
MÉTODOS DE PROGRAMAÇÃO INTEIRA APLICADOS AO PLANEJAMENTO DA AUTOMAÇÃO DE SISTEMAS DE MANOBRA EM REDES DE DISTRIBUIÇÃO

Mauricio Sperandio*
sperandio@labplan.ufsc.br

Jorge Coelho*
coelho@labplan.ufsc.br

*LabPlan EEL UFSC
Caixa Postal 476
CEP 88040 900 Florianópolis SC

ABSTRACT

Integer Programming Methods Applied to the Automation Planning of Manoeuvre Systems in Distribution Networks

This work presents a comparison between two integer programming methods applied to the problem of automation planning of loop controlled distribution feeders in large systems. It is necessary to decide which points to automatize amongst some feeders of different regions, minimizing de costs attaining to quality goals for each consumer cluster. The methods verified are the Branch-and-Bound and the Genetic Algorithm.

KEYWORDS: Integer Programming, Distribution Networks, Reliability Indexes, Planning, Automation.

RESUMO

Neste trabalho é apresentada uma comparação entre dois métodos de programação combinatória inteira formulados para resolver um problema de planejamento da automação de sistemas de manobra em redes de distribuição de grande porte. É preciso decidir quais pontos automatizar dentre vários alimentadores de diferentes regiões, minizando os custos e respeitando as metas de

qualidade de cada conjunto de consumidores. Os métodos verificados foram o Branch-and-Bound e o Algoritmo Genético.

PALAVRAS-CHAVE: Programação Inteira, Redes de Distribuição, Índices de Continuidade, Planejamento, Automação.

1 INTRODUÇÃO

Os sistemas de energia elétrica são divididos em três segmentos: geração, transmissão e distribuição. Neste último, a subestação é o ponto inicial de onde partem os alimentadores de média tensão, que abastecem os transformadores que irão levar a energia em baixa tensão até o ponto de conexão dos consumidores. Esse sistema é essencialmente radial, ou seja, há uma única fonte de energia, a subestação (SE), e todos os componentes que interligam essa fonte ao consumidor estão associados em série. Problemas decorrem naturalmente dessa operação, tais como o excesso de carregamento nos condutores dos alimentadores mais próximos da subestação e quedas de tensão acentuada (subtensão) nos pontos mais distantes.

A radialidade característica dos sistemas de distribuição facilita a configuração e manutenção dos equipamentos de proteção, que têm por finalidade impedir a propagação de falhas, especialmente curtos-circuitos. No entanto, uma falha no trecho principal de um alimentador

Artigo submetido em 29/07/2009 (Id.: 01020)
Revisado em 25/09/2009, 18/02/2010, 01/03/2010
Aceito sob recomendação do Editor Associado Prof. Antonio Carlos Zambroni de Souza

(tronco), causa a interrupção de todos os consumidores do mesmo. Para minimizar o efeito desse tipo de falha existem os sistemas de manobra.

Os sistemas de manobra são constituídos por chaves que podem estar em dois estados, normalmente fechadas (NF) ou normalmente abertas (NA). As chaves do tipo NF são posicionadas ao longo do tronco do alimentador, e tem como finalidade seccioná-lo no caso de falhas a frente delas, ou seja, a jusante do fluxo de energia elétrica. Já as chaves NA são uma opção de caminho alternativo dentro do próprio alimentador, ou podem realizar a conexão com um alimentador adjacente, caso haja uma falta de energia a montante. Nesse caso, é feita a transferência de cargas de um alimentador para outro, realizando uma manobra de abertura de uma chave NF e fechamento de uma chave NA. Esse tipo de manobra é condicionado a restrições de capacidade das linhas e aos limites de tensão regulamentados. Normalmente os sistemas de manobra são operados manualmente, o que implica na necessidade de equipes de manutenção, que percorrem o alimentador em busca do local onde ocorreu a falha, identificam o problema, estabelecem uma interface verbal com os centros de operação da distribuição, de forma que haja coordenação das ações em campo e se determine a sequência de abertura e fechamento das chaves. Tal circunstância expõe a operação do sistema a fatores externos cujos resultados podem fugir ao controle da coordenação desta atividade. Fatores como tempestades, congestionamentos de trânsito, quebras de veículos, acidentes de trabalho, falhas na comunicação, longas distâncias a serem percorridas e vias de acesso desfavoráveis são alguns dos indicativos de que a operação do sistema elétrico, dada a essencialidade do seu produto, precisa evoluir para um método mais automatizado. Atualmente, a quantidade de sistemas de manobra automatizados nas concessionárias de distribuição de energia elétrica brasileiras ainda é extremamente pequena. O principal motivo para a não utilização desse recurso é a falta de uma metodologia para avaliar os benefícios que podem advir com a automação. Sem uma modelagem adequada da influência das chaves de manobra automáticas nos indicadores de continuidade do sistema é impossível realizar uma análise de viabilidade do investimento.

1.1 Chaves automáticas

O emprego de chaves de manobra automatizadas possibilita a alimentação alternativa de forma imediata, por meio da transferência de carga entre alimentadores. A transferência automática, ou mesmo telecomandada, aumenta consideravelmente a disponibilidade de energia

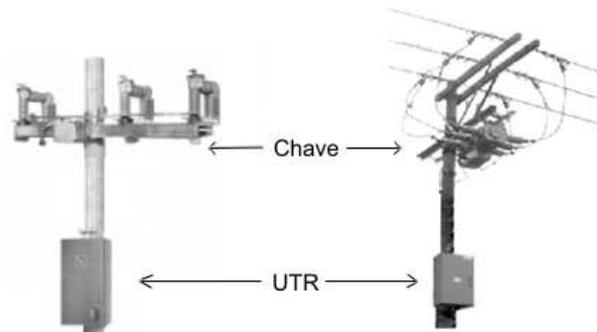


Figura 1: Chaves de manobra automáticas

para os consumidores, principalmente em áreas longe do centro de manutenção ou com acessos difíceis para a equipe de manutenção, ou seja, áreas com altos tempos de atendimento. Observa-se ainda em (QIN et alii, 1997) que essas chaves também podem funcionar para melhoria do perfil de tensão e alívio de carregamento, realizando a transferência de cargas entre alimentadores. Outros vários benefícios da automação de redes de distribuição são destacados em (GRUENEMEYER, 1991; RIGLER et alii, 1999; RONG-LIANG e SABIR, 2001; GILBERT, 2004). Chaves automáticas, como as que aparecem na Figura 1, podem abrir e fechar o circuito por ação de seus relés locais, ou serem controladas a partir do Centro de Operação da Distribuição (COD) via um canal de comunicação. Muitas dessas chaves já possuem equipamentos para aquisição de dados de tensão e correntes, que transmitidos ao COD pela Unidade Terminal Remota (UTR) permitem o monitoramento em tempo real do carregamento ao longo do alimentador.

A automação de circuitos em anel aberto é o tipo de configuração mais empregada nas empresas internacionais que adotam automação em suas redes (SPERANDIO et alii, 2005). O circuito em anel, ou *loop*, diminui as áreas afetadas por faltas através da transferência de carga entre alimentadores. O esquema é baseado em um circuito com duas fontes de alimentação e um ponto normalmente aberto (NA), como esquematizado na Figura 2. Os alimentadores interligados podem partir de uma mesma SE ou de SEs diferentes. O primeiro caso confere menor confiabilidade, pois caso haja uma falha na própria SE ambos os alimentadores ficarão sem suprimento.

As chaves automáticas possuem relés de detecção de sobrecorrente (curto-circuito) e de falta de tensão, e podem trabalhar tanto normalmente fechadas quanto normalmente abertas. Estas chaves dependem de um meio de comunicação para coordenarem a configuração mais

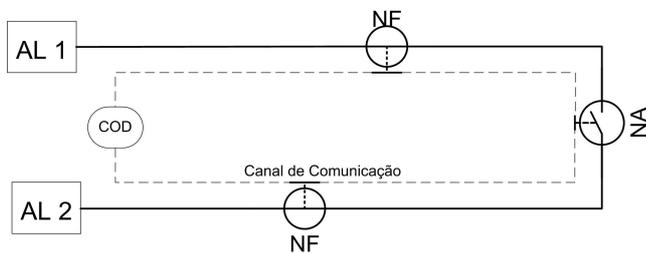


Figura 2: Esquema de chaves automáticas em anel aberto

adequada a cada situação, e agilizar a isolação do trecho com falha e recomposição dos trechos sob falta. A topologia de comunicação pode ser variada, assim como o gerenciamento do sistema, que pode ser centralizado, distribuído ou misto (SPERANDIO et alii, 2005). No exemplo da Figura 2 existem quatro zonas de proteção independentes, e caso os condutores sejam dimensionados para atender a todas elas é possível se obter a máxima eficiência do sistema. Na ocorrência de uma falha permanente, apenas o trecho entre duas chaves é imediatamente isolado para aguardar reparo, com as cargas a montante do trecho avariado sendo restabelecidas após o seccionamento, e as cargas a jusante sendo transferidas para o outro alimentador. Também podem ser usadas funções de religamento automático para evitar desligamentos por falhas temporárias, o que torna o esquema muito confiável, diminuindo ainda mais os indicadores de continuidade do conjunto.

1.2 Planejamento

Para o sistema funcionar de forma eficiente, o planejamento deve ser feito com base em requisitos técnicos que respeitem uma filosofia de proteção e controle, não em função de um equipamento escolhido ou já instalado, como muitas vezes acontece. Equipamentos existentes devem influenciar a instalação da automação de forma a coabitarem no sistema por certo período, porém não devem desviar o objetivo final do projeto. Este planejamento exige estudos de fluxo de potência e de confiabilidade, considerando critérios de operação, econômicos e regulatórios. Por exemplo, ao se buscar as posições de chaves automáticas para a configuração em anel aberto é preciso verificar se os cabos existentes suportam a transferência de carga do outro alimentador, e se os níveis de tensão permanecem dentro dos limites estabelecidos pelo órgão regulador.

No Brasil, a Agência Nacional de Energia Elétrica (ANEEL) regula os serviços de distribuição, cuja qualidade do fornecimento é avaliada por meio de indicadores de continuidade que expressem os valores vincu-

lados a conjuntos de unidades consumidoras, expostos nas resoluções 024 de 2000 e 075 de 2003. Os principais indicadores são o DEC e o FEC, respectivamente, Duração Equivalente de Interrupção por Unidade Consumidora e Frequência Equivalente de Interrupção por Unidade Consumidora. Um conjunto de unidades consumidoras abrange uma área geográfica contígua que é atendida pela concessionária, geralmente sendo considerado o município como um todo.

As chaves automatizadas são equipamentos de custo elevado, entre U\$ 9.000,00 e € 20.000,00 (CHAO-SHUN et al., 2006; DE TUGLIE et al., 2004), e que demandam um sistema de comunicação e software de gerenciamento. Devido a esse alto custo, não é viável, em um primeiro momento, a sua alocação generalizada em todos os alimentadores de uma rede de distribuição abrangente. Por isso, é necessário um planejamento adequado que considere a viabilidade técnica das manobras e maximize os ganhos possíveis com um número mínimo de chaves. Desta forma, é preciso determinar a alocação ótima dessas chaves dentre uma grande quantidade de possibilidades, o que exige uma análise combinatória complexa, como destacam Matsumoto et alii (2002). Esse problema é resolvido por métodos de programação inteira, como os apresentados a seguir.

2 MÉTODOS DE PROGRAMAÇÃO INTEIRA

A solução para o problema de alocação de chaves de manobra é não-linear, inteira e não-diferenciável, o que impede a aplicação da maioria das técnicas de otimização clássica. Algoritmos para solução desse tipo de problema não possuem um tempo polinomial definido, ou seja, não é possível se determinar o tempo de processamento necessário até sua solução. Neste trabalho adota-se a premissa de utilizar apenas uma chave por alimentador, o que possibilita utilizar resultados pré-calculados para cada posição de uma chave em um alimentador, transformando o problema em uma otimização combinatoria.

Programação Inteira (PI) é o nome dado às rotinas para solucionar problemas nos quais as variáveis devem ter apenas valores inteiros. Este tipo de problema ocorre geralmente quando se quer modelar a escolha de certas opções dentre um conjunto finito de alternativas, ou no caso de decisões do tipo sim ou não. Com relação a este último caso, o problema também é binário. Isso quer dizer que as variáveis podem assumir apenas dois valores ou estados, normalmente 0 ou 1, sendo o 0 relacionado à opção negativa (não fazer, ou ausente) e o 1 à opção positiva (fazer, ou presente), neste caso, estando relac-

onado a alocação ou não de uma chave automática em determinada posição de um alimentador.

Segundo Korte e Vygen (2008), problemas que envolvem a Programação Inteira são mais difíceis de solucionar do que os de Programação Linear (PL), pois para resolver PLs existem algoritmos de uso geral que são computacionalmente eficientes, como os métodos Simplex e de Pontos Interiores, o que não existe para PIs. Os métodos de solução para PIs possuem quatro características, duas quanto à aplicação e duas quanto à qualidade da solução:

Uso Geral: podem resolver qualquer PI, mas são altamente ineficientes do ponto de vista computacional e se aplicam apenas a problemas de pequeno porte;

Uso Especial: são formulados para resolver um tipo particular de PI de forma muito mais eficiente;

Ótimo: um algoritmo ótimo é aquele que garante matematicamente que a solução ótima pode ser encontrada;

Heurística: os algoritmos heurísticos são capazes de encontrar boas soluções factíveis, geralmente próximas do ótimo, com boa eficiência computacional. Também podem encontrar o ponto ótimo, entretanto não há garantias de que isso aconteça sempre.

Assim, os algoritmos para resolver PIs podem ser de quatro categorias:

Uso Geral, Ótimo: nesta categoria tem-se basicamente dois tipos de algoritmos, a enumeração completa (busca exaustiva) e o Branch-and-Bound (Busca em árvore). O primeiro é factível apenas para problemas de pequeno porte, com poucas variáveis, pois a explosão combinatória torna praticamente impossível a avaliação de todas as possibilidades de problemas maiores. Já o segundo algoritmo realiza uma enumeração sistemática, geralmente resultando na avaliação de uma pequena fração das alternativas, e dentro de determinadas condições garante encontrar a solução ótima.

Uso Geral, Heurística: são algumas técnicas aplicadas aos algoritmos de uso geral que visam diminuir o número de avaliações e restringir o tempo computacional para encontrar soluções factíveis.

Uso Especial, Ótimo: são algoritmos especialmente desenvolvidos para um problema específico, baseados na busca em árvore e gerando restrições via:

- ajuste de multiplicadores;
- ascendente dual;

- otimização de subgradiente;
- relaxação lagrangeana.

Uso Especial, Heurística: existe uma série de algoritmos desenvolvidos para solução de PIs que dependem de ajustes muito específicos de acordo com o problema em questão, são exemplos disto:

- algoritmos de intercâmbio;
- algoritmos genéticos ou evolucionários;
- busca gananciosa;
- busca tabu;
- heurísticas lagrangeanas;
- recozimento simulado.

Entendidos os tipos de métodos aplicáveis para a resolução de problemas que envolvem exclusivamente variáveis inteiras, foram escolhidos dois tipos de algoritmo para resolver a alocação de chaves automáticas. Estes algoritmos foram eleitos com base na abrangência de informações e aplicações, facilidade de implementação e disponibilidade de pacotes prontos de programação, e por serem de características distintas. O primeiro é o Branch-and-Bound, um método de uso geral e ótimo. Já o segundo é o Algoritmo Genético, que é definido como uma heurística de uso especial.

A seguir são abordadas as teorias relativas aos métodos, e no item 5 serão apresentados resultados da aplicação de ambos os algoritmos no problema de alocação de chaves automáticas em redes de distribuição, definindo os parâmetros e técnicas que mais se adequam ao objetivo.

2.1 Branch-and-Bound (BB)

O método Branch-and-Bound é geralmente usado para a solução de problemas combinatórios de grande porte com variáveis inteiras. O método consiste na enumeração sistemática de soluções candidatas, gerando uma árvore de opções, e a cada passo ou ramificação (*branching*) é resolvido um problema de programação linear com um subconjunto de variáveis, em que a restrição de valor inteiro é relaxada, para se obter os limites superior e inferior (*bounds*) que a função objetivo pode atingir. Quando é verificado que os limites não podem levar a uma solução melhor do que a anterior, a sequência de geração é interrompida, e o processo retrocede ao passo anterior para que uma nova sequência de geração seja tentada. Nesse caso, todas as opções que incluem

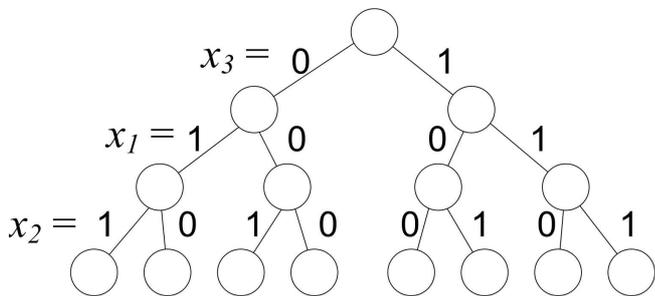


Figura 3: Árvore de busca para o caso binário com 3 variáveis

aquele valor para a variável podem ser ignoradas, isto é, não precisam ser avaliadas e a árvore é dita podada. O algoritmo:

1. Procura por um vetor com solução inteira viável;
2. A medida que a árvore cresce, se atualiza a melhor solução encontrada sempre que aparece uma solução melhor;
3. Verifica se nenhuma solução inteira melhor é possível por meio de uma série de sistemas lineares, impedindo que combinações desse tipo sejam testadas, mudando então para outros ramos da árvore de busca.

2.1.1 Branching

Uma ramificação é uma divisão do espaço de busca original em subconjuntos. Para criar a árvore do espaço de busca o algoritmo adiciona repetidamente restrições ao problema, e a cada ramificação uma variável cujo valor ainda não é inteiro é escolhida para receber uma restrição. Este processo é chamado de dicotômico, e os subconjuntos gerados são disjuntos. A convergência do algoritmo é garantida se o tamanho de cada subconjunto for menor do que o anterior, e o número de soluções para o problema inicial for finito (CLAUSEN, 1999). Por exemplo, no caso binário onde a variável pode assumir apenas os valores 0 e 1, a restrição $x_j = 0$ formaria um ramo e $x_j = 1$ o outro. Este processo pode ser representado por uma árvore como a da Figura 3, em que os arcos ilustram as restrições adicionadas. Essa figura demonstra todas as possibilidades para um problema com três variáveis binárias, x_1 , x_2 e x_3 . A ordem que as variáveis seguem nos níveis da árvore não é necessariamente a ordem dos seus índices, pois depende da estratégia de ramificação adotada.

Em cada nó o algoritmo resolve um problema de programação linear relaxado usando as restrições daquele

nó, e decide quando ramificar ou se mover para outro nó dependendo do resultado obtido. Existem três possibilidades:

1. Se o problema de programação linear relaxado no nó atual é inviável, ou o seu valor ótimo é maior que o do melhor ponto inteiro armazenado, o algoritmo remove o nó da árvore e nenhum nó abaixo deste é avaliado. Então o algoritmo se move para um novo nó de acordo com a estratégia de busca.
2. Se o algoritmo encontra um novo ponto inteiro viável com um valor objetivo menor que o melhor ponto inteiro armazenado, então o algoritmo também remove o nó após atualizar o melhor ponto inteiro e passa para o próximo nó.
3. Se o problema de programação linear relaxado encontra um ótimo não inteiro cujo valor objetivo é menor que o do melhor ponto inteiro, o algoritmo realiza a ramificação de acordo com o método especificado.

Os métodos de ramificação mais comuns para decidir como criar novos nós na árvore de busca são a Inviabilidade Mínima e a Inviabilidade Máxima. O primeiro seleciona a variável com a menor diferença para atender a restrição, ou seja, a variável que está mais próxima de 0 ou 1. Já o segundo seleciona a variável com a maior diferença entre as restrições, ou seja, a mais próxima a 0,5.

A estratégia de busca que é usada para definir qual será o próximo nó da árvore a ser avaliado se resume em duas opções:

Busca em Profundidade: o algoritmo sempre procura por um nó que ainda não foi avaliado em um nível abaixo do nó atual na árvore de busca. Senão, o algoritmo vai para um nível acima e então desce para o próximo nó disponível.

Melhor Nó: o algoritmo segue a busca escolhendo o nó que apresenta o menor limite inferior, dentre os nós disponíveis na árvore. Essa estratégia geralmente resulta em um menor número de nós avaliados, e se sobrepõe ao uso de uma busca em largura, que prioriza nós em níveis superiores.

Também é possível se adotar uma combinação destas duas estratégias, envolvendo inicialmente uma busca em profundidade até que pelo menos uma solução viável seja encontrada, seguindo com uma mescla entre busca pelo melhor nó e profundidade para tentar provar a otimalidade e também encontrar melhores soluções viáveis

(WOLSEY, 1998).

2.1.2 Bounds

A solução do problema de programação linear relaxado fornece o limite inferior para o problema de programação inteira binária. Se esta solução já for um vetor binário, então será o limite superior. Com estes resultados, o algoritmo atualiza os limites inferiores e superiores da função objetivo enquanto a árvore de busca cresce com mais nós. Esses limites são usados pelo algoritmo para cortar ramos desnecessários.

2.1.3 Critérios de Parada

O algoritmo Branch-and-Bound pode potencialmente se aproximar de uma busca exaustiva e avaliar quase todo o espaço de 2^n vetores binários, sendo n o número de variáveis. Desta forma, para limitar o tempo de execução deste algoritmo pode-se usar os seguintes critérios de parada:

- máximo número de nós visitados;
- máximo número de iterações para o problema de programação linear relaxado executar em um nó;
- tempo máximo de execução.

2.2 Algoritmo Genético (AG)

Os fundamentos teóricos dos Algoritmos Genéticos (AG) foram apresentados por Holland (1975), com a ideia de imitar o processo evolucionário que ocorre com os organismos biológicos na natureza. O algoritmo se baseia no processo de seleção natural, onde os indivíduos mais bem adaptados ao meio conseguem se reproduzir com maior frequência, transmitindo suas características genéticas aos descendentes. A reprodução é o ponto chave, na qual a evolução se processa. Recombinando os códigos genéticos dos ancestrais, novos cromossomos são produzidos, os quais, eventualmente passam por um processo denominado de mutação. Com esses processos, os descendentes poderão vir a apresentar características distintas de seus ancestrais sendo que, eventualmente, tais características permitem que o indivíduo gerado venha a ter uma maior capacidade de adaptação ao meio em que vive (DAVIS, 1991). Segundo Goldberg (1989) os AG são métodos numéricos de otimização que apresentam aspectos peculiares, em relação aos demais, que garantem sua robustez, tais como: trabalhar com codificação de parâmetros ao invés de variáveis originais do problema; pesquisar soluções ótimas a partir de

um conjunto de soluções e não a partir de apenas um ponto inicial; empregar uma função de avaliação para as diferentes soluções encontradas e utilizar regras probabilísticas e não determinísticas na pesquisa de novas soluções. Assim, os AG são simples, flexíveis, robustos e particularmente úteis na solução de problemas sem tempo polinomial determinado, em que outras técnicas de otimização deparam com dificuldades. Os algoritmos genéticos apresentam uma estrutura similar ao que se observa na natureza, sendo um processo de busca estocástica sem o uso de derivadas, passível de aplicação em uma série de problemas combinatórios. Seus passos principais são:

1. Gerar uma população inicial
2. Avaliar a aptidão dos indivíduos da população
3. Repetir
 - i Selecionar ancestrais da população
 - ii Efetuar a recombinação entre os ancestrais
 - iii Realizar mutação em alguns descendentes gerados
 - iv Avaliar a aptidão dos descendentes gerados
 - v Substituir indivíduos da população pelos descendentes

Até que uma solução satisfatória tenha sido encontrada

2.2.1 Estrutura do Cromossomo

A estrutura do cromossomo é um dos elementos fundamentais na definição do processo evolutivo. Nos algoritmos genéticos é nessa estrutura que se encontram armazenadas as características das soluções, o que representa um ponto de busca no espaço de estados. Também denominado de indivíduo, o cromossomo é definido por um vetor de comprimento finito que contém uma combinação de elementos chamados genes. O cromossomo geralmente é uma representação codificada de parâmetros da função objetivo na forma de um vetor. A população inicial de indivíduos compostos por seus cromossomos pode ser obtida por meio da geração aleatória de vetores. Essa geração pode atender condições de contorno previamente estabelecidas pelo usuário, tendo em vista o seu conhecimento do problema a ser otimizado, ou pode ser obtida pelo escalonamento do número de indivíduos que compõem a população no intervalo especificado. O tamanho da população ainda é escolhido de forma heurística, que depende da experiência do usuário e do seu conhecimento sobre a função e/ou problema.

Contudo, é sabido que quanto maior o número de indivíduos, maior a probabilidade de convergência e maior o processamento demandado a cada iteração.

2.2.2 Avaliação da Aptidão

A aptidão de um cromossomo pode ser entendida como a capacidade do indivíduo de adaptar-se ao meio ambiente. No caso dos algoritmos genéticos, quando aplicados sobre problemas de otimização combinatória, a medida de aptidão costuma se relacionar com o valor da função objetivo das diferentes configurações. É uma quantificação da otimalidade de uma solução, que permite uma comparação direta entre diferentes indivíduos. O valor de aptidão é usado para selecionar pares de indivíduos para recombinação e para transcrever cromossomos para a próxima geração.

2.2.3 Processo de Seleção

A cada iteração do algoritmo, uma porção da população é selecionada para realizar cruzamentos de cromossomos e produzir novos indivíduos. As soluções são selecionadas por meio de processos baseados na avaliação da aptidão, onde indivíduos mais aptos têm maior probabilidade de serem escolhidos. Alguns métodos avaliam toda a população e fazem uma classificação decrescente, outros selecionam apenas uma amostra da população, se o processo de avaliação demandar muito tempo. A maioria das funções é baseada em processos estocásticos que visam permitir a seleção de certa quantidade de indivíduos com menor aptidão. Isso ajuda a manter a diversidade de soluções e evita uma convergência prematura do algoritmo para um ótimo local.

2.2.4 Processo de Recombinação

A recombinação é um dos principais mecanismos do AG, o qual gera novas soluções a partir da troca de genes entre dois cromossomos. Os ancestrais são selecionados de acordo com um dos processos descritos anteriormente, e então um dos operadores de recombinação é aplicado. O mais comum é a recombinação simples (*single point crossover*), que consiste em escolher um ponto de corte aleatório na estrutura dos cromossomos ancestrais, e combinar a parte à esquerda do corte de um ancestral com a parte à direita do corte do outro ancestral. Essa operação permite que, para cada par de ancestrais selecionados, dois novos descendentes sejam gerados.

2.2.5 Processo de Mutação

Ao contrário do processo de recombinação, o operador de mutação é um mecanismo importante para a diversificação do espaço de busca, o que ajuda o algoritmo a não ficar preso em soluções sub-ótimas. Por meio deste operador, informações que foram eliminadas pela seleção, ou que ainda não faziam parte da população, poderão ser introduzidas na próxima geração. No caso de cromossomos com listas ordenadas, o espaço de estados visitados fica limitado às combinações possíveis entre indivíduos da população inicial se não ocorrer mutação. Geralmente, a mutação ocorre a uma taxa constante na população, afetando de forma aleatória genes de indivíduos.

2.2.6 Convergência

Por se tratar de um processo estocástico, nos problemas formulados com AG não é possível determinar o tempo até sua convergência nem garantir o ótimo global, que pode ser encontrado desde que haja mutação. Geralmente a função objetivo tende a melhorar rapidamente nas primeiras iterações e então passa a estagnar por longos períodos. As condições mais comuns para admitir que o processo convergiu e apresentar a solução final são:

- atingir um valor da função objetivo;
- número máximo de gerações (iterações);
- limite de tempo;
- o valor da melhor aptidão não melhora por determinado número de gerações.

3 CÁLCULO DOS INDICADORES DE CONTINUIDADE

Os indicadores de continuidade do fornecimento de energia elétrica (valor esperado de energia não fornecida ou número esperado de consumidores-hora sem fornecimento de energia) representam a confiabilidade do sistema, e alguns como o DEC e o FEC são regulados pela ANEEL.

A formação dos modelos matemáticos para estimar esses indicadores para cada conjunto consumidor apresenta algumas dificuldades devido à complexidade da criação de expressões analíticas em função dos diferentes tipos e lugares de instalação dos equipamentos de proteção e controle. Para a simplificação da solução deste problema pode ser utilizada a Matriz Lógico-Estrutural –

MLE (ABAIDE, 2005). A formação desta matriz depende das definições das características de tempo de restabelecimento do fornecimento de energia, que não dependem somente da instalação de equipamentos de proteção, mas também da quantidade, composição e esquema de distribuição destes equipamentos.

O indicador de duração de falhas é obtido a partir da expressão (1) que aplica operações nos elementos da MLE:

$$DEC = \frac{\sum_{i=1}^n \left(\sum_{j=1}^n M_{i,j} \right) N_i}{N_c} \quad (1)$$

Em que:

DEC = Duração Equivalente por Consumidor;
 $M_{i,j}$ = Elemento da linha i e coluna j , da MLE;
 N_i = Número de consumidores do nó i ;
 N_c = Número de consumidores do conjunto;
 n = Número de nós.

Já o indicador ENS é calculado pela equação (2), onde o somatório dos elementos da MLE é multiplicado pela respectiva carga da linha e os resultados de todas as linhas são somados.

$$ENS = \sum_{i=1}^n \left(\sum_{j=1}^n M_{i,j} \right) L_i \quad (2)$$

Em que:

ENS = Energia Não Suprida;
 $M_{i,j}$ = Elemento da linha i e coluna j , da MLE;
 L_i = Carga do nó i ;
 n = Número de nós.

Para se obter o indicador de frequência equivalente de falhas por consumidor, a Matriz Lógico-Estrutural deve ser modificada. Os elementos não-nulos da MLE devem ser substituídos por 1, ou seja, em vez de tempo, teremos a ocorrência ou não de uma falha. Assim, a expressão para o FEC é a seguinte:

$$FEC = \frac{\sum_{i=1}^n \left(\sum_{j=1}^n M_{i,j}^* \right) N_i}{N_c} \quad (3)$$

Em que:

FEC = Frequência Equivalente por Consumidor;
 $M_{i,j}^*$ = Elemento da linha i e coluna j , da MLE modificada;
 N_i = Número de consumidores do nó i ;
 N_c = Número de consumidores do conjunto;
 n = Número de nós.

Os pontos candidatos a receberem uma chave NF automática resultarão da análise de fluxo de potência, transferindo cada ponto de carga de um alimentador, no caminho da chave NA até a subestação, para o alimentador adjacente, até que os níveis mínimos de tensão ou máximos de carregamento sejam atingidos. Após a obtenção de todos os pontos candidatos (PCs) de cada par de alimentadores, calculam-se os indicadores de continuidade de cada posição (SPERANDIO, 2008).

4 FORMULAÇÃO DO PROBLEMA

O problema de planejamento da automação de sistemas de manobra em redes de distribuição geralmente envolve um orçamento restrito para o investimento inicial. Sendo assim, deseja-se minimizar o número de chaves automáticas que serão alocadas, mas de forma a se obter o máximo retorno financeiro por meio da redução dos tempos de interrupção por consumidor e por carga. Outro objetivo é atender as metas para os indicadores de continuidade regulados pela ANEEL. Então, como o número de chaves será menor do que o número de pares de alimentadores, tem-se um problema combinatório. Para resolver esse problema, propõe-se como função objetivo a minimização dos custos de instalação das chaves mais os custos das perdas com a Energia Não Suprida (ENS). Como restrições são usadas as metas de DEC e FEC para cada conjunto consumidor afetado pela área que se deseja automatizar, e foi decidido se limitar a alocação a apenas uma chave NF automática por alimentador, para reduzir o espaço de busca e facilitar a formulação do problema, já que a dependência topológica restringe os ganhos com mais de uma chave por alimentador (CARVALHO et alii, 2005). Mesmo assim, ainda é possível se alocar mais de uma chave em um mesmo alimentador com esta formulação, mas para isso é necessário rodar novamente a otimização considerando os pontos já automatizados previamente na MLE com tempo de restabelecimento reduzido. Além do mais, esta abordagem permite que a simulação elétrica do sistema e os cálculos dos indicadores sejam feitos *a priori*, possibilitando que seus resultados sejam usados na formulação de problema combinatório da seguinte forma:

$$\begin{aligned} & \min \quad CC + CENS \\ \text{s.a.} \quad & \sum DEC_s < MetasDEC \\ & \sum FEC_s < MetasFEC \\ & nca = < 1 \end{aligned} \quad (4)$$

Em que:

CC = Custo de instalação das Chaves;
 $CENS$ = Custo da Energia Não Suprida de todo o sistema;

$DECs$ = indicadores de duração para cada conjunto;
 $FECs$ = indicadores de frequência para cada conjunto;
 $MetasDEC$ = metas para os indicadores de duração;
 $MetasFEC$ = metas para os indicadores de frequência;
 nca = número de chaves NF por alimentador.

Para aplicar o método Branch-and-Bound no problema de alocação de chaves é preciso montar a função objetivo (f) e as matrizes de restrições de desigualdade (A) e igualdade (Aeq), bem como os vetores de restrições (b e beq), na seguinte forma:

$$\begin{aligned} & \text{Min } f.x \\ \text{s.a. } & A.x = < b \\ & Aeq.x = beq \end{aligned} \quad (5)$$

O resultado é um vetor coluna x na forma binária, que multiplicado pelo vetor f resulta no valor mínimo para atender as condições impostas pelas restrições.

A princípio, a função objetivo seria a soma do custo das chaves alocadas mais o custo da ENS do sistema. Porém, este tipo de abordagem não se mostrou adequada computacionalmente, pois abrange um grande espaço de busca. Então, observou-se que o número de chaves alocadas poderia ser uma restrição de igualdade, o que resultou em maior agilidade na solução do problema, sendo reformulado conforme (6). Este artifício é válido porque o custo de uma chave é muito maior do que o seu benefício no curto prazo, sendo que o número de chaves mínimo é obtido em função das metas de DEC e FEC, e a sua posição define o mínimo ENS. Caso as chaves não tivessem custo, seriam alocadas em todos os alimentadores.

$$\begin{aligned} & \text{min } CENS \\ \text{s.a. } & \sum DEC_s = < MetasDEC \\ & \sum FEC_s = < MetasFEC \\ & \sum pca = 1 \\ & \sum c = nc \end{aligned} \quad (6)$$

Em que:

$CENS$ = Custo da Energia Não Suprida de todo o sistema;

$DECs$ = indicadores de duração para cada conjunto;
 $FECs$ = indicadores de frequência para cada conjunto;
 $MetasDEC$ = metas para os indicadores de duração;
 $MetasFEC$ = metas para os indicadores de frequência;
 pca = vetor binário com a posição da chave NF por alimentador;

c = chaves NF do sistema;

nc = número de chaves NF desejado.

Sendo assim, o vetor f contém os valores da ENS, adquiridos previamente via simulação de falhas, como se houvesse uma chave NF automática alocada naquele ponto do alimentador. A matriz A contém os valores de DEC e FEC relativos a cada conjunto para cada Ponto Candidato, obtidos com a simulação de falhas. Logo, essa matriz deve ter o número de linhas igual a duas vezes o número de conjuntos supridos pelo sistema em estudo, uma linha para cada indicador do conjunto, e o número de colunas é igual ao tamanho do vetor x . O vetor b contém as metas para limitar os indicadores. Essas metas não são necessariamente as definidas pelo órgão regulador para o conjunto, pois o resultado da multiplicação $A.x$ depende do número de alimentadores sendo analisados, que pode ser menor do que o total de alimentadores que suprem o conjunto. A matriz de restrições de igualdade define que só pode ser escolhido um ponto por alimentador, incluindo o ponto zero (sem chave). Logo, tem tantas linhas quanto for o número de alimentadores, mais uma para a restrição do número de chaves NF, cujo somatório deve ser igual ao valor desejado para a alocação.

Como o número de chaves é definido por uma restrição de igualdade, além das restrições de metas de DEC e FEC, para se minimizar a quantidade de chaves enquanto se busca também minimizar a ENS é preciso testar a viabilidade de resolução do problema várias vezes, diminuindo o valor da restrição de chaves enquanto for possível. Para isso, o problema é resolvido via programação linear relaxada (PLR). Se a melhor solução possível via PLR não atende às restrições, implica que também não existe uma solução com valores inteiros para x que não viole tais restrições. Assim, a otimização do vetor x será executada pelo Branch-and-Bound com o número de chaves obtido antes do problema se tornar inviável.

O tempo computacional para a resolução deste tipo de problema não pode ser previsto, porém apresentou-se proporcional ao número de variáveis. Outro fator que influenciou o tempo foi a estratégia de busca. Nas simulações feitas, a busca em profundidade exigiu em média um tempo 20 vezes maior para encontrar a solução do que a busca pelo melhor nó, sendo recomendável esta última.

Um Algoritmo Genético (AG) para resolver o problema da melhor alocação de chaves automáticas que atenda aos requisitos da função objetivo e restrições conforme formulado em (5) pode ser codificado por um cromossomo composto por tantos genes quanto forem os pontos candidatos a receberem uma chave. Dessa forma, cada gene poderia assumir o valor de zero ou um, indicando a

1	2	3	4	5	6	7	8	9	10
4	8	0	1	4	0	9	5	5	0

Figura 4: Codificação inteira para um cromossomo do problema de alocação

presença ou não de uma chave automática, semelhante ao vetor x do método Branch-and-Bound. Esse tipo de codificação resulta em um cromossomo muito longo e de difícil interpretação direta pelo usuário.

Devido a maior flexibilidade de modelagem do AG, optou-se por outro tipo de codificação, a inteira. Nesse caso, o tamanho do cromossomo é igual ao número de alimentadores disponíveis para automação, e cada gene pode assumir um valor inteiro definido entre zero e o número de pontos candidatos do alimentador. Assim, o usuário consegue fazer a leitura da solução apresentada pelo cromossomo de forma direta.

Por exemplo, para um sistema com 10 chaves NAs conectando alimentadores diferentes, e um total de 9 pontos candidatos por alimentador, existem 1010 combinações possíveis para alocar de zero a dez chaves NF. A Figura 4 mostra um cromossomo codificado para este caso, em que a solução proposta seria alocar um total de sete chaves, sendo uma no quarto ponto candidato do alimentador 1, outra no oitavo ponto do alimentador 2, os alimentadores 3, 6 e 10 ficariam sem chave, etc.

A função de recombinação utilizada foi a troca de genes por ponto simples, escolhido aleatoriamente ao longo do cromossomo. Alguns cuidados devem ser tomados na aplicação dos operadores genéticos com este tipo de codificação inteira, principalmente no caso de mutações. É preciso garantir que as mutações não resultem em valores fora do número de PCs para o alimentador. Com essa finalidade foi implementada uma operação de mutação que primeiramente escolhe a quantidade de genes que serão alterados em um cromossomo, isso ocorre de forma aleatória em 1, 3, 6 ou 12 genes com probabilidade 0,3, 0,4, 0,2 e 0,1, respectivamente (Figura 5). Escolhidos os genes que sofrerão mutação, eles são alterados com uma escolha aleatória equiprovável dentro do número de PCs do alimentador.

Os valores de DEC, FEC e ENS para cada ponto candidato foram calculados previamente e ficam armazenados em uma matriz que tem uma linha para cada indicador, e uma coluna para cada ponto candidato (mesma matriz de desigualdade usada no Branch-and-Bound). Os valores dos genes do cromossomo apontam para os elementos

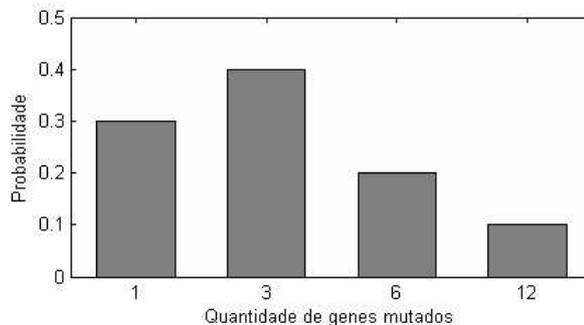


Figura 5: Função de mutação

dessa matriz. A aptidão é então avaliada pelo indicador ENS mais o custo do número de chaves, e é penalizada em soluções em que as metas dos indicadores são ultrapassadas. Para uma boa convergência do algoritmo, é necessário que o custo da chave seja maior que o valor da ENS do sistema, e a penalização das metas seja maior ainda. Dessa forma, o algoritmo busca primeiramente atender as metas, em seguida minimizar o número de chaves, e só então procura pela configuração que resulte no menor valor de Energia Não Suprida do sistema. Por exemplo, se o valor da ENS for da ordem de 10^7 , o custo de uma chave automática deve ser de pelo menos 10^8 , e múltiplo da diferença entre o indicador obtido e a sua meta deve ser igual a 10^9 .

Os parâmetros que se mostraram mais adequados para resolver um problema de alocação envolvendo 36 alimentadores com um total de 166 pontos candidatos foram: população de 50 indivíduos e 70% de recombinação com seleção estocástica.

O critério de parada pode ser um determinado número de gerações em que o melhor valor de aptidão não mais evolui, normalmente 100, ou um valor a ser atingido pela função objetivo.

5 RESULTADOS

Com a finalidade de avaliar a metodologia de alocação de chaves de manobra automáticas em redes de distribuição foram feitas simulações usando um sistema teste baseado em alimentadores reais. O sistema conta com 7 subestações que fornecem energia elétrica a 316.898 consumidores, agrupados em 4 conjuntos, através de 36 alimentadores de 13,8 kV. Cada conjunto possui metas distintas para os indicadores de continuidade DEC e FEC. O objetivo é minimizar a ENS do sistema com o menor número possível de chaves automáticas e de

forma a atender a uma redução de aproximadamente 10% nos indicadores.

Os alimentadores passaram por uma redução de pontos, e as respectivas taxas de falha e os tempos de reparo foram calculados por trecho. A determinação dos pontos candidatos à alocação de uma chave de manobra automática configurada na posição normalmente fechada (NF) é condicionada à existência de uma interligação com outro alimentador por uma chave normalmente aberta (NA), e à disponibilidade do sistema de comunicação escolhido. Também foram verificados, via fluxo de carga, os limites de tensão estabelecidos pela ANEEL e as correntes máximas suportadas pelos condutores ao se realizar a transferência de carga. Isso foi feito levando-se em consideração fatores de potência e demanda particulares para cada alimentador.

O cálculo dos indicadores de continuidade DEC e FEC e da ENS de cada alimentador foi feito por meio da Matriz Lógico-Estrutural (MLE), considerando-se as proteções existentes e tempos de manobras manuais. Então, para cada ponto candidato, essas variáveis foram recalculadas com a automação do conjunto NA-NF, ponderando a taxa de falha do meio de comunicação. Após serem obtidos os valores para todos os pontos candidatos, foi realizada uma análise de dominância para excluir os pontos não relevantes. Os indicadores dos pontos candidatos não-dominados são então utilizados na formulação de um problema de programação inteira, cujo objetivo é minimizar a ENS e o número de chaves, sujeito às metas determinadas para os indicadores DEC e FEC de cada conjunto. O número de chaves por alimentador é limitado a um. Para resolver o problema foram utilizados dois métodos, o Branch-and-Bound (BB) e o Algoritmo Genético (AG). Ambos os métodos usam as informações contidas nas matrizes de igualdade (Aeq) e desigualdade (A) da equação (5).

A unidade da ENS é kWh/ano, e para transformar a energia “recuperada” pela automação em uma importância monetária considerou-se o valor de 0,14391 \$/kWh, referente à diferença entre o valor médio da compra de energia pela concessionária e a tarifa básica cobrada pela distribuição aos consumidores cativos. A multa por ultrapassar os indicadores de continuidade é definida em (BARBOSA et al., 2006). No ano 2003, 26 empresas distribuidoras foram multadas por terem violado metas de DEC e FEC em 2002, totalizando R\$ 35,3 milhões.

5.1 Otimização com BB

A Tabela 1, a seguir, apresenta a ENS do sistema e os indicadores de continuidade DEC e FEC para os quatro

Tabela 1: Indicadores antes e depois da automação de 36 chaves, uma por alimentador.

Ind.	Antes	Restrição	Depois	Redução	%
ENS	15127.10 ³	Minimizar	11726.10 ³	3401.10 ³	22,5
DEC ₁	31,09	28	25,46	5,63	18,1
FEC ₁	24,48	22	20,70	3,77	15,4
DEC ₂	19,24	17	15,42	3,81	19,8
FEC ₂	19,13	17	15,84	3,29	17,1
DEC ₃	33,08	29	24,07	9,01	27,2
FEC ₃	35,24	32	28,16	7,08	20,0
DEC ₄	34,08	30	23,74	10,33	30,3
FEC ₄	30,91	28	27,37	3,53	11,4

conjuntos: antes de ter qualquer chave automática, as respectivas restrições impostas aos indicadores (metas), os resultados depois da alocação de 36 chaves automáticas NF (uma por alimentador), bem como as diferenças absolutas e relativas após a otimização.

Com a alocação de uma chave automática NF em cada alimentador do sistema teste é possível se recuperar em torno de 22,5% da Energia Não Suprida, o que resultaria na quantia de R\$ 489.466,42. Além disso, os indicadores de continuidade teriam uma redução média de quase 20%, o que implica em evitar multas pela transgressão das metas impostas pelo órgão regulador. Multas essas que podem chegar à cifra de R\$ 1.065.046,48 se considerarmos como metas as restrições do problema.

Após essa primeira análise, buscou-se minimizar o número de chaves automáticas a serem alocadas no sistema. Como esse número de chaves foi definido como uma restrição de igualdade, o algoritmo testa a viabilidade de solução do problema na primeira iteração, e caso seja viável, cancela a otimização e diminui o número de chaves previamente estabelecido. Esse processo iterativo continua até a solução ser inviável, e então o problema é resolvido com a última restrição que permite a viabilidade.

Esta abordagem de forçar uma restrição para a variável número de chaves, e testar a viabilidade do problema de forma iterativa, mostrou-se muito mais rápida do que deixar o algoritmo minimizá-la em conjunto com a ENS. Deste modo o problema fica muito mais restrito, diminuindo o espaço de busca e agilizando a sua solução.

O tempo computacional para resolução do problema de programação inteira não pode ser determinado, e o algoritmo Branch-and-Bound pode potencialmente avaliar as 2ⁿ possibilidades, em que n é o número de variáveis. Neste caso, tem-se 166 variáveis, sendo 130 pontos candidatos mais 36 opções de não se alocar uma chave, o que resulta em 9,35 · 10⁴⁹ possibilidades. Porém, as res-

trições do número de chaves e de se alocar apenas uma por alimentador, reduzem as possibilidades para uma ordem de 10^{15} .

Assim, consegue-se um número mínimo de 11 chaves para atender às metas de continuidade. Os resultados da ENS do sistema e dos indicadores estão na Tabela 2. Observando os valores dessa tabela é possível se verificar que os indicadores que ficaram mais próximos das suas restrições foram o DEC1 e o FEC4. O indicador com maior folga foi o DEC4, tendo uma redução de 21,86%. A diferença da ENS antes e após a automação proposta é de 1.747.725,60 kWh/ano, o que resultaria em um ganho de R\$ 251.515,19. Tal resultado, obtido com o número mínimo de chaves, é praticamente metade do que seria conseguido com uma chave por alimentador, uma diferença de R\$ 237.951,22. No entanto, o resultado é obtido com 25 chaves a menos.

Tabela 2: Indicadores antes e depois da automação com número mínimo de chaves via Branch-and-Bound.

Ind.	Antes	Restrição	Depois	Redução	%
ENS	15127.10 ³	Minimizar	13379.10 ³	1747.10 ³	11,5
DEC ₁	31,09	28	27,98	3,11	10,0
FEC ₁	24,48	22	21,72	2,75	11,2
DEC ₂	19,24	17	16,67	2,56	13,3
FEC ₂	19,13	17	16,84	2,29	12,0
DEC ₃	33,08	29	28,79	4,28	12,9
FEC ₃	35,24	32	31,62	3,62	10,3
DEC ₄	34,08	30	26,63	7,45	21,8
FEC ₄	30,91	28	27,98	2,92	9,5

5.2 Otimização com AG

O mesmo problema que foi otimizado com o método Branch-and-Bound, apresentado no item anterior, foi resolvido novamente por um Algoritmo Genético (AG). As mesmas metas para os indicadores foram adotadas, porém em vez de serem restrições, o fato de ultrapassá-las penaliza a função objetivo em 10^9 vezes a diferença. O número de chaves foi minimizado em conjunto com a ENS, sendo que o valor de uma chave foi ajustado para ser de 10^8 . De forma que os dois primeiros algoritmos do valor da função objetivo representam o número de chaves alocadas.

Os resultados obtidos após a convergência do algoritmo foram os mesmos da Tabela 2, e a Figura 6 mostra um gráfico da evolução dos valores da função objetivo do melhor indivíduo e da média da população ao longo das gerações.

Alguns resultados sobre o processamento efetuado pelos dois algoritmos de programação inteira empregados

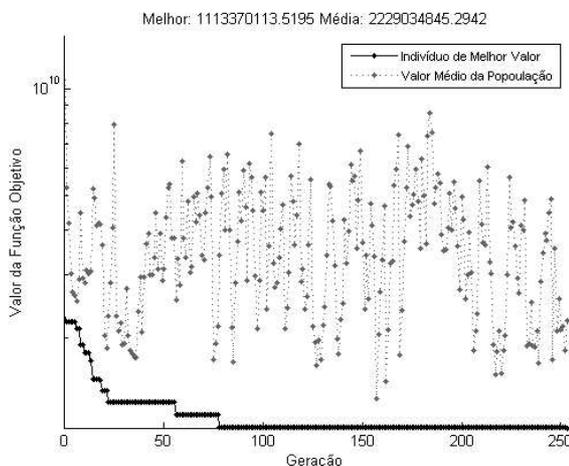


Figura 6: Evolução da população do AG

neste estudo de caso são exibido na Tabela 3, servindo de base para comparar seus desempenhos. As avaliações são referentes ao número de problemas lineares relaxados executados pelo Branch-and-Bound, e ao número de avaliações da função objetivo (fitness) do Algoritmo Genético. As iterações exprimem o número de nós avaliados pelo BB e o número de gerações do AG.

Pode-se observar na tabela a seguir que os algoritmos apresentaram um número de iterações próximo, porém o AG realizou muito mais avaliações, e mesmo assim foi mais rápido. Isso porque cada avaliação do AG é uma simples multiplicação matricial, enquanto cada PLR do BB exige um processamento muito maior.

Tabela 3: Comparação dos algoritmos

	BB	AG
Iterações	304	253
Avaliações	3613	17780
Tempo	23,921 s	6,341 s

O critério de parada do AG foi a não melhora do valor do melhor indivíduo por mais de 100 gerações. Constata-se pela Figura 6 que em torno de 75 gerações a melhor solução já estava bem próxima do seu valor final. Observa-se que parando nesse ponto, o AG levaria apenas um terço do tempo e obteria uma solução apenas 0,2% inferior.

6 CONCLUSÕES

Foram apresentadas algumas estratégias de formulação e verificados dois métodos distintos de programação inteira e busca combinatória a fim de resolver o problema

de alocação de chaves automatizadas em redes de distribuição.

O Branch-and-Bound, com uma abordagem generalista, que necessita basicamente da montagem das matrizes de restrições, mas que garante encontrar a solução ótima, serviu para balizar a resposta de uma heurística de uso especial, o Algoritmo Genético. Este último exige um maior ajuste de parâmetros, e um conhecimento do problema para formulação da função de aptidão e julgamento da convergência do algoritmo, no entanto, uma vez ajustado, permite encontrar a solução ótima, ou outras muito próximas do ótimo, com eficiência. Não obstante, se o usuário não dispuser de pacotes computacionais com estes métodos, a implementação de um AG é muito mais rápida e fácil.

Constata-se que mesmo com um investimento baixo, determinado pelo número mínimo de chaves automáticas necessário para atender às metas, o posicionamento ótimo dessas chaves pode trazer grandes benefícios para as concessionárias distribuidoras de energia elétrica e seus consumidores. O rápido reestabelecimento de seções da rede sob falta de energia permite um maior faturamento para a empresa e minimiza o tempo de desligamento, o que também ajuda a evitar multas por parte dos órgãos reguladores. Além do mais, é crescente o movimento das empresas de energia elétrica em direção ao conceito de “*Smart Grid*”, que proporciona o aumento da eficiência por meio da automação da rede.

REFERÊNCIAS BIBLIOGRÁFICAS

- Abaide, A. R. (2005) *Desenvolvimento de Métodos e Algoritmos para Avaliação e Otimização da Confiabilidade em Redes de Distribuição*. Tese de Doutorado, UFSM, Santa Maria - RS.
- Carvalho, P.M.S.; Ferreira, L.A.F.M.; Da Silva, A.J.C.A (2005). *Decomposition Approach to Optimal Remote Controlled Switch Allocation in Distribution Systems*. IEEE Transactions on Power Delivery, v.20,n.2, p.1031-1036..
- Chao-Shun, C., L. Chia-Hung, C. Hui-Jen, L. Chung-Sheng, H. Ming-Yang and H. Chia-Wen (2006). *Optimal placement of line switches for distribution automation systems using immune algorithm*. IEEE Transactions on Power Systems, vol.21, n.3, pp.1209-1217.
- Clausen, J. (1999). *Branch and Bound Algorithms - Principles and Examples*. Department of Computer Science, University of Copenhagen, Denmark.
- Davis, L. (1991). *Handbook of Genetic Algorithms*. New York, VanNostrand Reinhold.
- De Tuglie, E., M. La Scala, G. Patrono, P. Pugliese and F. Torelli(2004). *An optimal strategy for switching devices allocation in radial distribution network*. 7th AFRICON Conference in Africa, pp.683-689.
- Gilbert, D. (2004). After a Major Automation Rollout, the Benefits Roll In. *Transmission and Distribution World*, vol.56, n.5, pp.17-22.
- Goldberg, D.E. (1989). *Genetic Algorithm in Search, Optimization and Machine Learning*. Reading, Addison-Wesley.
- Gruenemeyer, D. (1991). *Distribution Automation: How Should It Be Evaluated?*. IEEE Rural Electric Power Conference, Dearborn - USA, pp.D3.
- Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, The University of Michigan Press.
- Korte, B.H. and J. Vygen (2008). *Combinatorial optimization: theory and algorithms*. 4th ed., Berlin, Springer.
- Matsumoto, Y., A. Ono, S. Hase, T. Kinoshita and K. Abe (2002). *Development of advanced distribution automation system using genetic algorithm*. IEEE/PES Transmission and Distribution Conference and Exhibition, Yokohama - Japan, pp.214-219.
- Qin, Z., D. Shirmohammadi and W.H.E. Liu (1997). *Distribution feeder reconfiguration for service restoration and load balancing*. IEEE Transactions on Power Systems, vol.12, n.2, pp.724-729.
- Rigler, D.M., W.R. Hodgkins and R.N. Allan (1999). *Quantitative reliability analysis of distribution systems: automation*. Power Engineering Journal, vol.13, n.4, pp.201-204.
- Rong-Liang, C. and S. Sabir (2001). *The benefits of implementing distribution automation and system monitoring in the open electricity market*. Canadian Conference on Electrical and Computer Engineering, Toronto - Canada, pp.825-830.
- Sperandio, M., J. Coelho e R. Ramos (2005). *Prospecção de Sistemas de Distribuição Automatizados, uma Visão Estratégica*. VI SIMPASE - Simpósio de Automação de Sistemas Elétricos, São Paulo - SP, pp.R074.

Sperandio, M. (2008). *Planejamento da Automação de Sistemas de Manobra em Redes de Distribuição*. Doutorado em Engenharia Elétrica, UFSC, Florianópolis - SC.

Wolsey, L.A. (1998). *Integer programming*. New York, Wiley