverem as cargas dos setores. Tal proximidade pode ser obtida com o agrupamento adequado das cargas na formação dos setores.

Alimentador	Chave Aberta	Chave Fechada	Número de Configurações
1	<i>s4</i>	Nenhuma	0
	<i>s3</i>	<i>s15</i>	1
	<i>s2</i>	<i>s14</i>	2
		<i>s15</i>	3
	<i>s1</i>	<i>s16</i>	4
		<i>s14</i>	5
		<i>s15</i>	6
10	<i>s13</i>	<i>s16</i>	7
	<i>s8</i>	<i>s14</i>	8
		<i>s17</i>	9
	<i>s7</i>	<i>s15</i>	10
		<i>s14</i>	11
		<i>s18</i>	12
		<i>s17</i>	13
		<i>s15</i>	14
	<i>s6</i>	<i>s14</i>	15
		<i>s18</i>	16
		<i>s17</i>	17
		<i>s15</i>	18
		<i>s14</i>	19
		11	<i>s12</i>
<i>s11</i>	<i>s17</i>		21
	<i>s18</i>		22
	<i>s16</i>		23
<i>s10</i>	<i>s17</i>		24
	<i>s18</i>		25
	<i>s16</i>		26
	<i>s17</i>		27
	<i>s18</i>		28

Tabela 2: Obtenção do número de configurações possíveis com um par de manobras para o sistema da Fig. 1.

Assim, aplicando-se (2) para o sistema da Fig. 1, obteve-se $n_{FAC} \approx 27$ (sabendo que $X = 5$, $Y = 13$, $n = 16$, $n_{al} = 3$ e $\sigma_e = 0$). Isso significa que aproximadamente $\frac{27}{153} 100\% \approx 17,6\%$ das configurações geráveis a partir da configuração do sistema da Fig. 1 são factíveis. Essa fração fica ainda menor para sistemas grandes. Para o sistema de São Carlos (veja a seção 3), obteve-se $n_{FAC} \approx 629$ (usando-se que

$X = 57$, $Y = 120$, $n = 142$, $n_{al} = 22$ e $\sigma_e = \sigma_c = 5.0$). Por outro lado, há $C_{177,2} = 15.576$ possíveis vetores V . Isto é, para este sistema o número de configurações factíveis, a serem analisadas pelo AG proposto, é menor que 4,0% dos vetores V .

É fácil ver que, com o aumento do número de pares de manobras, a porcentagem de configurações factíveis no total dos vetores V possíveis diminui. Desta forma, quanto maior o número de pares de manobras considerado, melhor deve ser o desempenho do AG com Cadeias de Grafo.

1.4.2 Redução do Espaço de memória RAM alocada

A representação dos grafos em computadores normalmente é feita por uma matriz de incidência (Wilson, 1972). Isso requer pelo menos n^2 bytes para representar uma configuração do sistema com n nós, isto é, o espaço de memória alocado é uma função quadrática de n . Até mesmo para sistemas de distribuição não-grandes, a memória alocada se torna um ponto crucial, pois várias dessas matrizes (indivíduos) são criadas pelo Algoritmo Genético. Por outro lado, a representação através de cadeias de grafo (matriz de cadeias) requer cerca de n bytes (veja a Tabela 1), e desta forma o espaço de memória alocado é aproximadamente uma função linear de n .

1.4.3 Tempo Gasto para Gerar uma Nova Configuração

Outra vantagem importante da representação por cadeias de grafo refere-se ao tempo gasto para gerar uma nova configuração. Quando uma mudança na configuração do sistema acontece, é necessário verificar se essa mudança produz uma configuração factível. Para fazer isso através da matriz incidência, é preciso usar de procedimentos como, por exemplo, os procedimentos (1) e (2) mencionados na seção 1.3. Já o método que usa AG como proposto não executa busca ou resolução de sistemas lineares durante a reconstrução de qualquer matriz de cadeias.

É importante observar que as vantagens da representação do sistema por cadeias de grafo é que torna a proposta de AG viável para ser aplicada em sistemas de distribuição de dimensão grande.

2 FORMULAÇÃO DO PROBLEMA

2.1 Fluxo de Carga

O modelo do sistema de distribuição usado neste artigo é monofásico com o fluxo de carga CC ($\theta_{ij} \approx 0$, sendo θ_{ij} a diferença dos ângulos das tensões nas barras i e j). O modelo de carga utilizado é de corrente constante. Isso torna o fluxo de carga usado nesta proposta de AG muito rápido (não iterativo). No problema de reconfiguração da forma como foi elaborado (usando os passos (i) e (ii), veja a seção 1.3.2), a cada reconfiguração recalcula-se o fluxo de carga somente para os alimentadores que trocam cargas (setores), reduzindo ainda mais o tempo de processamento.

Como a carga é de corrente constante, os estados das barras de um setor variam somente se a corrente que passa através daquele setor variar. Quando há adição ou remoção de trechos de um alimentador, os setores que não têm suas correntes alteradas não precisam ter os estados de suas barras recalculados.

Como os sistemas considerados são radiais, pode-se usar o fluxo de carga com uma rotina de ordenação de nós, conhecida como “pai-filho” (isto é, partindo dos nós dos ramos terminais até a raiz do alimentador). Porém, nem mesmo uma rotina de reordenação de nós é necessária se o sistema estiver representado na forma de cadeias de grafo (rotinas de ordenação, em geral, consomem tempos elevados de processamento). As cadeias de grafo já possuem os nós (setores) ordenados na forma “pai-filho”.

Acredita-se também que o uso de um fluxo de carga mais simplificado não traga prejuízos ao plano de restabelecimento de energia. Por outro lado, poderia-se usar um fluxo de carga mais elaborado somente para refinar as soluções encontradas.

2.2 Função Objetivo

No AG com Cadeias de Grafo implementado, dois objetivos são considerados: (I) reduzir o número de operações de chaves e (II) reduzir o desbalanço entre os alimentadores. Supõe-se que o balanceamento das cargas nos alimentadores terá um efeito no sentido de reduzir:

- a) a sobrecarga dos elementos do sistema, como uma restrição de corrente, (Chen and Hsu, 1989));
- b) a queda de tensão;

Como os objetivos (a) e (b) não são tratados diretamente na formulação do problema, é possível a geração de configurações que não respeitem algum limite de sobrecarga ou de queda de tensão. A permissão de se gerar algumas soluções aparentemente não apropriadas faz parte do procedimento dos AG para tentar aumentar a diversidade de soluções. O importante é que a(s) solução(ões) ótima(s) seja(m) configuração(ões) apropriada(s). (Caso isso ocorra, significa que não é possível restabelecer toda a área desenergizada).

Para verificar a capacidade de um AG atingir o ótimo global, usualmente determina-se tal ótimo usando um outro algoritmo capaz de garanti-lo. Como no restabelecimento de energia a função objetivo é fortemente não linear (inclusive descontínua), seria preciso executar uma busca e examinar todas as configurações possíveis para garantir o ótimo global. Para um sistema de distribuição grande, tal busca é impraticável. Desta forma, os resultados obtidos neste artigo se baseiam na capacidade bem conhecida de um AG alcançar o ótimo global (ou simplesmente de otimizar) de funções objetivo não lineares e mal comportadas (Delbem, Bretas and Carvalho, 1997), (Miranda, Ranito and Proença, 1994), (Nara *et al.*, 1992), (Fukuyama and Ueki, 1995), (Yeh, Venkata and Sumic, 1996), (Pereira and Murari, 1998), (Lin, Su and Tsay, 1998), (Siqing *et al.* 1998). (A existência de um modelo matemático do AG com RCG permitiria o cálculo da probabilidade deste algoritmo atingir o ótimo global em g gerações futuras.)

A função objetivo considerada é

$$z = \sigma_c + \beta \frac{n_{so}}{l_{max}} \quad (4)$$

onde σ_c é o desvio-padrão do carregamento dos alimentadores, n_{so} é o número de pares de operações de chaves, l_{max} é um limite desejável (escolhido arbitrariamente) de pares de operações de chaves de um plano de restabelecimento de energia em um sistema de distribuição

específico e β relaciona o número de pares de chaves (l_{max}) com desbalço entre alimentadores (comentários adicionais sobre o parâmetro β podem ser vistos na seção 3).

Outras formas de se tratar os objetivos de um problema de reconfiguração em sistemas de distribuição por AG podem ser vistas em (Miranda, Ranito and Proença, 1994) (Nara *et al.*, 1992),(Fukuyama and Ueki, 1995),(Yeh, Venkata and Sumic, 1996),(Pereira and Murari, 1998),(Lin, Su and Tsay, 1998),(Siqing *et al.* 1998).

2.3 Exemplo e Comentários Adicionais

O método de AG proposto começa com uma única estrutura na população inicial: a estrutura correspondente à configuração depois da zona faltosa ter sido isolada. O tamanho da população cresce a cada geração com a adição dos novos indivíduos gerados. As estruturas mais aptas são repetidas virtualmente muitas vezes, de modo que o tamanho de cada geração depende da aptidão da estrutura gerada.

O algoritmo pára quando ocorrem 3g gerações sem minimizar z, sendo g a última geração em que se minimizou z. Um critério mais eficiente de parada do algoritmo iria requerer um modelo matemático do AG usando a representação por cadeias de grafo proposta, o qual poderia ser considerado para um trabalho futuro.

Na Tabela 3 é ilustrado o funcionamento do AG proposto para o sistema da Fig. 1. Evidentemente as vantagens computacionais da representação por Cadeias de Grafo não podem ser completamente verificadas neste exemplo devido ao tamanho do sistema ser muito pequeno, apenas 18 chaves. Suponha que cada setor tem uma carga normalizada igual a 1. Suponha que o setor 14 (Fig. 1) é isolado devido a uma falta ocorrida nele, deixando o setor 15 desenergizado. O AG proposto começa com um único indivíduo na população pois só há uma forma de reconectar o setor 15 (pela chave *s18*). O AG basicamente deve:

- ◆ sortear um indivíduo da população existente (na Tabela 3, $I_0 \rightarrow I_1$ indica que o indivíduo I_0 foi sorteado gerando o indivíduo I_1);
- ◆ sortear uma conexão deste indivíduo a ser eliminada e uma outra a ser criada (esta última conexão é obtida de uma matriz com as possibilidades de conexões de cada setor, veja a Tabela 4 (onde 1º setor de cada linha da tabela pode se conectar com os demais setores na mesma linha)).
- ◆ avaliar o novo indivíduo.

A representação por Cadeias de Grafo permite fazer as alterações das conexões rapidamente, além de acelerar a identificação do conjunto de conexões que podem ser abertas e fechadas de forma a não gerar configurações (indivíduos) não factíveis (Teorema 1 e Corolário 1). É interessante observar que os novos indivíduos são gerados apenas com a troca de alguns ponteiros (estrutura característica da linguagem de programação C) e com poucos trechos de cadeias copiados. A Fig. 2 ilustra como o indivíduo I_3 é representado a partir do indivíduo I_2 , segundo uma estrutura de ponteiros.

É importante notar que neste trabalho não são realizadas trocas de trechos de cromossomos entre indivíduos diferentes (*crossover* convencional). Essa operação ainda não foi

desenvolvida para a representação por cadeias de grafo. As formas de troca de trechos de cromossomos entre indivíduos diferentes esquematizadas até então pelos autores não garantem que uma nova configuração seja sempre factível. Assim, seria preciso usar, por exemplo, rotinas de busca para verificar e corrigir possíveis configurações não factíveis. Diante disso, preferiu-se não implementar a operação de troca de cromossomos entre indivíduos diferentes, pois uma rotina de busca demandaria um tempo computacional muito elevado para sistemas de distribuição de tamanho grande.

População	RCG do Indivíduo Gerado	σ_c	Conexões Sorteadas	
			Eliminada	Criada
I_0	I_0 1 2 16 1 2 3 4 5 10 9 8 7 6 10 9 8 15 11 12 13	1,41	<i>s12</i> (para isolar a área faltosa)	<i>s18</i>
$I_0 \rightarrow I_1$	I_1 1 2 16 13 1 2 3 4 5 10 9 8 7 6 10 9 8 15 11 12	2,16	<i>s10</i>	<i>s16</i>
I_0, I_1 $I_0 \rightarrow I_2$	I_2 1 2 16 1 2 3 4 5 1 2 3 6 10 9 8 7 10 9 8 15 11 12 13	1,63	<i>s8</i>	<i>s14</i>
I_0, I_1, I_2 $I_2 \rightarrow I_3$	I_3 1 2 1 2 3 4 5 1 2 3 6 10 9 8 7 10 9 8 15 11 12 13 11 12 13 16	0,82	<i>s13</i>	<i>s16</i>

Tabela 3: Ilustração do funcionamento do AG proposto para o sistema da Fig. 1.

1 2	6 3 7	11 12
2 1 3 16	7 4 6 8 14	12 11 13
3 2 4 6	8 7 9 15	13 12 14 16
4 3 5 7	9 8 10	14 13 15 7
5 4	10 9	15 14 8
		16 2 13

Tabela 4: Possibilidades de conexões para todos os setores do sistema da Fig. 1.

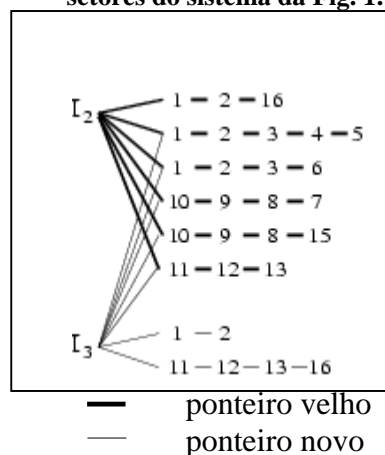


Fig. 2: Estrutura de ponteiros de um indivíduo (I_3) representado a partir de outro (I_2), referente aos dados da Tabela 3.

Devido à não implementação do *crossover* convencional e a troca de trechos de cromossomos entre cromossomos de um mesmo indivíduo ser o único operador implementado, não se fala em taxas de *crossover* e mutação.

3 RESULTADOS

O algoritmo proposto foi testado usando o sistema de distribuição, relativamente grande e complexo (177 chaves, 142 setores, 22 alimentadores e 75MW de capacidade) da cidade de São Carlos, SP. Sua representação por cadeias de grafo é mostrada na Tabela 5.

Primeiramente, o sistema da Tabela 5 foi replanejado usando o AG proposto com $\beta = 0,0$ (veja a equação (4)). As curvas mais claras das Fig. 3(a) e 3(b) representam o desempenho do AG neste replanejamento, obtendo-se uma configuração ótima com $z = \sigma_c = 1,85$. Para esta configuração ótima, o alimentador 100* (veja a Tabela 5) é, então, completamente desenergizado. O desbalanço e o número de pares de manobras obtidos pelo AG proposto ao longo das gerações correspondem às curvas mais escuras das Fig. 3(a) e 3(b) respectivamente. O Plano ótimo foi atingido em cerca de 260 gerações (aproximadamente 6 segundos) ($\beta = 3,0$, $l_{max} = 10$).

A seguir são apresentados mais resultados estatísticos e o efeito do parâmetro β (veja a equação (4)) no desempenho do AG proposto. Um β adequado deve garantir, com uma probabilidade apropriada, que o AG proposto irá encontrar no mínimo um ótimo local e um número mínimo de manobras. Todas os testes apresentados até o final desta seção relacionam-se com o desempenho do algoritmo proposto quando aplicado para minimizar z para o sistema da Tabela 5 (não previamente replanejado) quando seu alimentador 100* (com carga grande) está desenergizado e $l_{max} = 10$.

As Fig. 4, 5, 6 e 7 mostram o desempenho do algoritmo genético proposto para minimizar z usando $\beta = 0,0$, $\beta = 0,2$, $\beta = 0,4$ e $\beta = 0,6$, respectivamente. As Fig. 8, 9, 10 e 11 mostram o desempenho do algoritmo genético para manter o n_{so} pequeno quando $\beta = 0,0$, $\beta = 0,2$, $\beta = 0,4$ e $\beta = 0,6$, respectivamente.

22 81 80 23 21 15	69 70 68 64 61	116 117 115
23 31 22	70 69 68	117 129 125 118 116
24 25	71 72	118 120 119 117
25 26 24	72 77 74 73 71 58	119 118 114
26 34 27 25 13	73 80 72	120 118 48
27 28 26 21	74 72 58 18	121 122
28 29 27	75 76	122 129 124 123
29 41 31 30 28	76 83 77 75	123 122 99
30 37 29	77 78 76 72	124 125 122
31 45 42 32 29 23	78 79 77	125 126 124 117 48
32 87 31	79 80 78	126 125 98
33 34	80 81 79 73 22 16	127 128
34 35 33 26	81 84 80 22	128 129 127
35 40 37 34	82 83	129 130 128 122 117
36 37	83 90 84 82 76	130 134 131 129
37 39 36 35 30	84 85 83 81	131 132 130
38 39	85 86 84	132 133 131
39 51 40 38 37	86 87 85	133 132 108
40 42 41 39 35	87 92 88 86 32	134 142 135 130
41 40 29	88 87 45	135 142 136 134
42 43 40 31	89 90	136 135
43 45 44 42	90 91 89 83	137 138
44 54 53 43	91 93 92 90	138 139 137
45 88 54 47 46 43 31	92 91 87	139 142 140 138
46 48 45	93 94 91	140 142 139
47 94 45	94 96 95 93 47	141 142
		142 141 140 139 135 134

Tabela 6: Possíveis conexões para os setores do sistema usado nos testes.

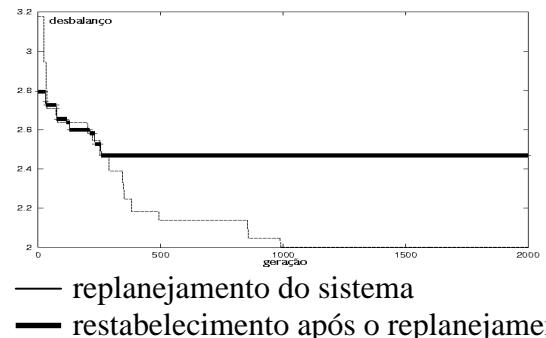


Fig. 3 (a): Minimização do desbalanço no replanejamento e no restabelecimento do sistema.

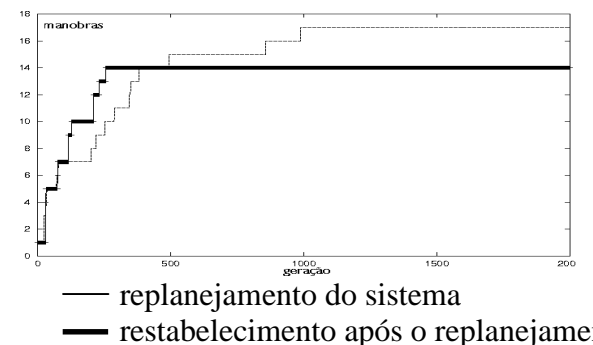


Fig. 3 (b): Número de pares de manobras para replanear e restabelecer o sistema.

Número da Cadeia	Cadeia	Número da Cadeia	Cadeia
1	1 2 3 4 5 6 7	19	75 76 77 78 79 80 81
2	8 9 10 11	20	82 83 84 85 86 87 88
3	12 13 14 15 16 17 18	21	89 90 91 92
4	19 20 21 22 23	22	89 90 91 93 94 95
5	24 25 26 27 28 29 30	23	89 90 91 93 94 96 97 98 99
6	24 25 26 27 28 29 31 32	24	100* 101 102 103 104 105
7	33 34 35		
8	36 37	25	100* 101 102 106 107 108 109
9	38 39 40 41	26	110 111 112 113
10	38 39 40 42 43 44	27	110 111 112 114
11	38 39 40 42 43 45 47	28	115 116 117 118 119
12	38 39 40 42 43 45 46 48	29	115 116 117 118 120
13	49 50 51 52 53 54 55 56	30	115 116 117 125 124 122 123
14	57 58 59 60 61	31	115 116 117 125 126
15	62 63 64 65	32	127 128 129 130 131 132 133
16	66 67 68 69 70	33	127 128 129 130 134 135 136
17	71 72 73	34	137 138 139 140
18	71 72 74	35	141 142

Tabela 5: Cadeias de grafo do sistema usado nos testes.

1 2	48 125 120 46	95 102 94
2 3 1	49 50	96 97 94
3 9 4 2	50 51 49	97 98 96
4 10 5 3	51 52 50 39	98 126 99 97
5 6 4	52 53 51	99 125 105 98
6 7 5	53 54 52 44	100 101
7 11 6	54 55 53 45 44	101 102 100
8 9	55 56 54	102 106 103 101 95
9 13 10 8 3	56 114 55	103 104 102
10 14 11 9 4	57 58	104 105 103
11 61 10 7	58 74 72 67 59 57	105 104 99
12 13	59 60 58	106 107 102
13 26 20 14 12 9	60 61 59	107 108 106
14 20 15 13 10	61 69 60 11	108 133 109 107
15 22 16 14	62 63	109 108
16 80 17 15	63 67 64 62	110 111
17 18 16 0	64 69 65 63	111 112 110
18 74 17	65 64	112 114 113 111
19 20	66 67	113 112
20 19 14 13	67 68 66 63 58	114 119 112 56
21 27 22	68 70 69 67	115 116

Na Fig. 4, como $\beta = 0,0$, somente a minimização de σ_c é exigida, veja a equação (4). Nesta figura, depois de 6000 gerações, todas as 30 amostras estão concentradas próximas de $\sigma_c = 3,0$ ou $\sigma_c = 2,7$. Esses dois valores de σ_c estão possivelmente próximos de ótimos locais ou globais. (Observe que não é possível, na prática, determinar o ótimo local e o ótimo global devido a grande dimensão do problema: 2^{142} configurações teriam de ser analisadas. Uma vez que todos os resultados dos testes estão próximos de pelo menos um ótimo local, todas as configurações obtidas no final dos testes da Fig. 4 minimizam satisfatoriamente o desbalanço entre os alimentadores. O maior desbalanço depois de 6.000 gerações é alcançado com $\sigma_c = 3,1$. Desta forma, será admitido que um

teste minimiza satisfatoriamente o desbalanço se σ_c for menor ou igual a $\sigma_c = 3,1$.

Quando $\beta = 0,2$, veja a Fig. 5, todos os testes alcançam $\sigma_c \leq 3,1$, então 100% das amostras minimizam satisfatoriamente o desbalanço. Quando $\beta = 0,4$ (veja a Fig. 6) 70% das amostras minimizam satisfatoriamente o desbalanço. Quando $\beta = 0,6$, veja a Fig. 7, somente 20% das amostras minimizam o desbalanço satisfatoriamente. Por outro lado, quando $\beta = 0,0$, o número máximo de pares de manobras (n_{so}^{max}) ocorrido é 62, veja a Fig. 8. Quando $\beta = 0,2$, $n_{so}^{max} = 36$, veja a Fig. 9. Quando $\beta = 0,4$, $n_{so}^{max} = 27$, veja a Fig. 10. Quando $\beta = 0,6$, $n_{so}^{max} = 18$, veja a Fig. 11.

Consequentemente, seria razoável escolher $0,2 \leq \beta \leq 0,4$, que corresponde a 70%-100% das amostras minimizadas satisfatoriamente e $27 \leq n_{so}^{max} \leq 36$, quando considerados os testes apresentados anteriormente. De fato, para $\beta = 0,3$, 97% dos testes atingem $\sigma_c = 3,1$ e $n_{so}^{max} = 30$ (veja as Fig. 12 e 13).

É importante observar que os testes mostrados acima, além de fornecerem uma análise estatística do desempenho do AG proposto, apresentam um exemplo de como encontrar um valor adequado para β .

Todas as experiências foram feitas em um Pentium100 com 32 MB de memória RAM. Com os dados das Tabelas 5 e Tabela 6 (na qual o 1º setor de cada linha da tabela pode se conectar com os demais setores na mesma linha), com a carga de cada setor igual 1 e com a função objetivo definida em (4), os resultados apresentados neste artigo podem ser estatisticamente reproduzidos.

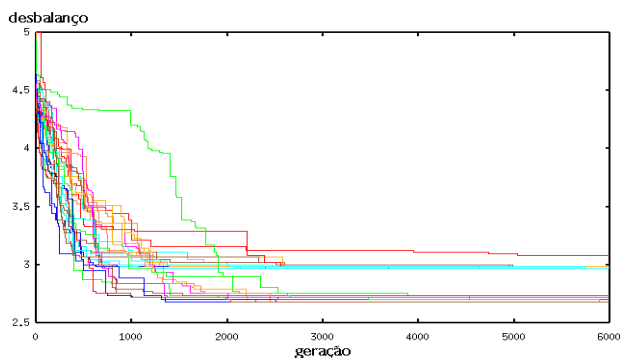


Fig. 4: Desbalanço versus geração quando $\beta = 0,0$. Trinta amostras.

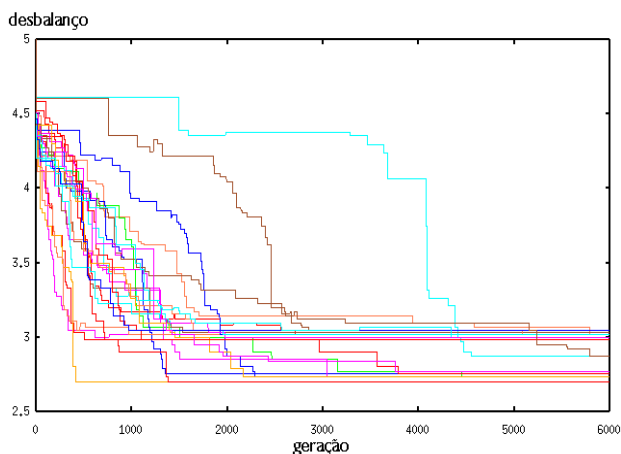


Fig. 5: Desbalanço versus geração quando $\beta = 0,2$. Trinta amostras.

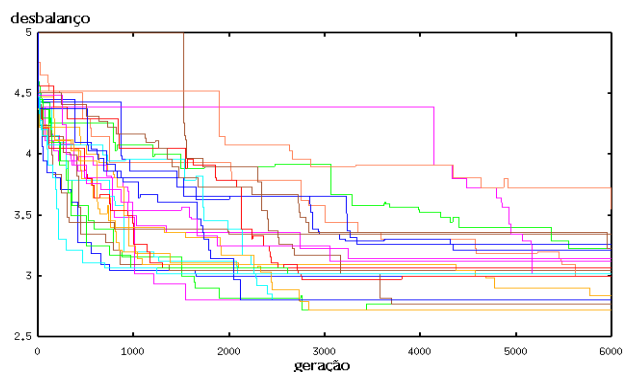


Fig. 6: Desbalanço versus geração quando $\beta = 0,4$. Trinta amostras.

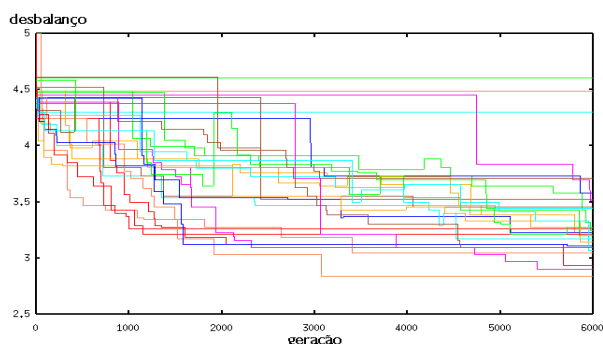


Fig. 7: Desbalanço versus geração quando $\beta = 0,6$. Trinta amostras.

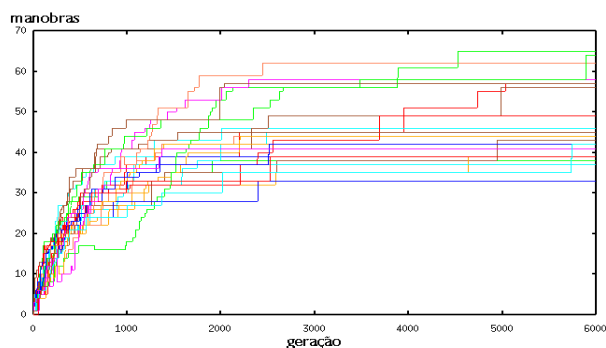


Fig. 8: Pares de manobras versus geração quando $\beta = 0,0$. Trinta amostras.

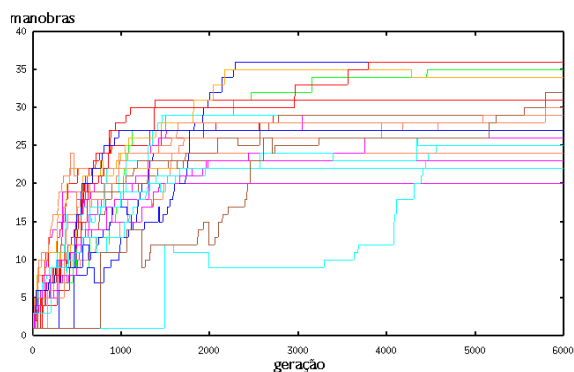


Fig. 9: Pares de manobras versus geração quando $\beta = 0,2$. Trinta amostras.

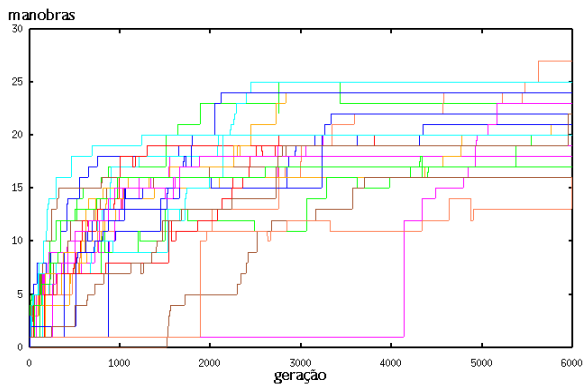


Fig. 10: Pares de manobras versus geração quando $\beta = 0,4$. Trinta amostras.

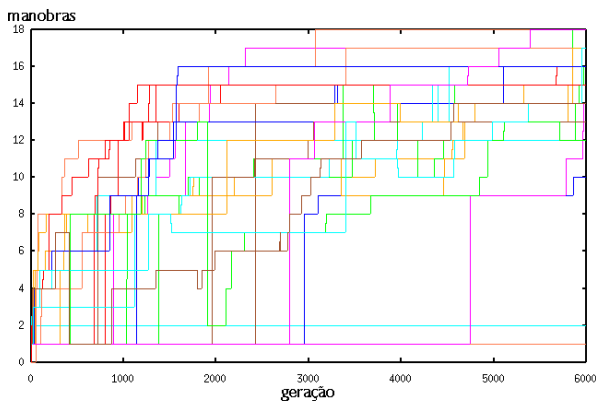


Fig. 11: Pares de manobras versus geração quando $\beta = 0,6$. Trinta amostras.

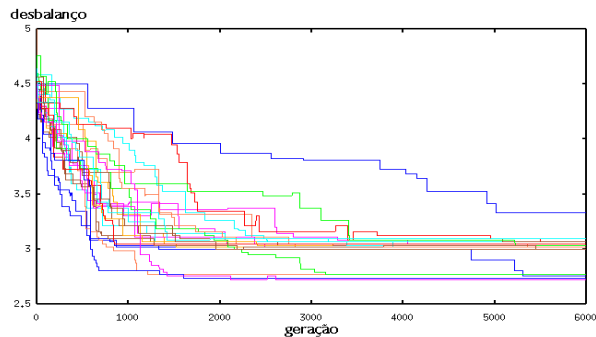


Fig. 12: Desbalanço versus geração quando $\beta = 0,3$. Trinta amostras.

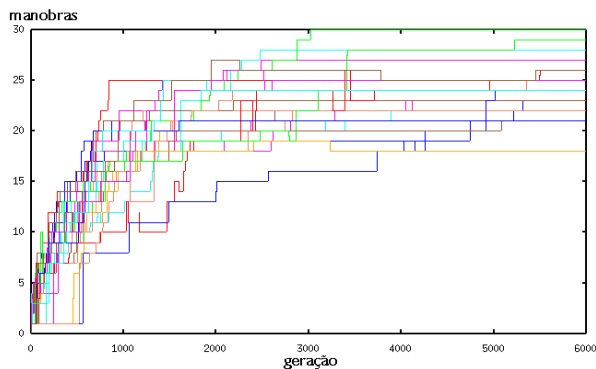


Fig. 13: Pares de manobras versus geração quando $\beta = 0,3$. Trinta amostras.

4 CONCLUSÕES

Neste trabalho, foi apresentado um método que usa AG para restabelecer energia em sistemas de distribuição após faltas

permanentes. A mesma proposta também pode ser usada para obter planos para manutenção do sistema. Os métodos para restabelecimento de energia e para planejamento ótimo de sistema de distribuição até então desenvolvidos podem tratar somente com otimizações locais e sistemas de distribuição não-grandes. Para superar este obstáculo, foi proposto neste artigo uma técnica baseada em AG com RCG para tornar possível obter um restabelecimento de energia ótimo para sistemas grandes e complexos em um curto intervalo de tempo. Isso se torna possível devido basicamente aos seguintes fatores:

- 1) a geração somente de configurações factíveis pelo algoritmo (veja as seções 1.3 e 1.4.1);
- 2) o número de configurações factíveis corresponder a uma pequena porcentagem do número de configurações possíveis com um par de manobras (veja a seção 1.4.1);
- 3) o desempenho do AG com RCG melhora em relação ao desempenho de um AG que não gera somente configurações factíveis conforme o número de pares de manobras aumenta (veja a seção 1.4.1);
- 4) a redução do espaço de memória alocada sem aumentar o custo computacional do algoritmo (veja a seção 1.4.2);
- 5) a redução do tempo para gerar uma nova configuração (veja a seção 1.4.3 e o 4º parágrafo da seção 2.3);

5 AGRADECIMENTOS

Este trabalho foi apoiado pela FAPESP.

6 REFERÊNCIAS BIBLIOGRÁFICAS

- Aoki, K., K. Nara, M. Itoh, T. Satoh, H. Kuwabara, July 1989, "A new algorithm for service restoration in distribution systems", *IEEE Trans. on Power Delivery*, vol.4, n.3, p.1832-1839.
- Castro, C. H., J. B. R. Bunch, M. Terry, March/April 1980, "Generalized algorithms for distribution feeder deployment and sectionalizing". *IEEE Trans. on PAS*, vol. PAS-99, n.2, p.549-557.
- Chen, J. L., Y. Y. Hsu, July 1989, "An expert system for load allocation in distribution expansion planning", *IEEE Trans. on Power Delivery*, vol.4, n.3, p.1910-1917.
- Delbem, A. C. B., N. G. Bretas, 2001, "Distribution System Reconfiguration Using Graph Chain Representation", Submetido ao *IEE Proceedings - Generation, Transmission and Distribution*.
- Delbem, A. C. B., N. G. Bretas, A. C. P. L. F. Carvalho, 1997, "Optimal energy restoration in distribution systems using genetic algorithms", *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'97)*, Gold Coast, Anais. p.150-154.
- Delbem, A. C. B., N. G. Bretas, A. C. P. L. F. Carvalho, 13-14 de Outubro, 1997, "Energy Restoration in Distribution Systems Using Search with Fuzzy Heuristics", *29th North American Power Symposium*, pp. 295-299.

- Delbem, A. C. B., N. G. Bretas, março, 1998, "Restabelecimento de Energia em Sistemas de Distribuição: Busca com Heurísticas Fuzzy e

- Algoritmos Genéticos com Cadeias de Grafo”, *Tese de Mestrado, Departamento de Engenharia Elétrica – EESC – USP*.
- Devi, V. S., D. P. S. Gupta, G. Anandalingam, January 1995, “Optimal restoration of power supply in large distribution systems in developing countries”, *IEEE Trans. Power Delivery*, vol. 10, No. 1, pp. 430-437.
- Fukuyama, Y., Y. Ueki, June 1995, “Application of genetic algorithms to service restoration in distribution-systems”, *Electrical Engineering in Japan*, vol. 115, No. 3, pp. 30-38.
- Goldberg, D. E., 1989, *Genetic algorithms in search, optimization and machine learning*, New York, Addison-Wesley Publishing Company.
- Hsu, Y. Y., H. Kuo, April 1994, “A heuristic based fuzzy reasoning approach for distribution system service restoration”, *IEEE Trans. on Power Delivery*, vol.9, n.2, p.948-953.
- Lin, W. M., Y. S. Su, M. T. Tsay, August, 1998, “Genetic algorithm for optimal distribution system planning”, *POWERCON’98 (1998 International Conference on Power System Technology Proceedings)*, pp. 241-245.
- Liu, C. C., S. J. Lee, S. S. Venkata, May 1988, “An expert system operational aid for restoration and loss reduction of distribution systems”, *IEEE Trans. on Power Systems*, vol.3, n.2, p.619-626.
- Miranda, V., J. V. Ranito, L. M. Proença, November 1994, “Genetic Algorithm in optimal multistage distribution network planning”, *IEEE Trans. on Power Systems*, vol. 9, No. 4, pp. 1927-1933.
- Morelato, A. L., A. Monticelli, October 1989. “Heuristic search approach to distribution system restoration”, *IEEE Trans. on Power Delivery*, vol. 4, No. 4, pp. 2235 - 2241.
- Nara, K., A. K. Deb, A. Shiose, M. Kitagawa, T. Ishihara, August 1992, “Implementation of genetic algorithm for distribution-systems loss minimum reconfiguration”, *IEEE Trans. on Power Systems*, vol. 7, No. 3, pp. 1044-1051.
- Pereira, M. A., C. A. F. Murari, setembro, 1998, “Aplicação de algoritmos genéticos em redes de distribuição de energia elétrica”, *III CONLADIS (Congresso Latino Americano de Distribuição de Energia Elétrica)*, pp. 592-596.
- Siqing, S., S. Youjiang, L. Yang, Z. Wenqin, Y. Yihan, August, 1998, “Integrating genetic algorithm with expert system for service restoration in distribution system”, *POWERCON’98 (1998 International Conference on Power System Technology Proceedings)*, pp. 265-269.
- Wang, J. C., H. Chiang, G. R. Darling, February 1996, “An efficient algorithm for real-time network reconfiguration in large scale unbalanced distribution system”, *IEEE Trans. on Power Systems*, vol.11, n.1, p.511-517.
- Wilson, R. J., 1972, *Introduction to graph theory*. Glasgow, Bell & Bain Ltd.
- Winston, P. H., 1984, *Artificial intelligence*. New York, Addison-Wesley.
- Yeh, E. C., S. S. Venkata, Z. Sumic, May 1996, “Improved distribution system planning using computational evolution”, *IEEE Trans. on Power Systems*, vol. 11, No. 2, pp. 668-674.
- Zbigniew, M., 1994, “Genetic algorithms + data structures = evolution programs”, New York, Springer-Verlag.

APÊNDICE

Algoritmos Genéticos

Os AG são algoritmos de busca baseados em mecanismos de seleção natural e genéticos (Goldberg, 1989). Eles são conhecidos como sendo uma busca e uma técnica de otimização muito robusta (têm desempenho bom para uma variedade de aplicações). Assim, eles podem ser usados tanto em problemas que requerem árvore de busca como em problemas que usam técnicas de otimização em geral. Como resultado, os AG demonstram ser uma ferramenta poderosa para lidar simultaneamente com restabelecimento de energia e otimização, isto é restabelecimento de energia ótimo em sistemas de distribuição.

Os conceitos básicos dessa técnica são: i) cada estrutura (criatura artificial) representa uma possível solução para o problema (em nosso caso, uma possível solução para o sistema de distribuição); ii) as estruturas são um pacote de *strings* (cromossomos) cujos *bits* (genes) alocam valores de característica (*alleles*) do indivíduo. Novas estruturas são geradas por mecanismos de troca que usam as partes daquelas *strings* mais aptas (ou adaptadas): *crossover* ou *mutação*. Este processo é chamado de reprodução. As estruturas mais aptas são selecionadas de acordo com o valor de sua função objetivo. A cada repetição do processo é obtida uma geração nova. Depois de várias gerações ou depois que a estrutura mais apta apresentar um valor desejado para função objetivo, o processamento pára. Uma explicação mais detalhada pode ser vista em (Miranda, Ranito and Proença, 1994).

Para o problema de restabelecimento de energia, cada estrutura foi considerada como uma configuração *radial* do sistema que alimenta todos seus setores (configuração factível). Uma estrutura nova é uma possível mudança feita em uma estrutura da geração anterior. Genes são setores (Delbem, Bretas, and Carvalho, 1997), (Delbem and Bretas, 1998) e cromossomos são alimentadores. Uma estrutura nova (filha) só é gerada trocando setores vizinhos (genes) entre alimentadores (*strings*) da mesma estrutura (mãe), como uma forma de evitar genes inativos (setores não energizados).

Teoria básica de Cadeias de Grafos

Esta seção começa definindo um grafo simples G (Delbem and Bretas, 1998) como um par $(N(G), E(G))$, onde $N(G)$ é um conjunto finito de elementos não-vazios chamados de nós e $E(G)$ é um conjunto finito de pares de elementos não ordenados distintos de $N(G)$ chamados de arcos; $N(G)$ é, às vezes, chamado de conjunto dos nós de G . Para ilustrar, a Fig. 14 representa um grafo simples G cujo conjunto de nós $N(G)$ é o conjunto $\{u, v, w, z\}$, e cujo conjunto de arcos $E(G)$ consiste nos pares $\{u, v\}$, $\{v, w\}$, $\{u, w\}$ e $\{w, z\}$. Dado um par $\{x, y\}$ é dito que este par une os nós x e y .

Dado qualquer grafo G , uma seqüência de arcos em G é uma seqüência finita de arcos da forma $\{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{m-1}, v_m\}$ (também denotada por $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_{m-1} \rightarrow v_m$). Uma seqüência de arcos na qual todas os arcos são distintos é chamada um caminho; se, além disso, os nós $v_0, v_1, v_2, \dots, v_m$ são distintos (exceto, possivelmente, $v_0=v_m$), então o caminho é chamado de cadeia. Uma cadeia ou um caminho é fechado se $v_0=v_m$. Um grafo é dito ser conexo se, dado um par de nós x, y de G , há uma cadeia de x para y . Um grafo arbitrário pode ser dividido em sub-grafos conexos disjuntos chamados componentes conexos. (Um grafo conexo tem só um componente).

Teorema 1: Sejam G_1 e G_2 dois grafos desconexos um do outro e sem cadeias fechadas. Adicionando um arco que os conecta resultará um grafo conexo sem cadeias fechadas (Delbem and Bretas, 1998).

Corolário 1: Se um arco é removido de um grafo conexo sem cadeias fechadas, resultará em dois grafos conexos sem cadeias fechadas (Delbem and Bretas, 1998).

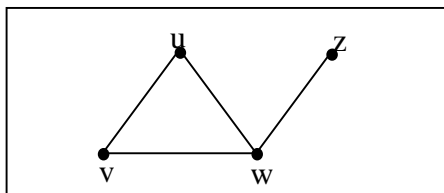


Fig. 14: Exemplo de um grafo.