

UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”
FACULDADE DE ENGENHARIA - CAMPUS DE ILHA SOLTEIRA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Interface Gráfica para o Planejamento da Expansão da Transmissão de Energia Elétrica

Andréa Barboza Proto

Orientador: Prof. Dr. Sérgio Azevedo de Oliveira

Dissertação apresentada à Faculdade de Engenharia - UNESP - Campus de Ilha Solteira, para obtenção do título de Mestre em Engenharia Elétrica.

Área de Conhecimento: Automação.

Ilha Solteira - SP
novembro/2009

FICHA CATALOGRÁFICA

Elaborada pela Seção Técnica de Aquisição e Tratamento da Informação
Serviço Técnico de Biblioteca e Documentação da UNESP - Ilha Solteira.

- | | |
|-------|---|
| P967i | <p>Proto, Andréa Barboza.</p> <p>Interface gráfica para o planejamento da expansão da transmissão de energia elétrica / Andréa Barboza Proto. -- Ilha Solteira : [s.n.], 2009. 102 f. : il.</p> <p>Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Automação, 2009</p> <p>Orientador: Sérgio Azevedo de Oliveira
Bibliografia: p. 82-83</p> <p>1. Planejamento da expansão da transmissão. 2. Heurísticas construtivas. 3. Meta-heurísticas. 4. Software. 5. Tcl/Tk.</p> |
|-------|---|



UNIVERSIDADE ESTADUAL PAULISTA
CAMPUS DE ILHA SOLTEIRA
FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA

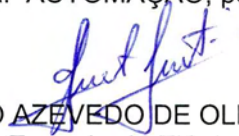
CERTIFICADO DE APROVAÇÃO

TÍTULO: INTERFACE GRÁFICA PARA O PLANEJAMENTO DA EXPANSÃO DA TRANSMISSÃO DE ENERGIA ELÉTRICA

AUTORA: ANDRÉA BARBOZA PROTO

ORIENTADOR: Prof. Dr. SERGIO AZEVEDO DE OLIVEIRA

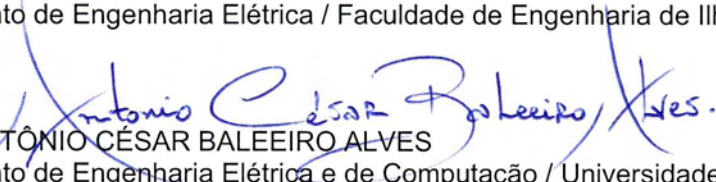
Aprovada como parte das exigências para obtenção do Título de MESTRE em ENGENHARIA ELÉTRICA, Área: AUTOMAÇÃO, pela Comissão Examinadora:


Prof. Dr. SERGIO AZEVEDO DE OLIVEIRA

Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira


Prof. Dr. RUBEN AUGUSTO ROMERO LAZARO

Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira


Prof. Dr. ANTÔNIO CÉSAR BALEEIRO ALVES

Departamento de Engenharia Elétrica e de Computação / Universidade Federal de Goiás

Data da realização: 20 de novembro de 2009.

DEDICATÓRIA

Aos meus entes queridos que se foram enquanto estive ausente de minha cidade natal, mas espero vê-los algum dia...

Dedico este trabalho ao meu querido e amado avô Dr. Lásaro Barbosa de Faria, ao meu tio querido Petrônio Leda Vêras e ao meu primo Aguimar Proto.

AGRADECIMENTOS

A Deus por me dar saúde e vitalidade durante todo período de dedicação ao mestrado.

A toda a minha família pelo seu apoio incondicional, compreensível, sobretudo nos momentos mais críticos.

Em especial ao Professor Dr. Sérgio Azevedo de Oliveira, por toda colaboração e dedicação no meu processo de aprendizagem durante o desenvolvimento deste trabalho.

Aos grandes amigos, Tércio Alberto dos Santos, Virgílio Fries Müller, Edilton Furquim Goulart Sobrinho, Rudi Alexis Solano, Wagner Pereira Lopes, Fernando Rodrigo Lopes Sanchez, Hélio Clementino dos Santos, Jaqueline Ferreira da Silva, Lucimar Garcino Lemes, Renato Cardoso dos Santos, Élcio Alteris dos Santos, Alfredo Bonini Neto, Juan Carlos Galvis Manso, Ápio Carniello, Carlos Alberto Febres Tapia.

A todos do DEE e do PPGEE que de uma forma ou de outra contribuíram para a conclusão deste trabalho.

E também quero agradecer a agência de fomento à pesquisa - Capes, pelo apoio financeiro com a manutenção da bolsa de auxílio.

“Ciclos se encerram,
amizades perpetuam...”

(Hamsuol)

RESUMO

Neste trabalho propõe-se o desenvolvimento de uma interface gráfica voltada para a resolução do problema de planejamento da expansão das linhas de transmissão, que utiliza-se de programas para a obtenção das soluções ótimas ou de boas soluções. A grande dificuldade encontrada por usuários ao interagir com estes programas, motivou o desenvolvimento de uma aplicação com interface gráfica, a qual disponibiliza ao usuário diversas metodologias para a resolução do problema do planejamento da expansão das linhas de transmissão de energia elétrica. Espera-se que software como este, agregado ao sistema de ensino tradicional que normalmente é utilizado num curso de engenharia elétrica, possa fortalecer o processo de aprendizagem do aluno. Assim, o software educacional Transmission Expansion Planning (TEP 1.0) está sendo desenvolvido visando propiciar um ambiente agradável para a realização de simulações e testes e favorecer a aprendizagem dos conceitos que envolvem o problema do planejamento da expansão da transmissão. É possível através deste software fazer simulações para os sistemas: Garver (6 barras/ 15 ramos), Sul brasileiro (46 barras/79 ramos) e Norte-Nordeste brasileiro (87 barras/179 ramos). O software se beneficia de recursos oferecidos por programas que são executados em *background*, bem como da utilização de meta-heurísticas e do ambiente de processamento de máquinas paralelas virtuais, as quais podem ser selecionadas para realização dos testes em determinado sistema.

Palavras chave: Planejamento da Expansão da Transmissão. Heurísticas Construtivas. Meta-heurísticas. Software Educacional. Tcl/Tk.

ABSTRACT

This work proposes the development of a computational tool aimed at solving the problem of the transmission expansion planning, which uses programs in the background to obtain optimal solutions or good solutions. The great difficulty for users to interact with these programs, motivated the development of an application with a graphical interface, which provides the user with various methodologies for solving the problem of expansion planning of transmission lines of electricity. It is expected that software like this, added to the traditional school system which is normally used in electrical engineering courses, can strengthen the process of student learning. Thus, the educational software Transmission Expansion Planning (TEP 1.0) is being developed to provide a pleasant environment for simulations and testing and promote the learning of concepts involving the issue of transmission expansion planning. It is possible using this software to do simulations for the systems: Garver (6 nodes / 15 branches), South Brazilian (46 nodes / 79 branches) and North-Northeast Brazilian (87 nodes / 179 branches). The software takes advantage of capabilities offered by programs that run on background, and using meta-heuristics and the processing environment, parallel virtual machine, which can be selected to test on a system.

Keywords: Transmission Expansion Planning. Constructive Heuristics. Meta-heuristics. Educational Software. Tcl/Tk.

LISTA DE FIGURAS

1	Recombinação e mutação (ROMERO; ESCOBAR; GALLEG0, 2006).	p. 30
2	Estratégia de intensificação (ROMERO; ESCOBAR; GALLEG0, 2006).	p. 39
3	Tela inicial da interface TEP.	p. 53
4	Sistemas testes.	p. 53
5	Sistema Garver com redespacho.	p. 54
6	Opção para navegação do programa <i>TEP</i> .	p. 54
7	Dados de entrada do sistema Garver.	p. 55
8	Escolha do método para a inicialização do problema.	p. 55
9	Escolha do método para a resolução do problema.	p. 56
10	Tela referente ao método Algoritmo Genético.	p. 56
11	Ajuste de parâmetros.	p. 57
12	Ajuste do custo do corte de carga.	p. 57
13	Resultado intermediário do Algoritmo Genético / Sistema Garver.	p. 58
14	Menu Graphic.	p. 59
15	Menu Tutorial.	p. 59
16	Menu Tutorial (detalhes).	p. 60
17	Menu Help.	p. 60
18	Menu About.	p. 61
19	Menu Back.	p. 61
20	Escolha das máquinas para processamento paralelo.	p. 62
21	Tela para montagem de um A-Team INIC.	p. 63
22	Submenu <i>Parallel methods</i> .	p. 63

23	Menu para a escolha de visualização de configuração ótima.	p. 64
24	Configuração ótima para o sistema Garver com redespacho.	p. 64
25	<i>Canvas</i> do sistema Sul brasileiro.	p. 69
26	<i>Canvas</i> do sistema Norte-Nordeste brasileiro.	p. 69
27	Recurso do <i>Scrollbar</i> para figuras grandes.	p. 70
28	Opções do menu <i>Parameter</i>	p. 71
29	Manipulação de arquivos.	p. 72
30	Manipulação de arquivos, diretório <i>ganor</i>	p. 72
31	Manipulação de arquivos, arquivo <i>saida.dat</i>	p. 73
32	Telas de saída de texto e gráfica.	p. 74
33	Escolha do sistema teste.	p. 75
34	Manipulação dos arquivos <i>mpop.dat</i> e <i>mbloco.dat</i>	p. 76
35	Linhas no diagrama unifilar.	p. 77
36	Arquivo <i>saída.dat</i>	p. 77
37	Solução ótima para o sistema Garver.	p. 78
38	Simulação do sistema Sul brasileiro.	p. 79
39	Simulação do sistema Norte-Nordeste brasileiro.	p. 79
40	Interpretador <i>wish</i>	p. 84
41	Diagrama Caso de Uso.	p. 85
42	Sistema Garver - Configuração Inicial.	p. 88
43	Sistema Garver - Configuração Ótima com Redespacho	p. 89
44	Sistema Garver - Configuração Ótima sem Redespacho	p. 89
45	Sistema Sul brasileiro - Configuração Inicial.	p. 94
46	Sistema Sul Brasileiro - Config. Ótima com Redespacho.	p. 95
47	Sistema Sul Brasileiro - Config. Ótima sem Redespacho.	p. 95

LISTA DE TABELAS

1	Dados de Barras - Garver	p. 87
2	Dados de Linhas - Garver	p. 87
3	Dados de Barras - Sul Brasileiro	p. 90
3	Dados de Barras - Sul Brasileiro	p. 91
4	Dados de Linhas - Sul Brasileiro	p. 91
4	Dados de Linhas - Sul Brasileiro (continuação)	p. 92
4	Dados de Linhas - Sul Brasileiro (continuação)	p. 93
5	Dados de Barras - Norte-Nordeste Brasileiro	p. 96
5	Dados de Barras - Norte-Nordeste Brasileiro (continuação)	p. 97
5	Dados de Barras - Norte-Nordeste Brasileiro (continuação)	p. 98
6	Dados de Linhas - Norte-Nordeste Brasileiro	p. 98
6	Dados de Linhas - Norte-Nordeste Brasileiro (continuação)	p. 99
6	Dados de Linhas - Norte-Nordeste Brasileiro (continuação)	p. 100
6	Dados de Linhas - Norte-Nordeste Brasileiro (continuação)	p. 101
6	Dados de Linhas - Norte-Nordeste Brasileiro (continuação)	p. 102

SUMÁRIO

1	INTRODUÇÃO	p. 14
2	O PROBLEMA DO PLANEJAMENTO ESTÁTICO DA EXPANSÃO DA TRANSMISSÃO	p. 16
2.1	INTRODUÇÃO	p. 16
2.2	A MODELAGEM MATEMÁTICA	p. 17
2.2.1	Modelo DC	p. 17
2.2.2	Modelo de Transportes	p. 19
2.2.3	Modelo Híbrido	p. 20
2.3	TÉCNICAS DE RESOLUÇÃO	p. 20
2.3.1	Introdução	p. 20
2.3.1.1	Métodos Exatos	p. 21
2.3.1.2	Métodos Aproximados	p. 22
2.3.2	Métodos Heurísticos Construtivos	p. 23
2.3.2.1	Método de Garver	p. 23
2.3.2.2	Método do Mínimo Esforço	p. 23
2.3.2.3	Método de Mínimo Corte de Carga	p. 24
2.3.3	Meta-heurísticas	p. 24
2.3.3.1	Algoritmo Genético	p. 25
2.3.3.2	AG Aplicado ao Problema do Planejamento	p. 28
2.3.3.3	“Simulated Annealing”	p. 32
2.3.3.4	SA Aplicado ao Problema do Planejamento	p. 33

2.3.3.5	Busca Tabu	p. 35
2.3.3.6	BT Aplicado ao Problema do Planejamento	p. 37
3	SOFTWARE EDUCACIONAL	p. 42
3.1	INTRODUÇÃO	p. 42
3.2	TÓPICOS FUNDAMENTAIS	p. 42
3.3	PRINCIPAIS CARACTERÍSTICAS DO SOFTWARE EDUCACIONAL	p. 44
4	A INTERFACE GRÁFICA PROPOSTA	p. 47
4.1	INTRODUÇÃO	p. 47
4.2	A LINGUAGEM TCL/TK	p. 48
4.3	A INTERFACE GRÁFICA PROPOSTA	p. 52
5	RESULTADOS E DISCUSSÕES	p. 68
5.1	INTRODUÇÃO	p. 68
5.2	EXEMPLO DE SIMULAÇÃO	p. 74
6	CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS	p. 80
	REFERÊNCIAS	p. 82
	Apêndice A – Instalação da interface gráfica TEP	p. 84
	Apêndice B – Diagrama Caso de Uso da TEP	p. 85
	Apêndice C – Questionário de avaliação	p. 86
	Anexo A – Dados dos Sistemas Testes (DE OLIVEIRA, 2004)	p. 87
A.1	Sistema Garver (6 Barras/15 Ramos)	p. 87
A.2	Sistema Sul Brasileiro (46 Barras/79 Ramos)	p. 90
A.3	Sistema Norte-Nordeste Brasileiro (87 Barras/179 Ramos)	p. 96

1 INTRODUÇÃO

Para resolver o problema de planejamento da expansão das linhas de transmissão é geralmente necessário a utilização de programas que ajudam na obtenção de boas soluções. E a grande dificuldade encontrada por usuários ao interagir com estes programas inspirou o desenvolvimento de uma aplicação com interface gráfica, a qual disponibiliza diversas metodologias para a resolução do problema do planejamento da expansão da transmissão.

É uma aplicação direcionada à alunos do ensino superior a nível de graduação e pós-graduação em engenharia elétrica, com ênfase em Sistema de Energia Elétrica. Espera-se que a sua utilização agregada ao sistema de ensino tradicional, possa fortalecer o processo de aprendizagem do acadêmico.

Portanto, o software educacional “Transmission Expansion Planning” (TEP 1.0) foi desenvolvido visando propiciar um ambiente *agradável* para a realização de simulações e testes e favorecer a aprendizagem dos conceitos que envolvem o problema do planejamento da expansão da transmissão.

É possível através deste software fazer simulações para os sistemas: Garver (6 barras/15 ramos), Sul brasileiro (46 barras/79 ramos) e Norte-Nordeste brasileiro (87 barras/179 ramos).

O software se beneficia de recursos oferecidos por programas que são executados em “background”, bem como da utilização de meta-heurísticas e do ambiente de processamento de máquinas paralelas virtuais, as quais podem ser selecionadas para realização dos testes em determinado sistema.

Neste trabalho, o capítulo 2 apresenta a modelagem do problema de planejamento estático da expansão das linhas de transmissão de energia, à longo prazo. Sendo este um problema cuja modelagem matemática corresponde a um problema de programação não linear inteira mista (PNLIM) que apresenta o fenômeno da explosão combinatória, exigindo alta capacidade de processamento das máquinas. No mesmo capítulo são abor-

dadas as técnicas de resolução utilizadas, a fim de obter resultados satisfatórios no referido problema, onde destacam-se as meta-heurísticas: “Simulated Annealing”, Algoritmos Genéticos e Busca Tabu. Estas, quando possuem boas configurações iniciais, diminuem o esforço computacional e exploram melhor o espaço de busca.

Na sequência, no capítulo 3, é apresentada a Tcl/Tk (“Tool command language, Tool kit”), linguagem de programação utilizada para o desenvolvimento desta ferramenta educacional, enfatizando-se os motivos que favorecem a sua escolha como linguagem de programação aplicada ao desenvolvimento deste projeto.

A interface gráfica é apresentada no capítulo 4, juntamente com a elaboração das simulações e dos testes realizados durante o desenvolvimento da mesma.

No capítulo 5, é feita uma análise da interface a fim de se conseguir uma ferramenta de fácil interação com o usuário através das informações adicionais: o componente ajuda e o componente tutorial. Sendo o primeiro responsável por explicar a funcionalidade de todos os recursos presentes na interface ao usuário, enquanto o tutorial apresenta os fundamentos básicos das metodologias utilizadas.

No último capítulo, são apresentadas as conclusões e sugestões para futuros trabalhos, finalizando-se com as referências bibliográficas utilizadas.

2 O PROBLEMA DO PLANEJAMENTO ESTÁTICO DA EXPANSÃO DA TRANSMISSÃO

2.1 INTRODUÇÃO

O crescimento da demanda de energia no sistema elétrico determina que se façam investimentos na construção de novas unidades geradoras, bem como na construção de novos circuitos de transmissão para o escoamento desta energia até os centros consumidores. Este processo é conhecido como planejamento da expansão do sistema elétrico. Este planejamento global normalmente é dividido no planejamento da geração (onde consideram-se fixos os custos de expansão da transmissão), no planejamento da transmissão (que considera predeterminado o plano de geração) e no planejamento da distribuição.

O planejamento da expansão de redes de transmissão de sistemas de energia elétrica determina *quando*, *onde* e *quantos* circuitos devem ser instalados na rede a fim de que o sistema opere adequadamente para uma demanda futura predeterminada e realizando o investimento de menor custo possível. O planejamento dinâmico (*quando*) em geral é decomposto em subproblemas estáticos que tratam das questões *onde* e *quantos* (planejamento em um estágio; de um ano inicial a um ano final, preestabelecidos).

O problema de planejamento estático da expansão da transmissão num horizonte de longo prazo (PPET) pode ser formulado como um problema de programação não linear inteiro misto (PNLIM) e, dadas as dimensões que o problema assume para casos práticos em geral, observa-se o fenômeno da explosão combinatória (pertencente ao conjunto de problemas NP-completo, de difícil tratamento). Sendo que, para uma alternativa de investimento (uma dada configuração), o problema se reduz a um problema de programação linear cujo objetivo é verificar a factibilidade desta alternativa. Estes conceitos

matemáticos serão abordados com peculiaridade na secção 2.3.

Este problema é solucionado com o auxílio da modelagem matemática e de técnicas de resolução adequadas.

2.2 A MODELAGEM MATEMÁTICA

A formulação matemática do problema do planejamento estático da expansão de sistemas de transmissão pode utilizar diferentes modelos matemáticos para sua resolução. O modelo DC é a modelagem mais utilizada no planejamento de sistemas de transmissão. Outros dois modelos destacam-se com o mesmo propósito: o Modelo de Transporte e o Modelo Híbrido.

2.2.1 Modelo DC

O modelo DC é uma generalização do modelo de fluxo de carga DC que está amplamente desenvolvido em (MONTICELLI, 1983). Ideal para ser usado em planejamento de sistemas de transmissão, pois leva em conta as duas leis de Kirchhoff para o sistema elétrico, ou seja, todas as barras do sistema e todos os laços existentes devem satisfazer tanto a primeira como a segunda lei, respectivamente.

A formulação do problema usando o modelo DC leva a um PNLIM o qual pertence ao conjunto de problemas chamados NP-completo de difícil tratamento, que apresenta o problema da explosão combinatória, pois geralmente existem muitos caminhos candidatos e além disso, em cada caminho podem ser alocadas várias linhas. Assim, o modelo está ligeiramente modificado comparado com o convencional, alterando-se a função objetivo a fim de facilitar a resolução do problema, no qual se leva em conta uma geração fictícia r_i multiplicada por um fator α_i de penalização, conforme apresentado a seguir:

$$\begin{aligned}
\text{Min } v &= \sum_{(i,j) \in \Omega} c_{i,j} n_{i,j} + \alpha \sum_i r_i \\
s.a. & \\
& Sf + g + r = d \\
& f_{i,j} - \gamma_{ij}(n_{i,j}^0 + n_{i,j})(\theta_i - \theta_j) = 0 \\
& |f_{i,j}| \leq (n_{i,j} + n_{i,j}^0) \bar{f}_{i,j} \\
& 0 \leq g \leq \bar{g} \\
& 0 \leq r \leq d \\
& 0 \leq n_{i,j} \leq \bar{n}_{i,j} \\
& n_{i,j} \text{ inteiro,} \\
& f_{i,j} \text{ irrestrito } \forall (i,j) \in \Omega
\end{aligned} \tag{2.1}$$

Descrição das variáveis:

- v Investimento devido a adição de circuitos;
- c_{ij} Custo de um circuito no ramo (i, j) ;
- n_{ij} Número de circuitos adicionados no ramo (i, j) ;
- α Parâmetro adequado de transformação de unidades;
- r_i Geração fictícia na barra (i) ;
- S Matriz de incidência nó-ramo transposta;
- f vetor de fluxo de potência ativa nos ramos;
- g Vetor de gerações;
- r Vetor de geradores fictícios ou artificiais;
- d Vetor de demandas;
- $f_{i,j}$ Fluxo de potência ativa total pelos circuitos nos caminhos (i, j) ;
- γ_{ij} Susceptância de uma linha no ramo (i, j) ;
- n_{ij}^0 Número de circuitos iniciais no ramo $(i, j) \in \Omega$;
- θ Vetor de ângulos das tensões nodais;
- $\bar{f}_{ij}$ Fluxo máximo de potência ativa por circuito no ramo (i, j) ;
- $\bar{g}$ Vetor de limites de geração;
- $\bar{n}_{ij}$ Número máximo de circuitos adicionados no ramo (i, j) ;
- Ω Conjunto de todos os ramos definidos pelos circuitos existentes e as alternativas de expansão.

2.2.2 Modelo de Transportes

Este modelo foi formulado por Garver (GARVER, 1970) e representa uma versão relaxada do modelo DC. No modelo de transportes utiliza-se somente a primeira lei de Kirchhoff. Isto favorece a sua aplicação em programas computacionais desenvolvidos para o planejamento da expansão da energia elétrica, os quais não requerem grande esforço computacional para convergir para boas soluções, pois são fáceis de resolver, quando comparado com outros modelos existentes. No entanto, sua solução (se ótima) pode ser inadequada comparada com o modelo DC, pois a solução do modelo de transporte poderá apresentar corte de carga inadequado.

No modelo de transportes deseja-se encontrar uma configuração que produz o menor investimento no plano de expansão do sistema elétrico e condições adequadas de operação desse sistema. A modelagem matemática do problema, considerando geração fictícia para todas as barras de carga, assume a seguinte forma:

$$\begin{aligned}
 \text{Min } v &= \sum_{(i,j) \in \Omega} c_{i,j} n_{i,j} + \alpha \sum_i r_i \\
 \text{s.a.} & \\
 & Sf + g + r = d \\
 & |f_{i,j}| \leq (n_{i,j}^0 + n_{i,j}) \bar{f}_{i,j} \\
 & 0 \leq g \leq \bar{g} \\
 & 0 \leq r \leq d \\
 & 0 \leq n_{i,j} \leq \bar{n}_{i,j} \\
 & n_{ij} \text{ inteiro} \\
 & f_{i,j} \text{ irrestrito } \forall (i,j) \in \Omega
 \end{aligned} \tag{2.2}$$

Atualmente este modelo vem sendo utilizado como ferramenta de apoio no processo de busca da solução, podendo estar presente em algoritmos inicializadores, os quais geram as configurações iniciais para o uso de meta-heurísticas na resolução do problema do planejamento da expansão da transmissão.

2.2.3 Modelo Híbrido

Trata-se de uma combinação do modelo de transportes e o modelo DC, introduzido por Villasana e Garver em (VILLASANA; GARVER; SALON, 1985). Este apresenta o mesmo grau de complexidade para obter a solução ótima, comparada com o modelo DC, no entanto é comumente utilizado somente como ferramenta de apoio no processo de resolução do modelo DC.

$$\begin{aligned}
 \text{Min } v &= \sum_{(i,j) \in \Omega} c_{i,j} n_{i,j} + \alpha \sum_i r_i \\
 \text{s.a.} & \\
 & Sf + S^0 f^0 + g + r = d \\
 & f^0_{i,j} - \gamma_{i,j}(n^0_{i,j} + n_{i,j})(\theta_i - \theta_j) = 0, \forall (i,j) \in \Omega_0 \\
 & |f^0_{i,j}| \leq n^0_{i,j} \bar{f}_{i,j}, \forall (i,j) \in \Omega_0 \\
 & |f^0_{i,j}| \leq n_{i,j} \bar{f}_{i,j}, \forall (i,j) \in \Omega \\
 & 0 \leq g \leq \bar{g} \\
 & 0 \leq n_{i,j} \leq \bar{n}_{i,j} \\
 & f^0_{i,j} \text{ irrestrito } \forall (i,j) \in \Omega_0 \\
 & f_{i,j} \text{ e } \theta_j \forall (i,j) \in \Omega
 \end{aligned} \tag{2.3}$$

Em que:

- S^0 Matriz de incidência nó-ramo dos circuitos existentes na topologia base;
- f^0_{ij} Fluxo de potência ativa nos circuitos existentes na topologia base;
- n^0_{ij} Número de circuitos existentes no ramo (i, j) na topologia base;
- Ω_0 Conjunto dos circuitos presentes na topologia base.

2.3 TÉCNICAS DE RESOLUÇÃO

2.3.1 Introdução

Atualmente, devido ao avanço tecnológico dos computadores, pode-se resolver problemas da vida real através do ambiente computacional, no entanto existem problemas que dificultam este processo de informatização.

Para o problema de otimização matemática que apresenta todas relações lineares, como o problema de programação linear (PL), existe o método simplex capaz de solucioná-lo; porém este método não é tão eficiente se aplicado para solucionar problemas de engenharia, os quais possuem um elevado número de variáveis e restrições. Contudo, em vários problemas de otimização que não podem ser resolvidos pelos métodos exatos, quando aplicam-se a eles os métodos aproximados, os mesmos são capazes de determinar boas soluções com esforço computacional relativamente aceitável (ROMERO; ESCOBAR; GALLEGU, 2006).

Os problemas combinatórios são exemplo de problemas que têm grande complexidade computacional para solucioná-los, pois nos mesmos ocorrem o fenômeno da explosão combinatória, ou seja, conforme aumenta o número de variáveis do problema cresce o seu esforço computacional de forma exponencial. Eles, são divididos em dois grupos: os problemas tipo P e os problemas NP. Para o primeiro existe algoritmos com esforço computacional polinomial para resolvê-los, enquanto que para os NP não se conhecem algoritmos capazes de resolvê-los com complexidade computacional do tipo polinomial.

Existe ainda problemas NP altamente complexos, os quais são denominados NP-completos.

Mediante esta divisão ao buscar soluções para determinado problema, deve-se realizar uma análise, identificando qual tipo de problema a ser trabalhado, e posteriormente aplicar o algoritmo adequado conforme abaixo:

1. - Identificar se este problema é P ou NP;
2. - Se este problema for NP, desenvolver algoritmos aproximados.

2.3.1.1 Métodos Exatos

Estes geralmente utilizam técnicas de decomposição matemática. São eficientes quando aplicados em problemas de planejamento de pequeno e médio porte, pois em sistemas de grande porte geram problemas de convergência e requer elevado esforço computacional. Este fato torna o método restrito a pequenas instâncias. Com o intuito de contornar esta desvantagem, pode-se aplicar a técnica de decomposição de Benders usada por vários algoritmos, explorando a decomposição natural do problema em duas partes: um subproblema de investimento e um subproblema de operação, sendo esta técnica um exemplo de método de otimização clássica.

Surgiram na década de 80, com o propósito de tentar solucionar o problema de planejamento da expansão da transmissão de maneira ótima tendo como ferramentas as técnicas de decomposição matemática. Originando-se assim a metodologia de Benders, acima mencionada, que decompõe o problema de planejamento da expansão das linhas de transmissão em duas partes. Em uma das partes escolhe-se o plano de expansão candidato e, calculam-se os custos de investimento associado ao mesmo, enquanto que na outra parte refere-se a um subproblema de operação testando o plano de expansão candidato em termos do adequado atendimento da carga.

A busca por um ótimo global é feita através de uma busca iterativa das resoluções separadas dos subproblemas abordados.

2.3.1.2 Métodos Aproximados

Os métodos aproximados podem ser classificados em métodos heurísticos e meta-heurísticas.

Os métodos heurísticos são algoritmos aproximados desenvolvidos para resolver problemas NP, os quais possuem a capacidade de encontrar boas soluções para este tipo de problema. São fáceis de implementar e não exigem grande esforço computacional, contudo devem ser implementados quando não existe a necessidade de obter o ótimo do problema, do ponto de vista prático, ou quando existir limitações de tempo e espaço. Estes métodos heurísticos podem ser implementadas visando conseguir configurações iniciais factíveis para o problema (ROMERO, 1999), para serem utilizadas posteriormente na inicialização das meta-heurísticas.

As meta-heurísticas são um conjunto de técnicas de otimização capazes de solucionar problemas complexos para os quais o esforço computacional cresce de forma exponencial, principalmente aqueles para os quais as heurísticas construtivas não são eficientes.

Neste trabalho, pretende-se trabalhar com os métodos heurísticos construtivos e com as meta-heurísticas, resumidos a seguir.

2.3.2 Métodos Heurísticos Construtivos

O algoritmo heurístico construtivo, a partir de uma configuração base, permite adicionar um ou vários circuitos a cada passo realizado, de modo que o conjunto de adições realizadas permita que o sistema elétrico opere de forma adequada.

O circuito escolhido a cada passo é denominado circuito candidato, o qual é identificado por um critério de sensibilidade. O índice que determina a escolha do circuito é feito pelo critério de sensibilidade (ou índice de desempenho).

As diferenças entre os vários algoritmos construtivos residem na forma de encontrar o indicador de sensibilidade e, obviamente, no modelo escolhido (DC, transportes ou híbrido) (DE OLIVEIRA, 2004).

2.3.2.1 Método de Garver

O modelo desenvolvido por Garver (GARVER, 1970), consiste na resolução do modelo de transportes, relaxando a integralidade das variáveis de investimento e a segunda lei de Kirchhoff, ou seja ele permite que variáveis inteiras do problema possam ser tratadas como variáveis reais durante sua resolução.

No PPET este modelo permite que o número de linhas entre barras pode ter uma parte inteira e parte fracionária, assim na solução do PPL pode-se obter o circuito mais atrativo que deve ser adicionado no sistema.

A vantagem deste modelo está na simplicidade de implementação do seu algoritmo, em contrapartida, não se garante a solução ótima do sistema.

2.3.2.2 Método do Mínimo Esforço

O *método do mínimo esforço* (MONTICELLI et al., 1982) baseia-se no fato de que a distribuição dos fluxos em uma rede segue uma *lei de mínimo esforço* que minimiza o produto das susceptâncias de cada ramo pelo quadrado do respectivo fluxo. Esta função de mínimo esforço, é utilizada como um índice de desempenho para ordenar as adições mais atrativas, da seguinte maneira:

$$IS_{me} = \Delta Z_{ij} = -\frac{1}{2}(\theta_i - \theta_j)^2 \Delta \gamma_{ij} \quad (2.4)$$

em que:

$(\theta_i - \theta_j)$ é a diferença angular do ramo ij antes da adição;

$\Delta \gamma_{ij}$ é a variação da susceptância de um circuito no ramo ij .

2.3.2.3 Método de Mínimo Corte de Carga

De maneira semelhante ao método de mínimo esforço, este método realiza a adição de linhas selecionadas de acordo com um índice de sensibilidade que permite encontrar linhas mais atrativas (PEREIRA et al., 1985).

Este algoritmo é similar ao método de mínimo esforço, com as seguintes diferenças:

- Calcula-se o corte de carga, usando programação linear e o índice de mínimo corte de carga de cada linha do sistema;
- Realiza a classificação das linhas, em ordem decrescente, de acordo com o índice de mínimo corte de carga;
- Se o corte de carga é menor que um limite preestabelecido, finaliza-se o processo. Caso contrário, seleciona-se a primeira, segunda ou terceira linha conforme uma lista ordenada, adiciona-se a linha ao sistema e repete-se o processo.

2.3.3 Meta-heurísticas

Meta-heurísticas são amplamente aplicadas e com êxito em problemas de engenharia e no campo da pesquisa operacional.

As meta-heurísticas utilizam os mecanismos de melhoria da vizinhança e estratégias inteligentes, com o propósito de garantir que o processo saia de um ótimo local e encontre a solução ótima global durante sua execução, percorrendo um espaço de busca pequeno de um conjunto grande.

A estratégia utilizada na procura de uma configuração ótima é o que difere as diversas meta-heurísticas existentes, assim o projetista após definir a meta-heurística que deverá ser utilizada na resolução de determinado problema, deverá adaptá-la de acordo com o problema a ser resolvido. Para, implementar as meta-heurísticas deve-se seguir os seguintes passos: (GALLEGO, 1997)

- Especificar uma forma de representar uma proposta de solução dentro do espaço de busca do problema (codificação);
- Definir a forma de encontrar a função objetivo ou seu equivalente para cada proposta de solução;
- Descrever com detalhes a vizinhança da solução corrente;
- Determinar de forma precisa e explícita se a forma de realizar as transições deve ser realizada a partir de um único ponto ou de um conjunto de pontos;
- Definir se o processo de busca deve ser realizado através de soluções factíveis ou podem ser consideradas também soluções infactíveis.

Os algoritmos “Simulated Annealing”, Busca Tabu e Genéticos, entre outros, são meta-heurísticas de alta qualidade e amplamente aplicados na resolução de problemas do planejamento estático da expansão das linhas de transmissão.

Utilizando conceitos da Mecânica Estatística, o algoritmo “Simulated Annealing” inspira-se no processo físico de obtenção de estruturas cristalinas através do superaquecimento e resfriamento lento da substância. O Algoritmo Genético é uma técnica de busca baseada na mecânica de seleção natural das espécies. Busca Tabu apoia-se em regras

geradas na inteligência artificial, o qual possui um esquema de busca local para explorar o espaço de soluções além do ótimo local (ROMERO; ESCOBAR; GALLEG0, 2006).

2.3.3.1 Algoritmo Genético

Este é um método que utiliza os princípios básicos da genética natural no processo de otimização matemática (ROMERO; ESCOBAR; GALLEG0, 2006), agregando conceitos de técnicas evolutivas, as quais inicializam seu processo de evolução através da população (conjunto de soluções possíveis). Esta população evoluirá mediante diferentes mecanismos, que caracterizam uma nova população. De fato, esta população deverá apresentar melhores soluções comparada com sua antecessora. Assim este processo é repetido até que se cumpra o critério de parada estabelecido.

As técnicas evolutivas necessitam de uma população inicial para poderem evoluir, embora existam técnicas que permitem a elaboração desta população inicial, a qual é comumente gerada de forma aleatória, principalmente em sistemas de médio e/ou pequeno porte. A população inicial gerada de forma aleatória no problema do planejamento da expansão da transmissão é obtida controlando o número de ligações que possuem valor diferente de zero e este deve estar entre zero e o valor máximo permitido por ligações. Contudo, em sistemas de grande porte, a população inicial é obtida utilizando os métodos aproximados (ROMERO, 1999). Posteriormente, cada indivíduo da população é analisado e classificado com um valor correspondente com sua função objetivo. Este indivíduo é representado por um único cromossomo; esses são normalmente implementados em forma de vetores ou lista de atributos, sendo que cada atributo é denominado de gene e seus valores de alelos.

Para que a população evolua, é permitido que o indivíduo de melhor qualidade possa gerar descendentes, sendo o número de descendentes permitido por indivíduo estatisticamente proporcional a qualidade de seu correspondente “fitness”, a isto aplica-se o processo de seleção, no qual os melhores sobrevivem e geram descendentes.

Após realizada a seleção sobre a população de descendentes serão aplicados os processos de recombinação e mutação, os quais definem uma nova população.

Um algoritmo genético estará definido após ser realizada as seguintes etapas: gerada a população inicial, definido o método de seleção, os parâmetros de controle (tamanho da população, taxa de recombinação, taxa de mutação) e o critério de parada.

O referido algoritmo deve apresentar os seguintes dados: a população inicial, o mecanismo de codificação, os parâmetros de controle, a função “fitness”, o esquema de seleção e os operadores genéticos do ciclo geracional (GALLEGO, 1997).

O mecanismo de codificação é realizado através da escolha do tipo de codificação que melhor se enquadra ao problema e que permitirá obter boas soluções. Existem diferentes mecanismos de codificação. Segundo a literatura especializada a codificação binária conduz a um desempenho insatisfatório e a escolha do mecanismo de codificação é de fundamental importância.

No PPET, a codificação poderá ser implementada pela representação binária ou representação decimal. Em sistemas de pequeno ou médio porte pode-se representar as variáveis de decisão pela codificação binária, no entanto, aplicar esta codificação em sistemas grandes, como por exemplo o norte-nordeste brasileiro, tornaria uma opção inviável. Isto porque a codificação binária apresenta uma característica desfavorável, o penhasco de Hamming, o qual exige esforço computacional para representar números consecutivos, como por exemplo os números três e quatro que são representados em binário por 011 e 100, enquanto que a codificação decimal não possui este problema.

A eficiência de um algoritmo genético é estabelecida através dos ajustes dos parâmetros de controle, pela forma de obter a população inicial e o estabelecimento do critério de parada (ROMERO; ESCOBAR; GALLEGO, 2006). Os parâmetros de controle são: o tamanho da população, a taxa de mutação e a taxa de recombinação.

Definir o tamanho da população exige analisar o esforço computacional requerido e a velocidade de execução do algoritmo; logo, definir este parâmetro com valor elevado implicará na redução da velocidade de resposta do algoritmo, em contrapartida aplicar valores pequenos não favorece o paralelismo implícito característico dos algoritmos genéticos. Segundo literaturas especializadas, o valor deste parâmetro se encontra entre 30 e 200. A taxa de recombinação e a taxa de mutação são parâmetros que dependem da sensibilidade do programador para serem definidos.

A taxa de recombinação varia entre 0,5 e 1, e a taxa de mutação entre 0,001 e 0,050. Com o propósito de selecionar os melhores indivíduos o algoritmo genético pode adotar um dos mecanismos de seleção existentes, como exemplo o método de seleção proporcional. Neste método, cada configuração tem o direito de gerar um número de descendentes proporcional a qualidade de sua função objetivo (GALLEGO, 1997), ou seja as configurações que apresentam maior valor da função objetivo participam da geração de novas configurações. Deste modo, para implementar o mecanismo de seleção proporcional a roleta

pode ser agregada ao método. Neste, cada configuração é disposta em um círculo da roleta, proporcional ao valor correspondente de sua função aptidão, na qual um processo aleatório determina o número de descendentes que cada configuração poderá ter.

Como mencionado, terminado o processo de seleção e determinando o número de descendentes de cada configuração da população atual, deve-se implementar a recombinação. Sendo que uma vez determinada a seleção, escolhe-se um par de indivíduos que irão gerar outros descendentes. Esta escolha pode ser realizada de forma aleatória ou escolhendo os indivíduos que podem gerar maior número de descendentes.

O processo de recombinação pode ser realizado de várias formas diferentes, de acordo com o ponto de recombinação escolhido. Podendo ser um ponto, dois pontos ou múltiplos pontos.

Na recombinação de um ponto escolhe-se aleatoriamente o ponto de recombinação, assim este ponto delimita o cromossomo em duas partes. O mesmo processo é feito no segundo cromossomo e posteriormente é feita a troca das partes entre os cromossomos, originando um novo cromossomo que fará parte da nova população.

Na recombinação de dois pontos, são escolhidos aleatoriamente dois pontos do cromossomo, delimitando uma faixa, a qual será trocada com a faixa do outro cromossomo que participa da recombinação.

Por último, a recombinação de múltiplos pontos é unitária, nesta cada elemento (bit) é selecionado de forma aleatória nas duas configurações, em seguida a troca é realizada.

A mutação é o último processo a ser realizado nos algoritmos genéticos, controlado pelo parâmetro denominado taxa de mutação como foi abordado anteriormente. Possui o objetivo de gerar a diversidade da população e varia entre 0 e 1 (GALLEGO, 1997).

É de fundamental importância ressaltar que o problema do planejamento é formulado como um problema de minimização, mas na formulação tradicional os algoritmos genéticos trabalham com problemas de maximização. No PPET, a transformação do problema de minimização em um problema de maximização foi realizada da seguinte maneira:

$$\min v \Leftrightarrow \max [K - v] \quad (2.5)$$

sendo K uma constante que pode ser fixa ou variar durante o processo de otimização, seu valor é baseado nos valores das funções objetivo das configurações iniciais, ou no caso de

variar durante o processo, pode assumir o valor da função objetivo de maior valor.

2.3.3.2 AG Aplicado ao Problema do Planejamento

Os Algoritmos genéticos apresentam grande capacidade para resolver problemas complexos e de grande porte, tendo alta probabilidade de encontrar a solução ótima. Como o problema do planejamento e expansão das redes de transmissão é um caso de difícil tratamento, onde as linhas assumirem valores inteiros, se torna um problema combinatorial que apresenta um grande número de soluções ótimas possíveis, na resolução desse problema o AG tem sido utilizado com bastante sucesso.

Para elaborar o AG são necessários a realização prévia de alguns tópicos, os quais são: determinar a representação e a codificação ideal para o problema de planejamento, escolher entre os tipos de mecanismos geracionais existentes qual será utilizado no algoritmo, determinar como será gerada a população inicial e definir os parâmetros de controle e o critério de parada. Esses tópicos são de fundamental importância para garantir a qualidade do AG.

Na representação determina-se o modo de calcular a função objetivo, que é gerada em problemas do planejamento com a utilização da modelagem DC.

A modelagem DC apresenta uma parcela correspondente aos custos das linhas adicionadas e a outra parcela referente ao corte de carga que gera determinada configuração. Assim, no AG a configuração será gerada somente com as variáveis de investimento em paralelo com o corte de carga do sistema e outras variáveis contínuas de operação. Estas variáveis de investimentos são inteiras, e determinam o número de linhas a serem inseridas em cada ramo candidato.

O valor obtido da função objetivo é o fator que permite avaliar o indivíduo, classificando se o elemento possui qualidade ou não. Seguindo a lógica do AG elementos de melhor qualidade participam com mais frequência dos mecanismos que geram uma nova configuração.

As variáveis que definem o número de linhas a serem adicionadas no sistema são inteiras, logo a codificação que se enquadra a representação do problema é a decimal. Outros tipos de codificações poderiam ser utilizadas, mas a decimal garante que o AG não tenha alto esforço computacional e não apresente o “penhasco de Hamming”.

No mecanismo geracional é realizada a seleção, a recombinação e a mutação; na seleção, a partir de uma população corrente obtém-se uma nova população. A Figura 1, apresenta as operações de recombinação e mutação para duas configurações de uma rede simples usando a codificação decimal, onde o tamanho do vetor é correspondente ao número de barras do sistema, e cada elemento do vetor determina o número de linhas adicionadas entre as barras existentes.

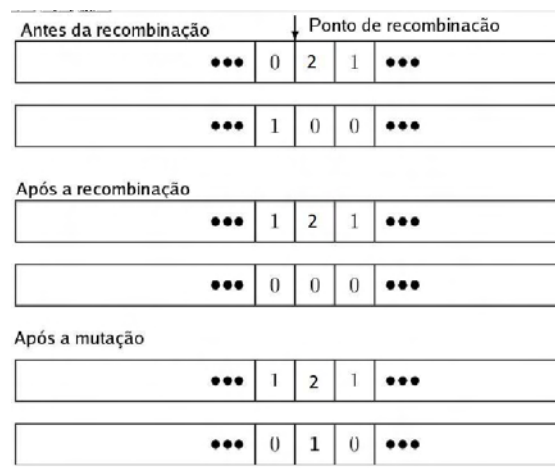


Figura 1: Recombinação e mutação (ROMERO; ESCOBAR; GALLEG0, 2006).

Primeiramente é selecionando o ponto de cruzamento, posteriormente é realizada a troca entre as configurações, e em seguida em uma determinada configuração é feita a mutação.

A população inicial pode ser gerada de duas maneiras: de forma aleatória, normalmente utilizada em sistemas de pequeno/médio porte, ou através dos métodos heurísticos contrutivos abordados no capítulo 3, para sistemas de grande porte. O desempenho do algoritmo genético está relacionado com os parâmetros de controle, taxa de recombinação, a taxa de mutação e o tamanho da população. A qualidade da solução depende da sensibilidade do programador.

É possível realizar diversas modificações no algoritmo genético, permitindo seu desenvolvimento com características específicas do problema de planejamento de sistemas de transmissão. Estas modificações são:

1. Implementação de uma taxa de mutação variável;
2. Taxa de mutação controlada por “Simulated Annealing” (SA);

3. “Building blocks” e rede inicial não convexa;
4. Parâmetro de penalidade α variável.

O algoritmo genético básico trabalha com a taxa de mutação como um fator secundário e de valor baixo, entretanto a taxa de mutação com valor elevado e variável encontra regiões de busca mais atrativas. E a utilização de mutação múltipla vem sendo aplicada com sucesso em aplicações de engenharia.

A taxa de mutação controlada por “Simulated Annealing” é um enriquecimento ao AG básico, sendo que na taxa de mutação controlada $\Delta v = e^{-\frac{\Delta E}{T}}$, onde Δv é a variação da função objetivo e T é o parâmetro de temperatura do SA. Em outras palavras, primeiramente determina-se Δv , se ele for menor que zero, então aceita a mutação, caso contrário, ela será aceita de forma probabilística. Em pesquisas realizadas este tipo de mutação quando presente no algoritmo genético básico conduz a melhores soluções quando aplicados no problema de planejamento de sistemas de transmissão (GALLEGO, 1997).

O mecanismo de blocos construtivos (“building blocks”) permite ao AG encontrar soluções ótimas ou quase-ótimas dentro das configurações de determinada população. Para compreender a definição de blocos construtivos é necessário o entendimento da sistemática denominada “schema”. Essa fixa as posições de todas possíveis configurações que apresentam os mesmos bits em determinadas posições, ou seja, o “schema” * *000 representa um subconjunto de todos os números binários de cinco dígitos que possuem três posições fixas, assim ele representará quatro configurações: 00000, 01000, 10000, 11000. Em seguida é calculado o valor médio da função objetivo de cada “schema”. Outro critério também é utilizado em blocos construtivos, que é a distância de definição do “schema” (“defining length”), sendo este a distância entre duas posições fixas mais externas do “schema”. Assim, um “building blocks” é o “schema” que apresenta valor objetivo médio de boa qualidade pequena distância de definição, que garantirá que o “schema” não seja destruído pelos operadores de recombinação e mutação. Este mecanismo é a essência fundamental do AG.

Tratando-se do parâmetro α de penalidade variável, este penaliza a função objetivo com o corte de carga; assim se este parâmetro assumir valor elevado, fará com que o AG trabalhe com configurações sem corte de carga, restringindo o uso de configurações com corte de carga que poderiam conduzir a regiões atrativas. Logo, para que se faça a melhoria na exploração do espaço de busca, é aconselhável iniciar com uma α menor e ir incrementando seu valor após cada geração.

2.3.3.3 “Simulated Annealing”

Técnica de otimização combinatória que buscou na física inspiração para obter boas soluções em diversos problemas de difícil solução. Baseia-se no processo de transformação de um sólido em cristais, o qual é conduzido a elevada temperatura e posteriormente resfriado lentamente, controlando a energia liberada até a obtenção de estrutura cristalina perfeita.

Surgiu no início da década de 80, baseado no algoritmo de *Metropolis*, onde observa-se a energia armazenada em cada molécula que estão distribuídas em vários níveis de energia dentro do material. Apresenta excelentes resultados quando aplicada ao planejamento da expansão de sistemas de transmissão com os métodos heurísticos construtivos (ROMERO; ESCOBAR; GALLEG0, 2006).

A utilização do SA na resolução do problema da transmissão é baseada nos seguintes procedimentos:

- Representação do problema;
- Mecanismo de transição; e
- Programa de resfriamento.

Pode-se também resumir os algoritmos de SA nos passos que se seguem.

PASSO 1 - Defina a função de minimização (maximização) do problema a ser resolvido;

PASSO 2 - Gere uma solução inicial aleatória;

PASSO 3 - Enquanto a temperatura não é zero, faça:

PASSO 4 - Enquanto não for achado o “equilíbrio térmico”, faça:

PASSO 5 - Aplique uma perturbação aleatória no estado atual, criando um novo estado;
avale ΔE (variação da função objetivo);

PASSO 6 - Se $(\Delta E < 0)$ atualize o estado atual com o novo estado. Caso contrário, atualize o estado atual com o novo estado se $e^{-\frac{\Delta E}{T}} > \text{random}[0,1]$;

PASSO 7 - Diminua a temperatura;

PASSO 8 - Retorne o estado de mínima energia (solução com o valor mínimo da função de minimização).

O mecanismo de transição é o processo que consiste em realizar a transição da configuração corrente para a configuração subsequente dentro de uma estrutura de vizinhança. Esta transição sempre ocorrerá para uma melhora no valor da função objetivo da incumbente. Para uma configuração de qualidade inferior, a transição só será realizada se for aceito um critério probabilístico que varia ao longo da execução do algoritmo.

O programa de resfriamento é a estratégia geral de controle do processo SA. Para o problema do planejamento os parâmetros foram definidos como segue: *Temperatura inicial* - A temperatura inicial deve ser calculada de tal maneira que só será aceito um número determinado de configurações propostas; destas, as que tiverem pior valor da função objetivo não poderão ultrapassar uma porcentagem de piora em relação a função objetivo que a originou. Isto, de tal forma que a configuração não se afaste para regiões pouco atrativas, o qual implicaria mais tempo de computação, ao tentar retornar às regiões mais atrativas. *Taxa de resfriamento* - Na literatura é recomendado taxas de resfriamento no intervalo $[0,8 ; 0,99]$, que correspondem a um resfriamento lento. A fim de que se permita uma exploração mais intensa nas temperaturas baixas, poderá considerar-se um N_k crescente quando decrescer T_k . *Temperatura final* - Os critérios de parada estão baseados no argumento de que a execução do processo deve ser interrompida se a melhoria esperada na função objetivo, no caso de se continuar executando o algoritmo, é pequena.

2.3.3.4 SA Aplicado ao Problema do Planejamento

Como abordado, três critérios devem ser previamente definidos para a elaboração do SA, sendo estes independentes e essenciais para o funcionamento do algoritmo. Por existir diferentes tipos de critérios, surgem distintos algoritmos SA. O primeiro critério a ser analisado é a adequada representação do problema, sendo que para o problema do planejamento da expansão de sistemas de transmissão é utilizado o modelo DC, apresentado anteriormente. Os outros dois critérios são: o mecanismo de transição e o processo de resfriamento.

No modelo DC, a factibilidade é mantida pelas restrições e a função custo é adquirida pela somatória de dois termos, o primeiro é referente ao custo das linhas adicionadas e o segundo termo identifica a penalidade devido ao corte de carga. Esta modelagem ligeira-

mente modificada comparada com a original, apresenta maior flexibilidade para mudar de uma configuração corrente para uma nova configuração, pois qualquer configuração proposta de adição de linhas é factível, assim considera-se as configurações que apresentam corte de carga e as que não apresentam corte de carga. No entanto, as configurações com corte de carga são fortemente penalizadas, devido ao fato de que o custo da função objetivo apresentará um valor elevado, enquanto configurações com pequeno corte de carga apresentam mais chances de serem aceitas.

O mecanismo de transição permite identificar a configuração candidata através da configuração corrente. Este mecanismo é implementado definindo uma estrutura de vizinhança para o problema. Existem três formas de realizar a transição, sendo que esta escolha determina o tipo de transição a ser realizada. As formas são: adicionar um circuito em um caminho candidato; ou ainda trocar circuitos, adicionando um circuito em um caminho candidato e retirando outro circuito em outro caminho candidato, ou somente retirar um circuito do caminho candidato. Estes tipos ainda podem ser trabalhados em conjunto, denominado de mecanismos de transição adição-troca-retirada.

Para o problema do planejamento de sistemas de transmissão o mecanismo de adição-troca-retirada pode ser implementado da seguinte forma:

- Escolhe-se aleatoriamente a linha i como sendo candidata a adição. A configuração candidata é obtida adicionando-se uma linha na posição i da configuração corrente. Se esta configuração é aceita pelos critérios do SA então foi obtida a configuração candidata e deve-se repetir o processo. Caso contrário, tenta-se a troca de linhas.
- Escolhe-se aleatoriamente a linha j entre aquelas linhas que foram adicionadas na configuração atual. A troca consiste em adicionar uma linha na posição i e retirar uma linha na posição j da configuração corrente, para se ter a configuração candidata. Se esta configuração é aceita pelos critérios do SA então foi obtida a configuração candidata e deve-se retornar ao primeiro item. Caso contrário, tenta-se a retirada da linha no item seguinte.
- A linha j escolhida no item anterior é considerada candidata a ser retirada. Logo esta configuração candidata é obtida retirando-se uma linha na posição j da configuração corrente. Se esta configuração é aceita pelos critérios do SA então foi obtida a configuração subsequente e deve-se retornar ao primeiro item. Caso contrário, não teve êxito as três tentativas (adição-troca-retirada) e retorna-se também ao primeiro item.

O mecanismo adição-troca-retirada tenta primeiramente adicionar uma linha, depois a troca e posteriormente a retirada da linha. Considerando o problema do planejamento, que se inicia com o corte de carga elevado, ocorrerá inicialmente a adição de linhas, logo quando não existe corte de carga deve-se realizar a retirada de linhas. Conclui-se que esta sequência de operações: adição-troca-retirada deve ser invertida em determinados momentos de execução do SA, visando aumentar o seu desempenho. Esta sequência deve ser invertida quando houver um corte de carga maior do que uma tolerância especificada.

Para que tenha maior eficácia no processo de geração de configurações do sistema elétrico, é necessário trabalhar com os dados da topologia da rede, garantindo a conectividade do sistema durante o processo de adição e retirada das linhas, evitando que existam linhas ineficientes ou ainda retiradas linhas de novos caminhos que deixem outras linhas sem possibilidade de transportar energia elétrica.

O processo de resfriamento é realizado com a determinação ou escolha de quatro parâmetros: o valor da temperatura inicial, o número de tentativas de transição a cada nível de temperatura, a taxa de diminuição da temperatura e o critério de parada ou a temperatura final. Estes parâmetros são fundamentais para o desempenho do algoritmo SA, sendo que sua eficiência depende da escolha adequada destes parâmetros.

2.3.3.5 Busca Tabu

É um procedimento meta-heurístico de busca local, que tem a finalidade de evitar que o processo estacione em um ótimo local, através da exploração do espaço de busca existente.

Foi criado por Fred Glover na década de 80 (ROMERO; ESCOBAR; GALLEGOS, 2006), com o propósito de solucionar problemas complexos embasando conceitos da inteligência artificial, contendo dois tipos de memória: memória de curto prazo e a memória de longo prazo. Sendo a primeira responsável por armazenar os eventos ocorridos recentemente e a segunda por armazenar o número de vezes que ocorreram determinados eventos.

Para a BT também aplica-se os processos de intensificação e diversificação. Na intensificação ocorre uma exploração ao redor das boas soluções, enquanto que na diversificação ocorre a exploração de novos subespaços de busca.

Quando um movimento foi anteriormente classificado como tabu e depois de ser analisado, este produz uma função objetivo de melhor qualidade (incumbente) do que um valor

referencial selecionado; então aplica-se o critério de aspiração que consiste em cancelar a proibição e aceitar o movimento (ROMERO; ESCOBAR; GALLEG0, 2006).

Para aprimorar estas estratégias aplica-se o “path relinking”, na qual é feita o estudo das trajetórias que conectam boas soluções. Outra estratégia de importância na BT é a *oscilação estratégica*. Esta controla os movimentos até um limite de factibilidade e depois durante o processo permite cruzar este limite retornando ao limite novamente em sentido oposto.

Inicia-se o processo de busca igual aos outros algoritmos heurísticos existentes. Na busca local, dada uma configuração x define-se uma configuração vizinha de x . No problema do planejamento da expansão da transmissão pode-se definir o termo vizinho de uma configuração x todas aquelas configurações que podem ser obtidas a partir de x com a adição de um circuito, a retirada de um circuito ou troca dos circuitos (retirada de um circuito e adição de outro), e passa-se para uma configuração vizinha quando esta apresenta diminuição na função objetivo (para os problemas de minimização).

Assim, a BT possui as seguintes características que o diferem no processo de busca local conforme mencionado.

- Utilização da lista tabu;
- Estratégia de diversificação e intensificação;
- “Path relinking” e configurações de elite;
- Redefinir o conjunto $N(x)$.

A memória de curto prazo da Busca Tabu armazena em uma lista atributos de configurações que foram modificadas recentemente, denominada memória baseada em recência. Esta possui o objetivo de evitar que o processo de busca visite lugares que já foram percorridos recentemente, considerando todos atributos acionados em um passado recente como proibidos, logo estes são excluídos do conjunto de configurações do espaço de busca local.

O critério de aspiração pode ser definido se uma possível configuração possui uma função objetivo melhor do que a incumbente, então deve-se eliminar a proibição. A BT, visa reduzir o conjunto de configurações candidatas a serem vizinhas, visto que em sistemas reais isto significa que $N(x)$ pode ter muitas configurações candidatas, as quais são obtidas através da adição e/ou remoção dos circuitos, mas isto requer a resolução de vários problemas de programação linear, não sendo aplicado na prática.

A memória de longo prazo envolve três aspectos fundamentais, os quais são: memória baseada em frequência, a estratégia de intensificação e a diversificação.

A primeira consiste em armazenar o número de vezes em que um atributo foi selecionado para participar da formação de configurações, durante o processo BT. Esta divide-se em: frequência de transição, a qual armazena o número de vezes em que determinado atributo é retirado ou adicionado nas configurações no processo BT, e a frequência de residência armazena o número de vezes em que um atributo permanece em novas configurações geradas durante a execução da BT.

Na estratégia de intensificação é feita a troca dos critérios de seleção das novas configurações candidatas, isto pode ser realizado eliminando vizinhos e incorporando novos vizinhos na configuração atual ou ainda modificando a configuração vizinha redefinindo $N(x)$; mas a forma mais atraente de realizar a intensificação é voltar às regiões atrativas identificando a configuração de elite. Após determinar essas regiões de elite, deve-se implementar novamente um novo processo de busca, conduzido por uma estratégia BT de memória de curto prazo. A intensificação por decomposição impõe novas restrições aos atributos, estruturas ou partes do problema, visando diminuir o espaço de busca (ROMERO; ESCOBAR; GALLEGOS, 2006).

Na estratégia de diversificação, conduz-se o processo de busca às regiões novas e atrativas. Através do mecanismo de oscilação estratégica, pode-se percorrer regiões factíveis e infactíveis em busca de configurações factíveis para poder aplicar o processo de intensificação na procura de um ótimo local correspondente a região; posteriormente pode-se aplicar uma rápida diversificação para encontrar novamente a região factível, reiniciando a oscilação entre as regiões.

A BT de memória de longo prazo pode possuir a estratégia de “path relinking” a qual busca boas soluções através das trajetórias traçadas entre configurações de elite, selecionando uma destas configurações chamando-a de configuração base e outra denominada configuração guia. As novas configurações são obtidas agregando atributos presentes nas configurações guias podendo usar intensificação ou diversificação.

2.3.3.6 BT Aplicado ao Problema do Planejamento

A BT apresenta o mecanismo de busca determinístico, diferente do AG e do SA que apresenta movimentos aleatórios. O processo é semelhante ao processo de busca local da

heurística de busca através de vizinhança tradicional, onde define-se uma vizinhança de x como sendo o conjunto de todas configurações de $x' \in N(x)$, nesta fase através de uma configuração corrente obtém-se a configuração vizinha que provoca uma diminuição da função objetivo. Este processo é repetido até que não exista melhora na função objetivo, encontrando-se assim uma solução ótima local. Entretanto, BT tem uma estratégia que permite sair de um ótimo local.

O mecanismo de transição da BT para o problema do planejamento do sistema de transmissão de energia elétrica trabalha com o princípio de adição-troca-retirada de linhas do circuito, como abordado no SA.

Na BT a estrutura de vizinhança é realizada dinamicamente em estrutura e tamanho para realizar a busca de forma eficiente e inteligente durante todo processo de otimização. Apresenta também a vantagem de evitar que o processo esteja em locais que já foram visitados, evitando-se a denominada ciclagem.

Como o tamanho da vizinhança não é um fator estático, a BT apresenta mecanismos para que esta vizinhança não se altere de forma desgovernada, pois o seu crescimento implica no aumento do esforço computacional. Assim, existe estratégias para a diminuição da vizinhança em cada transição. No PPET ao determinar-se a nova vizinhança se exigirá a execução de vários PL's para analisar todas as configurações candidatas, assim aplicar a redução do número das configurações candidatas favorecerá sua utilização no referido problema.

Como mencionado anteriormente, quatro estratégias são essenciais para o funcionamento da BT: memória de curto prazo, memória de longo prazo, estratégia de oscilação e "path relinking".

A memória de curto prazo consiste em proibir que atributos selecionados em um passado recente sejam escolhidos para serem configurações vizinhas, por esta razão também pode ser denominada de memória baseada em recência ou fatos recentes.

No PPET o termo atributo refere-se aos circuitos que fazem parte da configuração, assim o circuito que for adicionado na configuração será proibido de ser retirado durante as k próximas configurações geradas, sendo o k um número inteiro escolhido. Seguindo esta premissa, o atributo que for retirado da configuração corrente deverá estar proibido de ser adicionando nas próximas k configurações, desta forma existindo duas lista de atributos proibidos.

Ao tornar proibido os atributos visitados recentemente, pode-se não permitir que

a BT visite configurações novas de excelente qualidade, pois uma configuração visitada pode ter um atributo proibido que compartilha com a configuração visitada recentemente. Para amenizar esta rigidez de elementos proibidos a BT apresenta o critério de aspiração. Neste processo uma configuração que apresenta atributos proibidos poderá ter a proibição descartada, desde que se cumpra o critério de aspiração.

A BT apresenta quatro estratégias para realizar o processo de redução de configurações candidatas: aspiração adicional, formação de uma lista reduzida de configurações de elite, filtros sucessivos e estratégia de mudança cotada (ROMERO; ESCOBAR; GALLEGU, 2006).

Para o problema do planejamento de sistemas de transmissão a estratégia de redução de configurações candidatas é mais eficiente. Os elementos (atributos) que identificam as configurações vizinhas é constituído por um grupo reduzido de atributos (circuitos), determinados com o auxílio de algoritmos heurísticos construtivos usados no PPET.

A intensificação para o problema do planejamento de sistemas de transmissão é realizada em três etapas, apresentadas na Figura 2:

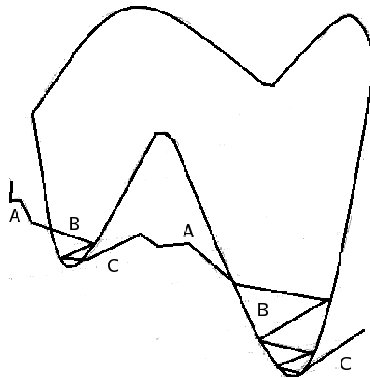


Figura 2: Estratégia de intensificação (ROMERO; ESCOBAR; GALLEGU, 2006).

- A parte A realiza a retirada/troca de circuitos se a configuração corrente apresentar corte de carga;
- A parte B realiza a retirada/troca de circuitos se a configuração corrente não apresentar corte de carga;
- A parte C realiza a retirada de circuitos quando não existe uma configuração factível vizinha (sem corte de carga) melhor que a configuração corrente (também sem corte).

Esta estratégia da memória de curto prazo permite determinar configurações vizinhas de elite, atrativas e de tamanho reduzido, ao contrário das técnicas tradicionais de

caracterização de vizinhança.

A diversificação visa conduzir o processo a regiões de busca não visitadas, com o intuito de encontrar configurações de qualidade. Este processo pode ser conduzido de duas formas:

- Realizar alterações na configuração através de adições, retiradas ou troca de linhas;
- Fazer “path relinking”.

A decisão de realizar uma das formas de diversificação é feita de forma aleatória. Caso for escolhido o primeiro item, deve-se analisar se a configuração apresenta corte de carga, se apresentar deverá ser realizado o processo de adição ou troca de circuitos; caso contrário, deve-se realizar o processo de retirada ou troca de circuitos.

A “path relinking” é realizada entre a configuração candidata à diversificação e uma das configurações de elite. Em outras palavras, este mecanismo explora trajetórias que conectam soluções de elite para obter novas soluções, o ponto de início do processo é denominado solução inicial, e através da trajetória a busca será conduzida até as denominadas soluções guia.

A memória de longo prazo da BT, visa armazenar as informações baseadas na frequência, realizando penalizações e incentivos. Como abordado, a BT apresenta a frequência de transição e a frequência de residência, sendo que a primeira armazena em um registro o número de vezes que se trocam os atributos, enquanto a segunda mantém em registro o número de iterações que os atributos permanecem nas configurações.

Existe diferentes mecanismos para trabalhar a frequência na BT para PPET. Sendo que em todas são realizadas de forma sistemática, durante a execução do processo é contabilizado o número de vezes em que, ao adicionar ou retirar uma linha, melhora a função objetivo e também o número de vezes em que uma linha reside nas configurações analisadas, juntamente também é armazenada a informação referente às trocas que melhoram e que não melhoram a função objetivo. Todas estas informações são armazenadas em vetores e matrizes. Tipos de frequências utilizados no PPET:

- *Frequência por adição*, determina o número de vezes em que uma linha ao ser adicionada melhora o valor da função objetivo;
- *Frequência por retirada*, determina o número de vezes em que uma linha ao ser retirada melhora o valor da função objetivo;

- *Frequência por residência*, determina o número de vezes em que uma linha está presente na configuração analisada;
- *Frequência por troca que melhora*, determina o número de vezes em que uma troca de linhas melhora o valor da função objetivo;
- *Frequência por troca que não melhora*, determina o número de vezes em que uma troca de linhas não melhora o valor da função objetivo.

Os diferentes tipos de frequência são aplicáveis ao PPET, sendo os seus atributos armazenados em vetores que trabalham de forma dinâmica e seu valor é aumentado à medida que o processo evolui.

3 *SOFTWARE EDUCACIONAL*

3.1 INTRODUÇÃO

Softwares educacionais possuem o objetivo de suplementar o ensino aplicado em sala de aula, fortalecendo o aprendizado de determinado conteúdo conhecido pelo aluno, mas não inteiramente dominado. Assim, o aluno poderá realizar simulações inferindo conhecimento sobre o conhecimento existente, realizando aprendizagem por descoberta. Essa descoberta será guiada pela existência de recursos motivacionais, os quais possuem o propósito de despertar o interesse do aluno em interagir com a ferramenta.

Logo, a inclusão da informática ao método tradicional de ensino permite preparar os alunos para o mundo de trabalho atual, o qual é marcado por inovações tecnológicas e totalmente informatizado.

Baseado nesta necessidade educacional, tem-se a motivação para a elaboração desta dissertação de mestrado, que tem como finalidade fortalecer o processo de aprendizagem do aluno, este estimulado e orientado pelo professor.

3.2 TÓPICOS FUNDAMENTAIS

Com a realização da primeira conferência nacional aplicada ao ensino superior foi possível expor a necessidade do uso da informática na educação. Este evento foi realizado em 1981 e nele o computador foi reconhecido como meio de ampliação das funções do professor, jamais como forma de substituí-lo.

Seguindo esta premissa, os softwares educacionais são utilizados com finalidade educativa. Esses sistemas computacionais voltados ao ensino são conhecidos como CAI (“Computer Assisted Instruction”) e apresentam o objetivo de transmitir informações sobre um

assunto. As instruções mediadas pelo computador poderão ajudar o usuário a aumentar suas habilidades de trocar informação com o computador, de modo apropriado para resolução de problemas (MORAES, 1993).

Existem diferentes softwares usados na educação, segundo classificação de Taylor(1980): tutor, ferramenta e tutelado (ZEM-MASCARENHAS; CASSIANI, 2001).

Na classificação *tutor*, o computador é direcionado ao aluno, desempenhando o papel do professor. Os softwares educacionais classificados como *ferramenta* propiciam ambiente para que os alunos possam adquirir e desenvolver conhecimentos através da interação com a interface da ferramenta, atuando como co-autor no processo de aprendizagem, enquanto os softwares tutores visam substituir o professor, ensinando através de perguntas e as respostas conduzem o aluno para etapas posteriores, sendo esta classificação comumente utilizada em jogos educativos.

O software *tutelado*, bem como os softwares educativos abordados, também trabalha com o desenvolvimento cognitivo do estudante, porém exige que o mesmo tenha uma base de conhecimento específica para interagir com o software, de forma que ele seja conduzido a formar e concretizar conhecimentos.

Quando o computador é utilizado como máquina de ensinar pode-se surgir categorias específicas desta modalidade, que são: os exercícios e prática, tutoriais, jogos e simulações (ZACHARIA, 2007).

Softwares *de exercícios e prática* são versões eletrônicas dos exercícios realizados em sala de aula, envolve memorização, repetição e fixação dos conhecimentos.

Os *tutoriais* visam ensinar e controlar o processo de aprendizagem servindo como um tutor individual e possui a vantagem de apresentar animações, imagens, sons, ou seja, possui características diferentes das apresentadas em sala de aula. Sendo em termos técnicos e pedagógicos bastante superiores aos exercícios repetitivos de prática e fixação.

Os *jogos* são considerados métodos que buscam o prazer e o divertimento, visando ensinar conceitos difíceis de serem aprendidos em sala de aula, por não existir uma aplicação prática imediata.

As *simulações* permitem vivenciar situações com realismo, reproduzindo fenômenos do mundo real que dificilmente poderiam ser trabalhados em sala de aula. Este tipo de software se aproxima de uma ferramenta e distancia-se da categoria de tutorial.

Outra forma de que o computador pode ser utilizado é como ferramenta. Nesta

categoria o aluno poderá desenvolver atividades e a aprendizagem ocorre pelo fato do aluno estar executando determinada tarefa no computador. Para que se faça o desenvolvimento desta categoria de software, como em todo processo de desenvolvimento de sistemas existe uma sistemática a ser seguida com o intuito de que o programa tenha qualidade e que cumpra todos seus requisitos.

Segundo Pressam (ZEM-MASCARENHAS; CASSIANI, 2001), o processo de desenvolvimento do software se divide em quatro etapas fundamentais: análise, desenvolvimento, testes e manutenção. No entanto, para o desenvolvimento específico do software educacional CAI são definidos, segundo Gagné (ZEM-MASCARENHAS; CASSIANI, 2001), três estágios: planejamento inicial, planejamento e desenvolvimento do conteúdo instrucional, avaliação e revisão.

No planejamento inicial deve-se identificar o propósito do software, e determinar qual a sua categoria dentro do contexto de software educacional. Na fase de planejamento e desenvolvimento do software, deve-se definir a plataforma de execução do programa, enfim, identificar e definir os critérios que são de fundamental importância para garantir o bom funcionamento do software. Na terceira e última etapa a avaliação e revisão, é que se verifica se as metas e objetivos determinados nas etapas anteriores foram alcançados.

A realização da avaliação do software educacional é de fundamental importância, sendo esta etapa um processo contínuo, realizado em três fases: inicialmente é necessário realizar a avaliação do software educacional quanto a compatibilidade de máquinas e sistemas operacionais, analisando o tempo de resposta, a qualidade das telas e a qualidade da interface do software em vários equipamentos. No segundo estágio é avaliado a qualidade do conteúdo, qualidade audiovisual e adequação ao público alvo. Na terceira etapa a avaliação deverá ser realizada por alunos, sendo avaliado a eficácia do mesmo quanto à adequação instrucional, a do programa e a sua estética.

3.3 PRINCIPAIS CARACTERÍSTICAS DO SOFTWARE EDUCACIONAL

Grandes avanços tecnológicos ocorreram na área de informática nos últimos anos, como exemplo o desenvolvimento de diversas tecnologias que favorecem o uso de computadores por usuários não-especializados na área.

Um dos fatores favoráveis ao crescimento do número de adeptos aos computadores pessoais é a engenharia de software, responsável pela elaboração da interface gráfica de sistemas. Peculiaridade que começou a ser explorada na década de 70, quando projetistas de sistemas, observaram que a interface gráfica “amigável” garantia maior aceitação do produto no mercado.

Isto acontece devido ao fato de que a interface gráfica tem como objetivo facilitar a utilização de qualquer sistema, tornando-o acessível a qualquer tipo de usuário.

Para que isto ocorra é estudado os mais diversos elementos que compoem a interface gráfica, visando sua utilização correta, tais como: a atribuição de cores, recursos multimídia, tutoriais dentre outros, os quais possam facilitar este processo de interação entre homem-máquina.

Conforme literatura especializada (LIANG, 1987), existem características elementares à uma interface gráfica, conforme segue.

1. A *diversidade*: define que a interface seja capaz de desenvolver interatividade a qualquer tipo de usuário;
2. A *complacência*: garante que a interface permita ao usuário recuperar-se de situações de erros;
3. A *eficiência*: garante que a interface minimize o esforço gasto para executar determinada tarefa;
4. A *conveniência*: garante que a interface permita fácil acesso a todas as operações do sistema;
5. A *flexibilidade*: permite que a interface provê diversas maneiras do usuário efetuar uma operação;
6. A *consistência*: determina padrões a serem seguidos, os quais facilitam a usabilidade da aplicação;
7. A *prestimosidade*: responsável por garantir o desenvolvimento de interface prestativa, fornecendo ajuda quando requisitada ou quando perceber que o usuário se encontra em dificuldades. A ajuda pode ser: na forma de mensagens de erro, conselhos, etc., devendo ser clara e precisa, não conduzindo o usuário a situações embaraçosas, nem exigindo dele conhecimentos que ele não tenha ou não possa obter pelo próprio sistema. Assim, o usuário não deve necessitar de outros recursos,

fora os oferecidos pela interface, para efetuar sua tarefa e os problemas devem ser notificados ao usuário tão logo sejam detectados, se possível antes que ocorram.

8. A *satisfação*: garante que a interface deverá satisfazer o usuário, não frustrando-o. Ela não deve demorar na resposta e deve permitir que o usuário obtenha ajuda em qualquer ponto da interação.

No *software* a ser desenvolvido almeja-se manter presente os seguintes itens acima especificados: 2, 3, 4, 6, 7, 8. A característica 1 não é obtida neste caso, pois o usuário tem que ter um mínimo de conhecimento prévio do assunto para poder usar a ferramenta. E a característica 5 também não se aplica, pois cada operação a ser executada será realizada de uma única maneira.

Contudo, pode-se observar que a interface deve abstrair dados complexos e ao mesmo tempo tornar o usuário capaz de manipular sistemas com diferentes objetivos, tornando o usuário autodidata, ou seja, quanto mais ele interagir com a interface, maior será o seu conhecimento e domínio sobre um sistema específico a ser simulado.

Assim, a interface gráfica é um forte aliado no processo de ensino-aprendizado, conduzindo ao desenvolvimento de softwares educacionais; os quais deverão motivar o aprendizado dos alunos que vivenciam o método tradicional de ensino. Os softwares educacionais devem ser utilizados como ferramenta de apoio ao ensino de forma moderna e interativa.

Para o desenvolvimento de qualquer interface gráfica um aspecto importante é a escolha da linguagem de programação adequada. Na seção que se segue, além da apresentação da interface gráfica proposta, são apresentadas as principais características da linguagem de programação Tcl/Tk, adotada neste trabalho.

4 A INTERFACE GRÁFICA PROPOSTA

4.1 INTRODUÇÃO

Na década de 70, os desenvolvedores de software não seguiam metodologias predeterminadas na criação de novos programas; o processo de desenvolvimento era feito de forma arbitrária, apresentando grandes riscos relacionados ao cumprimento das estimativas de tempo e custo.

Atualmente, ao se desenvolver um software, existe a preocupação em seguir um conjunto de regras denominadas metodologias que garantem o sucesso do projeto.

Durante o processo de desenvolvimento de software são caracterizadas três etapas fundamentais e sequenciais: a análise, a codificação e os testes.

Na fase de análise identifica-se todos os requisitos do software, sendo também definido nesta fase a linguagem de programação empregada. Na fase de codificação o programador transfere todos os requisitos para o ambiente computacional e, posteriormente, são realizados testes para conferir se foi atingido as características desejáveis definidas durante a fase de análise.

Neste trabalho, a escolha da linguagem de programação foi realizada de forma a cumprir todos os requisitos da interface gráfica identificados durante a fase de análise e os testes foram realizados acompanhando a evolução modular do projeto. Ou seja, para cada incremento de funcionalidades testes foram realizados.

4.2 A LINGUAGEM TCL/TK

Para o desenvolvimento deste trabalho, utilizou-se a ferramenta Tcl/Tk, um “kit” de programação composto por dois pacotes, o Tcl e o Tk. O primeiro garante a programação do sistema e o Tk é responsável por provêr o desenvolvimento da parte gráfica (OUSTERHOUT, 1994).

A escolha da linguagem de programação foi feita com base nos requisitos do software, que normalmente apresenta como requisito primordial o objetivo de ser uma aplicação multiplataforma, ou seja, capaz de operar em diferentes sistemas operacionais. No entanto, existem diversas linguagens que permitem suprir esta necessidade, mas a escolha foi realizada mediante as seguintes características: buscar uma linguagem de fácil aprendizado, “open source” (código-fonte aberto) e orientada à objetos.

O paradigma de programação orientado à objetos, é utilizado por diversos programadores desde a década de 80, mas foi amplamente usado quando desenvolvedores de software sentiram a necessidade de minimizar o custo de tempo e monetário nos projetos a serem desenvolvidos. A elaboração destes projetos era altamente complexa, pois conforme evoluía geravam grande quantidade de código-fonte. No entanto o paradigma orientado à objetos permite desenvolver projetos complexos de forma que durante o seu processo evolutivo tornem fáceis de entender, corrigir e modificar (DEITEL, 2001).

O termo objeto é usado como mecanismo de identificação único, o qual pode sofrer uma ação (ou comportamento). Os objetos quando apresentam o mesmo comportamento são agrupados em uma classe; assim a classe é uma abstração que descreve propriedades importantes para uma aplicação, e esta descreve um conjunto de objetos individuais, na qual cada objeto é nomeado de instância da classe.

Os objetos podem comportar-se diferentemente em diferentes classes, ou ainda, a operação de uma mesma classe pode ser implementada por mais de um método, gerando o chamado polimorfismo.

A herança é o mecanismo que permite compartilhar atributos e operações entre classes baseado no relacionamento hierárquico, surgindo termos de subclasses, as quais herdam todas características de sua superclasse, reduzindo assim a repetição do código em projetos.

O termo encapsulamento tem o objetivo de esconder informação, ou seja, ele separa aspectos externos do objeto, os quais são acessíveis a outros objetos, dos detalhes internos

de implementação do objeto, os quais permanecem escondidos em outros objetos. Sua usabilidade consiste em permitir que a implementação de determinado objeto possa ser modificada sem afetar as aplicações que usam este objeto.

Deste modo, o paradigma de orientação à objetos consiste em estruturar um programa através de objetos, enfatizando aspectos de abstração, encapsulamento, herança e polimorfismo, enquanto que no paradigma procedural é convencional a existência de um programa principal, que invoca procedimentos auxiliares para executar certas tarefas.

Programas que utilizam conceitos de orientação à objetos, ao invés de definir funções independentes, utilizadas em conjunto, dividem conceitualmente o problema em partes independentes (objetos), que podem conter atributos que os descrevem e que implementam o comportamento do sistema através de funções definidas nestes objetos (métodos).

Objetos (e seus métodos) fazem referência a outros objetos e métodos; o termo ‘envio de mensagens’ é utilizado para descrever a comunicação que ocorre entre os métodos dos diferentes objetos.

Entre as linguagens que possuem todas as características desejadas e comumente utilizada por programadores que trabalham em ambiente Linux, destaca-se a linguagem Tcl, criada no final da década de 80 por John Ousterhout, professor de Ciência da Computação na Universidade da Califórnia, juntamente com sua equipe. Foi desenvolvida durante a realização de seu doutorado e implementada com uma biblioteca de procedimentos em C, com o propósito de ser utilizada em diversas plataformas. Posteriormente foi adicionado ao Tcl um “kit” de ferramentas, o Tk.

O Tk é uma biblioteca que permite a elaboração das GUIs (“Graphical User Interfaces”) para interagir com o usuário, sendo provida de *widgets*, elementos que permitem a criação de barras de rolagem, janelas, botões, menus e vários outros recursos da programação visual.

A linguagem Tcl/Tk é uma linguagem de “scripts”. Sendo o “script” uma forma rápida e segura de gerar agilidade em tarefas repetitivas realizadas pelo usuário, no qual se pode inserir dentro do mesmo vários comandos que serão executados por linhas de comando (OUSTERHOUT, 1994). Este tipo de recurso teve seu apogeu com o uso dos sistemas UNIX, devido ao fato deste sistema operacional ser utilizado por linhas de comando.

O modelo de sistema operacional convencional começou a cair em desuso com o início das interfaces gráficas, assim houve a necessidade de uma padronização destes elementos gráficos; como exemplo, o botão X em qualquer sistema operacional possui o mesmo

propósito.

Linguagens de “scripts” requerem ambientes computacionais que sejam capazes de desenvolver sistema de software apenas com um conjunto de bibliotecas pré-definidas. A mesma utiliza um conjunto de componentes instalados pelo sistema operacional, os quais foram desenvolvidos em outras linguagens.

Com a existência das bibliotecas, programar em Tcl/Tk evita a preocupação com características específicas de hardware, mas exige estar centrado nos objetos criados e como estes devem interagir entre si. O nível de abstração é muito alto, sendo raro desenvolver programas complexos e/ou que possuam estrutura de dados; estes geralmente estão presentes nos componentes por elas manipulados. Estas linguagens são fracamente tipadas, não havendo declaração explícita de variáveis e/ou das estruturas de dados e são interpretadas.

No Tcl/Tk, durante a execução do programa fonte, o interpretador avalia e executa as operações nele descritas. Este processo não é realizado linha a linha como antigamente e sim a cada execução do programa. O interpretador exige que o microprocessador reconheça os comandos de alto nível da linguagem; assim, a cada comando de entrada, o comando será interpretado, caso seja apresentado com sintaxe correta, e posteriormente será convertido em um conjunto de instruções equivalentes em linguagem de máquina para que o microprocessador o compreenda.

Na linguagem compilada utiliza-se um compilador que gera um arquivo de linguagem de máquina que se torna o tradutor entre a linguagem de alto nível e o microprocessador. A linguagem compilada realiza a execução de determinado programa de forma rápida comparada com as linguagens interpretadas. Por exemplo, na criação de software utilizando linguagens compiladas ao realizar uma edição no código fonte o programador deverá compilar o sistema para geração do arquivo executável.

Em linguagens interpretadas este processo não é necessário, pois o código fonte é interpretado a cada interação, em contrapartida o desempenho é menor se comparados com linguagens compiladas. No entanto, atualmente existem compiladores Tcl disponíveis na internet, agrupados em três tipos: os *Tcl-Wrap*, os *Tcl-C* e os *Tcl-Bytecodes*.

Compiladores *Tcl-Wrap* geram programas executáveis com o interpretador, seu código Tcl e todas as imagens, bibliotecas e arquivos necessários para execução do programa, logo este são chamados empacotadores de “scripts”.

O *Tcl-C*, converte o código Tcl em código C (linguagem compilada), o qual deverá

posteriormente ser recompilado por um compilador C.

Os compiladores *Tcl-Bytecodes* compilam o código Tcl diretamente para *bytecodes*, eles são gerados pelo compilador e podem ser usados em diferentes plataformas otimizando a velocidade de execução do programa.

A linguagem Tcl interpretada não precisa ser compilada ou “linkada”, podendo o programa ser executado de imediato através de um interpretador Tcl como é o caso do *tclsh*, uma aplicação shell que lê comandos Tcl do arquivo. Quando o *tclsh* é evocado no modo interativo, funciona como um dispositivo terminal como um *prompt*. Neste caso são lidos comandos do *standard input* e impressas mensagens de erro e resultados dos comandos no *standard output*.

Uma alternativa ao *tclsh* é a shell *wish*, que pode ser usada do mesmo modo que o *tclsh*. O *tclsh* trata-se de um interpretador em modo texto, enquanto o *wish* é interpretador de modo gráfico.

A Tcl/Tk apresenta apoio de comunidades Tcl e engenheiros da Scriptics, este último trata-se de um renomado site do desenvolvedor, no qual os mesmos contribuem para o aperfeiçoamento da linguagem. Assim, foi desenvolvido o *Tcl/Tk browser plugin* voltado para aplicações cliente/servidor, o qual permite executar aplicações nos navegadores *Netscape* e o *Internet Explorer*, bem como o desenvolvimento do *Secure-Tcl, plugin* que foi acrescentado ao *Tcl/Tk browser plugin* tornando o ambiente seguro para executar *Tclets* (aplicações cliente-servidor em Tcl/Tk).

A versão Tcl/Tk 8.0, é inovada com a presença do compilador *run-time*, gerando um programa executável a partir do código-fonte, cuja finalidade é reduzir o tempo de execução deste código-fonte. A versão mais recente do Tcl/Tk, a versão 8.6a3, segue o paradigma de programação orientado à objeto buscando maior rapidez no tempo de resposta durante a resolução de determinado problema (TCLER'S WIKI, 2008).

No entanto alguns comandos Tcl/Tk foram modificados para melhorar seu comportamento perante os novos compiladores, surgindo algumas incompatibilidades, as quais podem ser facilmente contornadas. Nesta versão há a inserção de novos *widgets* dotados de aspectos Windows.

O Tcl/Tk propicia fácil sintaxe de programação e aprendizado, podendo ser usada como uma aplicação *standalone* ou embutida em outras aplicações. Sendo portáteis e executados em todos sabores do UNIX como exemplo: o Linux, o Solaris, etc.

Assim o Tcl/Tk supre a necessidade encontrada atualmente por desenvolvedores de

software, quando deparam com uma diversidade de arquitetura de máquinas existentes nas empresas. Logo, o Tcl/Tk está sendo utilizada como plataforma de integralização (TCL/TK BRASIL, 2007).

A referida linguagem de programação pode ser utilizada em diversos tipo de aplicações, atualmente pode-se verificar o seu enorme sucesso em grandes companhias, como a Pixar, a Motorola, a Cisco System Inc. e outras, além de que possui extensões disponíveis para utilizar todas as bases de dados padrões: Mysql, Oracle, PostgreSQL, etc. (TCL/TK, 2007).

São várias as vantagens que a Tcl/Tk oferece, e com base neste sucesso e como mencionado, esta linguagem supre todas as necessidades do software a ser desenvolvido.

4.3 A INTERFACE GRÁFICA PROPOSTA

A interface gráfica TEP - (“Transmission Expansion Planning”) tem como objetivo interagir de forma direta com os programas executados em “background” de acordo com o ajuste de parâmetros necessários, mediante a heurística aplicada e a escolha de determinado sistema teste, apresentando todas as linhas adicionadas e/ou removidas do sistema, durante a execução dos programas envolvidos na simulação. Isto favorece o processo de aprendizagem do usuário, o qual poderá compreender cada situação por ele criada, através da utilização da interface.

Assim, com o propósito de agregar as características mencionadas no processo de desenvolvimento de um software educacional, a interface TEP foi elaborada visando facilitar o ambiente de navegação através de uma janela principal que contém as principais funções do programa, como mostrado na Figura 3.

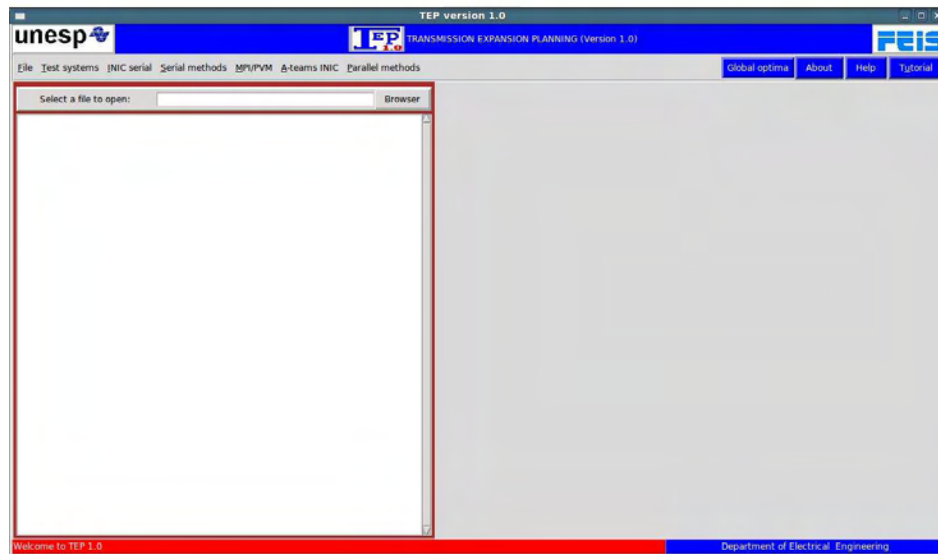


Figura 3: Tela inicial da interface TEP.

Na barra de menus da interface, partindo-se da esquerda para a direita, temos o menu *Exit*, esse quando acionado fecha todo o programa; o segundo menu, *Test systems* determina a escolha do sistema teste que deverá ser utilizado durante a interação com o programa, bem como uma opção de poder adicionar futuramente outros sistemas como pode ser visto na Figura 4. Os sistemas testes disponíveis atualmente são: Garver (6 barras/15 ramos), Sul brasileiro (46 barras) e Norte-Nordeste brasileiro (87 barras/179 ramos).

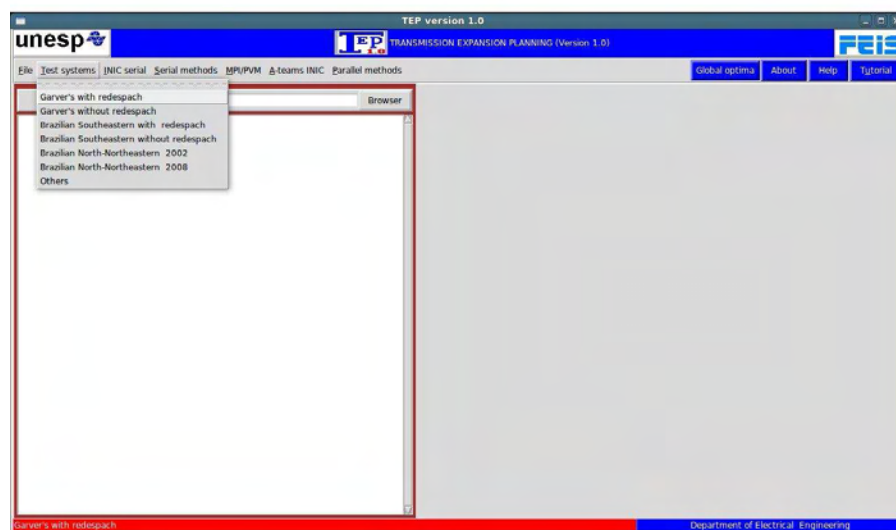


Figura 4: Sistemas testes.

A escolha de determinado sistema aciona uma janela dentro da interface, que apresenta

o diagrama unifilar do sistema teste correspondente. Na Figura 5 foi escolhido para a simulação o sistema Garver com redespacho.

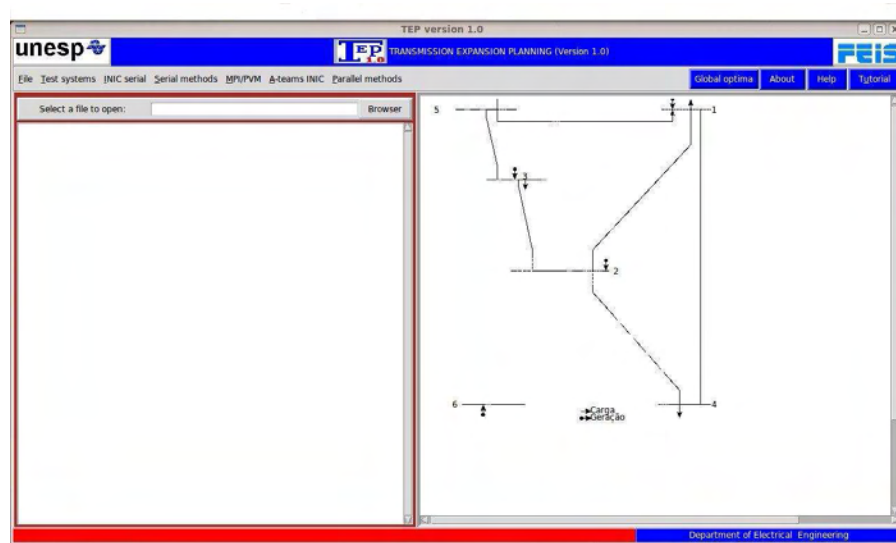


Figura 5: Sistema Garver com redespacho.

Carregado o sistema teste, o usuário pode ler os dados de entrada do mesmo através da opção de navegação presente na TEP à esquerda; clicando no botão *Browser* (Figura 6).



Figura 6: Opção para navegação do programa *TEP*.

Os dados do sistema serão visualizados na janela interna à esquerda da TEP, após a escolha do arquivo de dados, este processo pode ser visto na Figura 7.

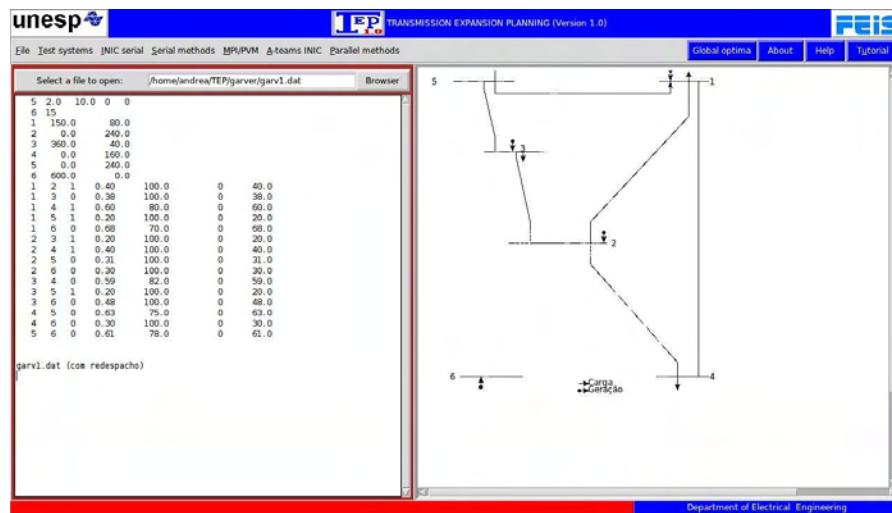


Figura 7: Dados de entrada do sistema Garver.

Definido o sistema teste, deve-se escolher através do menu *INIC Serial* com qual método serão calculadas as configurações iniciais factíveis daquele sistema teste a serem utilizadas na inicialização da meta-heurística que será escolhida posteriormente.

Na Figura 8, pode-se escolher um dentre os métodos heurísticos construtivos disponíveis, além da inicialização aleatória, para a inicialização do problema.

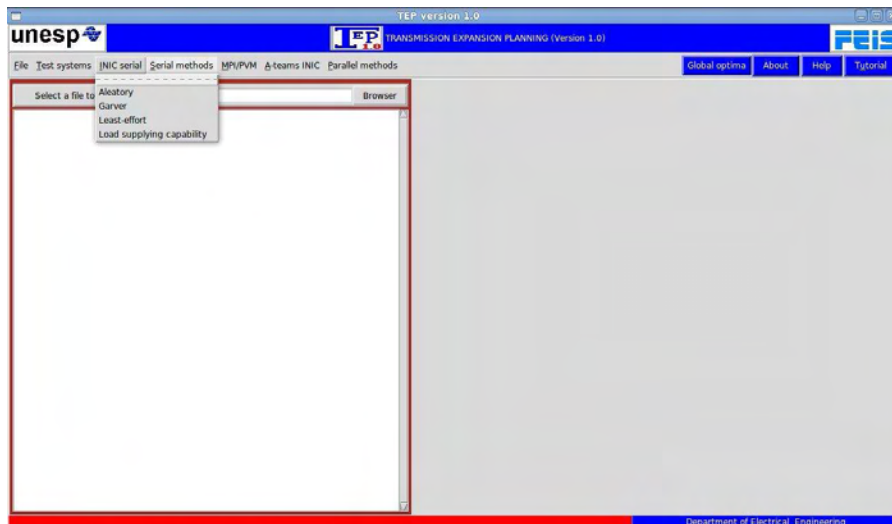


Figura 8: Escolha do método para a inicialização do problema.

Uma vez obtidas as configurações iniciais, deve-se definir o método de resolução que deverá ser aplicado através do menu *Serial methods*, conforme pode ser visto na Figura 9.

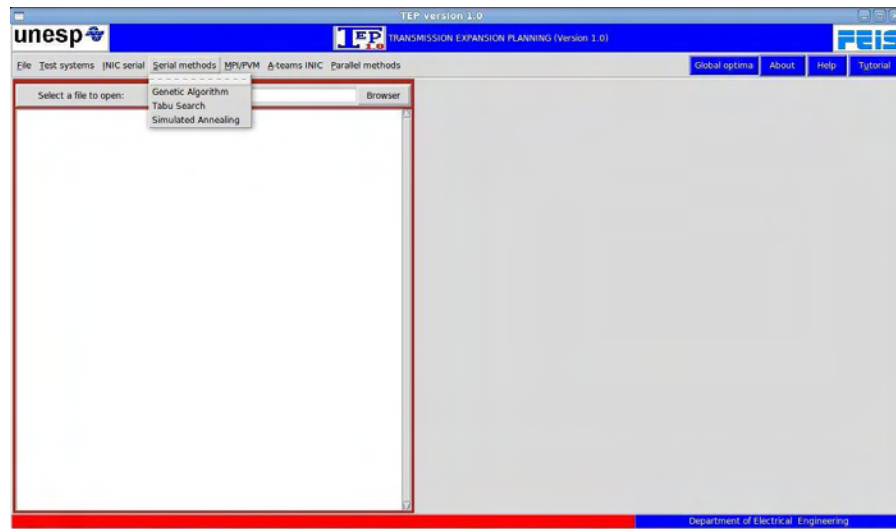


Figura 9: Escolha do método para a resolução do problema.

Na sequência, escolheu-se como exemplo o método algoritmo genético e na Figura 10 é mostrado a nova tela referente ao método e sistema teste escolhidos, ou seja, sistema teste Garver com redespacho e o método Algoritmo Genético.

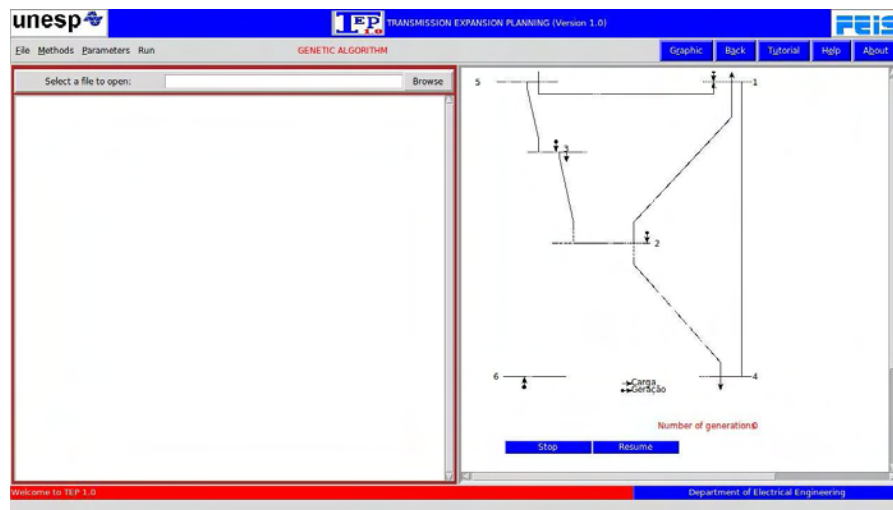


Figura 10: Tela referente ao método Algoritmo Genético.

Nesta nova janela, tem-se a opção do menu *Parameters*, o qual apresenta todos os parâmetros de calibração do sistema para a realização de simulações através do referido método; conforme pode ser visto na Figura 11.

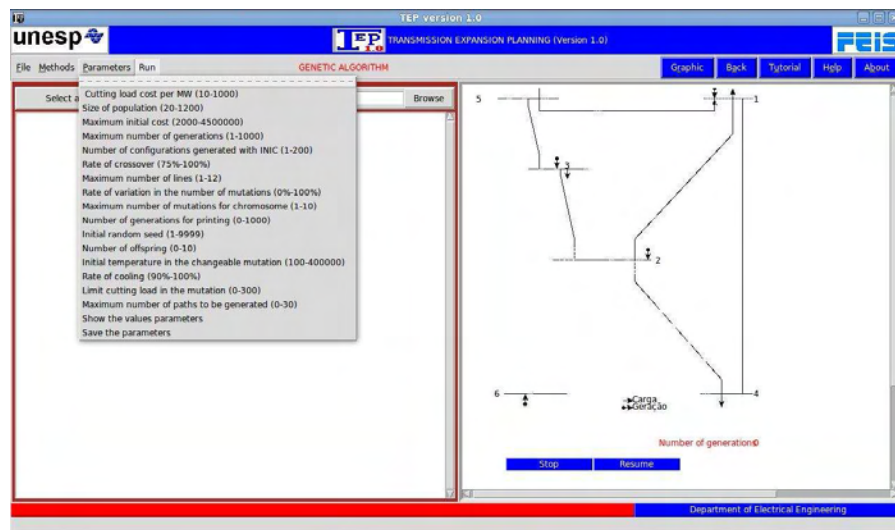


Figura 11: Ajuste de parâmetros.

Todos os parâmetros podem ser facilmente manipulados, como por exemplo o ajuste do custo de corte de carga; para isso deve-se alterar o valor do parâmetro desejado e, em seguida acionar o botão *OK*, conforme mostrado na Figura 12.

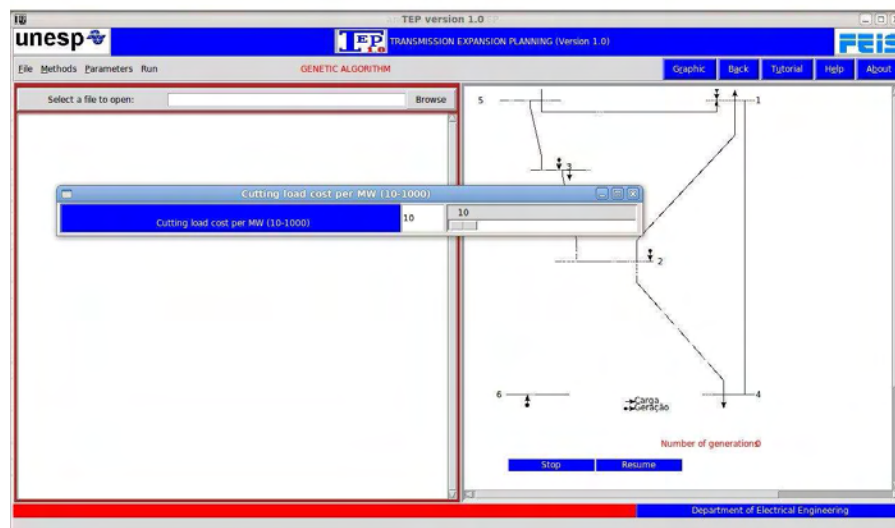


Figura 12: Ajuste do custo do corte de carga.

O ajuste de todos os outros parâmetros segue este mesmo estilo de funcionalidade. Nos dois últimos itens do menu *Parameters* (Figura 11), o primeiro tem a função de mostrar os valores dos parâmetros já calibrados e o segundo possui o objetivo de salvar estes parâmetros para serem utilizados posteriormente.

Após serem calibrados os parâmetros desejados, deve-se recorrer ao menu *Run*, onde

o usuário terá acesso a um submenu com as opções das diferentes versões de algoritmos baseados na meta-heurística selecionada. Escolhido o algoritmo (neste caso o *Ganord*), o sistema será simulado e visualiza-se a cada geração a configuração incumbente (melhor solução) na janela gráfica da direita, conforme a Figura 13. Nesta mesma figura, através do botão *Browser*, pode-se acessar o arquivo texto de saída com diversas informações da simulação, na janela da esquerda.

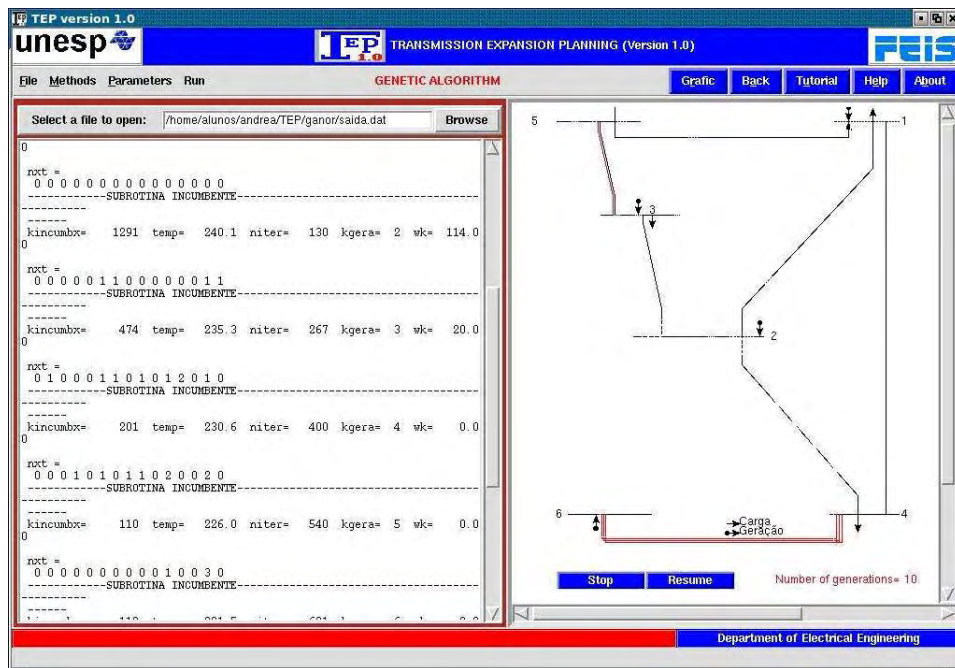


Figura 13: Resultado intermediário do Algoritmo Genético / Sistema Garver.

A interface possui botões que permitem manipular o processo de execução do algoritmo *Ganord*. Com estes, o usuário pode facilmente parar a execução, caso queira analisar com cautela cada iteração realizada. Para realizar esta tarefa basta que o usuário acione o botão *Stop* para interromper a execução do programa; para retornar a execução deve-se acionar o botão *Resume*.

Assim, durante a execução desta meta-heurística, o usuário visualizará passo a passo a adição e/ou retirada de circuitos do sistema testes em análise, o número de gerações e terminando com a visualização da solução ótima global, neste exemplo.

A TEP também possui o menu *Graphic*, sendo este capaz de plotar gráficos referentes a solução incumbente encontrada em cada instante. Nesta versão da interface gráfica, funcionando para o Linux, utiliza-se dos aplicativos *Gnuplot* e *Xfig* para a visualização da solução incumbente. Isto é feito através de um *script* que é interpretado internamente pelo Linux ao acionar o menu *Graphic*. Na Figura 14, pode-se verificar o gráfico da evolução

da solução, visualizado através do aplicativo *gv*.

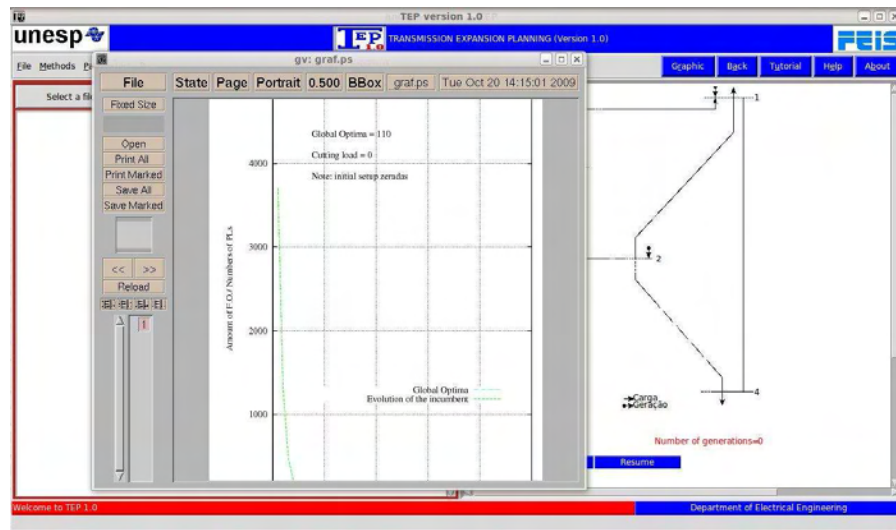


Figura 14: Menu Graphic.

O menu *Tutorial*, outro recurso da interface, aborda conceitos técnicos agregados ao PPET, explica o que são cada um dos conceitos e posteriormente descreve como utilizá-los, conforme visto na Figura 15.

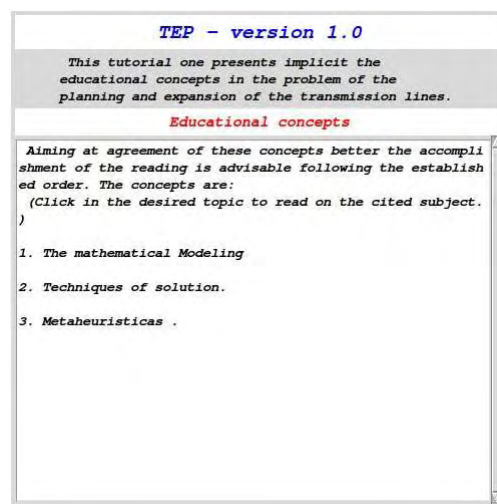


Figura 15: Menu Tutorial.

As informações de cada item específico do tutorial, por exemplo, a modelagem matemática, podem ser visualizadas com a associação de um arquivo em pdf, via aplicativo *kpdf* de acordo com o item selecionado pelo usuário, conforme mostrado na Figura 16.

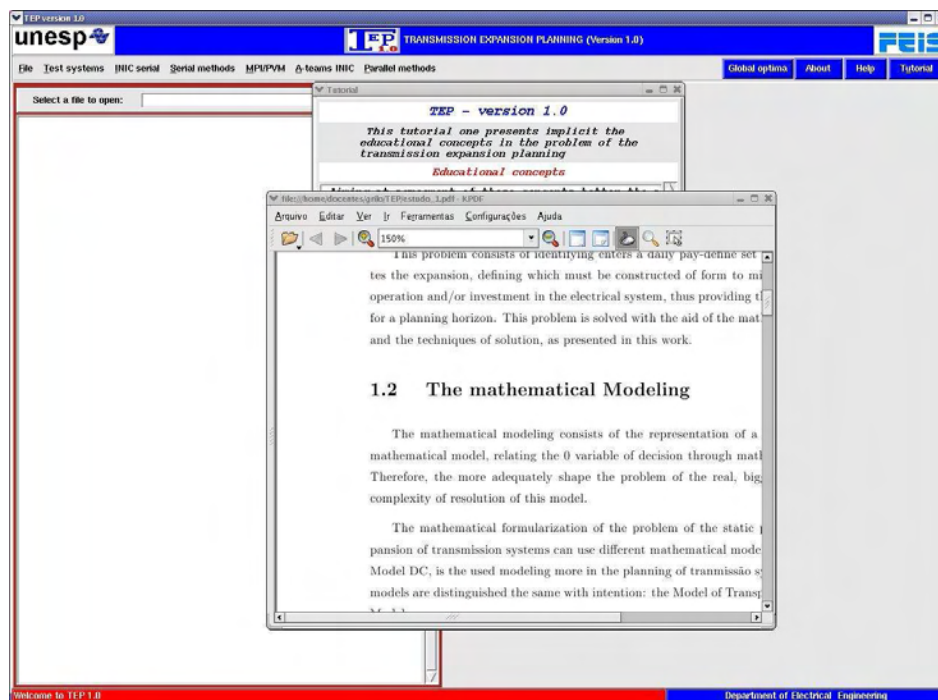


Figura 16: Menu Tutorial (detalhes).

O menu *Help* apresenta os passos básicos para se conseguir interagir com a interface TEP, como apresentado na Figura 17.

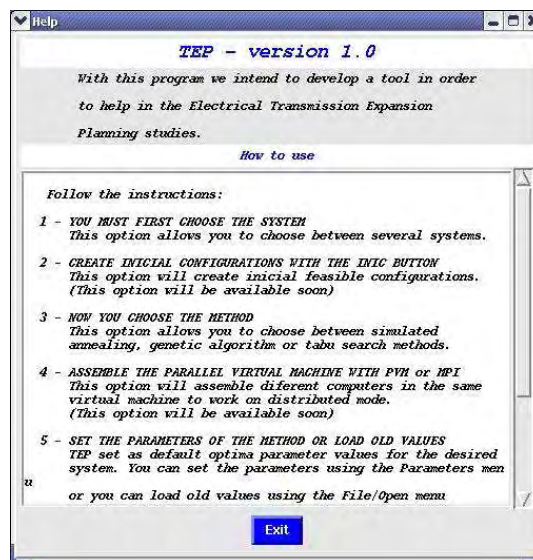


Figura 17: Menu Help.

O menu *About* foi idealizado para informar ao usuário qual a versão do programa e os autores do mesmo, bem como o email de contato do orientador do projeto, como visualizado na Figura 18.



Figura 18: Menu About.

Voltando-se a tela inicial do TEP, através do menu *Back*, conforme mostrado na Figura 19, o usuário pode selecionar outros atributos de uma nova simulação, escolhendo um novo sistema teste ou outro método de resolução do PPET.

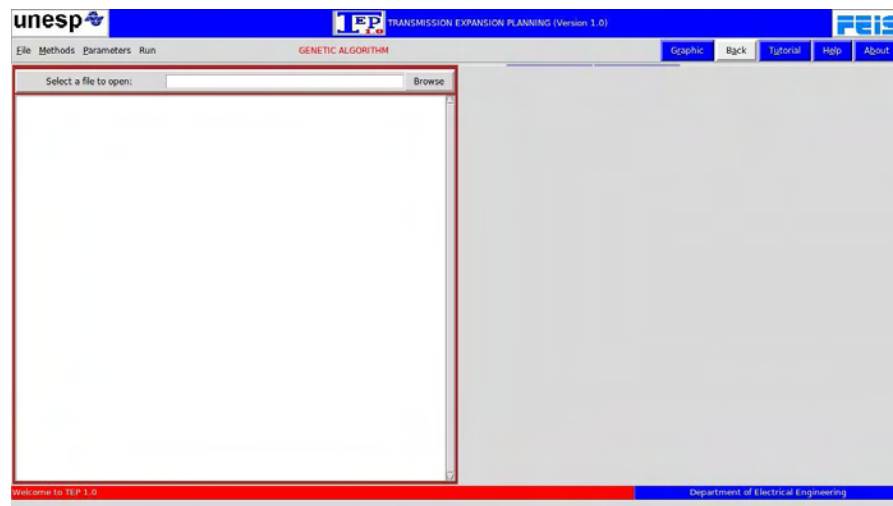


Figura 19: Menu Back.

Desde o desenvolvimento dos primeiros computadores existe a preocupação com a velocidade de execução das instruções em determinado intervalo de tempo, logo, várias pesquisas e projetos foram idealizados visando obter maior velocidade de execução dos dados. As instruções executadas de maneira sequencial exigem maior tempo de uso do processador, enquanto que as instruções executadas em paralelo diminuíam esse tempo de processamento, acarretando no aumento da velocidade da execução das instruções.

Assim, na janela principal do TEP, Figura 3, página 53, tem-se o menu *MPI/PVM*, que possibilita a escolha prévia de máquinas disponíveis na rede computacional para posterior execução de algoritmos que utilizam bibliotecas de processamento paralelo.

Na tela apresentada a seguir (Figura 20), o usuário escolheu as máquinas *fiona* e *robim* disponíveis no LLPP - Laboratório de Linux e Processamento Paralelo, ambiente em que foram feitas as simulações.

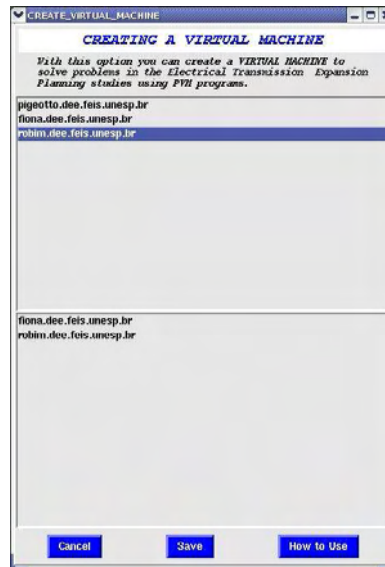


Figura 20: Escolha das máquinas para processamento paralelo.

Após a escolha, os nomes das mesmas são armazenados em um arquivo texto denominado *maqs*, que posteriormente é usado para “desovar” as instâncias dos algoritmos paralelos. Isto é feito pelo usuário através do botão *Save* na tela.

Realizada a montagem das máquinas o usuário deverá retornar a janela principal e acionar um dos submenus *Run PVM* ou *Run MPI* dentro do menu *MPI/PVM*.

O menu *A-teams INIC* executará os times assíncronos constituídos por agentes de algoritmos heurísticos construtivos paralelos e que possibilitam a inicialização do problema, ou seja, neste menu, o usuário poderá montar através da opção *Create A-TEAM INIC* um time assíncrono inicializador. A Figura 21 mostra o submenu *A-teams INIC*.

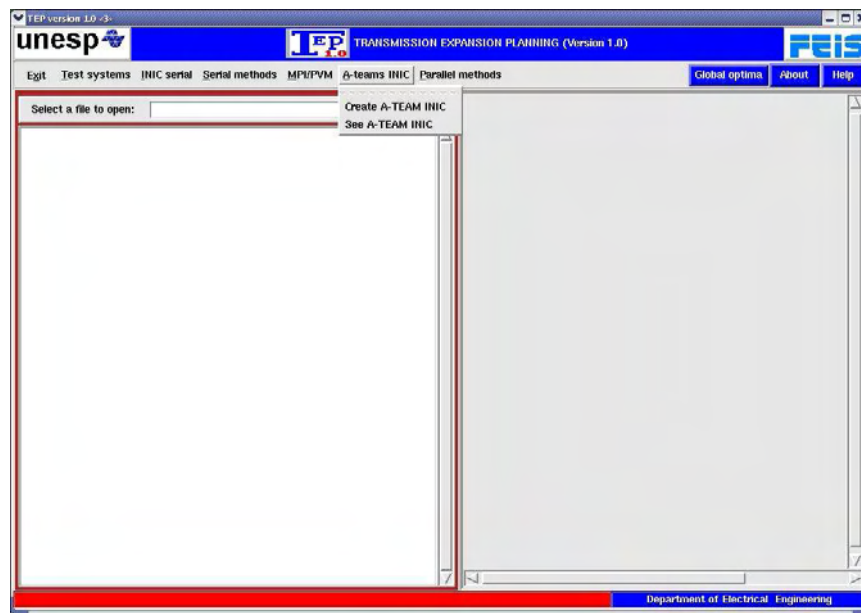


Figura 21: Tela para montagem de um A-Team INIC.

O menu *Parallel methods* executará os algoritmos paralelos desenvolvidos para as meta-heurísticas algoritmos genéticos, *simulated annealing* e busca tabu. Esses algoritmos podem ser escolhidos através do submenu apresentado na Figura 22.



Figura 22: Submenu *Parallel methods*.

Ainda dentro da janela principal, ao acionar um dos itens do menu *Global optima*, Figura 23, consegue-se visualizar na parte gráfica a solução ótima de qualquer dos sistemas testes, facilitando ao usuário a comparação das soluções encontradas com a solução ótima global ou com a melhor solução existente para cada sistema.

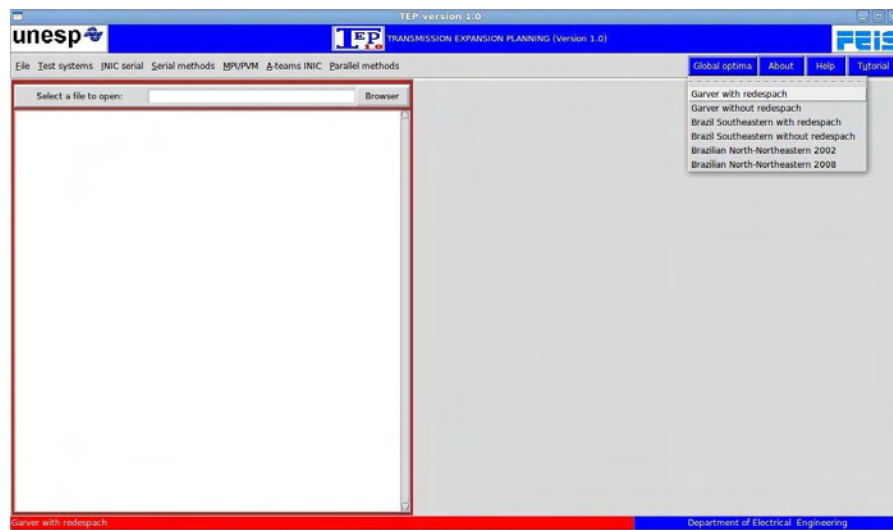


Figura 23: Menu para a escolha de visualização de configuração ótima.

A Figura 24, a seguir, mostra a configuração ótima para o sistema Garver com redes-pacho.

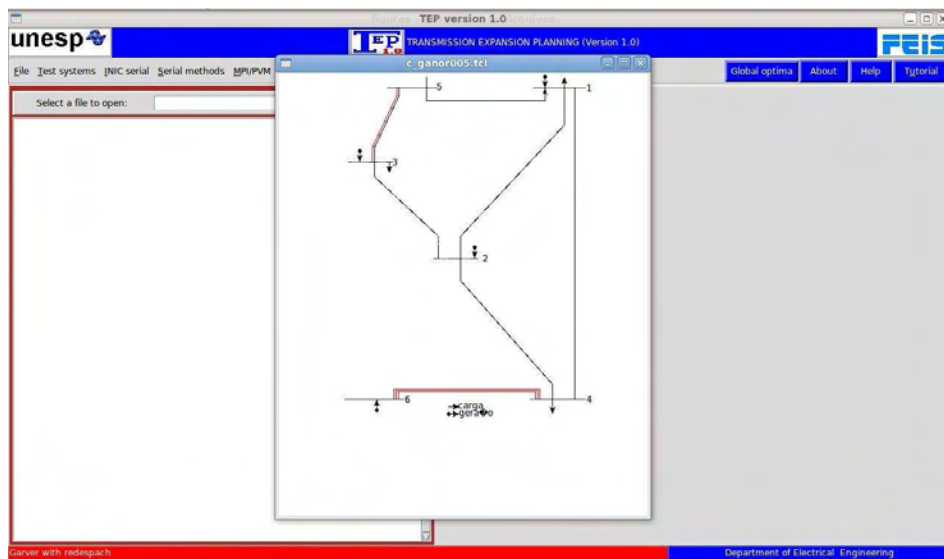


Figura 24: Configuração ótima para o sistema Garver com redespatcho.

Na interface gráfica TEP, o usuário poderá usar durante a realização da simulação o algoritmo genético serial Ganor (Gallego, de Oliveira, 1997b) detalhado em (Gallego, 1997), com alterações realizadas no código fonte para possibilitar a geração dos gráficos necessários para melhor análise dos resultados obtidos pelo algoritmo.

O algoritmo Ganor apresenta as seguintes estruturas básicas: a população inicial é gerada com o auxílio de um inicializador INIC para os diferentes sistemas testes, a codi-

ificação do problema é a decimal e o problema é representado pela função de normalização $\min v \Leftrightarrow \max [K - v]$, onde K é uma constante variável.

Os parâmetros de controle do algoritmo são:

1. n_{pop} - tamanho da população;
2. y_{cross} - taxa de recombinação;
3. n_{des} - número máximo de descendentes, em porcentagem;
4. n_{mutar} - taxa de mutação;
5. $temp$ - temperatura inicial do processo SA .

Os blocos construtivos são preservados no processo de mutação e de recombinação. Neste algoritmo também é feita a seleção limitada, evitando a convergência prematura de descendentes, assim o parâmetro n_{des} (número mínimo de descendentes) representa um limite, em porcentagem da população total.

A recombinação aplicada foi a recombinação dupla, que consiste em selecionar aleatoriamente dois pontos de recombinação e posteriormente é feita a troca entre estes dois pontos. O critério de parada é definido por um número máximo de gerações predefinido.

Assim o algoritmo poder ser descrito nos seguintes passos:

- Passo 1: É feita a leitura do tamanho da população n_{pop} e o número de configurações iniciais n_{para} (determinado com o algoritmo inicializador INIC), e a leitura de K_{gera} (número máximo de gerações permitidas);
- Passo 2: Para cada configuração é determinada o valor da função objetivo ($fitness$) correspondente, usando o programa de PL (MINOS). Atualizar a incumbente, caso seja possível. A variável K_2 conta o número de gerações.
- Passo 3: Se $K_2 > K_{gera}$, PARE; pois o número de gerações foi ultrapassado. Caso contrário, vá ao passo 4.
- Passo 4: Através do mecanismo de seleção proporcional é determinando o número de descendentes.
- Passo 5: É feita a recombinação entre todas as configuração que devem participar na formação da nova geração, até gerar o número de descendentes igual ao tamanho

da população. A taxa de recombinação (*ycross*) é que controla estatisticamente o número de configurações que serão recombinadas.

- Passo 6: A mutação é realizada de forma sistemática podendo ser mutações simples ou duplas, e este processo é controlado pela taxa de mutação. Voltar ao passo 2.

A TEP disponibiliza também várias versões de algoritmos paralelos para serem executados durante a realização de simulações, no entanto será abordado neste trabalho o algoritmo *gapar2b* (DE OLIVEIRA, 2004). Este consiste em um algoritmo que resolve os PL's do problema do planejamento utilizando n microprocessadores, de forma que consiga executar o problema com mais rapidez do que algoritmos sequenciais.

Este algoritmo trabalha no paradigma de programação paralela Mestre/Escravo usando primitivas PVM. Neste a unidade designada como mestre coordena as atividades realizadas pelas máquinas escravas que encontram dentro de uma rede de computadores.

O algoritmo *gapar2b* é definido pelos seguintes passos:

- Passo 0 - mestre: Definir o tamanho da população (*npop*) e com o auxílio do programa inicializador *INIC* é gerada a população inicial. Nesta fase também é necessário definir o número máximo de gerações permitidas (*Kgera*), bem como estabelecer o número de gerações para troca de mensagens entre os microprocessadores (K_1);
- Passo 1 - mestre: O processo mestre deverá enviar para cada processador escravo n/p configurações da população;
- Passo 1 - escravo: este passo é referente aos processos escravos, os quais devem determinar os valores das funções objetivos das n/p configurações atribuídas pelo mestre através do programa de PL (MINOS);
- Passo 2 - escravo: Fazer $N = \text{int}(K_2/K_1)$, onde k_2 é o número de gerações e se for possível atualizar a incumbente;
- Passo 3 - escravo: Se $K_2 = kgera$ ou $K_2 = N * K_1$, ir ao passo 7-escravo, senão ir ao passo 4-escravo;
- Passo 4 - escravo: Através do mecanismo de seleção determinar o número de descendentes correspondente a cada configuração;

- Passo 5 - escravo: Realizar a recombinação entre as configurações que irão participar da nova geração, até que o número de descendentes atinja o tamanho da população, controlada pela taxa de recombinação;
- Passo 6 - escravo: Realizar a mutação, controlada pela taxa de mutação, e voltar ao passo 2 - escravo;
- Passo 7 - escravo: Enviar a incumbente para o processo mestre e se $K_2 = k_{gera}$, enviar aviso para o mestre PARAR, senão ir para o passo 8-escravo;
- Passo 2 - mestre: Receber de cada processo escravo a incumbente e determinar entre elas qual a melhor, se tem aviso para PARAR então PARE senão voltar ao passo 3 - mestre.
- Passo 3 - mestre: Enviar a melhor configuração para os processos escravos. Voltar ao passo 2-mestre.
- Passo 8 - escravo: Receber a melhor configuração do mestre e voltar ao passo 2-escravo.

5 *RESULTADOS E DISCUSSÕES*

5.1 INTRODUÇÃO

Todo o desenvolvimento da interface gráfica TEP foi realizado no LLPP - Laboratório de Linux e Processamento Paralelo, do Departamento de Engenharia Elétrica da UNESP, Campus de Ilha Solteira, sob Linux Fedora 5.0 (kernel 2.6.17_1.2174_FC5smp), Tcl versão 8.4.13-1.1 e Tk versão 8.4.13-1.1.

Os programas relativos as heurísticas e meta-heurísticas foram desenvolvidos na linguagem Fortran e o compilador utilizado é o *ifort* da Intel, versão 8.1.

A interface gráfica TEP foi desenvolvida com o objetivo de facilitar a interação do usuário com ferramentas computacionais aplicadas ao estudo do problema do planejamento da expansão da transmissão do sistema elétrico, estudo este que requer conhecimentos específicos e a correta calibração dos algoritmos utilizados para a resolução do problema. Assim a interface gráfica do TEP deve auxiliar na abstração desta complexidade e aprimorar o ambiente de aprendizado.

O sistema teste Sul brasileiro, que antes era visualizado somente como figura estática no programa, agora foi agregado ao TEP na forma de *canvas*, tornando-se possível visualizar o diagrama unifilar deste sistema com todos os seus componentes (Figura 25). Assim, alterações topológicas, obtidas pelos algoritmos disponíveis na interface, serão mostradas dinamicamente durante a execução destes algoritmos.

Na sequência do processo de desenvolvimento do TEP, foi feita a elaboração dos *canvas* do sistema teste Norte-Nordeste brasileiro. Este *canvas* inicialmente foi desenvolvido isoladamente, ou seja, externo ao TEP, com o intuito de facilitar a programação. A imagem do referido canvas é apresentada na Figura 26, e o mesmo foi incorporado posteriormente ao TEP.

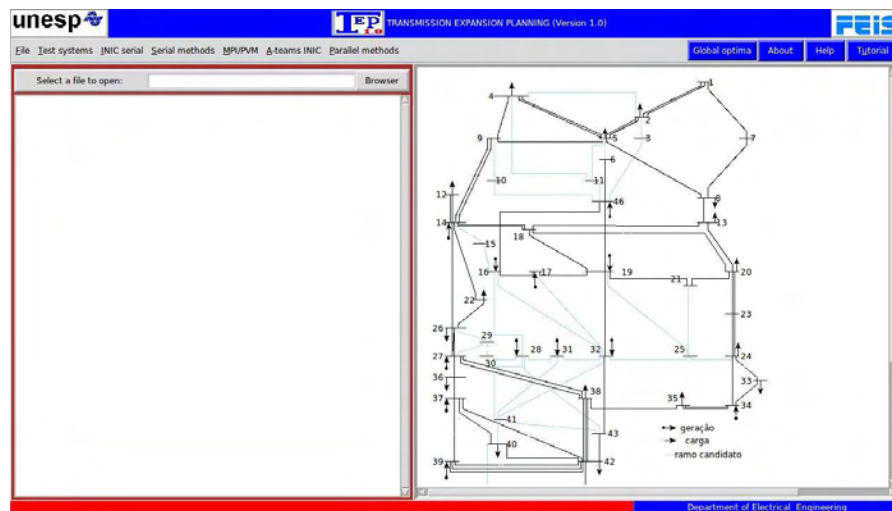


Figura 25: *Canvas* do sistema Sul brasileiro.

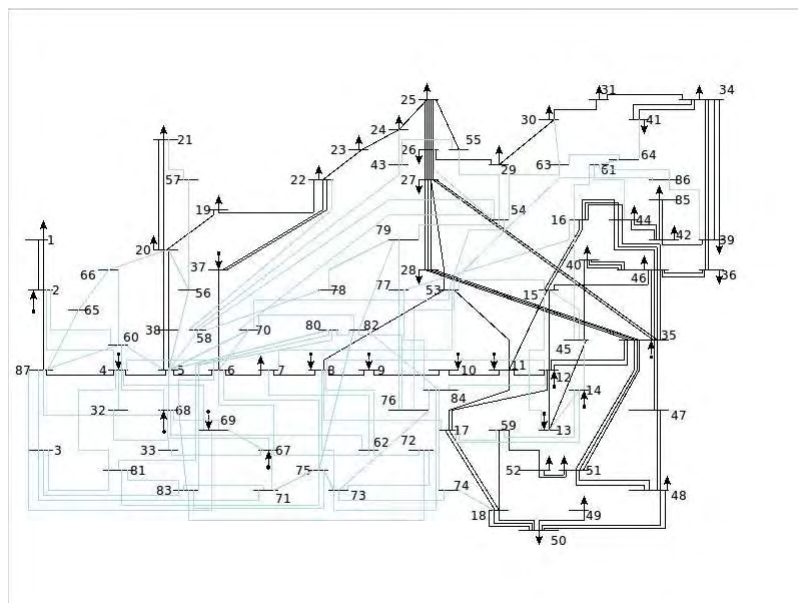


Figura 26: *Canvas* do sistema Norte-Nordeste brasileiro.

Com o objetivo de melhorar a visualização dos sistemas testes de grande porte implementou-se o recurso de *Scrollbar*, o qual permite o uso de barras de rolagens na janela na qual foi inserido o *canvas* referente a este sistema teste.

A Figura 27, a seguir, mostra este recurso já incorporado ao *canvas* do sistema teste Norte-nordeste brasileiro. Este recurso também está presente no *canvas* do sistema Sul brasileiro (Figura 25).

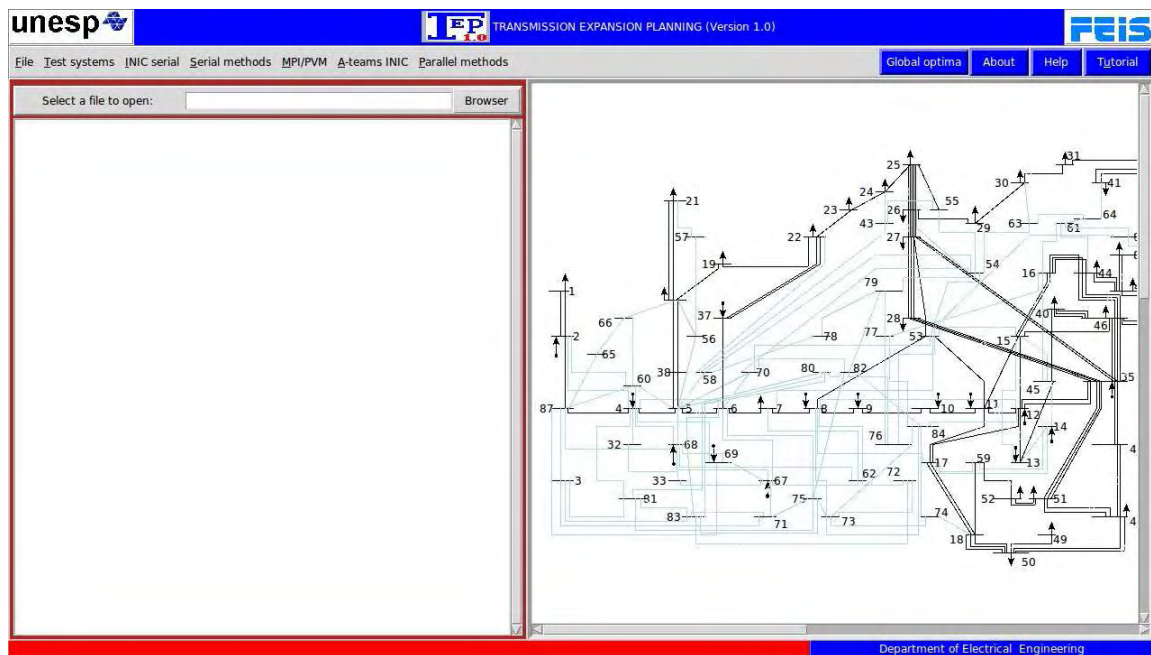


Figura 27: Recurso do *Scrollbar* para figuras grandes.

Como mencionado anteriormente, o *canvas* do sistema Norte-Nordeste brasileiro, foi elaborado de forma isolada e posteriormente agregado ao TEP. Nesta etapa existiu a dificuldade de programação, pois foi necessário fazer com que a interface como um todo trabalhasse com dois processos diferentes de visualização das linhas a serem adicionadas ou removidas durante a execução dos algoritmos em *background*. Ou seja, uma maneira de visualizar as linhas que devem ser adicionadas ou removidas é apresentando o desenho dessas linhas em cores diferentes entre as barras do sistema teste. A outra maneira é apresentar nos ramos entre as barras somente os números de linhas que devem ser adicionadas no sistema. A opção de visualizar os desenhos das linhas está disponível para sistemas de pequeno porte (sistema Garver), enquanto que a apresentação na forma de número de linhas está disponível para os demais sistemas.

Foi incluído também, na janela correspondentes a cada algoritmo, o botão *Back*, que permite a navegação entre as janelas. Assim o usuário pode retornar à janela principal da interface TEP quando desejar, podendo escolher outro algoritmo para simulação.

Na janela *Genetic Algorithm* no menu *Parameters*, as opções de ajustes dos parâmetros estavam escritos em português não seguindo o idioma padrão utilizado. Assim essas opções de ajustes foram traduzidas para o idioma padrão utilizado no TEP, como pode-se observar na Figura 28.

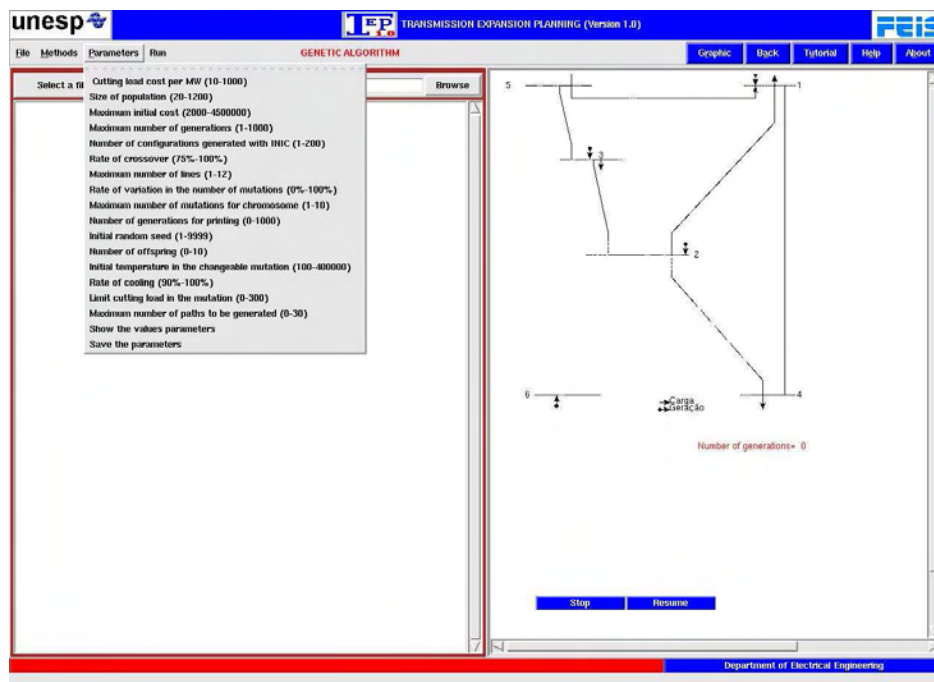


Figura 28: Opções do menu *Parameter*.

Na *release* anterior da TEP, era possível realizar simulações do algoritmo Ganor somente com o sistema Garver (6 barras/ 15 ramos). Nesta versão pode-se realizar simulações deste algoritmo com qualquer um dos sistemas testes presentes no TEP.

Nesta fase do desenvolvimento do software é possível realizar simulações de todos sistemas testes com as técnicas de resolução Algoritmos Genéticos e a Busca Tabu. Também foram implementados na interface gráfica os botões de manipulação da execução da simulação, o botão *Stop*, o botão *Resume*, o menu *Graphic* e o *label* informativo dentro do canvas.

O menu *Graphic* permite que o usuário visualize a melhor solução encontrada (incumbente) através de gráficos e, no *label*, o usuário poderá visualizar o número de gerações dentro da janela de simulação, não sendo necessário abrir o arquivo que contém esta informação.

Para abrir o arquivo que contém a informação do número de gerações é necessário que o usuário clique no botão *Browser* e em seguida deve-se selecionar a pasta correspondente a técnica de resolução que está sendo utilizada na realização da simulação. Depois de aberta esta pasta o usuário deverá selecionar o arquivo chamado *saida.dat*. O mesmo será visualizado no lado esquerdo da TEP.

Na Figura 29 pode-se visualizar o primeiro passo a ser feito para fazer a leitura do

número de gerações através de abertura de arquivos. O usuário deverá clicar no botão *Browser*.

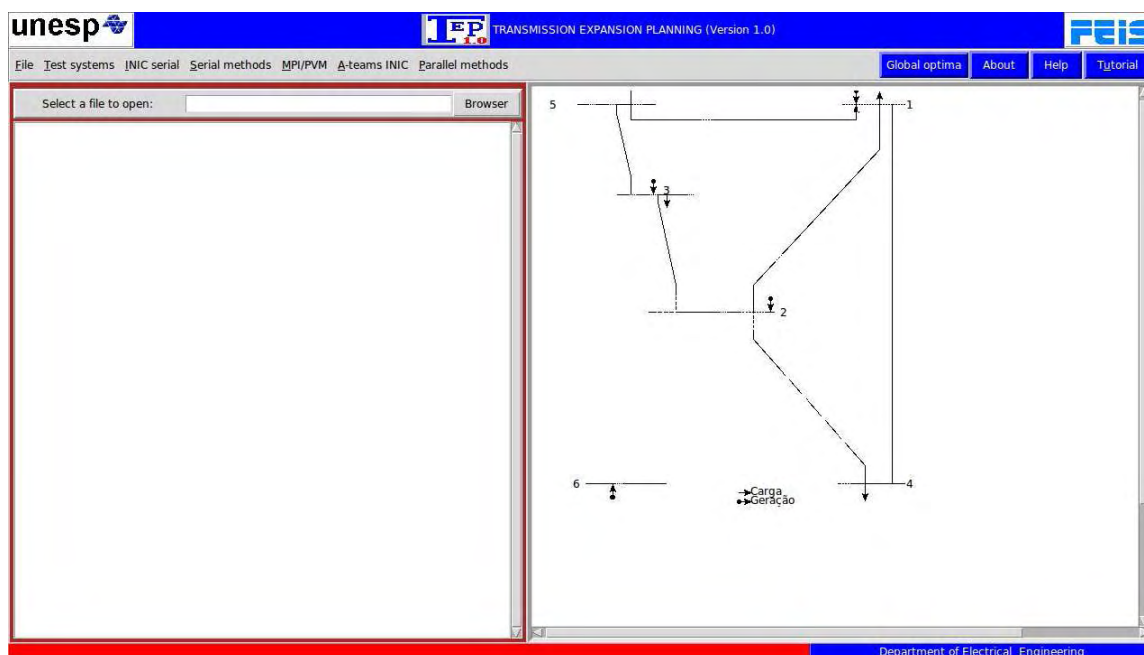


Figura 29: Manipulação de arquivos.

Em seguida o usuário deverá selecionar a pasta correspondente ao algoritmo utilizado na simulação, correspondente a Figura 30.

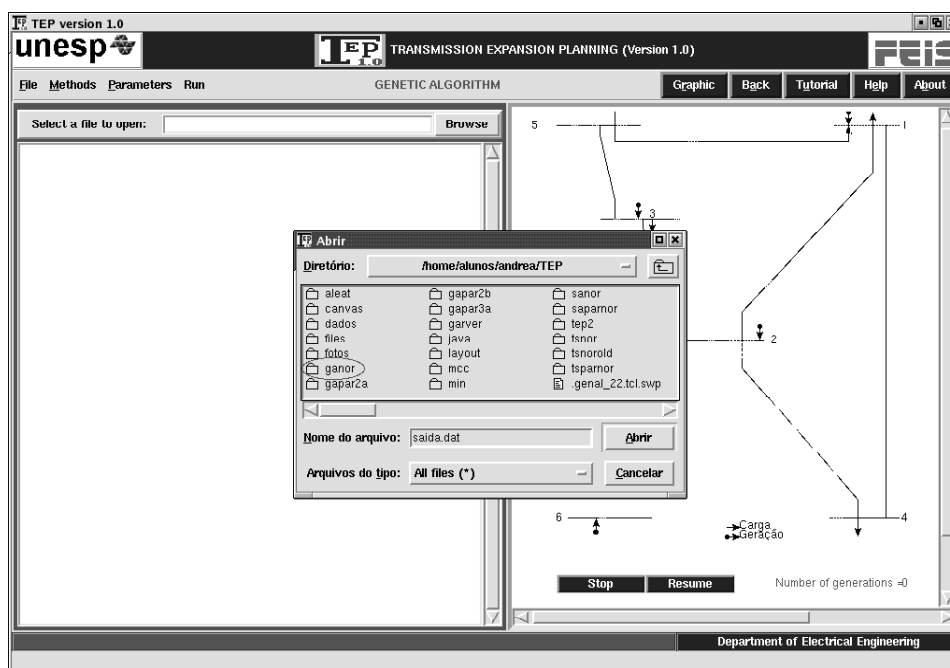


Figura 30: Manipulação de arquivos, diretório *ganor*.

E posteriormente selecionar o arquivo *saida.dat*, como mostrado na Figura 31.

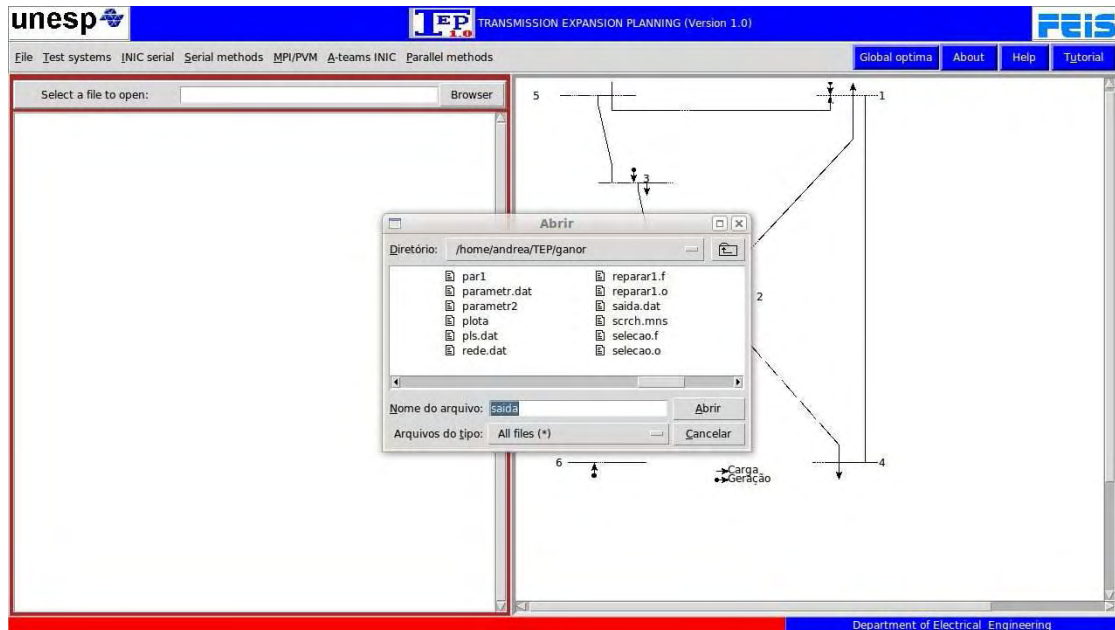


Figura 31: Manipulação de arquivos, arquivo *saida.dat*.

Na Figura 32 visualiza-se o *label* número de gerações dentro do *canvas*. Procedimento desenvolvido nesta versão da interface para que o usuário tenha esta informação no *canvas* e em tempo real de execução da simulação. Assim, o usuário terá maior interação com interface gráfica.

O usuário poderá também visualizar a informação com o arquivo *saída.dat* e acompanhando a informação no *label* dentro do *canvas* simultaneamente, como mostra a Figura 32.

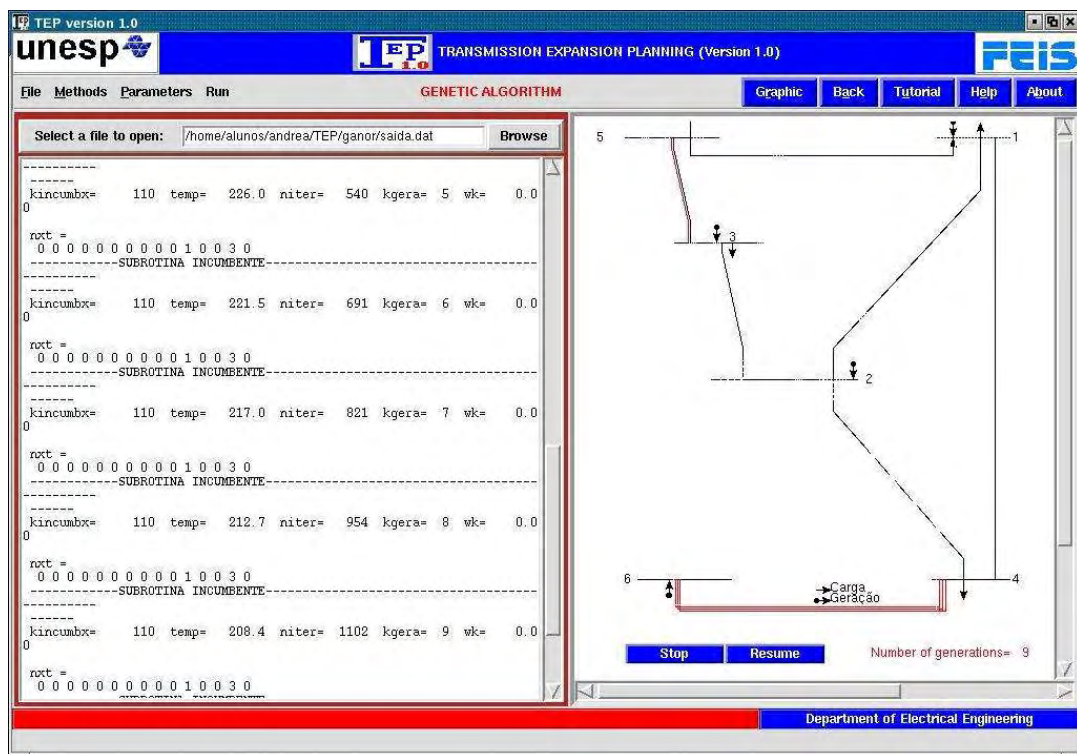


Figura 32: Telas de saída de texto e gráfica.

O botão *Stop* permite que o usuário pare a execução da simulação em um determinado instante, para poder analisar os resultados obtidos até o momento, através do arquivo saída.dat e comparar estes resultados com as configurações geradas na topologia gerada no *canvas*. E através do botão *Resume* é possível continuar a execução da simulação. Estes botões manipulam determinados processos executados pelo sistema operacional da máquina.

A estrutura de arquivos da TEP, que está em *background* também foi modificada, para permitir melhor manipulação de arquivos, além de evitar redundância e garantir melhor clareza para o programador.

5.2 EXEMPLO DE SIMULAÇÃO

A interface gráfica TEP permite ao usuário analisar o problema do planejamento da expansão da transmissão de energia elétrica de diferentes formas, dependendo de quatro critérios: do sistema teste, do método heurístico utilizado para geração de configurações iniciais, da meta-heurística e do ambiente paralelo.

Vários casos poderão ser criados com intuito de analisar o problema, podendo ou não envolver a computação paralela. Se o usuário não dispor dos recursos computacionais para a instalação das bibliotecas de processamento paralelo, a interface poderá ser executada de forma a não utilizar o recurso.

Um caso será simulado para melhor entendimento do funcionamento da ferramenta, mediante os seguintes passos:

Primeiramente o usuário deverá definir o sistema teste a ser utilizado dentre os sistemas testes disponíveis; neste caso será utilizado o sistema teste Garver, visualizado no lado direito da Figura 33.

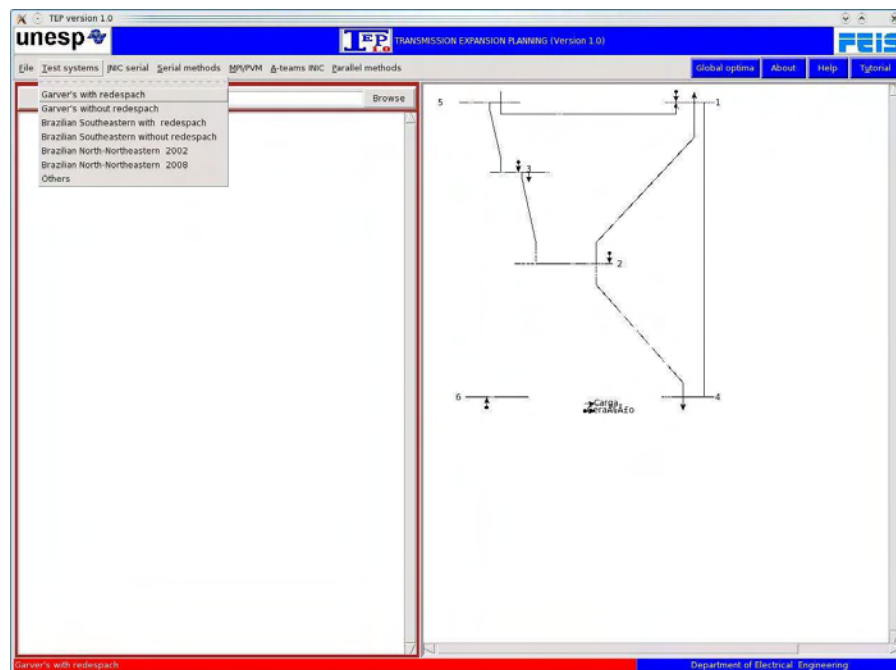


Figura 33: Escolha do sistema teste.

Depois de carregado o sistema teste o usuário deverá escolher o método heurístico inicializador das configurações iniciais. Nesta simulação, serão geradas as configurações iniciais através do método aleatório, como mostrado na Figura 8, na página 55.

Ainda nesta janela o usuário poderá visualizar as configurações geradas através de manipulação de arquivos, clicando no botão *Browser* e selecionando o arquivo que contém as configurações iniciais geradas (*mpop.dat*) e (*mbloco.dat*) conforme mostrado na Figura 34.

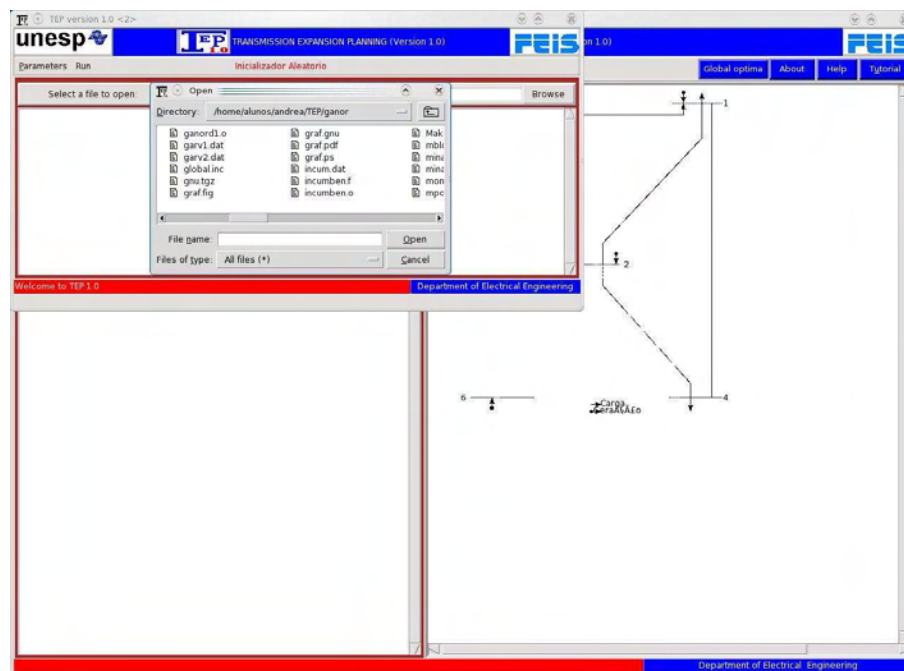


Figura 34: Manipulação dos arquivos *mpop.dat* e *mbloco.dat*.

Após serem geradas as configurações iniciais factíveis o usuário deverá escolher o método de resolução no menu *Serial methods* a ser utilizado na simulação, como mostrado na Figura 9, na página 56.

Na Figura 11, da página 57, vê-se que uma nova janela irá sobrepor a tela: o menu *Parameters*; nele o usuário poderá calibrar todos os parâmetros de calibração referente ao método escolhido, neste caso do Algoritmo Genético.

Após a calibração dos parâmetros o usuário deverá ir ao menu *Run* e acionar o botão *Ganor* para executar o programa. Durante essa execução as linhas adicionadas ou removidas são visualizadas no diagrama unifilar dentro da TEP.

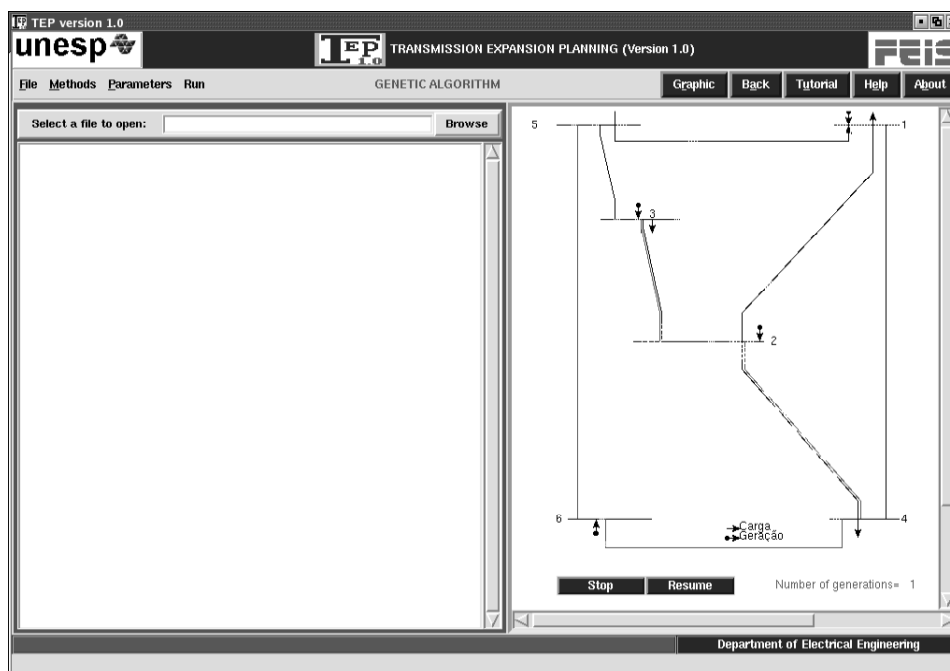
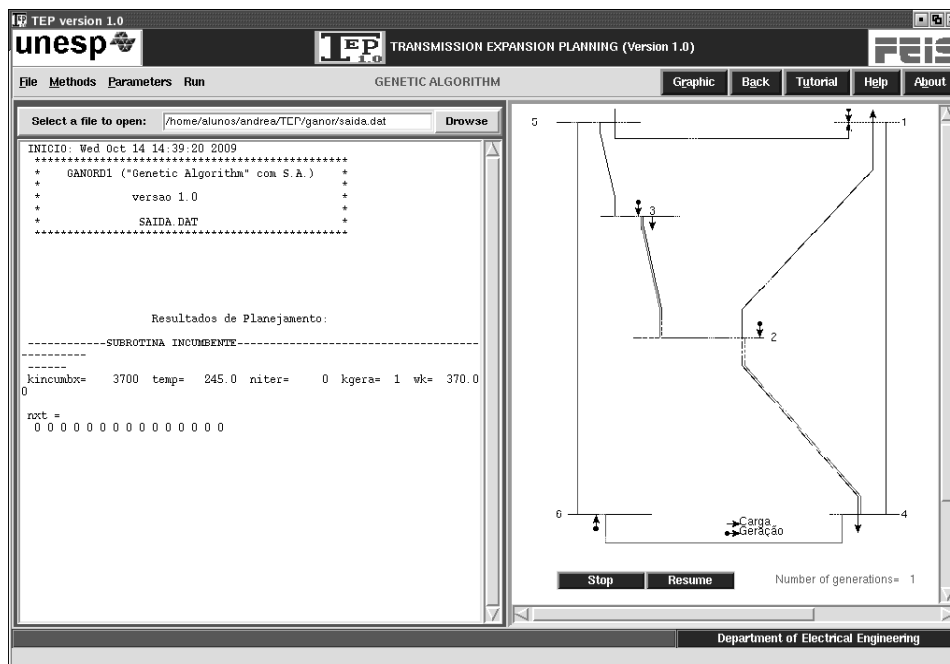


Figura 35: Linhas no diagrama unifilar.

Neste momento o usuário também pode acompanhar a adição/retirada das linhas através de arquivo texto. Clicando no botão *Browser* e selecionando o arquivo *saida.dat*, que apresenta todos os dados de saída do sistema.

Figura 36: Arquivo *saída.dat*.

Outro recurso presente na janela esquerda da interface é a manipulação do processo

que está sendo executado. O usuário pode em determinado momento parar a simulação e depois retornar, através dos botões *Stop/Resume*, como explicado anteriormente.

Pode-se observar que durante a realização da simulação a configuração ótima foi obtida na quinta geração, este dado poderá ser lido no arquivo texto, como mostrado na Figura 37, enquanto a simulação continua sendo executada até atingir o número máximo de gerações.

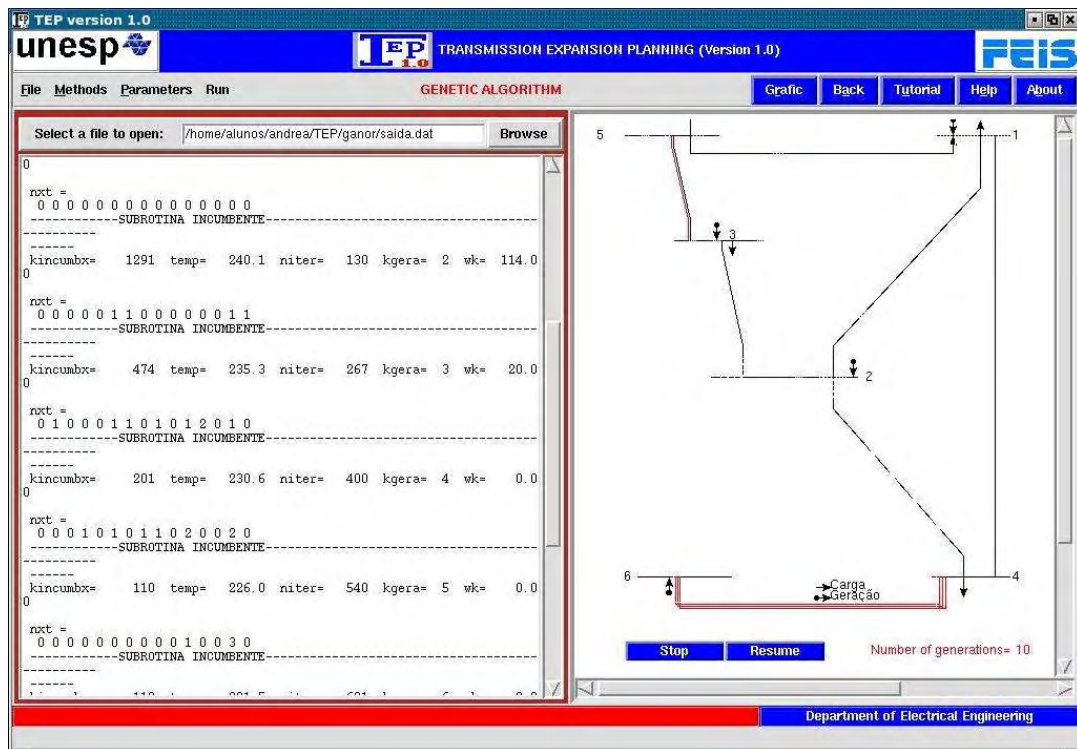


Figura 37: Solução ótima para o sistema Garver.

Resultados de outras simulações, com os sistemas testes: Sul brasileiro e Nordeste brasileiro, são apresentados nas Figuras 38 e 39, onde as linhas adicionadas aparecem no formato de números.

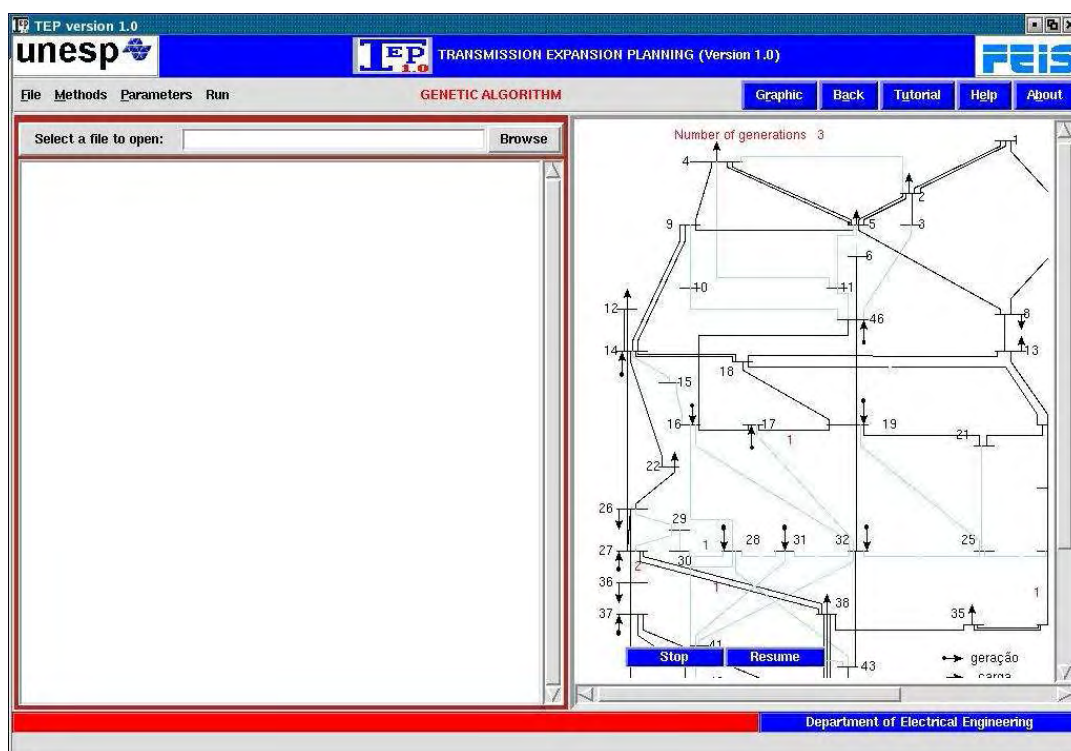


Figura 38: Simulação do sistema Sul brasileiro.

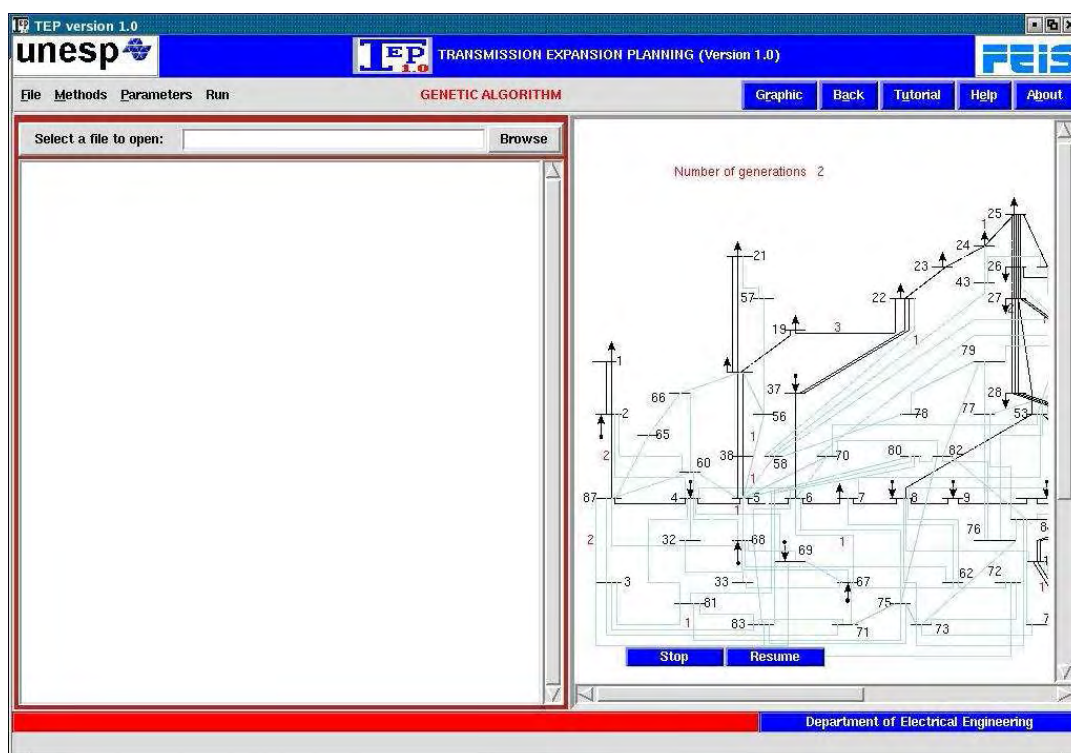


Figura 39: Simulação do sistema Norte-Nordeste brasileiro.

6 *CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS*

Considerando o estágio atual de desenvolvimento da interface gráfica TEP, a mesma já permite ao usuário ter uma visualização do comportamento dos algoritmos, simulados sobre os diferentes sistemas testes, muito melhor do que simplesmente a manipulação dos arquivos de dados e os arquivos de saída anteriormente disponíveis.

A inserção dos novos recursos ao programa, como os *canvas* com visualização das linhas ou dos números de linhas entre os ramos, o tutorial e outros, permite ao usuário um melhor entendimento de cada um dos algoritmos utilizados nas simulações, bem como da própria ferramenta computacional.

A interface gráfica em desenvolvimento requer a interação com os usuários, os quais podem auxiliar participando da fase de teste apontando suas dificuldades no uso da mesma. Assim, pode-se desenvolver um software que apresente maior probabilidade de aceitação no ambiente acadêmico. O software que tem como objetivo auxiliar o processo de aprendizagem do usuário, apresenta-se como uma ferramenta de fácil manuseio e propicia um ambiente favorável ao aprendizado. Ele deverá despertar no usuário o interesse em conhecer cada vez mais a ferramenta e o estimular na análise do problema, durante a realização das simulações. Buscou-se desenvolver o programa de forma que o mesmo se tornasse motivador ao usuário, o qual poderá realizar atividades prazerosas e atrativas.

O desenvolvimento da interface gráfica TEP é um processo evolutivo e constante, permitindo posteriormente a elaboração de várias *releases* melhoradas e cada vez mais eficientes. Com este propósito, algumas sugestões para a continuidade do trabalho são apresentadas a seguir, de forma resumida.

1. Implementar mecanismo que permita a visualização do arquivo *saída.dat* de forma *online*, este método certamente propiciará uma melhor visualização das configurações

geradas durante a simulação;

2. Desenvolver no código-fonte métodos capazes de desenhar de forma automática qualquer sistema teste novo que se pretenda analisar. Sendo esta uma tarefa muito difícil, mas não impossível;
3. Realizar a aprimoração do Tutorial, que poderá ser realizada implementando recursos audiovisuais;
4. Incrementar o programa com um time assíncrono de meta-heurística, além dos algoritmos paralelos, ou ainda incorporar outras meta-heurísticas, bem como algoritmos híbridos das meta-heurísticas utilizadas neste trabalho.

Concluí-se portanto que este trabalho é uma iniciativa de criação de recursos computacionais didáticos, para o uso no curso de engenharia elétrica, especificamente na área de sistema de potência. Este projeto também é um ambiente propício para o ensino dos conceitos de rede de computadores e computação paralela, que podem ser utilizados em universidades ou concessionárias de energia elétrica.

REFERÊNCIAS

- DE OLIVEIRA, S. A. *Metaheurísticas aplicadas ao planejamento da expansão da transmissão de energia elétrica em ambiente de processamento distribuído*. 2004. 315 f. Tese (Doutorado em Engenharia Elétrica) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2004.
- DEITEL, H. *C++: como programar*. Porto Alegre: Bookman, 2001. 656 p.
- GALLEGO, R. A. R. *Planejamento a longo prazo de sistemas de transmissão usando técnicas de otimização combinatorial*. 1997. 255 f. Tese (Doutorado em Engenharia Elétrica) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 1997.
- GARVER, L. Transmission network estimation using linear programming. *IEEE Transactions on Power Apparatus and Systems*, [s.l.], v. PAS-89, n. 7, p. 1688–1697, Sept./Oct. 1970.
- LIANG, T. *User interface design for decision support systems: a self-adaptive approach*. New York: Information & Management, 1987.
- MONTICELLI, A.; SANTOS JR., A.; PEREIRA, M. V. F.; CUNHA, S. H.; PRAÇA, J. C. G.; PARK, B. J. Interactive transmission network planning using a least-effort criterion. *IEEE Transactions on Power Apparatus and Systems*, [s.l.], v. PAS-101, n. 10, p. 3919–3925, Oct. 1982.
- MONTICELLI, A. J. *Fluxo de carga em redes de energia elétrica*. São Paulo: Edgar Blücher, 1983.
- MORAES, M. C. *Informática educativa no Brasil: um pouco de história*. Brasília: [s.n.], 1993.
- OUSTERHOUT, J. K. *Tcl and the Tk toolkit*. Massachusetts: Addison-Wesley, 1994. 460 p.
- PEREIRA, M. V. F.; PINTO, L. M. V. G.; OLIVEIRA, G. C.; CUNHA, S. H. F. A decomposition approach to automated generation/transmission expansion planning. *IEEE Transactions on Power Apparatus and Systems*, [s.l.], v. PAS-104, n. 11, p. 3074–3083, Nov. 1985.
- ROMERO, R. *Planejamento a longo prazo da expansão de sistemas de transmissão de energia elétrica*. 1999. Tese (Livre Docência) — Faculdade de Engenharia de Ilha Solteira, Universidade Estadual Paulista, Ilha Solteira, 1999.
- ROMERO, R. A.; ESCOBAR, A.; GALLEGOS, R. A. *Técnicas de optimización combinatorial*. Pereira: Universidad Tecnológica de Pereira, 2006.

TCLER'S WIKI. *Programadores Tcl/TK*. [S.l.], 2008. Disponível em: <<http://wiki.tcl.tk/>>. Acesso em: 12 nov. 2008.

TCL/TK. *Tcl Developer Xchange*. [S.l.], 2007. Disponível em: <www.tcl.tk>. Acesso em: 05 maio 2007.

TCL/TK BRASIL. *Tcl Brasil*. [S.l.], 2007. Disponível em: <www.tclbrasil.cipsga.org.br/tcltk.htm>. Acesso em: 05 maio 2006.

VILLASANA, R.; GARVER, L. L.; SALON, S. J. Transmission network planning using linear programming. *IEEE Transactions on Power Apparatus and Systems*, [s.l.], v. PAS-104, n. 2, p. 349–356, Feb. 1985.

ZACHARIA, V. L. C. *Centro de Referência Educacional*. [S.l.], 2007. Disponível em: <<http://www.centrorefeducacional.com.br/sofedu.html>>. Acesso em: 01 jun. 2009.

ZEM-MASCARENHAS, S. H.; CASSIANI, S. H. *Desenvolvimento e avaliação de um software educacional para o ensino de enfermagem pediátrica*. [S.l.], 2001. Disponível em: <www.revistausp.cipsga.org.br/tcltk.htm>. Acesso em: 01 jun. 2009.

APÊNDICE A – Instalação da interface gráfica TEP

Para a utilização da ferramenta TEP deverá ser preparado o ambiente de execução do programa, para isso é necessário a instalação prévia do compilador ou interpretador Tcl/Tk, de bibliotecas Fortran e bibliotecas de processamento paralelo. Posteriormente, os seguintes procedimentos devem ser realizados (dependendo do sistema operacional utilizado).

Para usuários do sistema operacional Linux, após a preparação do ambiente deve-se iniciar com a execução do CD de instalação do software, carregando todos arquivos necessários para a execução do TEP para o computador, ou seja copiar a pasta TEP para a sua máquina. Posteriormente deverá abrir o *console* e ir ao diretório TEP e em seguida chamar o interpretador *wish* com o nome do programa principal, conforme Figura 40.

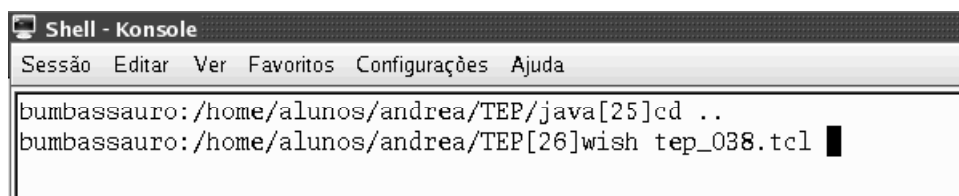


Figura 40: Interpretador *wish*.

Assim, será aberta a janela principal do TEP. Nesta, o usuário encontrará diversos recursos para realização das simulações.

No sistema operacional Windows, após a preparação do ambiente, dispor o CD de instalação do software na máquina, em seguida abrir o Internet Explorer e visualizar os arquivos do CD. Posteriormente copiar a pasta TEP para algum diretório dentro da sua máquina. No prompt do Windows digite o comando *wish* seguido com o nome do arquivo principal.

Posteriormente o usuário poderá usufruir dos recursos oferecidos pela interface gráfica.

APÊNDICE B – Diagrama Caso de Uso da TEP

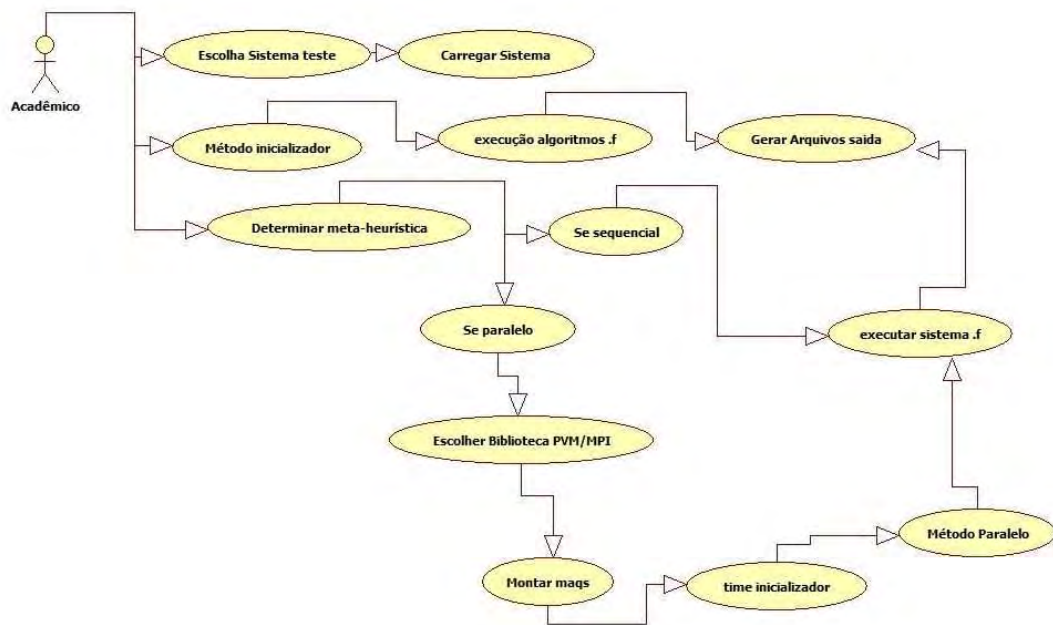


Figura 41: Diagrama Caso de Uso.

APÊNDICE C – Questionário de avaliação

O processo de avaliação da interface visa garantir a satisfação do usuário bem como identificar melhorias que possam ser realizadas em versões seguintes do programa.

Para a realização dessa avaliação é sugerido que usuários utilizem a ferramenta por um determinado período de tempo para que os mesmos sejam capazes de dar opiniões de melhorias, bem como detectar possíveis falhas, ou ainda relatar suas dificuldades de interagir com a interface.

O questionário foi elaborado de forma a avaliar os aspectos didáticos, a interação, a flexibilidade, o conteúdo, os recursos motivacionais, enfim todos critérios desejáveis em uma ferramenta educacional, abordada no capítulo 2.

As questões poderão ser as seguintes:

- 1.O usuário sabe a qualquer momento se localizar na sequência de interação ou na execução de uma tarefa? (Interação, aspectos didáticos e conteúdo)
- 2.A navegação no ambiente é fácil e rápida? (Interação)
- 3.O software fornece *feedback* imediato de todas as entradas de dados do usuário? Ou seja, no momento em que é executada determinada tarefa a interface avisa se o usuário está em situação de erro? (Interação)
- 4.O conteúdo da interface é significativa para o usuário? (Conteúdo)
- 5.A simulação permite visualizar eventos do mundo real? Eventos que não podem ser visualizados somente em sala de aula. (Correspondência com o mundo)
- 6.O vocabulário empregado no sistema lhe parece agradável ao seu entendimento? (Avaliação da interface)
- 7.A interface é adequada e pertinente em relação a uma disciplina específica? (Avaliação pedagógica)

Para que se consiga melhor análise do resultado dos questionários é aconselhável que o número de perguntas seja proporcional a cada item de avaliação, ou seja, se existir duas perguntas que analisem a interação, o número de perguntas deverá ser mantido para os outros critérios.

ANEXO A – Dados dos Sistemas Testes (DE OLIVEIRA, 2004)

A.1 Sistema Garver (6 Barras/15 Ramos)

Tabela 1: Dados de Barras - Garver

Barra	Capacidade de Geração (MW)	Geração Atual (MW)	Carga (MW)
1	150	50	80
2	0	0	240
3	360	165	40
4	0	0	160
5	0	0	240
6	600	545	0
Total	1.110	760	760

Tabela 2: Dados de Linhas - Garver

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo
1	1-2	1	0,40	100	40
2	1-3	0	0,38	100	38
3	1-4	1	0,60	80	60
4	1-5	1	0,20	100	20
5	1-6	0	0,68	70	68
6	2-3	1	0,20	100	20
7	2-4	1	0,40	100	40
8	2-5	0	0,31	100	31
9	2-6	0	0,30	100	30
10	3-4	0	0,59	82	59
11	3-5	1	0,20	100	20
12	3-6	0	0,48	100	48
13	4-5	0	0,63	75	63
14	4-6	0	0,30	100	30
15	5-6	0	0,61	78	61

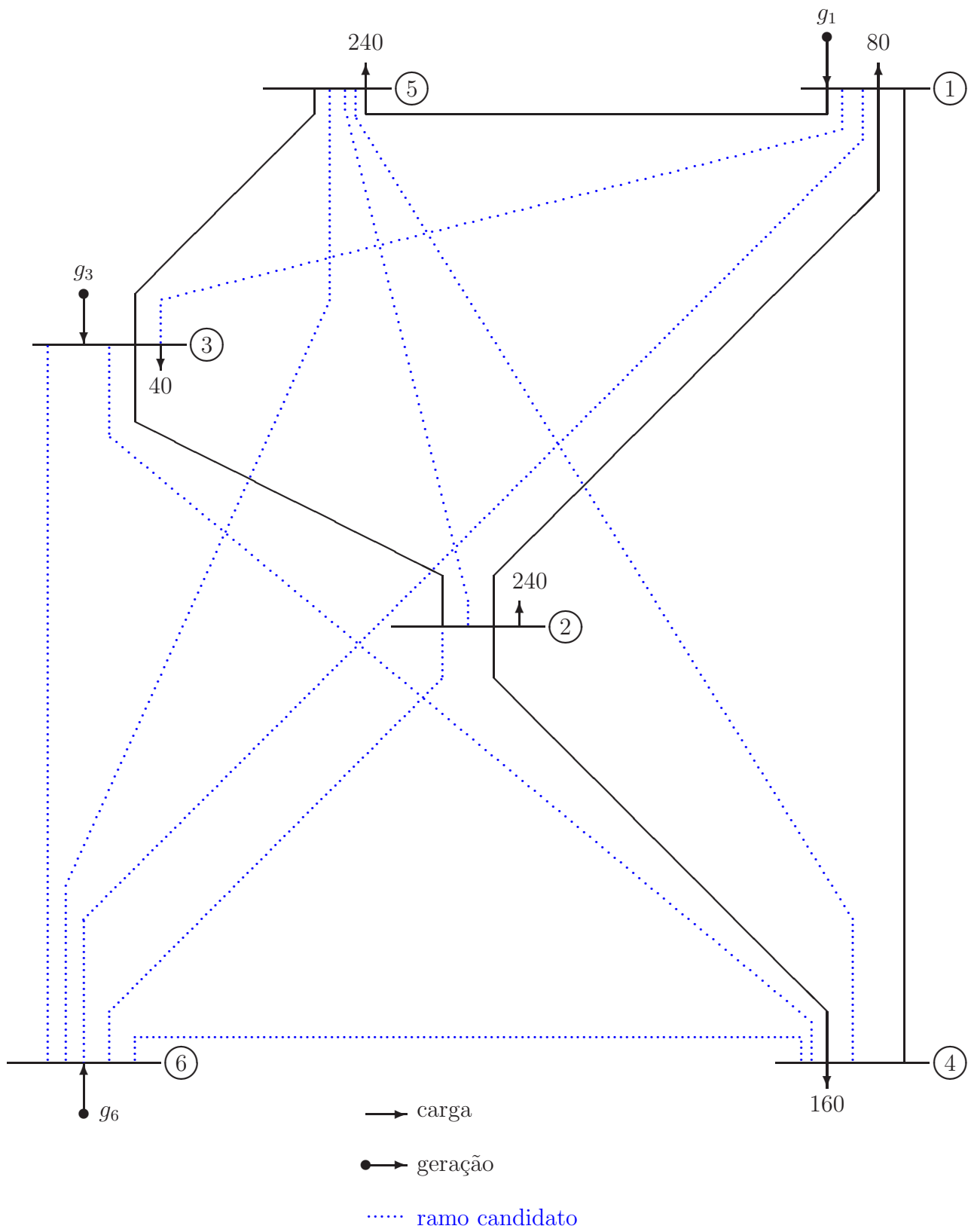


Figura 42: Sistema Garver - Configuração Inicial.

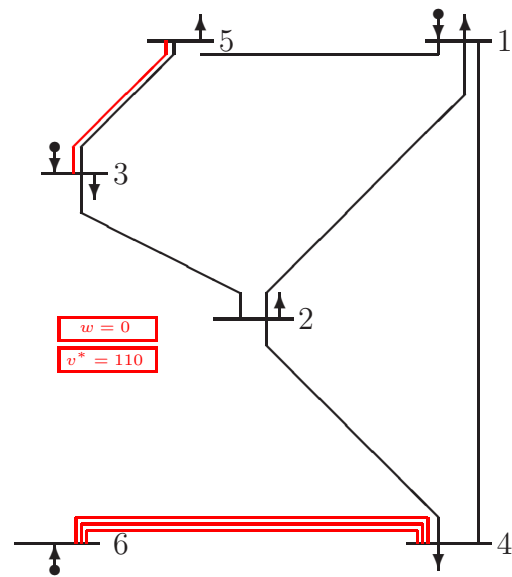


Figura 43: Sistema Garver - Configuração Ótima com Redespacho

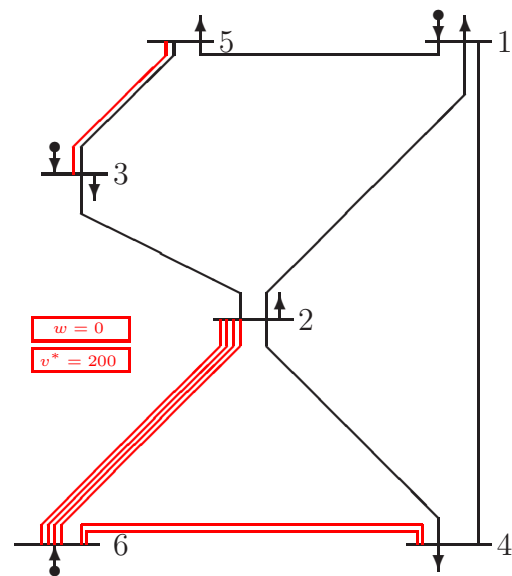


Figura 44: Sistema Garver - Configuração Ótima sem Redespacho

A.2 Sistema Sul Brasileiro (46 Barras/79 Ramos)

Tabela 3: Dados de Barras - Sul Brasileiro

Barra	Capacidade de Geração (MW)	Geração Atual (MW)	Carga (MW)
1	0,0	0,0	0,0
2	0,0	0,0	443,1
3	0,0	0,0	0,0
4	0,0	0,0	300,7
5	0,0	0,0	238,0
6	0,0	0,0	0,0
7	0,0	0,0	0,0
8	0,0	0,0	72,2
9	0,0	0,0	0,0
10	0,0	0,0	0,0
11	0,0	0,0	0,0
12	0,0	0,0	511,9
13	0,0	0,0	185,8
14	1.257,0	944,0	0,0
15	0,0	0,0	0,0
16	2.000,0	1.366,0	0,0
17	1.050,0	1.000,0	0,0
18	0,0	0,0	0,0
19	1.670,0	773,0	0,0
20	0,0	0,0	1.091,2
21	0,0	0,0	0,0
22	0,0	0,0	81,9
23	0,0	0,0	458,1
24	0,0	0,0	478,2
25	0,0	0,0	0,0
26	0,0	0,0	231,9
27	220,0	54,0	0,0
28	800,0	730,0	0,0
29	0,0	0,0	0,0
30	0,0	0,0	0,0
31	700,0	310,0	0,0
32	500,0	450,0	0,0
33	0,0	0,0	229,1
34	748,0	221,0	0,0
35	0,0	0,0	216,0
36	0,0	0,0	90,1
37	300,0	212,0	0,0
38	0,0	0,0	216,0
39	600,0	221,0	0,0

continua na próxima página

Tabela 3: Dados de Barras - Sul Brasileiro

Barra	Capacidade de Geração (MW)	Geração Atual (MW)	Carga (MW)
40	0.0	0.0	262.1
41	0.0	0.0	0.0
42	0,0	0,0	1.607,9
43	0,0	0,0	0,0
44	0,0	0,0	79,1
45	0,0	0,0	86,7
46	700,0	599,0	0,0
Total	10.545,0	6.880,0	6.880,0

Tabela 4: Dados de Linhas - Sul Brasileiro

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
1	01-07	1	0,0616	270	4,349.00
2	01-02	2	0,1065	270	7,076.00
3	04-09	1	0,0924	270	6,217.00
4	05-09	1	0,1173	270	7,732.00
5	05-08	1	0,1132	270	7,480.00
6	07-08	1	0,1023	270	6,823.00
7	04-05	2	0,0566	270	4,046.00
8	02-05	2	0,0324	270	2,581.00
9	08-13	1	0,1348	240	8,793.00
10	09-14	2	0,1756	220	11,267.00
11	12-14	2	0,0740	270	5,106.00
12	14-18	2	0,1514	240	9,803.00
13	13-18	1	0,1805	220	11,570.00
14	13-20	1	0,1073	270	7,126.00
15	18-20	1	0,1997	200	12,732.00
16	19-21	1	0,0278	1.500	32,632.00
17	16-17	1	0,0078	2.000	10,505.00
18	17-19	1	0,0061	2.000	8,715.00
19	14-26	1	0,1614	220	10,409.00
20	14-22	1	0,0840	270	5,712.00
21	22-26	1	0,0790	270	5,409.00
22	20-23	2	0,0932	270	6,268.00
23	23-24	2	0,0774	270	5,308.00
24	26-27	2	0,0832	270	5,662.00
25	24-34	1	0,1647	220	10,611.00
26	24-33	1	0,1448	240	9,399.00
27	33-34	1	0,1265	270	8,288.00
28	27-36	1	0,0915	270	6,167.00
29	27-38	2	0,2080	200	13,237.00

continua na próxima página

Tabela 4: Dados de Linhas - Sul Brasileiro (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
30	36-37	1	0,1057	270	7,025.00
31	34-35	2	0,0491	270	3,591.00
32	35-38	1	0,1980	200	12,631.00
33	37-39	1	0,0283	270	2,329.00
34	37-40	1	0,1281	270	8,389.00
35	37-42	1	0,2105	200	13,388.00
36	39-42	3	0,2030	200	12,934.00
37	40-42	1	0,0932	270	6,268.00
38	38-42	3	0,0907	270	6,116.00
39	32-43	1	0,0309	1.400	35,957.00
40	42-44	1	0,1206	270	7,934.00
41	44-45	1	0,1864	200	11,924.00
42	19-32	1	0,0195	1.800	23,423.00
43	46-19	1	0,0222	1.800	26,365.00
44	46-16	1	0,0203	1.800	24,319.00
45	18-19	1	0,0125	600	8,178.00
46	20-21	1	0,0125	600	8,178.00
47	42-43	1	0,0125	600	8,178.00
48	02-04	0	0,0882	270	5,965.00
49	14-15	0	0,0374	270	2,884.00
50	46-10	0	0,0081	2.000	10,889.00
51	04-11	0	0,2246	240	14,247.00
52	05-11	0	0,0915	270	6,167.00
53	46-06	0	0,0128	2.000	16,005.00
54	46-03	0	0,0203	1.800	24,319.00
55	16-28	0	0,0222	1.800	26,365.00
56	16-32	0	0,0311	1.400	36,213.00
57	17-32	0	0,0232	1.700	27,516.00
58	19-25	0	0,0325	1.400	37,748.00
59	21-25	0	0,0174	2.000	21,121.00
60	25-32	0	0,0319	1.400	37,109.00
61	31-32	0	0,0046	2.000	7,052.00
62	28-31	0	0,0053	2.000	7,819.00
63	28-30	0	0,0058	2.000	8,331.00
64	27-29	0	0,0998	270	6,672.00
65	26-29	0	0,0541	270	3,894.00
66	28-41	0	0,0339	1.300	39,283.00
67	28-43	0	0,0406	1200	46,701.00
68	31-41	0	0,0278	1.500	32,632.00
69	32-41	0	0,0309	1.400	35,957.00
70	41-43	0	0,0139	2.000	17,284.00
71	40-45	0	0,2205	180	13,994.00
72	15-16	0	0,0125	600	8,178.00

continua na próxima página

Tabela 4: Dados de Linhas - Sul Brasileiro (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
73	46-11	0	0,0125	600	8,178.00
74	24-25	0	0,0125	600	8,178.00
75	29-30	0	0,0125	600	8,178.00
76	40-41	0	0,0125	600	8,178.00
77	02-03	0	0,0125	600	8,178.00
78	05-06	0	0,0125	600	8,178.00
79	09-10	0	0,0125	600	8,178.00

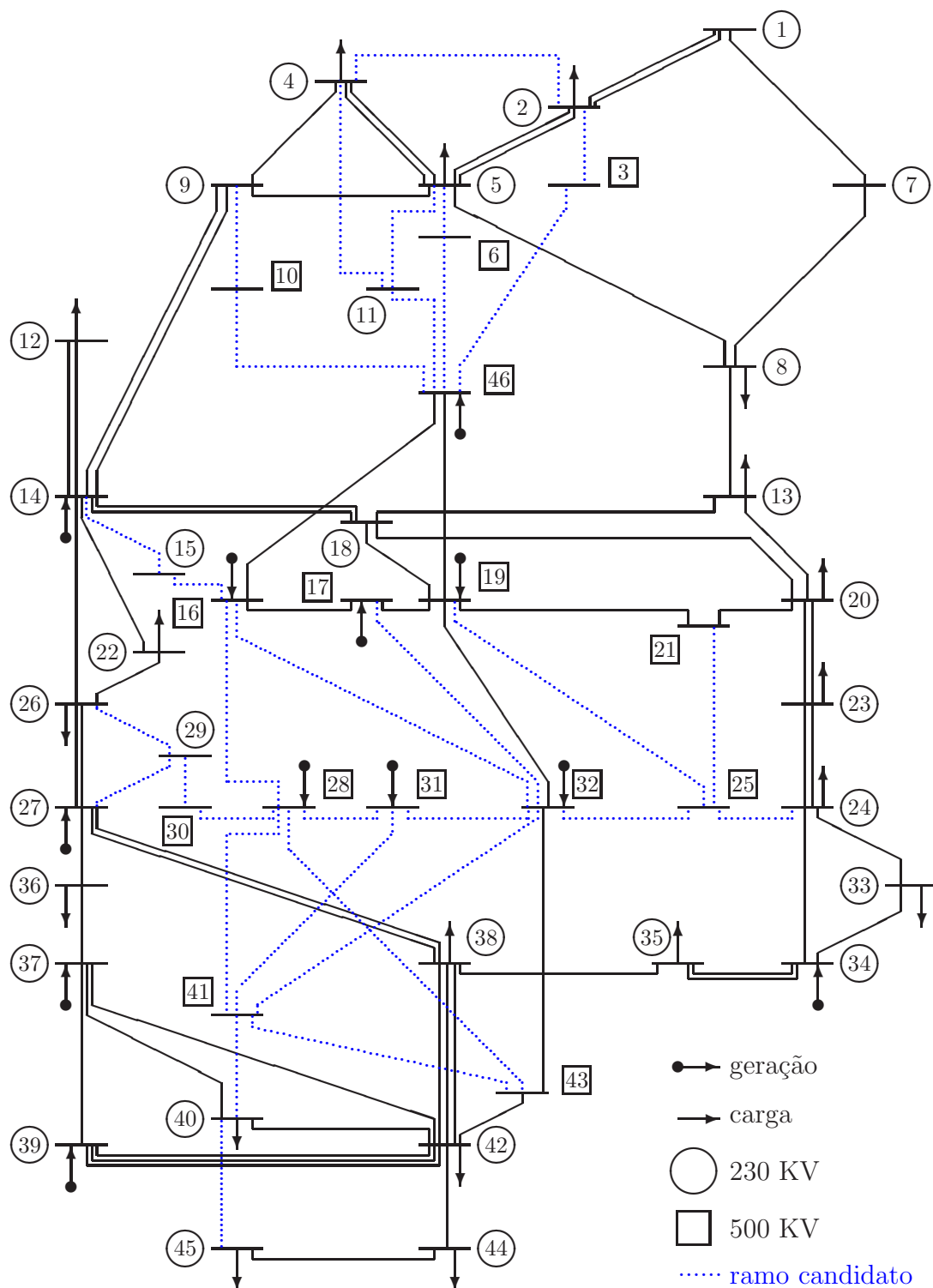


Figura 45: Sistema Sul brasileiro - Configuração Inicial.

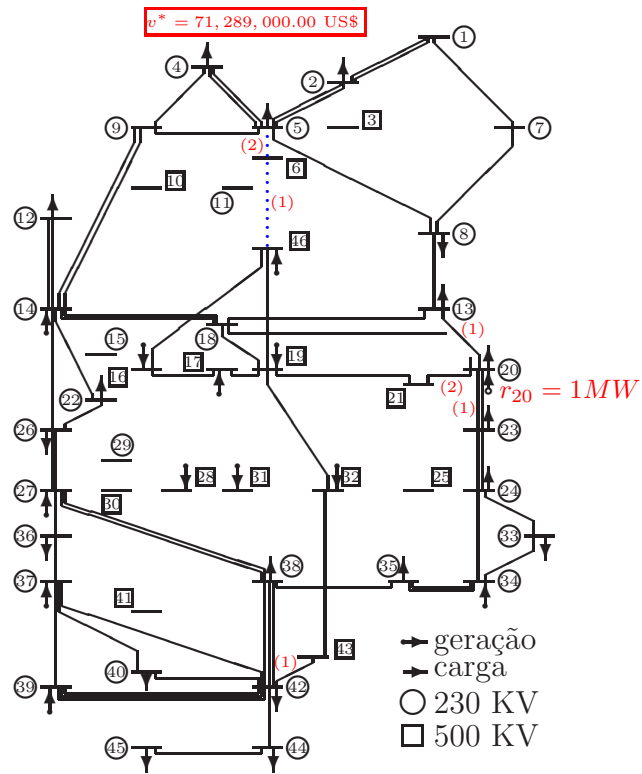


Figura 46: Sistema Sul Brasileiro - Config. Ótima com Redespacho.

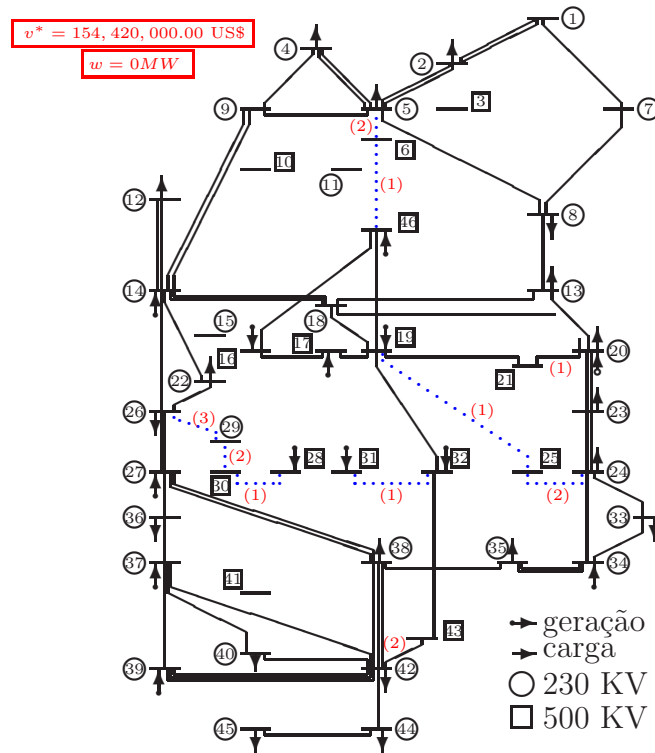


Figura 47: Sistema Sul Brasileiro - Config. Ótima sem Redespacho.

A.3 Sistema Norte-Nordeste Brasileiro (87 Barras/179 Ramos)

Tabela 5: Dados de Barras - Norte-Nordeste Brasileiro

Barra	Geração em 2002 (MW)	Carga em 2002 (MW)	Geração em 2008 (MW)	Carga em 2008 (MW)
1	0	1.857	0	2.747
2	4.048	0	4.550	0
3	0	0	0	0
4	517	0	6.422	0
5	0	0	0	0
6	0	0	0	0
7	0	31	0	31
8	403	0	82	0
9	465	0	465	0
10	538	0	538	0
11	2.200	0	2.260	0
12	2.257	0	4.312	0
13	4.510	0	5.900	0
14	542	0	542	0
15	0	0	0	0
16	0	0	0	0
17	0	0	0	0
18	0	0	0	0
19	0	86	0	125
20	0	125	0	181
21	0	722	0	1.044
22	0	291	0	446
23	0	58	0	84
24	0	159	0	230
25	0	1.502	0	2.273
26	0	47	0	68
27	0	378	0	546
28	0	189	0	273
29	0	47	0	68
30	0	189	0	273
31	0	110	0	225
32	0	0	0	0
33	0	0	0	0
34	0	28	0	107
35	1.635	0	1.531	0
36	0	225	0	325
37	169	0	114	0
38	0	0	0	0
39	0	186	0	269

continua na próxima página

Tabela 5: Dados de Barras - Norte-Nordeste Brasileiro (continuação)

Barra	Geração em 2002 (MW)	Carga em 2002 (MW)	Geração em 2008 (MW)	Carga em 2008 (MW)
40	0	1.201	0	1.738
41	0	520	0	752
42	0	341	0	494
43	0	0	0	0
44	0	4.022	0	5.819
45	0	0	0	0
46	0	205	0	297
47	0	0	0	0
48	0	347	0	432
49	0	777	0	1.124
50	0	5.189	0	7.628
51	0	290	0	420
52	0	707	0	1.024
53	0	0	0	0
54	0	0	0	0
55	0	0	0	0
56	0	0	0	0
57	0	0	0	0
58	0	0	0	0
59	0	0	0	0
60	0	0	0	0
61	0	0	0	0
62	0	0	0	0
63	0	0	0	0
64	0	0	0	0
65	0	0	0	0
66	0	0	0	0
67	1.242	0	1.242	0
68	888	0	888	0
69	902	0	902	0
70	0	0	0	0
71	0	0	0	0
72	0	0	0	0
73	0	0	0	0
74	0	0	0	0
75	0	0	0	0
76	0	0	0	0
77	0	0	0	0
78	0	0	0	0
79	0	0	0	0
80	0	0	0	0
81	0	0	0	0
82	0	0	0	0

continua na próxima página

Tabela 5: Dados de Barras - Norte-Nordeste Brasileiro (continuação)

Barra	Geração em 2002 (MW)	Carga em 2002 (MW)	Geração em 2008 (MW)	Carga em 2008 (MW)
83	0	0	0	0
84	0	0	0	0
85	0	487	0	705
86	0	0	0	0
87	0	0	0	0
Total	20.316	20.316	29.748	29.748

Tabela 6: Dados de Linhas - Norte-Nordeste Brasileiro

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
1	01-02	2	0,0374	1.000	44,056.00
2	02-04	0	0,0406	1.000	48,880.00
3	02-60	0	0,0435	1.000	52,230.00
4	02-87	1	0,0259	1.000	31,192.00
5	03-71	0	0,0078	3.200	92,253.00
6	03-81	0	0,0049	3.200	60,153.00
7	03-83	0	0,0043	3.200	53,253.00
8	03-87	0	0,0058	1.200	21,232.00
9	04-05	1	0,0435	1.000	52,230.00
10	04-06	0	0,0487	1.000	58,260.00
11	04-32	0	0,0233	300	7,510.00
12	04-60	0	0,0215	1.000	26,770.00
13	04-68	0	0,0070	1.000	10,020.00
14	04-69	0	0,0162	1.000	20,740.00
15	04-81	0	0,0058	1.200	21,232.00
16	04-87	1	0,0218	1.000	26,502.00
17	05-06	1	0,0241	1.000	29,852.00
18	05-38	2	0,0117	600	8,926.00
19	05-56	0	0,0235	1.000	29,182.00
20	05-58	0	0,0220	1.000	27,440.00
21	05-60	0	0,0261	1.000	32,130.00
22	05-68	0	0,0406	1.000	48,880.00
23	05-70	0	0,0464	1.000	55,580.00
24	05-80	0	0,0058	1.200	21,232.00
25	06-07	1	0,0288	1.000	35,212.00
26	06-37	1	0,0233	300	7,510.00
27	06-67	0	0,0464	1.000	55,580.00
28	06-68	0	0,0476	1.000	5,6920.00
29	06-70	0	0,0371	1.000	44,860.00
30	06-75	0	0,0058	1.200	21232.00
31	07-08	1	0,0234	1.000	29,048.00

continua na próxima página

Tabela 6: Dados de Linhas - Norte-Nordeste Brasileiro (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
32	07-53	0	0,0452	1.000	54,240.00
33	07-62	0	0,0255	1.000	31,460.00
34	08-09	1	0,0186	1.000	23,420.00
35	08-12	0	0,0394	1.000	47,540.00
36	08-17	0	0,0447	1.000	53,570.00
37	08-53	1	0,0365	1.200	44,190.00
38	08-62	0	0,0429	1.000	51,560.00
39	08-73	0	0,0058	1.200	21,232.00
40	09-10	1	0,0046	1.000	7,340.00
41	10-11	1	0,0133	1.000	17,390.00
42	11-12	1	0,0041	1.200	6,670.00
43	11-15	1	0,0297	1.200	36,284.00
44	11-17	1	0,0286	1.200	35,078.00
45	11-53	1	0,0254	1.000	31,326.00
46	12-13	1	0,0046	1.200	7,340.00
47	12-15	1	0,0256	1.200	31,594.00
48	12-17	1	0,0246	1.200	30,388.00
49	12-35	2	0,0117	600	8,926.00
50	12-84	0	0,0058	1.200	21,232.00
51	13-14	0	0,0075	1.200	10,690.00
52	13-15	0	0,0215	1.200	26,770.00
53	13-17	0	0,0232	1.200	28,780.00
54	13-45	1	0,0290	1.200	35,480.00
55	13-59	1	0,0232	1.200	28,780.00
56	14-17	0	0,0232	1.200	28,780.00
57	14-45	0	0,0232	1.200	28,780.00
58	14-59	0	0,0157	1.200	20,070.00
59	15-16	2	0,0197	1.200	24,760.00
60	15-45	0	0,0103	1.200	13,906.00
61	15-46	1	0,0117	600	8,926.00
62	15-53	0	0,0423	1.000	50,890.00
63	16-44	4	0,0117	600	8,926.00
64	16-45	0	0,0220	1.200	27,440.00
65	16-61	0	0,0128	1.000	16,720.00
66	16-77	0	0,0058	1.200	21,232.00
67	17-18	2	0,0170	1.200	21,678.00
68	17-59	0	0,0170	1200	21,678.00
69	18-50	4	0,0117	600	8,926.00
70	18-59	1	0,0331	1.200	40,170.00
71	18-74	0	0,0058	1.200	21,232.00
72	19-20	1	0,0934	170	5,885.00
73	19-22	1	0,1877	170	11,165.00
74	20-21	1	0,0715	300	6,960.00

continua na próxima página

Tabela 6: Dados de Linhas - Norte-Nordeste Brasileiro (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
75	20-38	2	0,1382	300	12,840.00
76	20-56	0	0,0117	600	8,926.00
77	20-66	0	0,2064	170	12,210.00
78	21-57	0	0,0117	600	8,926.00
79	22-23	1	0,1514	170	9,130.00
80	22-37	2	0,2015	170	11,935.00
81	22-58	0	0,0233	300	7,510.00
82	23-24	1	0,1651	170	9,900.00
83	24-25	1	0,2153	170	12,705.00
84	24-43	0	0,0233	300	7,510.00
85	25-26	2	0,1073	300	29,636.00
86	25-55	0	0,0117	600	8,926.00
87	26-27	2	0,1404	300	25,500.00
88	26-29	1	0,1081	170	6,710.00
89	26-54	0	0,0117	600	8,926.00
90	27-28	3	0,0826	170	5,335.00
91	27-35	2	0,1367	300	25,000.00
92	27-53	1	0,0117	600	8,926.00
93	28-35	3	0,1671	170	9,900.00
94	29-30	1	0,0688	170	4,510.00
95	30-31	1	0,0639	170	4,235.00
96	30-63	0	0,0233	300	7510.00
97	31-34	1	0,1406	170	8,525.00
98	32-33	0	0,1966	170	11,660.00
99	33-67	0	0,0233	300	7,510.00
100	34-39	2	0,1160	170	7,150.00
101	34-41	2	0,0993	170	6,215.00
102	35-46	4	0,2172	170	12,705.00
103	35-47	2	0,1327	170	8,085.00
104	35-51	3	0,1602	170	9,625.00
105	36-39	2	0,1189	170	7,315.00
106	36-46	2	0,0639	170	4,235.00
107	39-42	1	0,0973	170	6,105.00
108	39-86	0	0,0233	300	7,510.00
109	40-45	1	0,0117	600	8,926.00
110	40-46	3	0,0875	170	5,500.00
111	41-64	0	0,0233	300	7,510.00
112	42-44	2	0,0698	170	4,565.00
113	42-85	2	0,0501	170	3,465.00
114	43-55	0	0,0254	1.000	31,326.00
115	43-58	0	0,0313	1.000	38,160.00
116	44-46	3	0,1671	170	10,010.00
117	47-48	2	0,1966	170	11,660.00

continua na próxima página

Tabela 6: Dados de Linhas - Norte-Nordeste Brasileiro (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
118	48-49	1	0,0757	170	4,895.00
119	48-50	2	0,0256	170	2,090.00
120	48-51	2	0,2163	170	12,760.00
121	49-50	1	0,0835	170	5,335.00
122	51-52	2	0,0560	170	3,795.00
123	52-59	1	0,0117	600	8,926.00
124	53-54	0	0,0270	1.000	32,120.00
125	53-70	0	0,0371	1.000	44,860.00
126	53-76	0	0,0058	1.200	21,232.00
127	53-86	0	0,0389	1000	46,870.00
128	54-55	0	0,0206	1.000	25,028.00
129	54-58	0	0,0510	1000	60,940.00
130	54-63	0	0,0203	1.000	25,430.00
131	54-70	0	0,0360	1000	43,520.00
132	54-79	0	0,0058	1.200	21,232.00
133	56-57	0	0,0122	1000	16,050.00
134	58-78	0	0,0058	1.200	21,232.00
135	60-66	0	0,0233	300	7,510.00
136	60-87	0	0,0377	1.000	45,530.00
137	61-64	0	0,0186	1.000	23,420.00
138	61-85	0	0,0233	300	7,510.00
139	61-86	0	0,0139	1.000	18,060.00
140	62-67	0	0,0464	1.000	55,580.00
141	62-68	0	0,0557	1.000	66,300.00
142	62-72	0	0,0058	1.200	21,232.00
143	63-64	0	0,0290	1.000	35,480.00
144	65-66	0	0,3146	170	18,260.00
145	65-87	0	0,0233	300	7,510.00
146	67-68	0	0,0290	1.000	35,480.00
147	67-69	0	0,0209	1.000	26,100.00
148	67-71	0	0,0058	1.200	21,232.00
149	68-69	0	0,0139	1.000	18,060.00
150	68-83	0	0,0058	1.200	21,232.00
151	68-87	0	0,0186	1.000	23,240.00
152	69-87	0	0,0139	1.000	18,060.00
153	70-82	0	0,0058	1.200	21,232.00
154	71-72	0	0,0108	3.200	125,253.00
155	71-75	0	0,0108	3.200	125,253.00
156	71-83	0	0,0067	3.200	80,253.00
157	72-73	0	0,0100	3.200	116,253.00
158	72-83	0	0,0130	3.200	149,253.00
159	73-74	0	0,0130	3.200	149,253.00
160	73-75	0	0,0130	3.200	149,253.00

continua na próxima página

Tabela 6: Dados de Linhas - Norte-Nordeste Brasileiro (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
161	73-84	0	0,0092	3.200	107,253.00
162	74-84	0	0,0108	3.200	125,253.00
163	75-76	0	0,0162	3.200	185,253.00
164	75-81	0	0,0113	3.200	131,253.00
165	75-82	0	0,0086	3.200	101,253.00
166	75-83	0	0,0111	3.200	128,253.00
167	76-77	0	0,0130	3.200	149,253.00
168	76-82	0	0,0086	3.200	101,253.00
169	76-84	0	0,0059	3.200	70,953.00
170	77-79	0	0,0151	3.200	173,253.00
171	77-84	0	0,0115	3.200	132,753.00
172	78-79	0	0,0119	3.200	137,253.00
173	78-80	0	0,0051	3.200	62,253.00
174	79-82	0	0,0084	3.200	98,253.00
175	80-81	0	0,0101	3.200	117,753.00
176	80-82	0	0,0108	3.200	125,253.00
177	80-83	0	0,0094	3.200	110,253.00
178	81-83	0	0,0016	3.200	23,253.00
179	82-84	0	0,0135	3.200	155,253.00

FICHA DE IDENTIFICAÇÃO

Classificação de Segurança Livre		Documento no.	
Data (mês e ano) novembro/2009		Projeto no.	
Título e subtítulo Interface gráfica para o planejamento da expansão da transmissão de energia elétrica		No. do volume Único	
		No. da parte	
Título do projeto			
Entidade Executora (autor coletivo)		Autor(es) Andréa Barboza Proto	
Entidade patrocinada (cliente ou destinatário principal) PPGEE - UNESP Campus de Ilha Solteira			
Resumo Neste trabalho propõe-se o desenvolvimento de uma ferramenta educacional voltada para a resolução do problema de planejamento da expansão das linhas de transmissão, que utiliza-se de programas para a obtenção das soluções ótimas ou de boas soluções. A grande dificuldade encontrada por usuários ao interagir com estes programas, motivou o desenvolvimento de uma aplicação com interface gráfica, a qual disponibiliza ao usuário diversas metodologias para a resolução do problema do planejamento e expansão das linhas de transmissão de energia elétrica. Espera-se que software como este, agregado ao sistema de ensino tradicional que normalmente é utilizado num curso de engenharia elétrica, possa fortalecer o processo de aprendizagem do aluno. Assim, o software educacional Transmission Expansion Planning (TEP 1.0) está sendo desenvolvido visando propiciar um ambiente agradável para a realização de simulações e testes e favorecer a aprendizagem dos conceitos que envolvem o problema do planejamento da expansão da transmissão. É possível através deste software fazer simulações para os sistemas: Garver (6 barras/15 ramos), Sul brasileiro (46 barras/79 ramos) e Norte-Nordeste brasileiro (87 barras/179 ramos). O software se beneficia de recursos oferecidos por programas que são executados em <i>background</i>, bem como da utilização de meta-heurísticas e do ambiente de processamento de máquinas paralelas virtuais, as quais podem ser selecionadas para realização dos testes em determinado sistema.			
Palavras-chave Planejamento da Expansão da Transmissão. Heurísticas Construtivas. Meta-heurísticas. Software Educacional. Tcl/Tk.			
No. da edição	No. de páginas 102	ISSN (para relatórios publicados)	Classificação (CDC ou CDD)
Distribuidor		No. de exemplares 5 (cinco)	Preço -
Observações Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, UNESP - Campus de Ilha Solteira, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.			