

# OTIMIZAÇÃO DO ATENDIMENTO DE CARGAS DE SUBESTAÇÕES UTILIZANDO ALGORITMOS GENÉTICOS

<sup>2</sup>GERALDO C.C. ORREA, <sup>1</sup>HEITOR S.L. OPES.

<sup>1</sup>Centro Federal de Educação Tecnológica do Paraná  
Programa de pós-graduação em Engenharia e  
Informática  
Av. 7 de setembro, 3165-80230-901 – Curitiba, PR  
hslopes@cefetpr.br

<sup>2</sup>COPEL-Companhia Paranaense de Energia  
R. José Izidoro Biazeto 158 bloco B – Mossunguê  
81200-240 – Curitiba, PR  
gerald@onda.com.br

**Resumo**—Este trabalho apresenta a utilização de algoritmos genéticos na localização de subestações de distribuição de energia elétrica e faz uma comparação com a metodologia atualmente utilizada pela Copel, baseada em um algoritmo criado por Ford & Fulkerson. Apesar dos resultados como algoritmo de Ford & Fulkerson ter sido ligeiramente melhores, existem algumas necessidades que ainda não são atendidas pelo algoritmo. Com a abordagem por algoritmos genéticos tais limitações poderão ser consideradas.

**Abstract**—This paper presents the use of genetic algorithms in electric energy distribution substation allocation and compares with the methodology that has been used by Copel, based in Ford & Fulkerson algorithm. The results of Ford & Fulkerson algorithm were a little better; there are some needs that this algorithm does not support. These limitations may be considered with genetic algorithm approach.

**Keywords**—Substation allocation, genetic algorithms, loads

## 1 Introdução

Para atender à crescente demanda de energia elétrica, é necessário a construção de novas subestações e a ampliação das já existentes. Para tanto, as concessionárias de energia elétrica desenvolvem um plano diretor para definir a quantidade, potência e localização das subestações necessárias para atender uma demanda de carga conforme previsto pelo setor de Estudos de Mercado num horizonte de longo prazo que pode chegar a 25 anos.

A fim de executar estes procedimentos é necessário identificar qual a área que deve ser atendida por uma certa subestação, considerando a capacidade desta subestação, o valor das cargas e as distâncias entre as subestações e as cargas. O objetivo é atender as cargas com o menor custo de transporte. Este custo é medido através dos momentos dos produtos das cargas de uma região pelas respectivas distâncias até as subestações. A este custo de transporte é dado o nome de “Momento Elétrico”.

O algoritmo desenvolvido por Ford & Fulkerson (1962) tem sido utilizado pela Companhia Paranaense de Energia (Copel) para executar estes procedimentos. Como não existe outro algoritmo implementado, os usuários do sistema questionam se as soluções apresentadas estão realmente entre as melhores. O objetivo deste trabalho é montar uma solução de transporte de carga para subestações utilizando algoritmos genéticos (Lopes, 1999) e com

raros resultados como os valores já obtidos em um estudo feito como algoritmo de Ford-Fulkerson.

## 2 Fundamentação teórica

### 2.1 Subestação de carga elétrica

A energia gerada em uma Usina é transportada até os centros de consumo através de linhas de transmissão que trabalham normalmente com tensões acima de 69 kV (69.000 Volts), chamadas de altas tensões. Ao chegar nas grandes cidades as altas tensões devem ser reduzidas para níveis mais adequados às necessidades de segurança da população. A redução destas tensões para valores menores (13,8 kV ou 34,5 kV) é feita em uma subestação.

Todo o consumo de energia elétrica é considerado a carga da rede. Para efeito de estudos de atendimento de cargas de subestações, as cargas são agrupadas em quadrículas. Para efeito de estudos de planejamento de localização de subestações, a área em estudo é dividida em quadrículas de 1 km<sup>2</sup>. Toda a carga de uma quadrícula é agrupada no centro da quadrícula como se fosse carga pontual. O somatório dos produtos das cargas das quadrículas pelas respectivas distâncias até as subestações é denominado de “Momento Elétrico”.

### 2.2 Algoritmo de Ford-Fulkerson

O cálculo de alocação de carga em subestações considerando o menor Momento Elétrico utilizando

o algoritmo de Ford-Fulkerson (1962) necessitados dados de entrada:

- projeção de demanda de carga (até 25 anos) obtidos do setor de Estudos de Mercado. Esta projeção procura identificar qual será a carga em uma determinada quadrícula de 1 km<sup>2</sup> em um certo período no futuro.
- Localização e capacidade das subestações existentes.

O processador na possível simular diversas alternativas de atendimento de cargas das quadriculas em função das possibilidades de ampliação das subestações existentes e da criação de novas subestações.

### 3 Metodologia

#### 3.1 Informações necessárias

As mesmas informações básicas utilizadas no algoritmo de Ford-Fulkerson são necessárias para a utilização do algoritmo genético, isto é: a localização das cargas nas quadriculas de 1 km<sup>2</sup> da área em estudo (refletindo a demanda do mercado futuro) e a localização e capacidade das subestações existentes ou previstas.

#### 3.2 Variáveis do problema de codificação

Um exemplo ser utilizado corresponde a uma área com 3 subestações e 199 quadriculas. Considerando que uma quadrícula pode ser atendida por mais de uma subestação, será codificado um gene correspondente a cada quadrícula para cada subestação, isto é, 3 x 199 = 597 genes. As cargas são medidas em kVA (1000 Volt-Ampère) e, portanto, para reduzir o espaço de busca, o conteúdo dos genes será percentual da carga da quadrícula que será atendida pela subestação correspondente. Este percentual será considerado em intervalos de 20%, ou seja, o conteúdo da quadrícula poderá variar de 0 a 5 (correspondendo a 0%, 20%, 40%, 60%, 80% e 100%). Desta maneira, são necessários 3 bits para cada gene. Cada indivíduo (solução para o problema) será representado por um cromossomo binário de 3 x 199 = 1791 bits. Com este esquema de codificação, o genótipo cria um espaço de busca de 2<sup>1791</sup> = 1,39 x 10<sup>539</sup> o qual, evidentemente é intratável computacionalmente utilizando-se métodos de busca exaustiva.

### 3.3 Restrições

#### 3.3.1 Capacidade das subestações

Para cada subestação  $k$ , há uma restrição em relação à sua capacidade máxima de atendimento de carga, dada pela equação 1:

$$h_k = \text{Máx}\left\{0, \frac{\sum_{i=1}^q PQSE_{ik} \times CQ_i}{100} - \text{CapacSE}_k\right\} \quad [1]$$

onde:

$PQSE_{ik}$  = Percentual da carga da Quadrícula  $i$  que é atendida pela Subestação  $k$ ;

$CQ_i$  = Carga (em kVA) da quadrícula  $i$ ;

$\text{CapacSE}_k$  = Capacidade da Subestação  $k$ ;

$q$  = número de quadriculas.

#### 3.3.2 Atendimento das cargas

O algoritmo deve procurar atender o máximo possível de cargas respeitando a capacidade das subestações. Esta restrição é representada pela equação 2:

$$h_i = \frac{\sum_{k=1}^s \left( \text{Máx}\left\{0, \frac{CQ_i - \sum_{k=1}^s (PQSE_{ik} \times CQ_i)}{100}\right\} \right)}{q} \quad [2]$$

onde:

$s$  = número de subestações.

#### 3.3.3 Carga da quadrícula

As somatórias dos genes correspondentes a uma quadrícula deve ser menor ou igual a 100%, ou seja, a somatória dos percentuais da carga que são atendidas pelas 3 subestações deve ser menor ou igual a 100%. Para facilitar o tratamento da epistasia, as variáveis que correspondem a uma quadrícula ficarão contíguas no cromossomo. Esta restrição será atendida pela correção dos genes no cromossomo.

### 3.4 Função objetivo

A função objetivo é a minimização do produto dos Momentos Elétricos e dada pela equação 3:

$$FO = \prod_{k=1}^s \frac{\sum_{i=1}^q (PQSE_{ik} \times CQ_i \times DQSE_{ik})}{100} \quad [3]$$

onde:

$DQSE_{ik}$  = Distância em metros do centro da quadrícula  $i$  até a subestação  $k$ .

### 3.5 Normalização da função objetivo

O valor normalizador da função objetivo é calculado considerando-se a pior situação, isto é, que todas as quadrículas sejam atendidas pela subestação mais distante de cada uma. Este valor é calculado pela equação 4. Além disso, como o problema é tratado a minimização da função pelo algoritmo genético, por definição, trata a função de *fitness* como maximização, deve-se subtrair de 1 para maximizar.

$$ValorNormObj = \sum_{i=1}^q CQ_i \times MDQSE_i \quad [4]$$

onde:

$MDQSE_i$  = Distância da subestação mais distante da quadrícula  $i$ .

### 3.6 Normalização das restrições

Cada subestação tem uma capacidade máxima de atendimento de carga, conforme a equação 1 e o valor normalizador é calculado considerando-se a pior situação possível, onde todas as cargas estão sendo atendidas por somente uma subestação. Neste caso, o valor normalizador é dado pela equação 5, para cada subestação  $k$ .

$$ValorNormRestr_k = \left( \sum_{i=1}^q CQ_i \right) - CapacSE_k \quad [5]$$

O algoritmo deve procurar atender o máximo possível de cargas respeitando a capacidade da subestação. A restrição dada pela equação 2 já está normalizado, visto que o valor é dividido pelo número de quadrículas, não sendo, portanto, necessário outro valor normalizador desta restrição.

A restrição da carga na quadrícula não necessita de normalização como foi explicado no item 3.3.3.

### 3.7 Função de fitness

Para a definição da função de *fitness*, deve-se levar em conta as considerações dos itens 3.5 e 3.6. Além disso, a função de *fitness* deve incorporar a função objetivo normalizada e as restrições normalizadas. Isto é feito seguindo-se os princípios sugeridos por Goldberg (1989), mostrados na equação 6.

$$f(x) = g(x) + r \frac{\sum_{i=1}^{N-1} \phi(h_i(x))}{N} \quad [6]$$

Nesta equação,  $f(x)$  é a função de *fitness* final,  $g(x)$  a função objetivo normalizada,  $r$  o coeficiente de penalização (neste problema, -4),  $N$  o número de restrições normalizadas,  $h_i$  são as  $i$ -ésimas restrições

a problemas (normalizadas) e  $\phi$  uma função agregadora (em geral próprio somatório). Finalmente a função de *fitness*, a qual o algoritmo genético tentará maximizar é dada pela equação 7.

$$f = 1 - \frac{\prod_{k=1}^s \left( \sum_{i=1}^q (PQSE_{ik} \times CQ_i \times DQSE_{ik}) / 100 \right)}{ValorNormObj} + \frac{\sum_{k=1}^s \left( \text{Max} \left\{ 0, \frac{\left( \sum_{i=1}^q PQSE_{ik} \times CQ_i \right) / 100 - CapacSE_k}{ValorNormRestr_k} \right\} \right)}{s} + \frac{\sum_{i=1}^q \left( \text{Max} \left\{ 0, \frac{CQ_i - \sum_{k=1}^s \left( \frac{PQSE_{ik} \times CQ_i}{100} \right)}{CQ_i} \right\} \right)}{q} \quad [7]$$

onde:

$s$  = número de subestações.

### 3.8 Parâmetros do algoritmo genético

O método de seleção utilizado no algoritmo genético foi o da amostragem estocástica universal (*stochastic universal sampling*) por ser menos agressivo que o tradicional método de seleção proporcional (*roulette wheel*).

Os dois operadores genéticos tradicionais foram utilizados: a mutação (probabilidade variável) e o *crossover* (probabilidade de 90%). Além destes, dois outros operadores genéticos específicos foram desenvolvidos para este problema, descritos nas duas próximas subseções. Como objetivo de fazer um busca global mais eficaz, foi implementado um processo que faz o percentual de mutação variar de forma linear entre 1% (no início do processo) até 20% (no final do processo).

Resumo dos parâmetros:

- população 200
- crossover 90%
- mutação varia entre 1% e 20%
- método de seleção amostragem estocástica universal

### 3.8.1 Mutação agrupadora

Como objetivo de agilizar o processo de agrupamento das quadrículas a uma subestação, foi criado um operador genético de mutação que utiliza um percentual ( $pLSEmutgrupo$ ) para decidir se vai aplicar a mutação a um cromossomo ou não. São sorteadas 5 quadrículas por cromossomo. O operador vai, então, alocar toda a carga de cada quadrícula selecionada para a subestação mais próxima.

O valor da probabilidade  $pLSEmutgrupo$  é alterado ao longo das gerações. Inicia com 1% e aumenta de valor até chegar a 70% no final do processamento.

### 3.8.1 Operador de cargas não atendidas

Este operador tem a função de agilizar o processo de atendimento das cargas (evitando cargas não atendidas). Este operador identifica as cargas não atendidas e as aloca para a subestação mais próxima. A parâmetro  $pLSEnaoatendida$  contém o percentual (em decimal) de cromossomos a que a operação deve ser aplicada.

Assim como o operador de mutação agrupadora, o valor da variável  $pLSEnaoatendida$  é alterado ao longo das gerações, iniciando com 1% e aumentando até chegar a 90% no final do processamento.

## 4 Resultados

O sistema foi implementado utilizando-se como base o software domínio público GALOPPS, versão 3.2 (Goodman, 1996). Como foi utilizado um software de código aberto e bastante flexível, foi possível implementar operadores genéticos especiais (Mutação agrupadora e Operador de cargas não atendidas) que exploram o conhecimento específico do problema. Foram realizados os experimentos com 500, 1000 e 2000 gerações do algoritmo genético. O melhor resultado foi obtido na geração 682 (do processamento de 1000).

Para processar o algoritmo genético foi utilizado um computador com CPU AMD K6-II de 550 Mhz e 128 MB de RAM. O tempo de processamento obtido com 500, 1000 e 2000 gerações foi, respectivamente, 55 segundos, 1 minuto e 45 segundos, e 3 minutos e 19 segundos.

Nas figuras 1, 2 e 3 são apresentados os gráficos de evolução de  $fitness$ , ( $fitness$  do melhor indivíduo,  $fitness$  médio da população, e  $fitness$  do pior indivíduo) *on-line performance* e *off-line performance* para os processamentos com 500, 1.000 e 2.000 gerações.

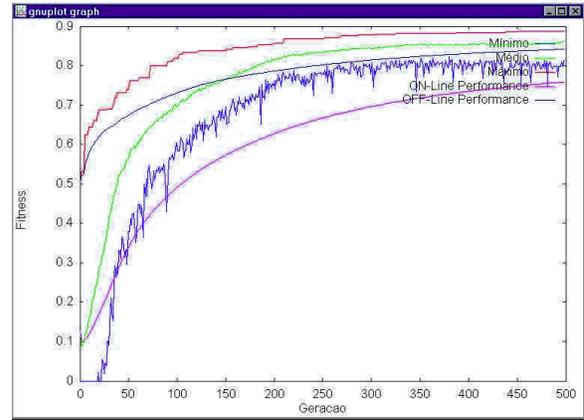


Figura 1 – Curva de evolução para a melhor rodada com 500 indivíduos.

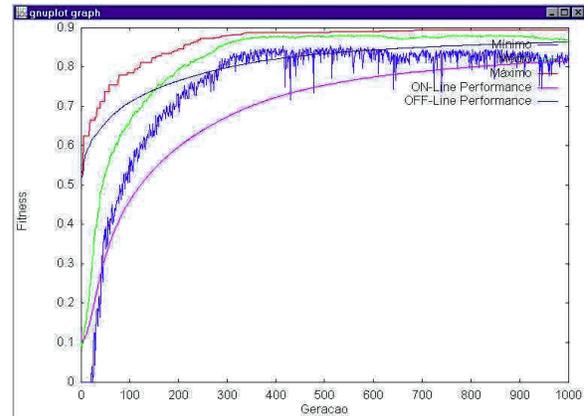


Figura 2 – Curva de evolução para a melhor rodada com 1000 indivíduos.

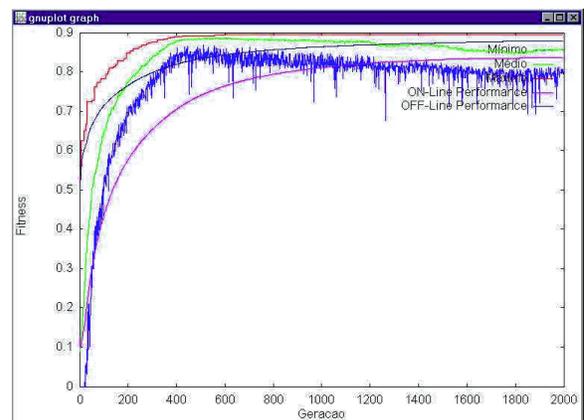


Figura 3 – Curva de evolução para a melhor rodada com 2000 indivíduos.

No processamento com 2000 gerações, o declínio da curva de  $fitness$  mínimos se deve a uma fatia da população que está aumentando gradativamente até atingir valores altos (20%).

## 5 Discussão e análise dos resultados

A figura 4 representa a mesma informação gerada pelo algoritmo Ford Fulkerson. As figuras 4b e 4c representam as soluções com respeito à disposição

do atendimento das cargas para o algoritmo genético com 500 e 1000 gerações. Como pode ser observado, o algoritmo genético conseguiu um ótimo resultado, por isso o algoritmo de Ford-Fulkerson chegou a uma solução ligeiramente melhor.

As diversas cores (azul, verde e amarelo) representam as áreas atendidas por uma certa subestação. A cor vermelha identifica uma área que contém cargas que não foram atendidas pelas subestações.

## 6 Conclusões e trabalhos futuros

Da forma como está o algoritmo de Ford & Fulkerson parece ser o mais indicado para trabalhar com locações de subestações, entretanto existem uma série de solicitações que este algoritmo ainda não atende, as principais são:

- O algoritmo trabalha com distâncias lineares entre o centro da quadrícula e a subestação. Na realidade as distâncias devem ser calculadas considerando o caminho dos cabos dos alimentadores já existentes ou em planejamento. Esta é uma diferença importante pois os alimentadores fazem muitas “voltas” para atender as cargas.
- O algoritmo considera que as quadrículas estão disponíveis para serem atendidas por qualquer subestação, quando na realidade já existem alimentadores atendendo aquelas quadrículas. É bastante oneroso modificar um alimentador a fim de atender a solução fornecida. O algoritmo poderia levar em consideração que algumas quadrículas já estão atendidas e somente poderia haver uma mudança se o benefício justificasse o custo das modificações ou se o usuário assim desejar.
- O algoritmo considera que todas as quadrículas podem ser atendidas por uma linha aérea. Na prática existem várias “áreas bloqueadas”, tais como parques e áreas de preservação, que devem ser contornadas, aumentando o custo de transporte da carga.

Na implementação destes itens deve ser considerado que alguns dos aspectos mencionados acima, como custo, criam um grande nível de dificuldade para o projeto. Com a abordagem por algoritmos genéticos tais limitações do algoritmo de Ford & Fulkerson poderão, com alguns esforços, serem consideradas na função objetivo. Tendo sido o resultado experimental da abordagem por algoritmos genéticos competitivo com o algoritmo de Ford & Fulkerson, conclui-se que há uma grande perspectiva positiva de que algoritmos genéticos possam substituir satisfatoriamente os procedimentos atuais.

## 7 Referências bibliográficas

- GOLDBERG, D.E., (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Reading: Addison-Wesley.
- FORD, L.R.J. & FULKERSON, D.R. Flows in Networks. Princeton, N.J.: Princeton University Press, 1962.
- GOODMAN E.D. (1996) Genetic Algorithm Optimized for Portability and Parallelism System. East Lansing, Michigan State University.
- LOPES, H.S. (1999) Algoritmos genéticos em projetos de engenharia: aplicações e perspectivas futuras. In: Anais do SBAt'99-IV. Simpósio Brasileiro de Automação Inteligente. São Paulo, 08-10/setembro/1999, p. 64-74

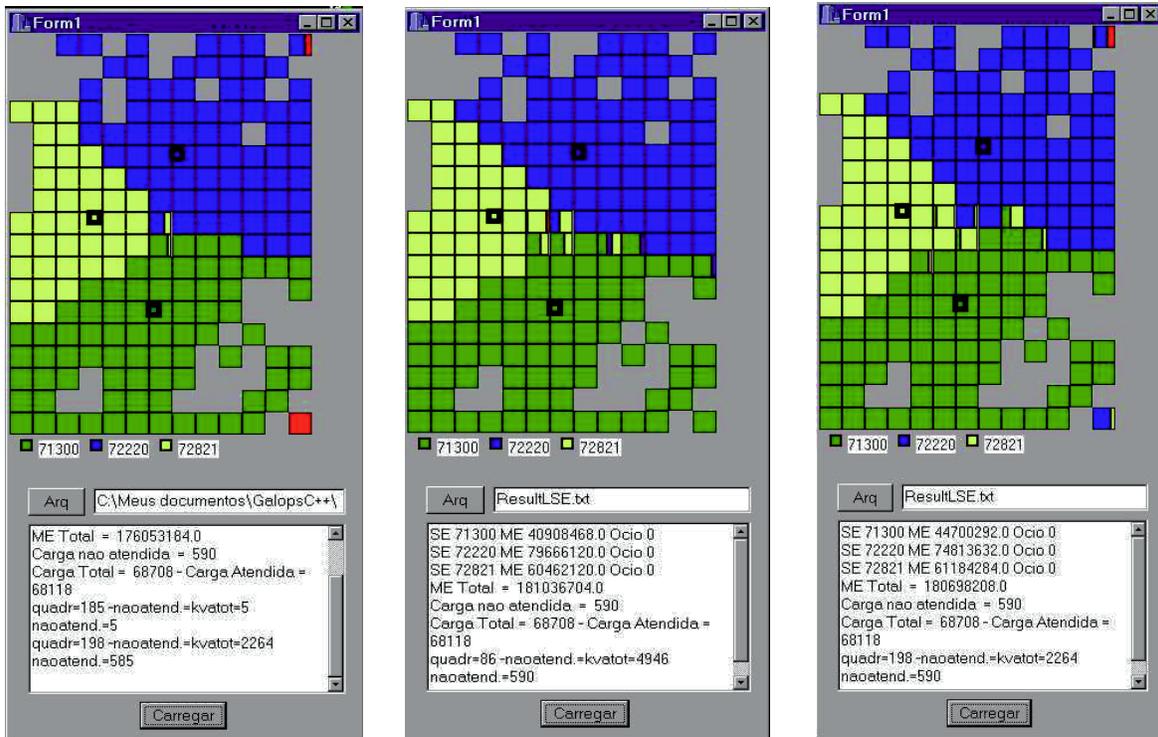


Figura4:(a) Resultado obtido para a área em estudo como algoritmo de Ford & Fulkerson;(b) Melhor resultado obtido como algoritmo genético com 500 gerações;(c) Melhor resultado obtido como algoritmo genético com 1000 gerações.