

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

**Times Assíncronos Inicializadores para o  
Planejamento da Expansão da Transmissão de  
Energia Elétrica Baseados no Modelo Híbrido Linear**

**Fernando Rodrigo Lopes Sanchez**

**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, 06 de junho de 2008

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.



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

## CERTIFICADO DE APROVAÇÃO

**TÍTULO:** Times Assíncronos Inicializadores para o Planejamento da Expansão da Transmissão de Energia Elétrica Baseados no Modelo Híbrido Linear

**AUTOR:** FERNANDO RODRIGO LOPES SANCHEZ

**ORIENTADOR:** Prof. Dr. SERGIO AZEVEDO DE OLIVEIRA

Aprovado como parte das exigências para obtenção do Título de MESTRE em ENGENHARIA ELÉTRICA 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. EDUARDO NOBUHIRO ASADA  
Departamento de Engenharia Elétrica / Escola de Engenharia de São Carlos-Usp

Data da realização: 06 de junho de 2008.



Presidente da Comissão Examinadora  
Prof. Dr. SERGIO AZEVEDO DE OLIVEIRA

FICHA CATALOGRÁFICA

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

S211t

Sanchez, Fernando Rodrigo Lopes.

Times assíncronos inicializadores para o planejamento da expansão da transmissão de energia elétrica baseados no modelo híbrido linear / Fernando Rodrigo Lopes Sanchez.

Ilha Solteira : [s.n.], 2008

96 f. : il.

Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Automação, 2008

Orientador: Sérgio Azevedo de Oliveira

Bibliografia: p. 69-72

1. Times assíncronos. 2. Algoritmos heurísticos construtivos. 3. Modelo híbrido linear. 4. Planejamento da expansão da transmissão. 5. Metaheurísticas. 6. Processamento paralelo (Computadores).

# *Agradecimentos*

A Deus que nos concede o direito à vida.

Aos meus pais, Pedro e Marly, pela paciência, apoio emocional e financeiro.

À minha irmã Priscila pelo suprimento técnico.

Ao Professor Dr. Sérgio Azevedo de Oliveira, por ser sempre um orientador e amigo.

À Cy, companheira de todas as horas, amante, amiga e exemplo.

Aos professores do DEE, especialmente ao professor Rubén, Shinoda, Dilson, Marcelo, Mantovani e ao professor José Marcos do departamento de matemática.

Às secretárias do DEE e PPGEE, pelo constante apoio nas partes burocráticas do processo.

Aos técnicos e amigos do laboratório de informática, Deoclécio e Beto, pelas trocas de informações.

Aos amigos Edgar e Mara que dedicaram parte do seu tempo contribuindo diretamente para a realização deste trabalho.

Aos amigos do DEE: Cuiabá, Apio, Andrea, Elisângela, Rosane, Silvia, Carlos, Shigeo e Hélio.

Agradeço a turma: Chicão, Riga, Costa, Lumbriga, Dalazoana, Pirulito, Quimba, Jorjão, Edilton, Nilton, Mofo, Togeder, Di, Dai, Menossi, Carpete e todos os outros que fizeram da permanência em Ilha Solteira algo agradável e que ficará para sempre guardado na memória.

À família de onde venho  
e à família que estou começando...

SANCHEZ, Fernando Rodrigo Lopes. **Times Assíncronos Inicializadores para o Planejamento da Expansão da Transmissão de Energia Elétrica Baseados no Modelo Híbrido Linear**. 2008. 96 f. Dissertação (Mestrado em Engenharia Elétrica - Automação) - Faculdade de Engenharia, Universidade Estadual Paulista, Ilha Solteira, 2008.

## *Resumo*

Neste trabalho foram implementados diversos agentes heurísticos construtivos, baseados no modelo híbrido linear, que fazem parte de um time assíncrono que tem como objetivo gerar configurações de boa qualidade para inicializar as metaheurísticas que resolvem o problema do planejamento da expansão da transmissão dos sistemas de energia elétrica. A teoria de times assíncronos foi aplicada para reunir as qualidades individuais dos métodos heurísticos, de uma maneira que, partindo de uma configuração base (sem adições) e utilizando um fluxo de dados cíclico, os agentes construtivos adicionassem circuitos a esta configuração de maneira sistemática e aleatória até que esta atenda as demandas de carga solicitadas pelo sistema elétrico em um horizonte futuro. Estas configurações foram então utilizadas por um algoritmo genético no intuito de validar a qualidade das mesmas. Os algoritmos foram implementados em Fortran, utilizando as rotinas de trocas de mensagens do LAM-MPI e simulados para sistemas teste de pequeno, médio e grande porte em ambiente de processamento distribuído. Os resultados comprovam que os times assíncronos de vários métodos heurísticos são mais eficazes comparados com uma única heurística.

**Palavras chave:** Times Assíncronos, Algoritmos Heurísticos Construtivos, Modelo Híbrido Linear, Planejamento da Expansão da Transmissão, Metaheurísticas, Processamento Paralelo.

# *Abstract*

In this study, it was implemented several constructive heuristic algorithms, based on hybrid linear model, which are part of an asynchronous team that aims to generate initial solutions with good quality for meta-heuristics that solve the transmission expansion planning problem of electric power systems. The theory of asynchronous team was applied to meet the individual qualities of each heuristic method, in a way that, starting from a base network configuration and using a cyclical flow of data, heuristic agents add circuits to its configuration in a systematic and random way until they meet the load demands requested by the electrical system on a future horizon. Then these configurations are utilized by a genetic algorithm in order to validate the quality of them. The algorithms were implemented in Fortran, using exchanging messages routines from LAM-MPI and simulated for small, medium and large size test-systems in distributed processing environment. The results show that the solutions obtained with asynchronous teams of several heuristic methods are more effective than the solutions with a single heuristic algorithm.

**Keywords:** Asynchronous Team, Constructive Heuristic Algorithms, Hybrid Linear Model, Transmission Expansion Planning, Meta-heuristics, Distributed Processing.



## *Lista de Figuras*

1	Time assíncrono - topologia básica. . . . .	p. 34
2	Memória central. . . . .	p. 41
3	Agente construtor. . . . .	p. 42
4	Exemplo de escalabilidade do time assíncrono com agente <i>vgsB</i> . . . . .	p. 51
5	Sistema Garver - Configuração Inicial. . . . .	p. 79
6	Sistema Garver - Configuração Ótima com Redespacho . . . . .	p. 80
7	Sistema Garver - Configuração Ótima sem Redespacho . . . . .	p. 80
8	Sistema Sul brasileiro - Configuração Inicial. . . . .	p. 81
9	Sistema Sul Brasileiro - Config. Ótima com Redespacho. . . . .	p. 82
10	Sistema Sul Brasileiro - Config. Ótima sem Redespacho. . . . .	p. 82

## *Lista de Tabelas*

3	Tempo total para gerar configurações iniciais. . . . .	p. 51
4	Sistema Garver - 8 configurações iniciais zeradas. . . . .	p. 52
5	Sistema Garver - 8 configurações iniciais geradas com 6 agentes mcchB.	p. 53
6	Sistema Garver - 8 configurações iniciais geradas com 6 agentes minhB.	p. 53
7	Sistema Garver - 8 configurações iniciais zeradas. . . . .	p. 54
8	Sistema Sul brasileiro - 40 configurações iniciais zeradas. . . . .	p. 55
9	Sistema Sul brasileiro - 40 configurações iniciais geradas com 6 agentes mcchB. . . . .	p. 55
10	Sistema Sul brasileiro - 40 configurações iniciais geradas com todos os agentes. . . . .	p. 56
11	Sistema Sul brasileiro - 40 configurações iniciais zeradas. . . . .	p. 56
12	Sistema Sul brasileiro - 40 configurações iniciais geradas com todos os agentes. . . . .	p. 57
13	Sistema Norte Nordeste (Plano 2002) - 180 configurações iniciais zeradas.	p. 58
14	Sistema Norte Nordeste (Plano 2002) - 180 configurações iniciais geradas com todos os agentes. . . . .	p. 59
15	Sistema Norte Nordeste (Plano 2008) - 180 configurações iniciais com 6 agentes vgsB. . . . .	p. 60
16	Sistema Norte Nordeste (Plano 2008) - 180 configurações iniciais geradas com todos os agentes. . . . .	p. 61
17	Resultados do GANORD com configurações iniciais zeradas. . . . .	p. 62
18	Resultados do GANORD - configurações iniciais com 6 agentes mcchB.	p. 62
19	Resultados do GANORD - configurações iniciais com 6 agentes minhB.	p. 63

20	Resultados do GANORD - configurações iniciais com 6 agentes vgsB. . .	p. 63
21	Resultados do GANORD - configurações iniciais com todos os agentes.	p. 64
22	Tabela comparativa dos resultados do GANORD. . . . .	p. 65
23	Dados de Barras - Garver . . . . .	p. 73
24	Dados de Linhas - Garver . . . . .	p. 74
25	Dados de Barras - Sul Brasileiro . . . . .	p. 74
25	Dados de Barras - Sul Brasileiro (continuação) . . . . .	p. 75
25	Dados de Barras - Sul Brasileiro (continuação) . . . . .	p. 76
26	Dados de Linhas - Sul Brasileiro . . . . .	p. 76
26	Dados de Linhas - Sul Brasileiro (continuação). . . . .	p. 77
26	Dados de Linhas - Sul Brasileiro (continuação) . . . . .	p. 78
27	Dados de Barras - Norte-Nordeste Brasileiro . . . . .	p. 83
27	Dados de Barras - Norte-Nordeste Brasileiro (continuação) . . . . .	p. 84
27	Dados de Barras - Norte-Nordeste Brasileiro (continuação) . . . . .	p. 85
28	Dados de Linhas - Norte-Nordeste Brasileiro . . . . .	p. 86
28	Dados de Linhas - Norte-Nordeste Brasileiro (continuação) . . . . .	p. 87
28	Dados de Linhas - Norte-Nordeste Brasileiro (continuação) . . . . .	p. 88
28	Dados de Linhas - Norte-Nordeste Brasileiro (continuação) . . . . .	p. 89
28	Dados de Linhas - Norte-Nordeste Brasileiro (continuação) . . . . .	p. 90
28	Dados de Linhas - Norte-Nordeste Brasileiro (continuação) . . . . .	p. 91

# *Lista de Abreviaturas*

(PPNLIM)	problema de programação não linear inteira mista;
(PPL)	problema de programação linear;
(PL)	programação linear;
(AG)	algoritmo genético;
(SA)	<i>simulated annealing</i> ;
( <i>A-Teams</i> )	times assíncronos;
(PVM)	<i>Parallel Virtual Machine</i> ;
(MPI)	<i>Message Passing Interface</i> ;
(FIFO)	<i>first in first out</i> ;
( <i>nafa</i> )	parâmetro de penalização dos cortes de cargas;
( <i>npop</i> )	tamanho da população;
( <i>kincumb</i> )	variável de transformação do problema de maximização em um problema de minimização;
( <i>jgera</i> )	número de gerações;
( <i>npara</i> )	quantidade de indivíduos da população iniciados com o <i>inic_refina</i> ;
( <i>ycross</i> )	taxa de recombinação;
( <i>nmax</i> )	número máximo de linhas por ramo;
( <i>bconst</i> )	taxa de variação no número de mutações;
( <i>nmutar</i> )	taxa de mutação;
( <i>kseed</i> )	semente inicial;
( <i>ndes</i> )	número máximo de descendentes, em porcentagem;
( <i>temp</i> )	temperatura inicial do processo <b>SA</b> ;
( <i>alfat</i> )	taxa de resfriamento;
( <i>mcor</i> )	limite de corte de carga na mutação;
( <i>ncl</i> )	número máximo de caminhos a ser testados;
(LLPP)	Laboratório de Linux e Processamento Paralelo.

# *Sumário*

<b>1</b>	<b>Introdução</b>	p. 14
<b>2</b>	<b>O Problema do Planejamento da Expansão da Transmissão de Energia Elétrica</b>	p. 16
2.1	Modelagem Matemática do Problema . . . . .	p. 17
2.1.1	Modelo DC . . . . .	p. 18
2.1.2	Modelo de Transportes . . . . .	p. 19
2.1.3	Modelo Híbrido . . . . .	p. 20
2.2	Técnicas de Solução . . . . .	p. 21
2.2.1	Métodos Exatos . . . . .	p. 22
2.2.2	Métodos Aproximados . . . . .	p. 22
2.2.2.1	Métodos Heurísticos Construtivos . . . . .	p. 23
2.2.2.1.1	Método de Garver . . . . .	p. 23
2.2.2.1.2	Método do Mínimo Esforço . . . . .	p. 24
2.2.2.1.3	Método de Mínimo Corte de Carga . . . . .	p. 25
2.2.2.2	Metaheurísticas . . . . .	p. 26
2.2.2.2.1	Algoritmo Genético . . . . .	p. 27
2.2.2.2.2	Algoritmo Genético para o Planejamento da Expansão da Transmissão . . . . .	p. 28
<b>3</b>	<b>Times Assíncronos</b>	p. 32
3.1	Times Assíncronos: Visão Histórica . . . . .	p. 32
3.2	Times Assíncronos: Visão Geral . . . . .	p. 32

3.2.1	Agentes de construção . . . . .	p. 34
3.2.2	Memória central . . . . .	p. 35
3.2.3	Agentes de destruição . . . . .	p. 35
3.2.4	Comunicações assíncronas . . . . .	p. 35
3.2.5	Fluxo de dados cíclico . . . . .	p. 35
3.3	O Ambiente distribuído . . . . .	p. 36
3.3.1	As Bibliotecas PVM e MPI . . . . .	p. 36
3.3.2	Máquina Paralela Virtual . . . . .	p. 38
<b>4</b>	<b>Algoritmos Propostos</b>	<b>p. 39</b>
4.1	Memória Central . . . . .	p. 39
4.2	Agentes construtores . . . . .	p. 42
4.2.1	Agentes de Villasana-Garver-Salon . . . . .	p. 42
4.2.2	Agentes Mínimo Esforço . . . . .	p. 43
4.2.3	Agentes Mínimo Corte de Carga . . . . .	p. 45
4.3	Algoritmo Genético Utilizado . . . . .	p. 46
4.3.1	Descrição do algoritmo GANORD . . . . .	p. 46
<b>5</b>	<b>Resultados</b>	<b>p. 49</b>
5.1	Geração das Soluções Iniciais . . . . .	p. 50
5.2	Simulações com o algoritmo GANORD . . . . .	p. 51
5.2.1	Sistema Garver . . . . .	p. 52
5.2.1.1	Sistema Garver com redespacho . . . . .	p. 52
5.2.1.2	Sistema Garver sem redespacho . . . . .	p. 53
5.2.2	Sistema Sul brasileiro . . . . .	p. 54
5.2.2.1	Sistema Sul brasileiro com redespacho . . . . .	p. 54
5.2.2.2	Sistema Sul brasileiro sem redespacho . . . . .	p. 56

5.2.3	Sistema Norte-Nordeste brasileiro . . . . .	p. 57
5.2.3.1	Sistema Norte-Nordeste brasileiro - Plano 2002 . . . . .	p. 57
5.2.3.2	Sistema Norte-Nordeste brasileiro - Plano 2008 . . . . .	p. 59
5.3	Resumo dos Resultados . . . . .	p. 61
<b>6</b>	<b>Conclusões e Sugestões para Trabalhos Futuros</b>	p. 67
	<b>Referências</b>	p. 69
	<b>Apêndice A – Dados dos Sistemas Testes</b>	p. 73
A.1	Sistema Garver (6 Barras/15 Ramos) . . . . .	p. 73
A.2	Sistema Sul Brasileiro (46 Barras/79 Ramos) . . . . .	p. 74
A.3	Sistema Norte-Nordeste Brasileiro (87 Barras/179 Ramos) . . . . .	p. 83
	<b>Apêndice B – Uso do Ambiente LAM-MPI</b>	p. 92
B.1	Começando com o LAM-MPI . . . . .	p. 92
B.1.1	O arquivo de Máquinas . . . . .	p. 92
B.1.2	Lamboot . . . . .	p. 92
B.1.3	Tping N . . . . .	p. 93
B.2	Compilando Programas MPI . . . . .	p. 93
B.2.1	Executando Programas no Ambiente LAM-MPI . . . . .	p. 94
B.3	Finalizando o LAM-MPI . . . . .	p. 95

# 1 *Introdução*

O problema do planejamento da expansão da transmissão de energia elétrica envolve uma gama de etapas e procedimentos. Num primeiro momento, é realizada análise estática a longo prazo, seguida da análise dinâmica, análise de contingências entre várias outras análises e testes antes da construção física das linhas de transmissão.

Esta dissertação trata do problema do planejamento *estático* da expansão da transmissão de energia elétrica num horizonte de *longo prazo*. Neste tipo de planejamento, dada uma configuração inicial, dados de geração e demanda do horizonte de planejamento, tenta-se determinar o plano de expansão com um custo mínimo, isto é, deve-se determinar *onde e que* tipos de novos equipamentos devem ser instalados afim de atender as futuras demandas de cargas.

Identificado o problema, se faz necessário determinar a melhor maneira de resolvê-lo. Para isto, as metaheurísticas têm mostrado desempenhos satisfatórios, porém, dada as dimensões do problema, quando se trata de sistemas de grande porte, elas não apresentam soluções aceitáveis se não partirem de configurações iniciais de boa qualidade. Na literatura especializada, o uso de métodos heurísticos construtivos na geração de soluções iniciais para as metaheurísticas apresentaram resultados satisfatórios.

Os vários métodos heurísticos construtivos existentes para resolver o problema do planejamento contribuem de maneira singular para a composição de uma solução de boa qualidade. O que se busca neste trabalho, é a utilização destas características individuais, montando times assíncronos compostos por estes métodos, trabalhando em paralelo e com o objetivo de gerar configurações iniciais de qualidade superior às obtidas com uma única heurística.

No capítulo 2 é feita uma introdução ao problema do planejamento como um todo, as modelagens matemáticas mais usadas, uma breve abordagem sobre os métodos exatos e uma introdução aos métodos aproximados, divididos em heurísticos construtivos e metaheurísticas. O capítulo 3 apresenta os times assíncronos e suas características, bem



---

como, as bibliotecas de troca de mensagens e a máquina paralela virtual. No capítulo 4 é feito o detalhamento dos algoritmos implementados, a memória central e os agentes construtores que formam o time assíncrono inicializador proposto. É realizado ainda no capítulo 4 uma revisão do algoritmo genético utilizado para testar as configurações iniciais geradas. O capítulo 5 traz testes, comentários e simulações de várias combinações de times assíncronos. As conclusões gerais estão dispostas no capítulo 6. Finalmente as referências bibliográficas são mostradas, seguidas do apêndice A, com dados dos sistemas testes utilizados, e B com informações sobre o uso do ambiente LAM-MPI.

## 2 *O Problema do Planejamento da Expansão da Transmissão de Energia Elétrica*

O planejamento da expansão de sistemas de transmissão de energia elétrica tem como objetivo determinar *onde*, *quantos* e *quando* serão adicionados novos circuitos e/ou transformadores no sistema de forma que o mesmo atenda as novas demandas do horizonte de planejamento. O planejamento pode ser estático, determinando somente *onde* e *quantos* circuitos serão adicionados ou planejamento dinâmico (multi-estágio), que determina também *quando* os circuitos serão instalados ao longo do tempo. Os dados do problema são: a topologia atual (ano base), os circuitos candidatos, a geração e a demanda para o ano-horizonte, as restrições de investimento, etc. Com estes dados e com a *finalidade de determinar o plano ótimo da expansão a um custo mínimo de implantação*, pode-se modelar o problema do planejamento como um problema de minimização.

Como todos os problemas de otimização matemática, o problema do planejamento de sistemas de transmissão deve ser separado em duas partes: modelagem matemática e a técnica de solução escolhida para resolver esse modelo.

A modelagem matemática ideal para o problema corresponde a um problema de programação não linear inteira mista (PNLIM), e que dada a configuração do sistema pode apresentar o fenômeno da explosão combinatória. Sendo que, para uma alternativa de investimento o problema se reduz a um problema de programação linear (PL), cujo objetivo é verificar a factibilidade desta alternativa, ou seja, verificar se a configuração escolhida atende as demandas de energia do horizonte de planejamento.

Sendo assim, as técnicas de resolução do problema de planejamento, têm como maior dificuldade a natureza combinatória do processo que normalmente leva a um número muito grande de alternativas de solução. Além disso, o problema de planejamento apresenta uma estrutura multimodal com um número muito elevado de ótimos locais o que leva a

maioria dos métodos aproximados a parar em uma solução ótima local e às vezes de pobre qualidade. Para contornar estes problemas foram apresentados na literatura especializada muitos modelos e técnicas de solução sendo que alguns serão brevemente explicados neste capítulo.

## 2.1 Modelagem Matemática do Problema

A modelagem matemática consiste na representação de um problema real que relaciona um conjunto de variáveis de decisão através de um conjunto de relações matemáticas que podem assumir formas e tipos variados. A modelagem matemática pode ser uma representação exata ou simplificada de um problema da vida real. Em geral, quanto mais exata a modelagem matemática que representa o problema da vida real, mais complexa é a resolução deste modelo matemático. Assim, deve existir um compromisso entre a modelagem matemática adotada e a técnica de solução escolhida para que se possa utilizar recursos computacionais aceitáveis, isto é, dentro dos limites desses recursos. Com o desenvolvimento das pesquisas, das técnicas de solução e/ou computadores mais velozes é provável que modelos atualmente considerados complexos se tornem adequados no futuro (ROMERO, 1999).

No planejamento estático da expansão de sistemas de transmissão parte-se de um sistema elétrico com uma topologia corrente e deseja-se encontrar o plano de expansão ótimo (construção de novos circuitos e/ou transformadores) para um horizonte de planejamento definido. Nesse contexto, a modelagem ideal para indicar a operação adequada seria a representação do problema por meio de relações matemáticas de fluxo de carga AC. Entretanto, a topologia inicial do sistema elétrico usado em planejamento é um sistema não conexo, isto é, o sistema apresenta um conjunto de barras isoladas ou ilhadas da parte principal do sistema, tornando mais difícil a análise do fluxo de carga AC.

Levando em conta as observações anteriores, atualmente, se considera a modelagem matemática conhecida como *modelo DC* como sendo a modelagem ideal para ser usada no planejamento de sistemas de transmissão. Porém, vários outros modelos foram propostos para o problema do planejamento da expansão dos sistemas de transmissão destacando-se o *modelo de transportes* e o *modelo híbrido linear*. A seguir, estes modelos são descritos mais detalhadamente.

### 2.1.1 Modelo DC

O modelo DC é uma generalização do modelo de fluxo de carga DC que está amplamente desenvolvido em (MONTICELLI, 1983). Este modelo é tido como ideal para ser usado em planejamento de sistemas de transmissão, pois leva em conta as leis de Kirchhoff para o sistema elétrico, ou seja, todas as barras do sistema e todos os laços existentes devem satisfazer à lei das correntes e a lei das tensões de Kirchhoff.

A formulação do problema usando o modelo DC leva a um problema de PNLIM que 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. A formulação do problema já considerando uma pequena modificação na função objetivo, onde se leva em conta uma geração fictícia  $r_i$  multiplicada por um fator  $\alpha_i$  de penalização, aumentando a região factível do problema, é apresentada na equação 2.1 a seguir.

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

Descrição das variáveis:

- $v$  Investimento devido a adição de circuitos;
- $c_{ij}$  Custo de um circuito no ramo  $(i, j)$ ;
- $n_{ij}$  Número de linhas adicionadas;
- $\alpha$  Parâmetro de penalização dos cortes de cargas;
- $r$  Vetor de geradores fictícios ou artificiais;
- $B$  Matriz de susceptâncias do sistema elétrico;

$\theta$	Ângulos das tensões nodais;
$g$	Vetor de gerações;
$d$	Vetor de demandas;
$f_{i,j}$	Fluxo de potência ativa total no ramo $(i, j)$ ;
$\gamma_{i,j}$	susceptância equivalente referente aos circuitos adicionados;
$\gamma_{ij}^0$	susceptância equivalente referente aos circuitos da configuração base;
$n_{ij}^0$	Número de linhas iniciais no ramo $(i, j) \in \Omega$ ;
$\bar{f}_{ij}$	fluxo máximo de potência ativa por circuito $(i, j)$ ;
$\bar{g}$	Vetor de limites de geração $(i, j)$ ;
$\bar{n}_{ij}$	Vetor de número máximo de adições no ramo $(i, j)$ ;
$\Omega$	Conjunto de todos os ramos definidos pelas linhas existentes e as alternativas de expansão.

Entretanto, quando são especificados os valores dos  $n_{i,j}$ , isto é, quando é escolhida (ou determinada) uma proposta de investimento ou configuração, a formulação se reduz a um simples problema de PL cuja única finalidade é verificar a factibilidade da proposta de investimento, isto é, verificar se o investimento proposto produz um corte de carga igual a zero.

## 2.1.2 Modelo de Transportes

O planejamento de grandes sistemas de energia elétrica envolve muitos aspectos dado a dimensão do problema. O uso da programação linear para a análise da rede elétrica em questão, determinando onde existe escassez de recursos e onde alocar novos circuitos para suprir estas deficiências energéticas, se mostrou uma solução inteligente e bastante viável.

Desenvolvido por Garver (GARVER, 1970), o modelo de transportes se tornou uma parte fundamental dos programas computacionais desenvolvidos para o planejamento da expansão da transmissão de energia elétrica e é usado até hoje por muitas empresas por convergir para boas soluções, próximas do ótimo, com pouco esforço computacional. O modelo desenvolvido por Garver também conhecido como *Modelo de Transportes*, pode ser considerado uma versão relaxada do *modelo DC*, onde não se leva em conta as restrições pertinentes a lei das tensões de Kirchhoff. Resolver a formulação do modelo de transportes significa estimar o transporte de carga no sistema (o somatório dos fluxos de potência que chega e sai de uma barra do sistema é igual a zero), obedecendo à capacidade de transmissão dos circuitos e que as unidades de geração operem dentro de seus limites

especificados. Com o objetivo de encontrar uma configuração que produz o menor investimento no plano de expansão do sistema elétrico a modelagem matemática do problema assume a forma que se segue.

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

Descrição das variáveis:

$S$  Matriz de incidência nó-ramo.

### 2.1.3 Modelo Híbrido

O *Modelo Híbrido* introduzido por Villasana e Garver em (VILLASANA; GARVER; SALON, 1985) é uma combinação do modelos de transportes e DC. No modelo original, o *modelo DC* é resolvido para todos os circuitos pertencentes a configuração corrente obedecendo as duas leis de Kirchhoff e o *modelo de transportes* é resolvido apenas para os circuitos de sobrecarga (novas regiões de busca), servindo como indicador das sobrecargas existentes e da quantidade de carga solicitada. Sendo assim, a modelagem proposta por Villasana se mostra muito próxima do modelo DC e com o mesmo nível de complexidade, isso explica porque poucos pesquisadores exploraram o método na sua forma original.

Uma variante do modelo híbrido, usada em (BINATO, 2000), (HAFFNER, 2000) entre outros, pode ser chamada de *Modelo Híbrido Linear*. Para criar o *Modelo Híbrido Linear* simplesmente foi adicionado à formulação 2.2 uma parcela das restrições correspondentes a lei das tensões de Kirchhoff (referentes as linhas iniciais), contornando as deficiências existentes no *modelo de transportes*. Desta maneira, os circuitos pertencentes a topologia base (inicial) devem cumprir as duas leis de Kirchhoff e os circuitos novos obedecem

apenas a lei das correntes de Kirchhoff, sendo que este modelo ainda é um problema de PLIM (Programação Linear Inteira Mista) como o *modelo de transportes*. O interessante neste modelo é que ele tem a complexidade parecida com a do *modelo de transportes*, mas os seus resultados são factíveis para o *modelo DC*. A formulação do *Modelo Híbrido Linear* utilizada neste trabalho é apresenta à seguir.

$$\begin{aligned}
 \text{Min } v &= \sum_{(i,j) \in \Omega} c_{i,j} n_{i,j} \\
 \text{s.a.} & \\
 & Sf + S^0 f^0 + g = d \\
 & |\theta_i - \theta_j| = f_{ij} / (\gamma_{ij} + \gamma_{ij}^0) \quad \forall (i,j) \in \Omega_1 \\
 & |f_{ij}| \leq (n_{ij} + n_{ij}^0) \bar{f}_{ij} \\
 & 0 \leq g \leq \bar{g} \\
 & 0 \leq n_{i,j} \leq \bar{n}_{i,j} \\
 & n_{ij} \text{ inteiro} \\
 & f_{ij} \text{ irrestrito} \\
 & \forall (ij) \in \Omega
 \end{aligned} \tag{2.3}$$

Descrição das variáveis:

- $f^0$  vetor de fluxo de potência ativa referente aos circuitos da configuração base;
- $S^0$  Matriz de incidência nó-ramo referente aos circuitos da configuração base;
- $\Omega_1$  Conjunto de todos os ramos definidos pelas linhas existentes na configuração base.

## 2.2 Técnicas de Solução

Ao longo dos mais de trinta anos de pesquisa operacional voltada ao planejamento da transmissão, foram apresentadas várias técnicas para resolver os modelos matemáticos desenvolvidos. Duas grandes áreas dividem essas técnicas em *Métodos Exatos* e *Métodos Aproximados*.

### 2.2.1 Métodos Exatos

Os métodos exatos exploram a decomposição do problema principal em problemas menores e relaxando temporariamente as restrições, visam tornar o problema mais fácil de resolver. A condição que deve ser mantida é que o conjunto de soluções viáveis do problema original sempre esteja contido no conjunto de soluções viáveis do problema relaxado.

A técnica de decomposição de Benders (BENDERS, 1962) explora a decomposição natural do problema de planejamento da expansão da transmissão em dois subproblemas, o de investimento e o de operação. A técnica começou a ser usada para o planejamento da transmissão por volta da década de 80, tendo suas principais aplicações em (GRANVILLE; PEREIRA, 1985), (PEREIRA et al., 1985) e (BINATO, 2000). Em (ROMERO; MONTICELLI, 1994) o emprego da técnica usando os modelos de transportes, híbrido e DC levou a soluções ótimas para sistemas de pequeno e médio porte que ainda eram desconhecidas na literatura especializada. Entretanto, na resolução de sistemas de grande porte esse método se mostrou ineficiente.

A idéia básica da metodologia Branch-and-Bound (LAND; DOIG, 1960) é a estratégia de dividir um problema sucessivamente em subproblemas relaxados até que eles sejam completamente resolvidos. A solução do problema original é a melhor solução encontrada dentro de um conjunto de soluções (MIYASAKI, 2002). Atualmente os algoritmos baseados na metodologia Branch-and-Bound são os mais utilizados dentre os métodos exatos conhecidos. Uma combinação do algoritmo de Branch-and-Bound com outras técnicas de decomposição foi apresentada em (PINTO; NUNES, 1990). As principais aplicações relacionadas podem ser encontradas em (HAFFNER et al., 2001), (HAFFNER et al., 2000), (CHOI; EL-KEIB; TRAN, 2005) e (LEE; HOCKS; HYNILICZA, 1974).

### 2.2.2 Métodos Aproximados

No princípio da década de 80 os pesquisadores voltaram sua atenção para uma nova área da pesquisa operacional, os métodos aproximados, que podem ser divididos em métodos *heurísticos construtivos* e *metaheurísticas*. Os métodos heurísticos construtivos são algoritmos relativamente simples que a cada iteração adicionam um circuito à configuração, a fim de suprir a demanda de carga. Já as metaheurísticas são algoritmos complexos e geralmente mais eficientes, com uma grande variedade de tempo de processamento e com parâmetros que podem ser calibrados para cada tipo de aplicação,



melhorando seu desempenho.

### 2.2.2.1 Métodos Heurísticos Construtivos

Um algoritmo heurístico construtivo é um procedimento passo-a-passo que, de maneira sistemática, encontra uma boa proposta de expansão do sistema elétrico, isto é, a partir da configuração base, em cada passo é adicionado um ou vários circuitos até que o conjunto de adições realizadas permita uma operação adequada do sistema elétrico. Portanto, em cada passo a configuração do sistema é modificada pela adição de um ou vários circuitos e esta configuração obtida é denominada configuração corrente. O circuito escolhido em cada passo para ser adicionado à chamada configuração corrente é o circuito que corresponde ao caminho mais atrativo identificado pelo critério de sensibilidade (ou índice de desempenho). Assim a diferença fundamental entre os vários algoritmos heurísticos construtivos reside no indicador de sensibilidade escolhido e, obviamente, no modelo escolhido (DC, transportes ou híbrido) (ROMERO, 1999). Dentre os vários métodos heurísticos construtivos existentes na literatura, serão detalhados neste capítulo o Algoritmo Heurístico de Garver, o Método do Mínimo Esforço e o Método do Mínimo Corte de Carga.

#### 2.2.2.1.1 Método de Garver

Em (GARVER, 1970) é proposta uma das primeiras técnicas de estimação de fluxo capaz de estudar qualquer arranjo da rede e determinar onde há a necessidade de novos circuitos. Garver sugeriu em seu trabalho a utilização do “linear loss function network model” uma modelagem agora chamada de *modelo de transportes* como a forma mais adequada para planejamento da expansão. Na época não era possível a utilização de modelos mais exatos, como o *fluxo de carga AC*, devido à topologia desconexa dos sistemas utilizados para o planejamento. O algoritmo de Garver resolve facilmente sistemas com barras ou regiões ilhadas; isso proporcionou o surgimento da fase dos algoritmos construtivos para planejamento da expansão, usados até hoje pelas concessionárias por convergir para soluções de boa qualidade com baixo esforço computacional.

A parte mais interessante nesta técnica é o fato de que as sobrecargas não aparecerão nos circuitos normais, mas em um novo tipo de rede fictícia chamada caminho de sobrecarga, de capacidade ilimitada. Desta maneira a adição de circuitos pode ser feita baseada nesses circuitos fictícios; além disso, o caminho de sobrecarga pode facilmente incluir na estimação do fluxo nós que ainda não pertencem à rede. Estes caminhos de sobrecarga

possuem um custo muito alto e não são atrativos para a solução final de um problema de minimização, por isso, a cada iteração o circuito fictício que possui a maior sobrecarga é substituído por um circuito real, buscando minimizar o custo final do problema.

O *modelo de transportes* explicado na seção 2.1.2 é ligeiramente diferente do modelo apresentado por Garver em (GARVER, 1970), mas conceitualmente equivalente. O problema 2.2 é um problema de PLIM que, quando tem a integralidade da variável  $n_{ij}$  relaxada, se torna um problema de programação linear (PL). Então, a estimação de fluxo de cargas do sistema elétrico modelado pode ser obtida utilizando programas computacionais específicos para resolver PL's. Desta maneira, adiciona-se ao sistema um circuito correspondente ao caminho que apresentar maior fluxo de sobrecarga e a configuração com o novo circuito adicionado passa a ser a *configuração corrente* da próxima iteração. O algoritmo encerra se não existirem mais caminhos com sobrecarga, ou seja, o sistema está operando adequadamente para o *modelo de transportes*.

### 2.2.2.1.2 Método do Mínimo Esforço

No início da década de 80 apareceram vários trabalhos relevantes utilizando o *modelo DC* e neste contexto novos algoritmos heurísticos construtivos foram desenvolvidos. Por se tratar de um modelo mais completo, observa-se uma maior complexidade em se resolver o *modelo DC* comparado com o *modelo de transportes*, em compensação as configurações encontradas são mais atrativas como alternativa de planejamento.

O *método do mínimo esforço* apresentado por Monticelli em (MONTICELLI et al., 1982) utiliza o *modelo DC* e leva em conta o padrão da distribuição dos fluxos na rede para determinar quais circuitos serão adicionados. O critério do mínimo esforço analisa o quadrado da diferença angular das tensões nodais multiplicado pelas susceptâncias dos circuitos, e em cada passo do processo de planejamento é adicionado ao sistema aquele circuito que produz o maior impacto na distribuição de fluxos na rede, segundo a equação a seguir.

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

em que:

( $\theta_i$  é o ângulo da tensão do nó  $i$  e  $\gamma_{ij}$  é a susceptância de um circuito no ramo  $ij$ ).

Nesta metodologia, a dificuldade em resolver a rede inicial desconexa é contornada

adotando-se uma “rede fictícia”, superposta à configuração do sistema, constituída por circuitos com uma capacidade de transmissão muito inferior a dos circuitos pertencentes a rede real. Desta maneira, a rede fictícia só é usada quando for impossível utilizar a rede real.

Mesmo usando uma modelagem mais completa, o algoritmo de mínimo esforço é uma heurística simples que realiza o plano de expansão passo a passo, adicionando os circuitos um a um ou em pequenos grupos.

### 2.2.2.1.3 Método de Mínimo Corte de Carga

O método desenvolvido em (PEREIRA; PINTO, 1985) é semelhante ao método de mínimo esforço; usa o *modelo DC* e realiza a adição de linhas selecionadas de acordo com um índice de sensibilidade que permite encontrar linhas mais atrativas. O diferencial é que no algoritmo do mínimo esforço são permitidas sobrecargas nos circuitos fictícios, já no método do mínimo corte de carga os circuitos devem permanecer dentro dos limites de operação e os cortes de cargas são contornados pelos geradores artificiais (em cada barra do sistema é considerada a existência de um gerador artificial). O problema do planejamento da expansão da transmissão é agora modelado segundo o *modelo DC modificado* mostrado na equação à seguir.

$$\begin{aligned}
 \text{Min } v &= \sum r_i \\
 \text{s.a.} & \\
 & B\theta + g + r = d \\
 & |\theta_i - \theta_j| \leq \frac{\bar{f}_{ij}}{\gamma_{ij}} \\
 & 0 \leq g \leq \bar{g} \\
 & 0 \leq r \leq d \\
 & \theta_j \text{ irrestrito}
 \end{aligned} \tag{2.5}$$

Desta maneira, como o próprio nome do método diz, em cada passo do algoritmo novos circuitos são adicionados com o objetivo de minimizar a utilização dos geradores fictícios, ou seja, minimizar o corte de carga do sistema. O processo de adição de circuitos encerra quando não houver cortes de carga no sistema. O critério para seleção dos circuitos que serão adicionados segue o índice de desempenho a seguir.

$$IS_{mcc} = (\pi_i - \pi_j)(\theta_i - \theta_j) \tag{2.6}$$

em que:

$\pi_j$  é o multiplicador de Lagrange da  $j$ -ésima restrição ( $B\theta + g + r = d$ );

$\theta_j$  são os ângulos de tensão da barra  $j$  obtidos na análise DC do sistema corrente, equação (2.5).

### 2.2.2.2 Metaheurísticas

Em se tratando de problemas de natureza combinatória, normalmente, o número de configurações possíveis é muito elevado. A procura por soluções de boa qualidade dentre essa gama de soluções levou os pesquisadores a desenvolver algoritmos baseados nas mais diversas áreas. Foram desenvolvidas técnicas que simulam processos da indústria metalúrgica, inteligência artificial, algoritmos baseados em situações como a organização dos formigueiros ou a evolução natural dos indivíduos, entre outros. A aplicação das metaheurísticas também atingiu várias áreas do conhecimento, algoritmos aproximados são largamente usados em pesquisas onde o grande espaço de busca torna inviável ou até mesmo impossível a resolução do problema utilizando métodos exatos.

Pode se mencionar as seguintes situações onde uso das metaheurísticas se mostra eficiente: busca por combustíveis fósseis, seqüenciamento de proteínas, aplicações financeiras, indústria têxtil, reconhecimento e classificação de padrões, resolução de problemas matemáticos, alocação de recursos, problemas de corte irregular entre várias outras. Com artifícios como memórias de incumbentes e recência, as metaheurísticas atingem soluções melhores que os métodos heurísticos construtivos, porém, demandam mais esforço computacional.

Técnicas de otimização como o “simulated annealing”, busca tabu, algoritmos genéticos, times assíncronos entre outros, são exemplos de metaheurísticas que estão sendo empregados com sucesso na solução do problema de planejamento da expansão da transmissão.

O “simulated annealing” (**SA**) (KIRKPATRICK; GELATTI; CECCHI, 1983) é uma técnica que faz analogia ao processo de esfriamento de metais. No **SA** básico, em cada nível de temperatura são realizadas  $k$  tentativas de mudança para uma nova configuração. Se a nova configuração for de melhor qualidade ela é aceita pelo algoritmo, se for de pior qualidade ela só é aceita segundo certa probabilidade. Isso permite que o algoritmo saia de regiões de ótimo local em busca de melhores configurações em outra vizinhança. Atingidas as  $k$  tentativas de mudança, o algoritmo passa a um próximo nível de temperatura repetindo o mesmo processo do nível anterior, mas agora com um novo  $k$  e com critérios

de aceitação mais rígidos. O algoritmo encerra com o congelamento ou quando atinge um número máximo de iterações.

Busca tabu, (GLOVER, 1989) e (GLOVER, 1990) utiliza conceitos da inteligência artificial para explorar os espaços de busca de soluções além do ótimo local. Busca tabu permite mover-se de forma agressiva entre as vizinhanças. Utilizando-se de melhorias adaptativas, o mesmo trabalha de maneira inteligente e com atributos tabus que impedem que o algoritmo reutilize configurações saturadas por um período, o algoritmo varre todo o espaço de soluções possíveis.

Algoritmo genético (**AG**) é uma técnica de busca utilizada na pesquisa operacional para achar soluções aproximadas em problemas de otimização. Os **AG** são uma classe particular de algoritmos evolutivos que usam técnicas inspiradas pela biologia evolutiva como: hereditariedade, mutação, seleção natural e recombinação (ou “crossing over”). No **AG**, os indivíduos (configurações) mais aptos são selecionados para se recombinar e com isso passam suas características para as próximas gerações, melhorando assim, a capacidade dos indivíduos (qualidade das configurações) ao longo do processo.

Time assíncrono pode ser considerado uma metaheurística onde vários agentes contribuem simultaneamente, paralelamente e de maneira assíncrona, para a formação de um superagente (desenvolvido para solucionar um determinado problema). Um time assíncrono é constituído de uma ou mais memórias que armazenam um conjunto de soluções, de vários agentes construtores que contribuem de maneira singular e em cooperação para o melhoramento das soluções e de agentes destruidores (reguladores) responsáveis por manter constante o número de soluções armazenado nas memórias.

Das várias metaheurísticas existentes, foi dado ênfase neste trabalho ao estudo dos times assíncronos. Especialmente os times assíncronos de métodos heurísticos construtivos, que serão descritos no capítulo 3. O algoritmo genético é brevemente discutido na seqüência, pois foi utilizado para testar a qualidade das soluções iniciais geradas com o time assíncrono proposto. Este algoritmo genético é proposto em (GALLEGO, 1997)

#### 2.2.2.2.1 Algoritmo Genético

A teoria da evolução como é conhecida nos dias de hoje combina a genética e a seleção natural, sendo Charles Darwin o pesquisador mais conhecido nesta área. Pode-se definir genética natural como a diversidade entre indivíduos em uma população de organismos que se reproduzem. Esta diversidade é produzida pela recombinação e pela inserção de material genético novo na população a cada geração (AVILA, 2002). Esta definição vem

sendo assimilada e utilizada principalmente no desenvolvimento das áreas que envolvem a biologia e a matemática, através de simulações de sistemas genéticos.

Os indivíduos de uma espécie são considerados melhores ou piores dependendo do tipo de situação. Se em uma dada situação, duas pessoas, uma magra e uma obesa, são perseguidas por um predador, a pessoa magra tem maior probabilidade de fugir. Mas se as mesmas pessoas ficarem ilhadas por vários dias, sem alimento, a pessoa obesa tem maior probabilidade de sobreviver. Sobrevivendo, os indivíduos têm maior probabilidade de se reproduzirem e passar suas características genéticas para as próximas gerações.

Em 1975, o engenheiro eletricitista John H. Holland escreveu o livro intitulado “Adaptation in natural and artificial systems” (HOLLAND, 1975), em que aborda diretamente os **AGs**, o que deu origem ao uso desta técnica para a otimização de sistemas. Posteriormente, a metodologia foi desenvolvida com mais detalhes por David E. Goldberg, antigo aluno de Holland. Os estudos de Goldberg foram publicados no seu livro “Genetic algorithms in search, optimization & machine learning” (GOLDBERG, 1989).

Uma comparação simples entre os problemas de otimização e a evolução natural pode ser feita da seguinte maneira: uma parte das soluções possíveis é comparada aos indivíduos de uma população, e a procura por soluções de boa qualidade ou o melhoramento das soluções atuais é comparado ao processo evolutivo que uma espécie sofre ao longo das gerações.

#### **2.2.2.2 Algoritmo Genético para o Planejamento da Expansão da Transmissão**

Os **AGs** são algoritmos de otimização estocástica que trabalham de forma “aleatória orientada” de acordo com regras probabilísticas baseadas numa analogia com os mecanismos da genética natural. Métodos aproximados como o **AG** vem sendo largamente usados na resolução de problemas combinatórios, como o planejamento da expansão da transmissão, pela sua capacidade de encontrar soluções de boa qualidade dentre a grande quantidade de soluções possíveis.

- **Codificação e representação do problema**

Para o planejamento da transmissão, o problema é melhor modelado utilizando-se o modelo DC, detalhado no capítulo 2.1, seção 2.1.1. A função objetivo do sistema de equações 2.1, considera o custo das linhas adicionadas  $n_{ij}$  mais uma penalidade correspondente ao custo do corte de carga no sistema. Desta maneira, uma confi-

guração da população está constituída unicamente pelas variáveis de investimento  $n_{ij}$ . Portanto, as variáveis que formam uma configuração são inteiras e correspondem ao número de linhas adicionadas em cada caminho.

O problema do planejamento é formulado como um problema de minimização, mas na formulação tradicional, os algoritmos genéticos são problemas de maximização. A transformação do problema de minimização em um problema de maximização é realizada da seguinte maneira:

$$\min v \Leftrightarrow \max [K - v]$$

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.

- **Geração da População Inicial**

No problema da expansão de redes de transmissão gera-se a população inicial de duas maneiras: a primeira corresponde a uma forma aleatória na qual é gerado um número aleatório igual ou inferior ao máximo número de linhas permitido para ser armazenado em posições também escolhidas aleatoriamente. Neste tipo de inicialização, é possível encontrar soluções de boa qualidade ou até mesmo a solução ótima para sistemas de pequeno e médio porte, obviamente, exigindo um maior número de iterações (gerações) e um maior tempo computacional. Já para sistemas de grande porte a inicialização aleatória mostra-se pouco efetiva, pois devido ao grande número de regiões no espaço de soluções o algoritmo pode ficar preso em alguma região de ótimo local. Na segunda e mais eficiente, a população inicial é gerada utilizando métodos heurísticos construtivos. Neste trabalho os métodos do mínimo esforço (MONTICELLI et al., 1982), método do mínimo corte de carga (PEREIRA; PINTO, 1985) e método de Garver (GARVER, 1970) foram combinados na forma de agentes de times assíncronos para gerar soluções iniciais de boa qualidade.

- **Seleção - Recombinação - Mutação**

Sobre a população corrente são realizados os processos de seleção, recombinação e mutação. A população corrente é então atualizada com os melhores indivíduos da configuração corrente mais os melhores indivíduos gerados, sempre respeitando um número máximo de indivíduos por geração, criando-se assim uma nova população.

O mecanismo de *seleção* é a primeira etapa no processo de obtenção de uma nova população a partir da população corrente. Nesta etapa, determina-se o número de descendentes a que tem direito cada configuração da população corrente, proporcional ao valor da função objetivo. Para evitar a convergência prematura num ótimo local no início da busca, limita-se o número máximo de descendentes para cada configuração independentemente da qualidade de sua função objetivo. Na segunda etapa, foi utilizada a *recombinação* dupla, de maneira que são selecionados aleatoriamente dois pontos em cada uma das configurações que se cruzam. O material genético entre os dois pontos é trocado pelos indivíduos “pais” segundo uma taxa de recombinação (porcentagem de configurações que serão efetivamente recombinadas). O resultado disso são duas novas configurações descendentes que passam agora pelo processo de mutação. Uma variação muito utilizada nos algoritmos genéticos é a incorporação de técnicas de outras metaheurísticas, como por exemplo, o processo de *mutação* controlado por “simulated annealing”. Se a configuração resultante da mutação for de melhor qualidade ela é aceita pelo algoritmo, se for de pior qualidade ela só é aceita segundo certa probabilidade controlada pelo critério de mutação a seguir:

$$P(\Delta v) = e^{-\frac{\Delta v}{T}}$$

em que  $\Delta v$  é a variação da função objetivo e  $T$  é o parâmetro de controle *temperatura* do processo “simulated annealing”, que diminui ao longo das gerações diminuindo a probabilidade dos indivíduos sofrerem mutações.

- **Parâmetros de Controle e Critérios de Parada**

A escolha adequada dos parâmetros de controle de um algoritmo genético é um tópico que ainda apresenta muitos pontos polêmicos entre os especialistas. Estes parâmetros não podem ser selecionados de maneira independente, pois os mesmos estão estreitamente relacionados e também dependem do tipo de problema a ser resolvido, das dimensões dos sistemas analisados, da qualidade das soluções iniciais e dos recursos computacionais disponíveis. Os parâmetros de controle de um **AG** básico são: tamanho da população, número máximo de gerações, taxa de recombinação e taxa de mutação.

Nos **AG** básicos implementados para o planejamento da expansão normalmente são empregados dois critérios de parada simples: o primeiro, por número máximo de



gerações, e o segundo, se durante as últimas  $k$  gerações a solução incumbente não apresenta melhoria.

## 3 *Times Assíncronos*

Neste capítulo é apresentada uma introdução à teoria de Times Assíncronos, fundamentos e características. Também é feito um comparativo entre as bibliotecas de processamento paralelo, bem como, é introduzido o conceito de Máquina Paralela Virtual.

### 3.1 Times Assíncronos: Visão Histórica

Em (ORTEGA; RHEINBOLDT, 1970) é realizado um trabalho que verifica a convergência de métodos iterativos combinados em um ambiente seqüencial. A partir da década de 80, com o surgimento dos sistemas distribuídos, é apresentado em (TALUKDAR; PYO; GIRAS, 1983) um dos primeiros trabalhos utilizando uma combinação de algoritmos de maneira assíncrona. Em (PYO, 1985), Pyo chama de “team approach” a técnica de montar um time assíncrono com algoritmos já existentes para resolver equações algébricas. Em (TALUKDAR; DE SOUZA, 1990) é introduzido o termo “Asynchronous teams” num algoritmo usado para resolver equações não-lineares.

Em (DE OLIVEIRA; DE ALMEIDA; MONTICELLI, 1998) e (DE OLIVEIRA; DE ALMEIDA; MONTICELLI, 1999) a idéia de times assíncronos é aplicada ao problema do planejamento da expansão da transmissão utilizando métodos heurísticos construtivos; os algoritmos propostos nesses artigos são mais detalhados em (DE ALMEIDA, 1998) e em (DE OLIVEIRA, 2004) é apresentado um time assíncrono de metaheurísticas para o problema do planejamento.

### 3.2 Times Assíncronos: Visão Geral

Segundo (TALUKDAR; DE SOUZA, 1992) a idéia de times assíncronos (*A-Teams*) surgiu de sistemas naturais e sintéticos que combinam características individuais dos agentes tais como: insetos, pássaros, peixes, humanos, livros ou computadores para formar superagentes, tais como: colônia de insetos, bandos de pássaros, cardumes de peixes, corporações,

bibliotecas ou clusters de computadores. Desta maneira, define-se *A-Team* como qualquer super-agente cujos agentes são autônomos, as comunicações são assíncronas e o fluxo de dados é cíclico.

A facilidade na utilização de times assíncronos se encontra na capacidade de inserir ou retirar os agentes autônomos sem a necessidade de um agente supervisor e a competência em encontrar soluções de boa qualidade pode ser associada à habilidade individual de cada agente e a utilização de recursos computacionais distribuídos.

Um *A-Team* consiste de uma rede cíclica onde agentes autônomos cooperam, trabalhando todo o tempo em paralelo, sem sincronismo e com a partilha permanente dos resultados entre os membros do time, ou seja, as soluções geradas por um processo podem ser reutilizadas por outro, (PEIXOTO; DE SOUZA, 1994b) (PEIXOTO; DE SOUZA, 1994a).

Para entender melhor o *A-Team* faz-se necessário uma observação das suas características:

- **Agentes de construção** - são a parte do time que resolve o problema. Esses agentes são autônomos, pois não dependem de outros agentes para realizar sua função dentro do time. Uma vez no time o agente contribui para a solução final, sendo que sua ausência não impede a solução do problema.
- **Memória central** - memória compartilhada pelos agentes, a qual é formada por uma população de configurações. O tamanho dessa população é constante, determinado pelo usuário e mantido com auxílio do agente destruidor.
- **Agente de destruição** - responsável por manter constante o tamanho da população, substituindo, por exemplo, uma melhor configuração encontrada no lugar da pior configuração armazenada na memória.
- **Comunicações assíncronas** - cada agente acessa a memória central de maneira assíncrona, ou seja, a solicitação de um dado da memória central não depende do estado dos outros agentes e sim se o agente está pronto para uma nova tarefa.
- **Fluxo de dados cíclico** - os dados (soluções contidas na memória) modificados por um agente podem ser modificados por outro agente do time; desta maneira, cada agente contribui com a sua característica (ou habilidade individual) para a melhoria das soluções como um todo.

A Figura 1, a seguir, apresenta um exemplo de *A-Team* composto por três agentes construtores, uma memória central e um agente destruidor, cujo funcionamento será discutido nas seções seguintes.

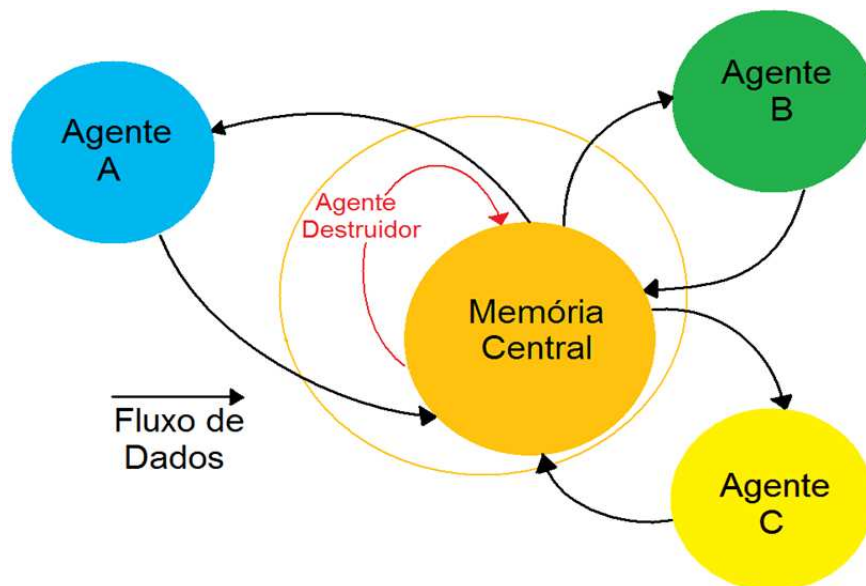


Figura 1: Time assíncrono - topologia básica.

### 3.2.1 Agentes de construção

Um *A-Team* é formado por agentes que comunicam entre si através de memórias compartilhadas. Um agente é dito autônomo se ele próprio decide sobre sua seleção de entrada, planejamento e política de alocação de recursos. Uma vez que os agentes autônomos são completamente independentes uns dos outros, novos agentes podem ser adicionados ao time ou agentes já existentes podem ser retirados sem qualquer aviso e sem qualquer alteração no trabalho dos outros agentes. Um agente consiste em um algoritmo que se propõe a resolver o problema (ou parte dele) e um protocolo de comunicação com a memória. Geralmente, devido ao porte e complexidade dos problemas tratados, os agentes despendem muito mais tempo processando dados que efetuando comunicações com as memórias. Esta característica, aliada a autonomia dos agentes e ao assincronismo das comunicações, torna os *A-Team* adequados ao processamento paralelo e distribuído. Nos *A-Team* não existem agentes supervisores. Cada agente é livre para escolher quando e que solução processar. Conseqüentemente, agentes podem entrar e sair do time a qualquer instante. A saída de um agente causa apenas uma degradação suave no time, não comprometendo de todo a geração de novas soluções (PASSOS; FONSECA, 2005).

### 3.2.2 Memória central

As memórias armazenam soluções ou informações relevantes à resolução do problema, e o fato de serem compartilhadas garante que os resultados produzidos por um agente estarão disponíveis aos demais. Os agentes podem ler e escrever nas memórias compartilhadas sempre que desejarem, conforme suas respectivas políticas de acesso as mesmas. Em um *A-Team* pode existir mais de uma memória separando, por exemplo, a qualidade das soluções armazenadas e deste modo, aprimorar a maneira que os agentes acessam a memória. Pode-se também criar grupos de agentes para agir em determinadas memórias, de modo que cada agente acesse os dados mais adequados a sua capacidade.

### 3.2.3 Agentes de destruição

Uma vez que as memórias compartilhadas não podem incorporar soluções indefinidamente, a cada uma estão associados um ou mais agentes denominados *destruidores*. As únicas funções de um agente destruidor são julgar e eliminar, baseado em uma política de destruição; o agente escolhe que solução eliminar a fim de abrir espaço a uma nova. Esta política de destruição é geralmente relacionada à qualidade dos dados. Nos *A-Team*, tão importante quanto a tarefa de produzir boas soluções é a de eliminar soluções não promissoras.

### 3.2.4 Comunicações assíncronas

O fluxo de dados é um importante parâmetro para um *A-Team* e normalmente está relacionado a quais algoritmos estão disponíveis. Dependendo da estrutura do time várias classes de memórias podem ser usadas, separando os dados já modificados e/ou melhorados de outros ainda grosseiros ou que já estão em uma fase final de refinamento, desta maneira se forma um ciclo de dados caracterizando o *A-Team*.

### 3.2.5 Fluxo de dados cíclico

No *A-Team*, cada agente tem a oportunidade de escrever sua melhor solução nas memórias, que pode então ser usada como entrada por outro agente do time. Esta interação entre os agentes leva a necessidade da existência de fluxo cíclico das soluções. Estes ciclos permitirão “feedback” e a possibilidade de um agente operar sobre uma solução criada previamente por ele e modificada pelos demais. Esta cooperação entre os agentes

aumenta as chances do time gerar soluções melhores, que não seriam geradas pelos agentes isoladamente.

## 3.3 O Ambiente distribuído

O ambiente computacional paralelo e distribuído pode ser manipulado utilizando-se várias técnicas como, por exemplo, computação em grade e paralelismo por troca de mensagens. Em se tratando de times assíncronos, a facilidade em explorar o ambiente paralelo leva a um estudo preliminar do ambiente distribuído, das bibliotecas de processamento paralelo e da máquina paralela virtual.

### 3.3.1 As Bibliotecas PVM e MPI

Grande parte das primeiras gerações de máquinas paralelas utilizava a arquitetura de troca de mensagens, uma vez que ela possuía um menor custo se comparada com a arquitetura de memória compartilhada (multiprocessadores). Neste contexto foram desenvolvidas várias bibliotecas distintas para exploração do paralelismo usando troca de mensagens. Assim, os fabricantes de máquinas paralelas disponibilizavam bibliotecas que funcionavam corretamente em seu hardware, porém eram incompatíveis com as máquinas de outros fabricantes. Muitas das diferenças encontradas nas rotinas para troca de mensagens eram apenas na sintaxe; entretanto freqüentemente essas diferenças causavam sérios problemas na portabilidade da aplicação paralela de uma biblioteca para outra.

O PVM (*Parallel Virtual Machine*) foi especialmente desenhado para interligar recursos computacionais numa rede, fornecendo ao utilizador uma plataforma paralela para rodar aplicações, independentemente da localização, da quantidade ou das diferenças entre os computadores na rede. Em PVM, um programa vê um conjunto heterogêneo de computadores de um modo uniforme como se tratasse um único computador paralelo.

O projeto PVM teve início no verão de 1989 no Oak Ridge National Laboratory com a construção de um protótipo PVM 1.0, por Vaidy Sunermam e Al Geist, que foi usado apenas internamente.

Em março de 1991, a Universidade de Tennessee desenvolveu a versão 2.0 que foi sucessivamente alterada, 2.1 - 2.4, como resultado das críticas e sugestões de usuários.

Em fevereiro de 1993 estava concluída a versão 3.3, distribuída livremente e utilizada em grande escala em projetos da época.

Em julho de 1997 foi lançada uma nova versão do PVM, presumivelmente a última. A biblioteca entrou em desuso com o aparecimento do MPI e a falta de atualizações.

O MPI (*Message Passing Interface*) tem opções mais avançadas (que o PVM), como envio de mensagens *broadcast* (para todas as máquinas do cluster) e *multicast* (para um grupo específico de máquinas), assim como um melhor controle sobre o tratamento que cada mensagem terá ao ser recebida por outro ponto do cluster (GRAMA et al., 1994).

O padrão MPI foi concluído no final do primeiro semestre de 1994 (versão 1.0) e atualizado na metade do ano seguinte (versão 1.1). O MPI-2 teve sua versão preliminar apresentada no Super Computing 96, e em abril de 1997 o documento MPI-2 foi unanimemente votado e aceito. Resultado de um esforço da comunidade científica para definir e padronizar a sintaxe e semântica de uma biblioteca de rotinas para troca de mensagens que pudesse ser implementada numa ampla variedade de máquinas paralelas. Alguns autores apontam sua utilização e aceitação atual pela colaboração dos membros que constituíram o *MPI Forum*.

O *MPI Forum* é constituído por pesquisadores, acadêmicos, programadores, usuários e fabricantes, representando aproximadamente 40 organizações. Entre os participantes do fórum pode-se destacar (CENAPAD, 2001):

- Representantes da Convex, Cray, IBM, Intel, Meiko, nCUBE, NEC e Thinking Machines;
- Membros de grupos de desenvolvedores de software tais como: PVM, P4, Zipcode, Chamaleon, PARMACS, TCGMSG e Express;
- Especialistas na área de processamento paralelo.

O MPI é uma das bibliotecas para exploração do paralelismo mais difundida, padrão para a comunicação paralela com trocas de mensagens em agregados de computadores. Ele é largamente utilizado pela disponibilidade de funcionamento em grande número das máquinas paralelas atuais. Em (REBONATO, 2004), é possível encontrar uma pequena introdução do uso da biblioteca MPI bem como a sua utilização em ambiente distribuído. Em (GROPP; LUSK, 1997) é realizada, com mais detalhes, uma comparação entre as bibliotecas de processamento paralelo PVM e MPI.

### 3.3.2 Máquina Paralela Virtual

O conceito utilizado pelas bibliotecas PVM e MPI para a execução de aplicações paralelas em multicomputadores é a criação de uma máquina paralela virtual composta dos recursos computacionais dos computadores que compõem a máquina física em questão. Por exemplo, no caso de um agregado de computadores composto de 16 máquinas mono-processadas, poderá ser composta uma máquina virtual paralela com até 16 processadores. No caso da memória, a organização em forma de multicomputador não possibilita seu compartilhamento. Dessa forma, o fato da memória estar fisicamente distribuída entre as máquinas do cluster não causará problema algum: o conjunto das memórias pertencentes às máquinas físicas que compõem o agregado também poderá ser considerada parte da máquina paralela virtual e as informações entre as memórias são passadas via troca de mensagens.

Após a máquina paralela virtual ser configurada e estar operante, deixa de existir um conjunto de máquinas interconectadas através de alguma rede passando a estar disponível uma máquina paralela (virtual). As aplicações paralelas escritas com PVM e MPI são executadas nessa máquina virtual em seus respectivos ambientes.



## 4 *Algoritmos Propostos*

Cada método heurístico construtivo possui suas particularidades, características que os fazem melhores para resolver determinados problemas ou determinadas partes de um problema maior. Partindo desta observação e baseado em uma estrutura de processamento paralelo, pode-se montar times assíncronos, formados por agentes autônomos, onde a busca por melhores resultados para a inicialização das metaheurísticas utilizadas na resolução do problema do planejamento da expansão pode ser realizada rapidamente e com soluções mais satisfatórias. Neste capítulo são apresentados os algoritmos do time assíncrono inicializador implementado e o algoritmo genético serial utilizado para testar as soluções iniciais geradas.

### 4.1 Memória Central

Em se tratando de times assíncronos, existem várias maneiras de se implementar memórias. Elas podem simplesmente armazenar as configurações possíveis ou incorporar várias tarefas importantes para o melhor desempenho do time. Neste trabalho foi implementada uma memória que além de possuir a função de armazenamento assume o papel do agente destruidor. O algoritmo correspondente a memória central, bem como o time assíncrono proposto, recebeu o nome de “*inic\_refina*” pois, ao final do processo de adição é realizado um refinamento da configuração retirando as linhas desnecessárias.

O agente destruidor é o responsável por manter constante o tamanho da população, ou seja, limitar a quantidade de configurações armazenadas na memória. Quando uma configuração é solicitada por um agente, o seu “lugar” fica reservado na memória e quando ela retorna é armazenada na mesma posição. Desta maneira, a memória assume a função do agente destruidor, mantendo constante o número de configurações disponíveis. Como neste caso específico o agente destruidor é uma função interna da memória, o termo “agente” utilizado daqui em diante fará referência somente aos agentes construtores.

Assim que o time assíncrono é iniciado, a memória já possui todas as configurações armazenadas e “zeradas” (sem a adição de nenhuma linha). Cada agente iniciado com o time solicita uma configuração na memória. Se mais de um agente solicita configuração na memória ao mesmo tempo, eles obedecem uma fila FIFO “*first in first out*” coordenada intrinsecamente pela função de envio e recebimento de mensagens do MPI. A escolha de qual configuração será enviada ao agente é realizada aleatoriamente. Uma vez escolhida uma configuração, a memória verifica se ela está bloqueada; se estiver, seleciona outra e envia para o agente.

Cada agente é responsável pela adição de um único circuito por vez; em consequência disto, a troca de informações entre memória e agentes é intensa, porém, por se tratar de arquivos pequenos e as comunicações serem assíncronas não apresentou o chamado “gargalo” na rede, nem trouxe prejuízos ao tempo final de simulação.

As configurações vão sendo preenchidas com circuitos à medida que elas passam pelos agentes. Uma vez que uma configuração atende o critério de parada, ou seja, o sistema está operando adequadamente para o modelo utilizado, ela é marcada pelo agente como “bloqueada” impedindo que outros agentes a modifiquem. Esta configuração retornada para a memória com o atributo de bloqueio passa então por um refinamento.

Durante o processo de adição de circuitos as diferentes estratégias de adição de circuitos utilizada por cada agente pode levar ao aparecimento de circuitos desnecessários em uma configuração. No refinamento a memória simula a remoção de todos os circuitos adicionados à configuração. Se a retirada do circuito não acarreta em um corte de carga ao sistema, atualiza-se a configuração sem os circuitos desnecessários e a armazena definitivamente na memória. A simulação de retirada dos circuitos também é feita segundo o modelo híbrido linear, ou seja, resolve-se o problema de PL referente a configuração do sistema modelada pela equação 2.3, resultando na estimação dos fluxos de carga na rede e se não existirem sobrecargas no sistema a configuração é atualizada sem o circuito verificado.

Quando todas as configurações estiverem marcadas com o atributo de bloqueio a memória escreve os dados de saída e finaliza o time assíncrono.

A Figura 2 mostra o fluxograma da memória central do time assíncrono implementado.

A seguir é apresentada a seqüência de decisões tomadas pelo algoritmo de memória central “*inic\_refina*”.

- **Passo 1:** Lê os dados do sistema teste, como: número de linhas  $nl$ , número de

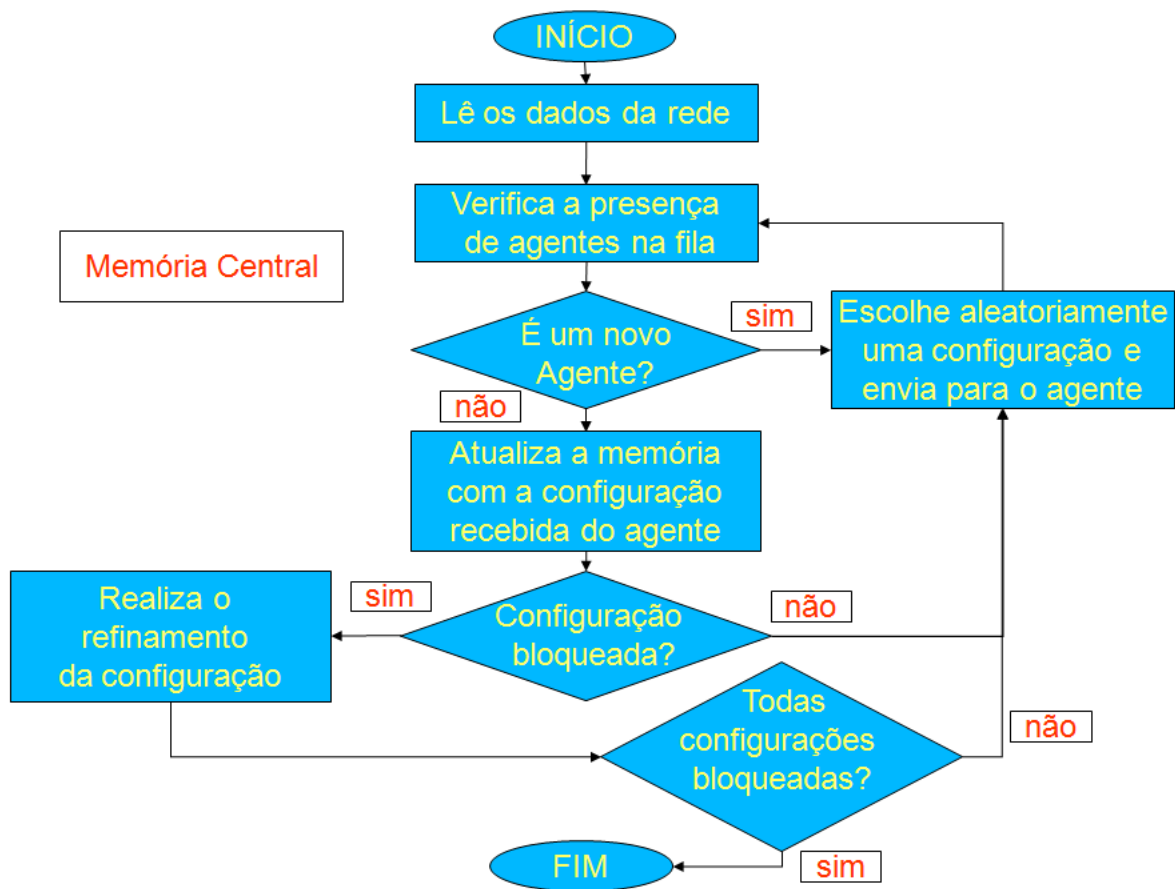


Figura 2: Memória central.

barras  $nbar$ , geração  $ger$  e demanda  $dem$  das barras, capacidade de transmissão  $maxflu$ , susceptância  $susc$  e custo das linhas  $cus$ ;

- **Passo 2:** Verifica a presença de agentes na fila. Se for um agente novo vai para o passo 5, caso contrário vai para o passo 3.
- **Passo 3:** Atualiza a memória com o configuração recebida do agente. Se a configuração estiver bloqueada vai para o passo 4, caso contrário vai para o passo 5.
- **Passo 4:** Realiza o refinamento da configuração recebida do agente. Se todas as configurações estiverem desbloqueadas PARE, caso contrário vai para o passo 5.
- **Passo 5:** Escolhe aleatoriamente uma dentre as configurações desbloqueadas, envia para o agente e retorna ao passo 2

## 4.2 Agentes construtores

Os agentes construtores são os responsáveis pela adição de circuitos às configurações. São baseados em métodos heurísticos construtivos e implementados como funções que permitem a sua comunicação com a memória central. Todos agentes foram implementados utilizando o *modelo híbrido linear* apresentado no capítulo 2, equação 2.3. O funcionamento básico dos agentes construtores implementados é o mesmo, tendo suas principais diferenças na seleção dos circuitos adicionados. O fluxograma da Figura 3 mostra a seqüência de decisões tomadas por um agente construtor.

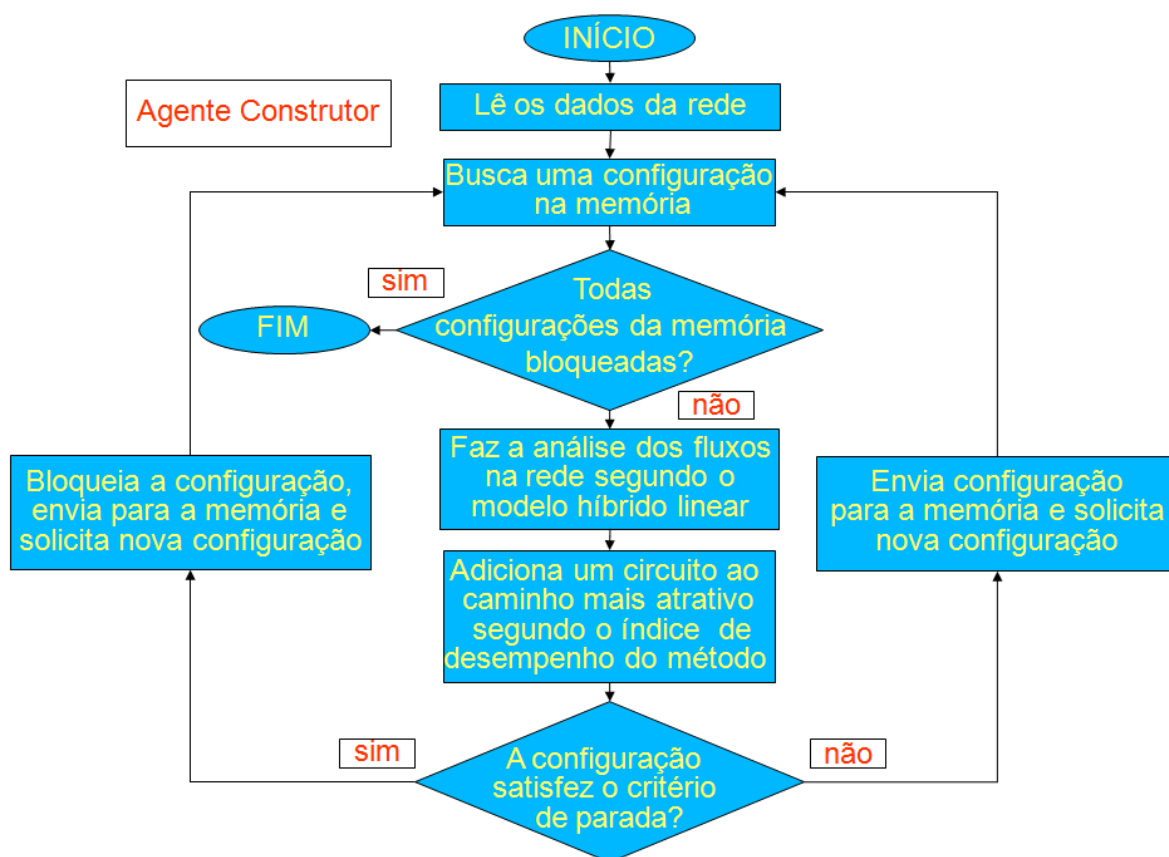


Figura 3: Agente construtor.

### 4.2.1 Agentes de Villasana-Garver-Salon

Na proposta de (VILLASANA; GARVER; SALON, 1985) foi apresentado um novo modelo matemático para o problema do planejamento, o *modelo híbrido*, e foi utilizado o mesmo critério de adição de circuitos apresentada em (GARVER, 1970). O índice de desempenho apresentado por Villasana foi mantido, ou seja, é adicionado um circuito ao caminho que apresentar o maior fluxo de sobrecarga. Baseado nesta metodologia foram implementa-

dos dois agentes construtores para o time assíncrono inicializador proposto, o “*vgsA*” e o “*vgsB*”. No agente “*vgsA*” o algoritmo sempre adiciona ao sistema o circuito que apresentar o maior fluxo de sobrecarga. Já no “*vgsB*” os fluxos de sobrecarga são ordenados de forma decrescente e é então adicionado aleatoriamente ao sistema um dos cinco circuitos mais sobrecarregados, introduzindo desta maneira uma maior diversidade nas soluções iniciais geradas. A seguir é apresentada a seqüência de decisões tomadas pelo algoritmo “*vgsA*”.

- **Passo 1:** Lê os dados do sistema teste, como: número de linhas  $nl$ , número de barras  $nbar$ , geração  $ger$  e demanda  $dem$  das barras, capacidade de transmissão  $maxflu$ , susceptância  $susc$  e custo das linhas  $cus$ ;
- **Passo 2:** Busca uma configuração na memória. Se existirem configurações disponíveis na memória vá para o passo 3. Caso contrário PARE.
- **Passo 3:** Realiza a análise do fluxo de carga segundo o *modelo híbrido linear* para todos os ramos da rede.
- **Passo 4:** Ordena os ramos de maneira decrescente segundo seus fluxos e adiciona ao sistema uma linha ao ramo mais sobrecarregado, ou seja o ramo que apresentar o maior fluxo de sobrecarga.
- **Passo 5:** Se não existirem mais sobrecargas no sistema, marca a configuração como bloqueada (impedindo que a memória envie esta configuração para outro agente), envia para a memória e retorna ao passo 2; caso contrário vá para o passo 6.
- **Passo 6:** Envia a configuração de volta à memória e retorna ao passo 2.

Esta seqüência de passos, exceto o passo 4, é repetida pelo algoritmo “*vgsB*” que no passo 4 adiciona ao sistema uma linha a um dos cinco ramos mais sobrecarregados escolhido aleatoriamente.

## 4.2.2 Agentes Mínimo Esforço

No método do mínimo esforço apresentado em (MONTICELLI et al., 1982) foi utilizado o *modelo DC*. Fundamentado neste modelo, o índice de desempenho empregado na identificação dos circuitos mais atrativos para a adição ao sistema leva em conta o quadrado da diferença angular das tensões nodais, resultado das restrições correspondentes

a lei das tensões de Kirchhoff. No algoritmo aqui proposto, não foi possível utilizar o índice de desempenho da maneira proposta originalmente, pois os circuitos candidatos a adição obedecem somente à lei das correntes de Kirchhoff. A adaptação deste índice de desempenho para utilizar o *modelo híbrido linear* da equação 2.3 pode ser observada nas formulações a seguir.

Índice proposto originalmente.

$$IS_{me} = Z_{ij} = -\frac{1}{2}(\theta_i - \theta_j)^2 \gamma_{ij} \quad (4.1)$$

porém,

$$(\theta_i - \theta_j) = \frac{f_{ij}}{\gamma_{ij}} \quad (4.2)$$

substituindo a equação 4.2 em 4.1, obtém-se:

$$IS_{me} = Z_{ij} = -\frac{1}{2} \frac{f_{ij}^2}{\gamma_{ij}} \quad (4.3)$$

Com base neste índice de desempenho modificado foram implementados dois agentes para o método de mínimo esforço. O agente “*minhA*” que adiciona ao sistema o circuito que apresentar o maior índice de desempenho e o “*minhB*” que escolhe aleatoriamente um dentre os cinco circuitos com melhor índice. A seguir é apresentada a seqüência de decisões tomadas pelo algoritmo “*minhA*”.

- **Passo 1:** Lê os dados do sistema teste, como: número de linhas *nl*, número de barras *nbar*, geração *ger* e demanda *dem* das barras, capacidade de transmissão *maxflu*, susceptância *susc* e custo das linhas *cus*;
- **Passo 2:** Busca uma configuração na memória. Se existirem configurações disponíveis na memória vá para o passo 3. Caso contrário PARE.
- **Passo 3:** Realiza a análise do fluxo de carga segundo o *modelo híbrido linear* para todos os ramos da rede.
- **Passo 4:** Ordena os ramos de maneira decrescente segundo seus índices de desempenho e adiciona ao sistema uma linha ao ramo que apresentar o maior índice.
- **Passo 5:** Se não existirem mais sobrecargas no sistema, marca a configuração como

bloqueada (impedindo que a memória envie esta configuração para outro agente), envia para a memória e retorna ao passo **2**; caso contrário vá para o passo **6**.

- **Passo 6:** Envia a configuração de volta a memória e retorna ao passo **2**.

Esta seqüência de passos, exceto o passo **4**, é repetida pelo algoritmo “*minhB*” que no passo “**4**” adiciona ao sistema uma linha a um dos cinco ramos com melhor índice de desempenho escolhido aleatoriamente.

### 4.2.3 Agentes Mínimo Corte de Carga

Como no método do mínimo esforço, o método do mínimo corte de carga original utiliza o *modelo DC* e um índice de sensibilidade para a adição de linhas baseado nos resultados da lei das tensões de Kirchhoff sobre os circuitos candidatos. Da mesma maneira que no método do mínimo esforço, o índice de sensibilidade teve de ser modificado para atender o modelo proposto. Outra modificação foi que na proposta original as faltas no sistema eram contornadas pela existência de geradores artificiais em todos os nós e no algoritmo proposto as sobrecargas aparecem nos circuitos candidatos, sendo assim, o agente implementado minimiza as sobrecargas no sistema analisado adicionando circuitos segundo o índice de sensibilidade a seguir.

$$IS_{mcc} = (\pi_i - \pi_j) \frac{f_{ij}}{\gamma_{ij}} \quad (4.4)$$

em que:

$\pi_j$  é o multiplicador de Lagrange da *j*-ésima restrição ( $B\theta + g + r = d$ ).

Deste método também foram implementados dois agentes o “*mcchA*” que adiciona um circuito ao ramo que apresentar o maior corte de carga, e o “*mcchB*” que faz o ordenamento dos circuitos de acordo com o índice da equação 4.4 e adiciona um dentre os cinco que apresentarem o maior corte de carga. A seguir é apresentada a seqüência de decisões tomadas pelo algoritmo “*mcchA*”.

- **Passo 1:** Lê os dados do sistema teste, como: número de linhas *nl*, número de barras *nbar*, geração *ger* e demanda *dem* das barras, capacidade de transmissão *maxflu*, susceptância *susc* e custo das linhas *cus*;
- **Passo 2:** Busca uma configuração na memória. Se existirem configurações disponíveis na memória vá para o passo **3**. Caso contrário PARE.

- **Passo 3:** Realiza a análise do fluxo de carga segundo o *modelo híbrido linear* para todos os ramos da rede.
- **Passo 4:** Ordena os ramos de maneira decrescente segundo seus índices de sensibilidade e adiciona ao sistema uma linha ao ramo que apresentar o maior índice.
- **Passo 5:** Se não existirem mais sobrecargas no sistema, marca a configuração como bloqueada (impedindo que a memória envie esta configuração para outro agente), envia para a memória e retorna ao passo **2**; caso contrário vá para o passo **6**.
- **Passo 6:** Envia a configuração de volta a memória e retorna ao passo **2**.

Esta seqüência de passos, exceto o passo **4**, é repetida pelo algoritmo “*mcchB*” que no passo “**4**” adiciona ao sistema uma linha a um dos cinco ramos com maior índice de sensibilidade (corte de carga) escolhido aleatoriamente.

## 4.3 Algoritmo Genético Utilizado

O algoritmo genético serial aqui estudado foi desenvolvido em (GALLEGO; DE OLIVEIRA, 1997), utilizando estações SUN com sistema operacional SOLARIS e largamente detalhado em (GALLEGO, 1997). O algoritmo que leva o nome de GANORD sofreu algumas modificações para rodar em sistemas Linux, mas somente em instruções de compilação e nomenclatura de subrotinas. As estruturas básicas para este AG são descritas no capítulo 2, seção 2.2.2.2.2. Na seqüência, é brevemente detalhado o **AG** do modo que foi utilizado neste trabalho.

### 4.3.1 Descrição do algoritmo GANORD

- **Passo 1:** Ler o tamanho da população  $n_{pop}$  e o número de configurações iniciais  $n_{para}$  (determinado previamente com o auxílio do programa inicializador *inic\_refina*). Ler  $j_{gera}$  (número máximo de gerações permitidas).
- **Passo 2:** Para cada configuração ou elemento da população, determinar o valor da função objetivo (“fitness”) correspondente, usando o algoritmo de PL (*MINOS*) (MURTAGH; SAUNDERS, 1987).

Atualizar, se possível, a incumbente.  $k_2$  conta o número de gerações.



- **Passo 3:** Se  $k_2 > k_{gera}$  PARE, pois foi excedido o número máximo de gerações permitidas. Caso contrário, vá ao passo 4.
- **Passo 4:** Usando o mecanismo de seleção proporcional ou o mecanismo de seleção com limitação de número de descendentes, determinar o número de descendentes (“offsprings”) correspondente a cada configuração.
- **Passo 5:** Realizar recombinação de maneira sistemática entre cada par de configurações que devem participar na formação da nova geração, até gerar um número de descendentes igual ao tamanho da população. A taxa de recombinação ( $ycross$ ), controla estatisticamente o número de configurações que serão recombinadas.
- **Passo 6:** Implementar a mutação de maneira sistemática. Cada configuração candidata realiza até  $n_m$  mutações (simples ou duplas) onde  $n_m$  é proporcional à taxa de mutação  $nmutar$ . Este processo também é controlado estatisticamente pela taxa de mutação. Voltar ao passo 2.

Os parâmetros de controle utilizados para calibrar este AG são:

- $nafa$  - parâmetro de penalização dos cortes de cargas;
- $npop$  - tamanho da população;
- $kincumb$  - variável de transformação do problema de maximização em um problema de minimização;
- $jgera$  - número de gerações;
- $npara$  - quantidade de indivíduos da população iniciados com o  $inic_refina$ ;
- $ycross$  - taxa de recombinação;
- $nmax$  - número máximo de linhas por ramo;
- $bconst$  - taxa de variação no número de mutações;
- $nmutar$  - taxa de mutação;
- $kseed$  - semente inicial;
- $ndes$  - número máximo de descendentes, em porcentagem;
- $temp$  - temperatura inicial do processo **SA**;

- *alfat* - taxa de resfriamento;
- *mcor* - limite de corte de garga na mutação;
- *ncl* - número máximo de caminhos a ser testados.

## 5 *Resultados*

Nas simulações realizadas neste trabalho utilizou-se os computadores do LLPP - Laboratório de Linux e Processamento Paralelo, do Departamento de Engenharia Elétrica da UNESP Campus de Ilha Solteira. O ambiente físico bastante heterogêneo possui duas máquinas com processador Intel Pentium 4 de 3,0 GHz com sistema operacional Fedora core 5, três máquinas com processador AMD Sempron 2800+ de 2,0 GHz com sistema operacional Fedora core 4, uma máquina com processador AMD Sempron 1900+ de 1,6 GHz com sistema operacional Fedora core 4 e uma máquina com processador AMD Sempron 2800+ de 2,0 GHz com sistema operacional Fedora core 5. Os algoritmos foram implementados em FORTRAN da INTEL versão 8.1, utilizando a biblioteca de processamento paralelo LAM-MPI.

Das máquinas disponíveis, uma é sempre utilizada para a memória central e as restantes foram utilizadas conforme os diferentes arranjos de times assíncronos, sendo que cada máquina é responsável por um único agente construtor a cada simulação.

Pela heterogeneidade do ambiente e como o laboratório é acessado por diversos usuários local ou remotamente, os algoritmos simulados concorrem com outros processos existentes nas máquinas. Isso leva a uma aleatoriedade nos tempos de acesso dos agentes à memória, contribuindo para a geração de soluções diversificadas, sendo esta uma qualidade desejada em um algoritmo inicializador, ou seja, o compartilhamento do ambiente com outros usuários durante o processamento dos times assíncronos não prejudica os seus desempenhos.

Neste trabalho, no intuito de validar as configurações iniciais geradas com os times assíncronos baseados no método híbrido linear, foi utilizado a metaheurística algoritmo genético, particularmente o algoritmo genético serial GANORD (GALLEGO; DE OLIVEIRA, 1997).

## 5.1 Geração das Soluções Iniciais

A busca por boas soluções para o planejamento da expansão da transmissão, utilizando metaheurísticas, deve começar com a escolha de configurações iniciais de boa qualidade, pois os desempenhos das metaheurísticas na obtenção dos ótimos globais ou das melhores soluções dependem de como os mesmos foram inicializadas.

Para validar o desempenho dos algoritmos propostos foram geradas configurações iniciais para os seguintes sistemas teste: Garver (6 barras/ 15 ramos) de pequeno porte, sistema Sul brasileiro (46 barras / 79 ramos) médio porte e sistema Norte-Nordeste brasileiro (87 barras / 179 ramos) de grande porte. Os dados destes sistemas são apresentados no apêndice A.

Em todos os testes realizados foram utilizadas oito configurações iniciais para o sistema Garver, quarenta para o sistema Sul brasileiro e cento e oitenta para o Norte-Nordeste, sendo estas configurações geradas com os diferentes arranjos de times. O número de configurações iniciais foi mantido também para as simulações do GANORD com inicialização zerada, isto é, iniciado apenas com a configuração base sem adições.

Mesmo sendo o planejamento da expansão uma tarefa que não exige processamento em tempo real, a natureza combinatória do problema leva a um elevado número de configurações possíveis e torna o processo muito lento, justificando o uso de técnicas para reduzir o tempo na busca de boas soluções. No gráfico da Figura 4 pode-se observar que o aumento no número de agentes  $vgsB$  no time assíncrono traz uma redução no tempo total gasto para gerar cento e oitenta configurações iniciais para o sistema Norte-Nordeste plano 2008, demonstrando a escalabilidade dos times assíncronos propostos.

A Tabela 3 mostra exatamente a importância dos times assíncronos de diferentes heurísticas, pois para o mesmo número de agentes e máquinas utilizados, o time composto por um representante de cada agente implementado gera as configurações em um melhor tempo quando comparado com times de uma única heurística. No sistema Garver com redespacho, por exemplo, observa-se que o tempo para as configurações geradas com o time de *todos* os agentes é dezessete segundos mais rápido que o tempo para o melhor time de uma única heurística ( $vgsB$ ). Sendo este comportamento observado em todos os sistemas teste.

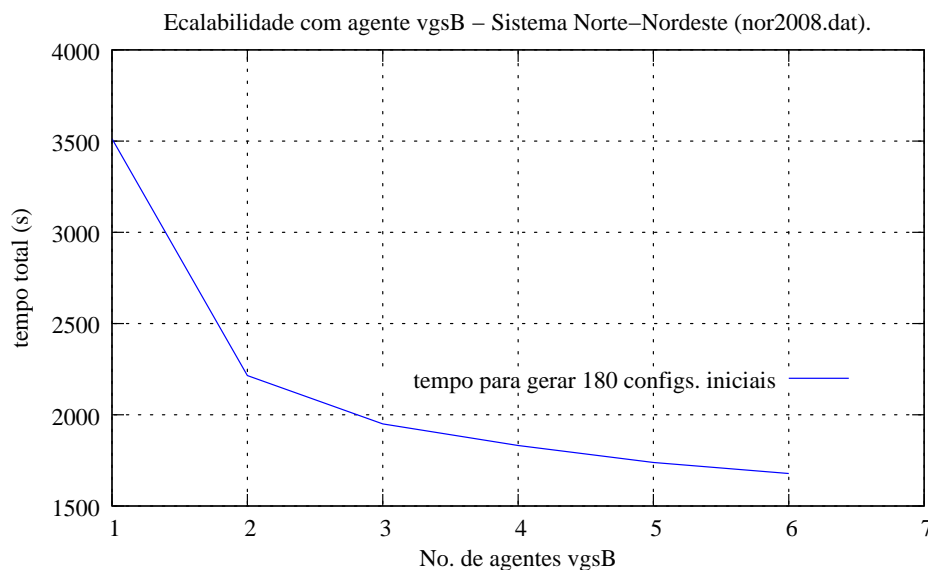


Figura 4: Exemplo de escalabilidade do time assíncrono com agente *vgsB*.

Tabela 3: Tempo total para gerar configurações iniciais.

Sistema teste	Tempo			
	Composição do Time Assíncrono			
	Todos os agentes	6 mcchB	6 minhB	6 vgsB
Garver c/ redesp.	0,26 s	1,28 s	0,53 s	0,43 s
Garver s/ redesp.	0,46 s	0,65 s	0,53 s	0,54 s
Sul Bras. c/ redesp.	7,39 s	10,30 s	10,77 s	9,62 s
Sul Bras. s/ redesp.	10,02 s	13,20 s	12,23 s	12,02 s
Nor-Nord.-2002	14,94 min	16,83 min	15,78	15,00 min
Nor-Nord.-2008	17,57 min	28,79 min	20,11 min	27,97 min

## 5.2 Simulações com o algoritmo GANORD

Dentre as várias metaheurísticas existentes, o algoritmo genético foi com certeza a mais investigada. A calibração de seus parâmetros relativamente mais simples, comparado com outras metaheurísticas, e o fato de já existirem trabalhos relacionados a algoritmos genéticos para o planejamento da expansão, com o uso de soluções iniciais de boa qualidade (GALLEGO, 1997) e (DE OLIVEIRA, 2004), foram determinantes na escolha do AG para validar as soluções geradas com os times assíncronos propostos.

Na seqüência, são apresentados alguns resultados para os sistemas Garver, com e sem redespacho; Sul brasileiro, com e sem redespacho; Norte-Nordeste brasileiro, planos 2002

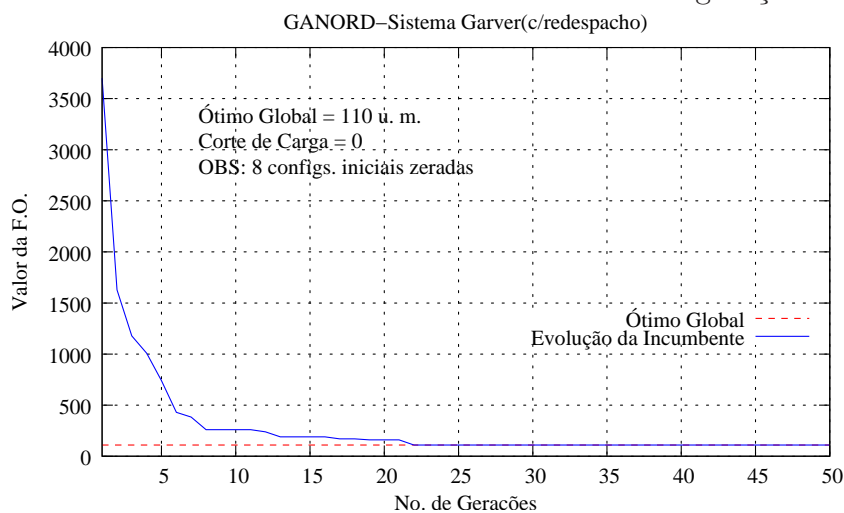
e 2008. Em cada tabela, são apresentados: um gráfico com a evolução da incumbente, os parâmetros utilizados na calibração do algoritmo genético e o vetor contendo as adições de linhas para a melhor configuração encontrada.

## 5.2.1 Sistema Garver

### 5.2.1.1 Sistema Garver com redespacho

O algoritmo genético GANORD, simulado para o sistema Garver com redespacho, convergiu para a configuração ótima em 22 gerações, com 591 PPLs em um tempo de 2,65 s quando inicializado com configurações zeradas (configuração base sem adições), como mostra o gráfico da Tabela 4. No gráfico da Tabela 5, observa-se a vantagem de se utilizar configurações iniciais de boa qualidade, pois o algoritmo genético convergiu em 4 gerações, com 60 PPLs em um tempo de 0,30 s quando simulado com configurações geradas pelo time inicializador composto por 6 agentes *mcchB*. E na Tabela 6 o gráfico mostra que o algoritmo convergiu em 2 gerações, com 30 PPLs em 0,11 s para o time de 6 agentes *minhB*. Para o time formado por 6 agentes *vgsB* e o time com *todos* os agentes (uma versão de cada agente descrito no capítulo 4) a configuração ótima foi encontrada durante a inicialização, ou seja, dentre as configurações geradas pelo time assíncrono já se encontra a melhor configuração possível.

Tabela 4: Sistema Garver - 8 configurações iniciais zeradas.



a) gráfico;

0	0	0	0	0	0	0	0	0	0	0	1	0	0	3	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

b) configuração ótima;

parâmetro	valor
nafa	10
npop	8
kincumb	4000
jgera	50
npara	8
ycross	0.97
nmax	4
bconst	0.996
nmutar	4
kseed	1112
ndes	5%
temp	250
alfat	0.93
mcor	0
ncl	0

c) parâmetros.

Tabela 5: Sistema Garver - 8 configurações iniciais geradas com 6 agentes mcchB.

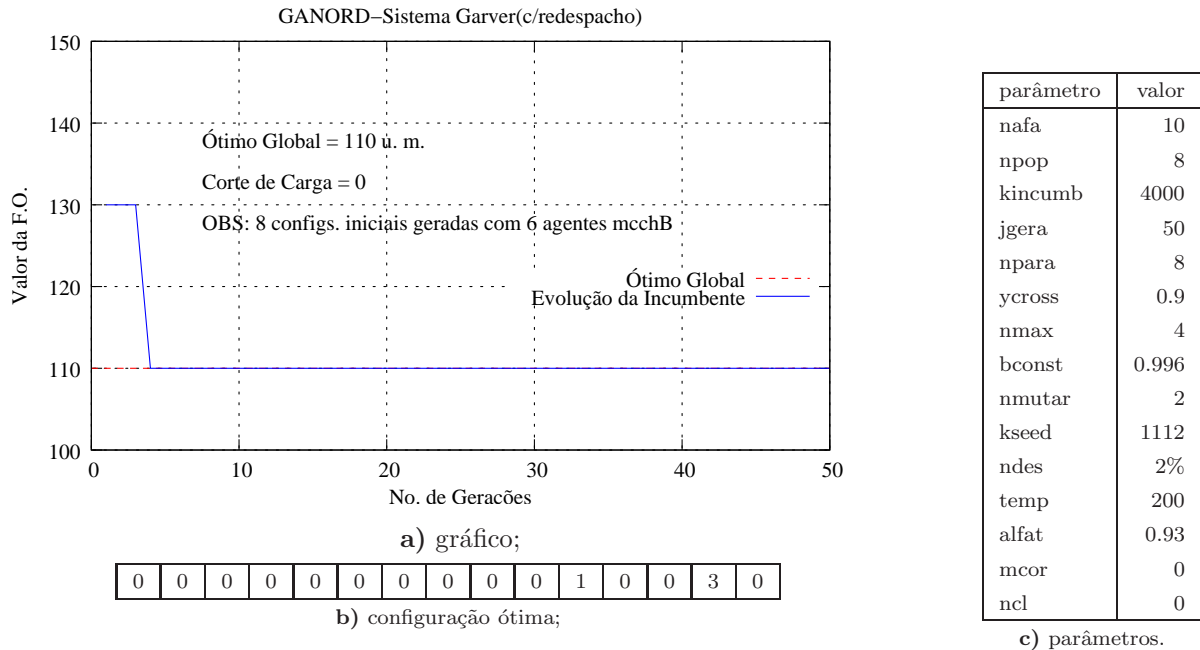
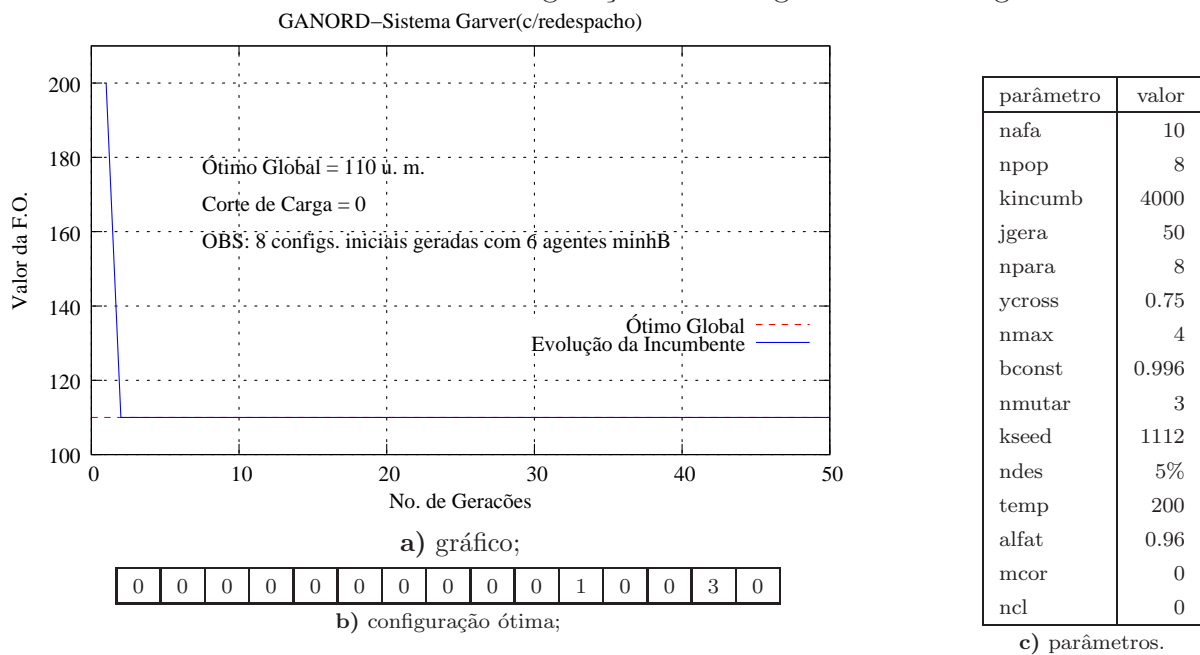


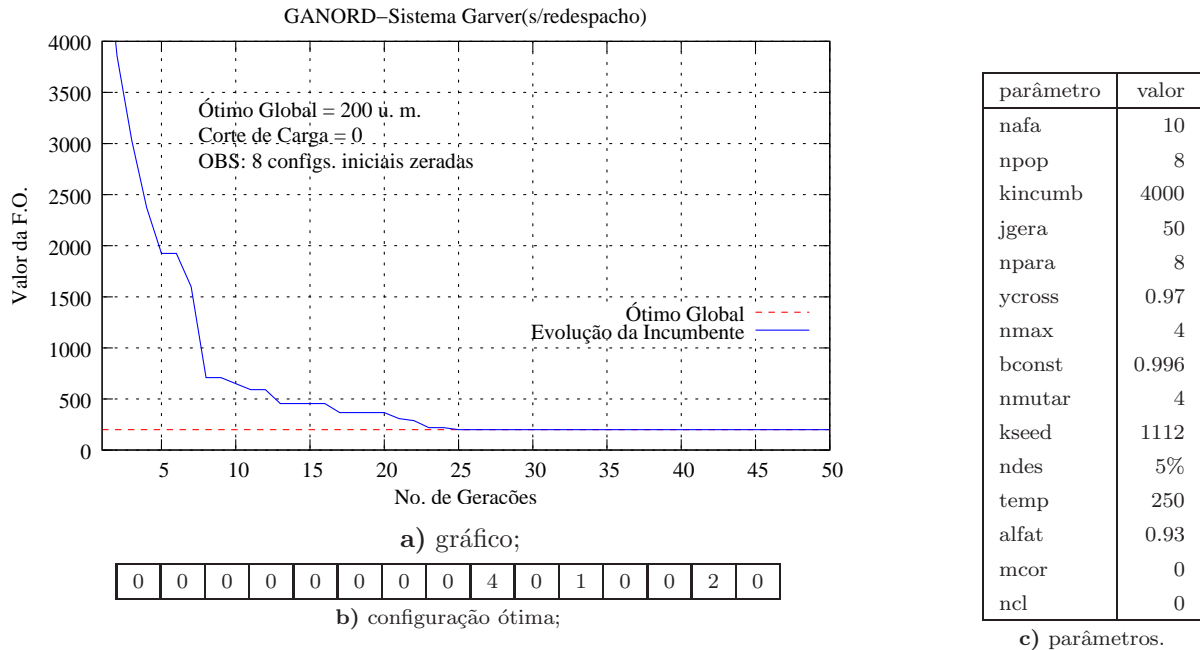
Tabela 6: Sistema Garver - 8 configurações iniciais geradas com 6 agentes minhB.



### 5.2.1.2 Sistema Garver sem redespacho

A Tabela 7 mostra o resultado da simulação em que o algoritmo GANORD convergiu em 25 gerações, com 667 PPLs em um tempo de 2,49 s, com 8 configurações iniciais zeradas para o sistema Garver sem redespacho.

Tabela 7: Sistema Garver - 8 configurações iniciais zeradas.



Não é necessária a apresentação dos gráficos dos arranjos de times assíncronos formados por agentes das três metodologias propostas, inclusive o time com todos os agentes, porque estes foram capazes de encontrar a solução ótima (200 u.m.) entre as configurações iniciais geradas para o sistema Garver sem redespacho.

## 5.2.2 Sistema Sul brasileiro

### 5.2.2.1 Sistema Sul brasileiro com redespacho

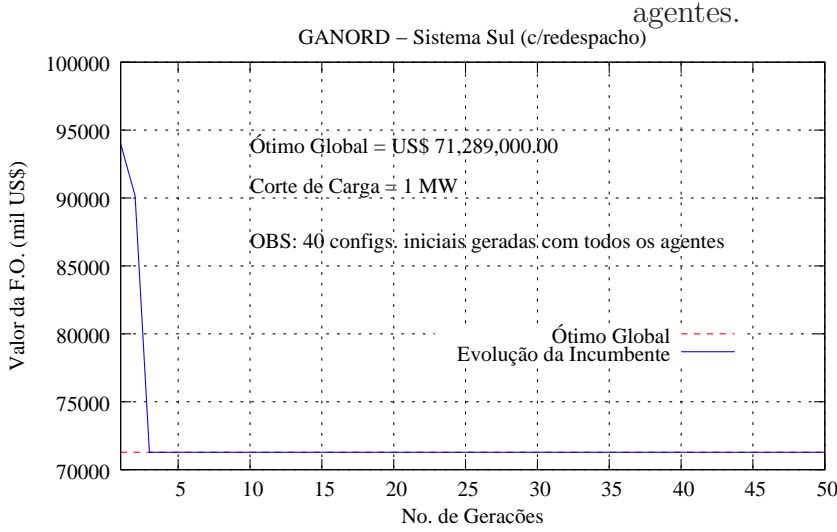
O gráfico da Tabela 8 mostra a evolução da incumbente do algoritmo genético, inicializado com configurações zeradas, que só atinge o ótimo depois de 13 gerações, 1.650 PPLs em 13,75 s. A Tabela 9 traz a simulação do AG com a inicialização do time de uma única heurística que apresentou o melhor desempenho, o *mcchB*, encontrando a configuração ótima em 8 gerações (770 PPLs em 6,16 s).

Pode-se salientar o melhor desempenho do time com *todos* agentes em relação ao time de agentes *mcchB*, observando o gráfico da Tabela 10 quando o algoritmo converge com 3 gerações (242 PPLs em 2,31 s).





Tabela 10: Sistema Sul brasileiro - 40 configurações iniciais geradas com todos os agentes.



a) gráfico;

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	1	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0

b) configuração ótima;

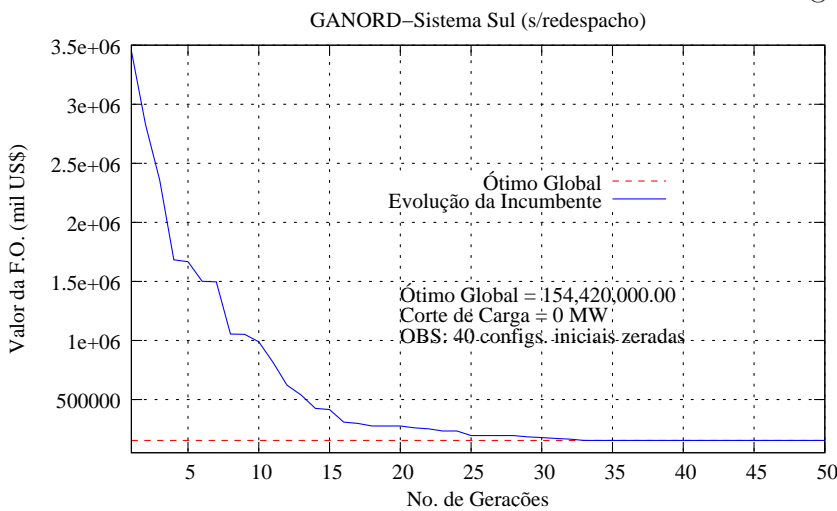
parâmetro	valor
nafa	1000
npop	40
kincumb	2500000
jgera	50
npara	40
ycross	0.92
nmax	4
bconst	0.996
nmutar	3
kseed	1112
ndes	4%
temp	25000
alfat	0.90
mcor	0
ncl	0

c) parâmetros.

### 5.2.2.2 Sistema Sul brasileiro sem redespacho

O gráfico da Tabela 11 mostra a evolução da incumbente ao longo das 33 gerações (3.801 PPLs em 32 s) que o algoritmo leva até encontrar a solução ótima quando inicializado com configurações zeradas.

Tabela 11: Sistema Sul brasileiro - 40 configurações iniciais zeradas.



a) gráfico;

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0	0	0	0	0	1	0	0	0	0	1	0	0
1	0	1	0	3	0	0	0	0	0	0	0	2	2	0	0	2	0	0	0

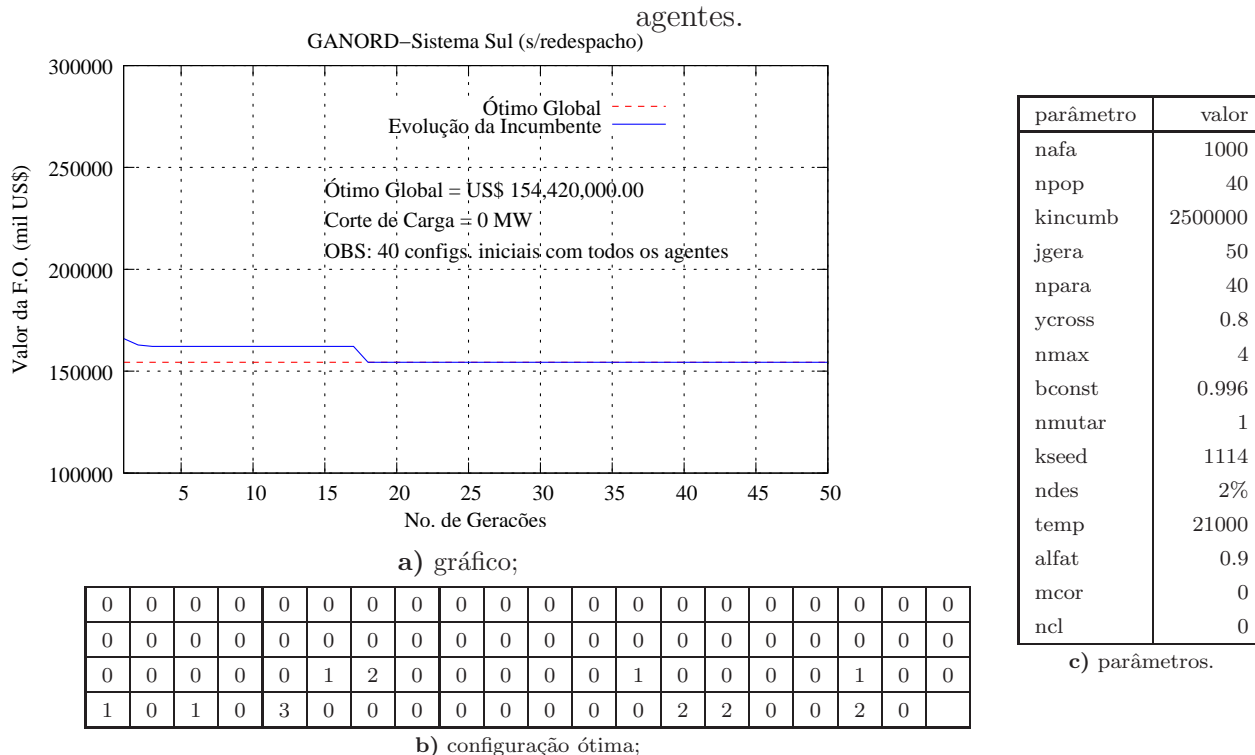
b) configuração ótima;

parâmetro	valor
nafa	1000
npop	40
kincumb	2500000
jgera	50
npara	40
ycross	0.92
nmax	4
bconst	0.996
nmutar	3
kseed	1112
ndes	5%
temp	34500
alfat	0.96
mcor	0
ncl	1

c) parâmetros.

O ótimo global foi atingido em 18 gerações (1.216 PPLs em 12,36 s) para a inicialização com *todos* os agentes, conforme mostra a Tabela 12.

Tabela 12: Sistema Sul brasileiro - 40 configurações iniciais geradas com todos os agentes.



Os times assíncronos compostos por agentes de uma única heurística geram configurações iniciais muito semelhantes. Isso impede que o algoritmo vá para novas regiões de busca, ficando então preso em regiões de ótimo local encontrado logo nas primeiras gerações. Os resultados das simulações realizadas com esses arranjos de times encontram-se nas tabelas 18, 19 e 20.

### 5.2.3 Sistema Norte-Nordeste brasileiro

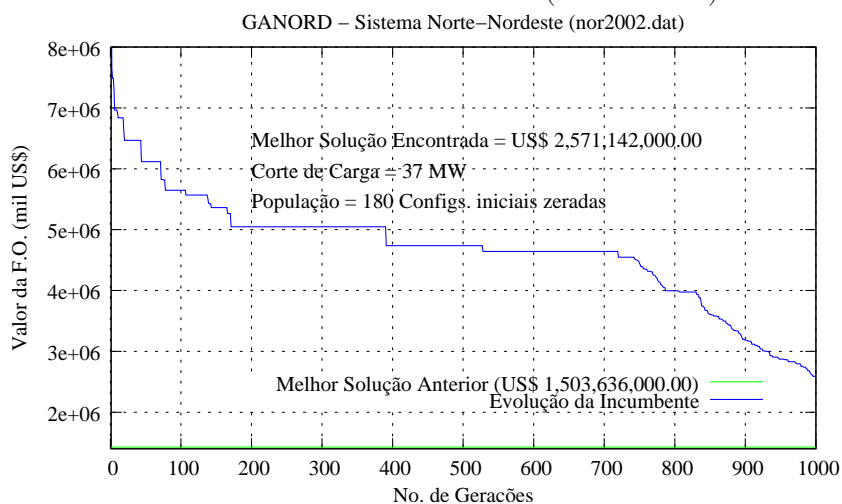
Para as simulações do sistema Norte-Nordeste brasileiro foram usados como valores de referência as melhores soluções encontradas com o algoritmo genético GANORD em (DE OLIVEIRA, 2004), pois as mesmas foram obtidas utilizando as soluções geradas com o time assíncrono inicializador desenvolvido em (DE OLIVEIRA; DE ALMEIDA; MONTICELLI, 1999).

#### 5.2.3.1 Sistema Norte-Nordeste brasileiro - Plano 2002

Por se tratar de um sistema de grande porte torna-se inviável a simulação do GANORD com configurações iniciais zeradas, porque mesmo depois de 1.000 gerações (236.997

PPLs em 71,24 min) a solução encontrada é mais de US\$ 1 bilhão mais cara que o valor de referência, como pode ser observado no gráfico da Tabela 13.

Tabela 13: Sistema Norte Nordeste (Plano 2002) - 180 configurações iniciais zeradas.



a) gráfico;

0	0	0	2	0	0	1	2	1	0	0	0	0	1	0	0	0	0	2	
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	1	1	2	0
0	0	3	1	1	0	0	0	6	0	3	0	0	1	1	0	0	0	0	0
1	0	0	1	0	2	0	0	1	0	0	0	0	0	1	2	0	0	0	0
0	0	0	0	0	1	0	0	1	0	2	0	0	1	2	0	0	1	0	0
2	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	2	0	0
0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

b) melhor configuração encontrada

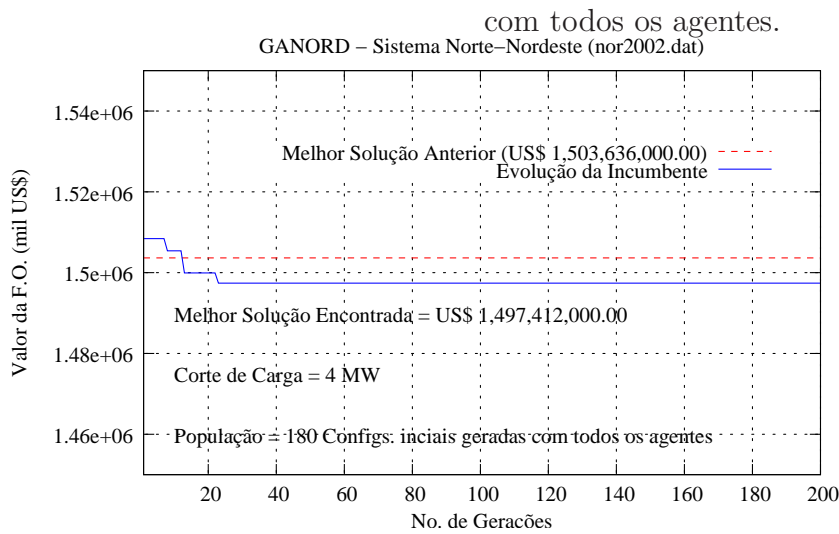
parâmetro	valor
nafa	1000
npop	180
kincumb	4500000
jgera	1000
npara	180
ycross	0.8
nmax	12
bconst	0.996
nmutar	5
kseed	1112
ndes	2%
temp	50000
alfat	0.997
mcor	300
ncl	30

c) parâmetros.

No gráfico da Tabela 14 pode-se observar que a solução encontrada em apenas 13 gerações (4.955 PPLs em 1,35 min.) inicializada com um times composto por *todos* os agentes propostos, é melhor que a solução de referência. Isso comprova a eficácia dos time assíncronos baseados no modelo híbrido linear quando aplicados a geração de soluções iniciais para a metaheurística AG.

Os resultados das simulações realizadas com o GANORD, com configurações iniciais geradas por times de uma única heurística, ficaram presos às soluções encontradas durante a geração das configurações, apresentando soluções de alto custo de investimento. Os resultados destas simulações se encontram nas tabelas 18, 19 e 20.

Tabela 14: Sistema Norte Nordeste (Plano 2002) - 180 configurações iniciais geradas



parâmetro	valor
nafa	1000
npop	180
kincumb	4500000
jgera	200
npara	180
ycross	0.8
nmax	12
bconst	0.996
nmutar	2
kseed	1211
ndes	3%
temp	20000
alfat	0.998
mcor	300
ncl	30

a) gráfico;

0	0	0	2	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	2	
0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	1	2	1
0	0	3	0	1	0	0	0	6	0	3	0	0	1	1	0	0	0	0	0	0
1	0	0	1	0	3	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	1	0	2	0	0	2	2	0	0	1	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	2	1	0	0
0	0	0	0	0	0	0	2	1	0	0	0	0	1	0	0	1	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

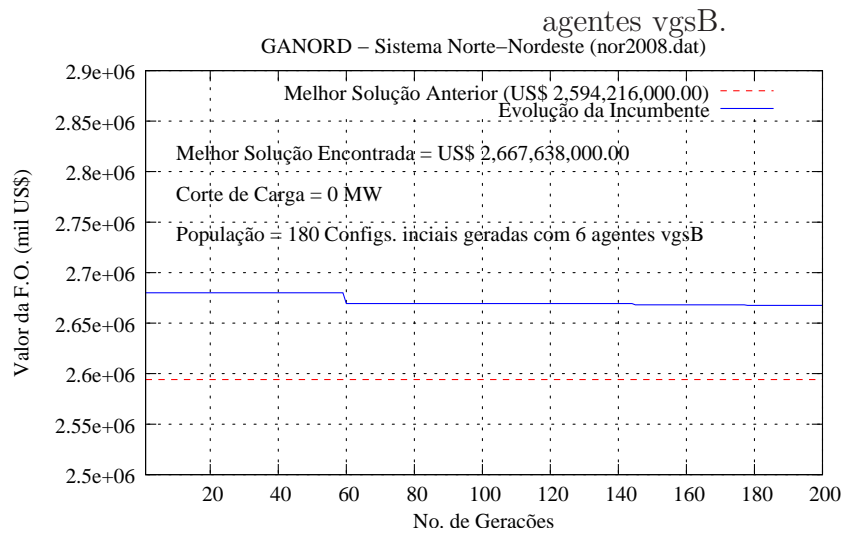
b) melhor configuração encontrada

c) parâmetros.

### 5.2.3.2 Sistema Norte-Nordeste brasileiro - Plano 2008

O gráfico da Tabela 15 traz o melhor resultado dentre as simulações realizadas com o sistema Norte-Nordeste plano 2008, convergindo para o ótimo local de US\$ 2,667,638,000.00 em 178 gerações (87.891 PPLs em 30,36 min), inicializado com configurações geradas pelo time de agentes *vgsB*. Este resultado foi ainda melhor que o resultado da simulação inicializada com todos os agentes, Tabela 16, que encontrou uma solução ótima local de valor US\$ 2,674,442,000.00 (62.083 PPLs em 20,72 min).

Tabela 15: Sistema Norte Nordeste (Plano 2008) - 180 configurações iniciais com 6



parâmetro	valor
nafa	1000
npop	180
kincumb	4500000
jgera	200
npara	180
ycross	0.8
nmax	12
bconst	0.996
nmutar	4
kseed	1211
ndes	2%
temp	150000
alfat	0.996
mcor	300
ncl	30

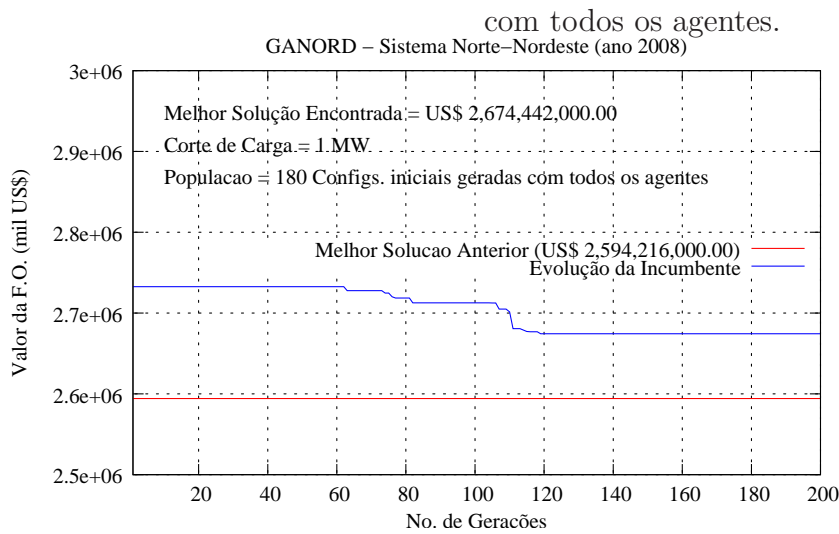
a) gráfico;

1	0	1	0	0	0	1	1	1	1	0	1	0	0	3	0	0	0	1	3
2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	4	0	0	0	0	0	1	4	1
1	0	6	0	1	0	0	0	11	0	6	0	0	0	0	0	0	2	0	0
1	0	0	1	0	4	0	1	0	0	0	0	0	1	1	2	0	0	0	0
1	0	0	1	1	3	0	0	2	0	2	1	0	3	3	0	0	1	0	0
4	0	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	3	0	0
0	0	1	0	0	1	0	0	1	2	0	0	0	0	0	0	1	1	2	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

b) melhor configuração encontrada

c) parâmetros.

Tabela 16: Sistema Norte Nordeste (Plano 2008) - 180 configurações iniciais geradas



a) gráfico;

1	0	0	1	0	0	1	2	2	0	0	0	1	0	3	0	0	0	1	3
0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	4	0	0	0	0	0	1	4	1
1	0	6	0	1	0	0	0	11	0	6	0	0	0	0	0	0	2	0	1
1	0	0	1	0	4	0	0	0	0	0	0	0	0	1	2	0	0	0	0
0	0	0	1	0	2	0	1	2	0	2	1	0	3	3	0	0	0	0	0
5	0	1	1	0	0	1	0	0	1	0	0	1	0	0	0	1	3	1	0
0	0	0	0	0	1	0	0	1	2	0	0	0	0	0	0	1	1	2	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

b) melhor configuração encontrada

parâmetro	valor
nafa	1000
npop	180
kincumb	4500000
jgera	200
npara	180
ycross	0.8
nmax	12
bconst	0.996
nmutar	4
kseed	1211
ndes	2%
temp	150000
alfat	0.996
mcor	50
ncl	30

c) parâmetros.

## 5.3 Resumo dos Resultados

Para cada sistema teste foram geradas configurações iniciais com quatro arranjos diferentes de times assíncronos, são eles:

- time com *todos* os agentes - (1 *vgsA*, 1 *vgsB*, 1 *mcchA*, 1 *mcchB*, 1 *minhA* e 1 *minhB*);
- time com 6 agentes *vgsB*;
- time com 6 agentes *mcchB*;
- time com 6 agentes *minhB*.

Desta maneira, para cada situação *sistema teste - arranjo de time*, foram realizadas diversas simulações no intuito de encontrar a calibração de parâmetros mais adequada para o algoritmo genético serial GANORD.

Nesta seção, são apresentados, em tabelas, os resultados referentes as simulações com as melhores calibrações. A Tabela 17 é referente as simulações com configurações iniciais zeradas; e as tabelas 18, 19, 20 e 21 são referentes as simulações do GANORD inicializado com os diferentes arranjos de times assíncronos propostos.

Tabela 17: Resultados do GANORD com configurações iniciais zeradas.

algoritmo itens sistemas	GANORD			
	Número de Gerações	Número de PPLs	Tempo da Solução Ótima (s)	Tempo Médio por PL (s)
Garver c/ redesp.	22	591	2,65	0,0045
Garver s/ redesp.	25	667	2,39	0,0037
Sul bras. c/ redesp.	13	1.650	13,75	0,0083
Sul bras. s/ redesp.	33	3.801	31,99	0,0084
Nor-Nord.-2002	1.000 <sup>a</sup>	236.997	71,24 min <sup>b</sup>	0,0180
Nor-Nord.-2008	— <sup>c</sup>	—	—	—

<sup>a</sup>simulado com 1.000 gerações

<sup>b</sup>ótimo local US\$ 2,571,142,000.00 - corte de carga 37 MW

<sup>c</sup>simulação não realizada

Tabela 18: Resultados do GANORD - configurações iniciais com 6 agentes mcchB.

algoritmo itens sistemas	GANORD			
	Número de Gerações	Número de PPLs	Tempo da Solução ótima (s)	Tempo Médio por PL (s)
Garver c/ redesp.	4	60	0,30	0,0049
Garver s/ redesp.	— <sup>a</sup>	—	—	—
Sul bras. c/ redesp.	8	830	7,30	0,0088
Sul bras. s/ redesp.	2	70	0,58 <sup>b</sup>	0,083
Nor-Nord.-2002	76	42.289	12,94 min <sup>c</sup>	0,0188
Nor-Nord.-2008	106	55.928	18,28 min <sup>d</sup>	0,0196

<sup>a</sup>ótimo global encontrado na inicialização

<sup>b</sup>ótimo local US\$ 162,188,000.00 - corte de carga 1 MW

<sup>c</sup>ótimo local US\$ 1,608,130,000.00 - corte de carga 0 MW

<sup>d</sup>ótimo local US\$ 2,789,859,000.00 - corte de carga 0 MW



Tabela 19: Resultados do GANORD - configurações iniciais com 6 agentes minhB.

algoritmo itens sistemas	GANORD			
	Número de Gerações	Número de PPLs	Tempo da Solução ótima (s)	Tempo Médio por PL (s)
Garver c/ redesp.	2	30	0,11	0,0038
Garver s/ redesp.	— <sup>a</sup>	—	—	—
Sul bras. c/ redesp.	8	770	6,16	0,0080
Sul bras. s/ redesp.	2	70	0,70 <sup>b</sup>	0,0110
Nor-Nord.-2002	1	1	0,02 <sup>c</sup>	0,0178
Nor-Nord.-2008	71	38.705	12,77 min <sup>d</sup>	0,0198

<sup>a</sup>ótimo global encontrado na inicialização

<sup>b</sup>ótimo local US\$ 162,188,000.00 - corte de carga 1 MW

<sup>c</sup>ótimo local US\$ 1,520,709,000.00 - corte de carga 0 MW

<sup>d</sup>ótimo local US\$ 2,783,149,000.00 - corte de carga 4 MW

Tabela 20: Resultados do GANORD - configurações iniciais com 6 agentes vgsB.

algoritmo itens sistemas	GANORD			
	Número de Gerações	Número de PPLs	Tempo da Solução ótima (s)	Tempo Médio por PL (s)
Garver c/ redesp.	— <sup>a</sup>	—	—	—
Garver s/ redesp.	— <sup>a</sup>	—	—	—
Sul bras. c/ redesp.	2	242	2,01	0,0083
Sul bras. s/ redesp.	2	70	0,65 <sup>b</sup>	0,0104
Nor-Nord.-2002	1	1	0,02 <sup>c</sup>	0,0164
Nor-Nord.-2008	178	87.891	30,36 min <sup>d</sup>	0,0207

<sup>a</sup>ótimo global encontrado na inicialização

<sup>b</sup>ótimo local US\$ 162,188,000.00 - corte de carga 1 MW

<sup>c</sup>ótimo local US\$ 1,518,343,000.00 - corte de carga 0 MW

<sup>d</sup>ótimo local US\$ 2,667,638,000.00 - corte de carga 0 MW

Tabela 21: Resultados do GANORD - configurações iniciais com todos os agentes.

algoritmo itens sistemas	GANORD			
	Número de Gerações	Número de PPLs	Tempo da Solução ótima (s)	Tempo Médio por PL (s)
Garver c/ redesp.	– <sup>a</sup>	–	–	–
Garver s/ redesp.	– <sup>a</sup>	–	–	–
Sul bras. c/ redesp.	3	242	2,31	0,0095
Sul bras. s/ redesp.	18	1.216	12,36	0,0102
Nor-Nord.-2002	13	4.955	1,35 min <sup>b</sup>	0,0186
Nor-Nord.-2008	116	62.793	20,72 min <sup>c</sup>	0,0198

<sup>a</sup>ótimo global encontrado na inicialização

<sup>b</sup>ótimo local US\$ 1,497,412,000.00 - corte de carga 4 MW

<sup>c</sup>ótimo local US\$ 2,674,442,000.00 - corte de carga 1 MW

A Tabela 22 mostra um comparativo entre os resultados obtidos com o algoritmo genético serial GANORD inicializado com as configurações geradas pelo time assíncrono proposto neste trabalho e os resultados obtidos com a inicialização gerada com o time assíncrono apresentado em (DE OLIVEIRA, 2004) para diferentes sistemas teste.

Tabela 22: Tabela comparativa dos resultados do GANORD.

algoritmo itens sistemas	GANORD		
	<i>in</i>	Número de Gerações	Número de PPLs
Garver c/ redesp.	(DE OLIVEIRA, 2004)	2	130
	algoritmos propostos	— <sup>a</sup>	—
Garver s/ redesp.	(DE OLIVEIRA, 2004)	6	200
	algoritmos propostos	— <sup>a</sup>	—
Sul bras. c/ redesp.	(DE OLIVEIRA, 2004)	16	2.665
	algoritmos propostos	3	242
Sul bras. s/ redesp.	(DE OLIVEIRA, 2004)	22	3.395
	algoritmos propostos	18	1.216
Nor-Nord.-2002	(DE OLIVEIRA, 2004)	187	66.342 <sup>b</sup>
	algoritmos propostos	13	4.955 <sup>c</sup>
Nor-Nord.-2008	(DE OLIVEIRA, 2004)	481	278.196 <sup>d</sup>
	algoritmos propostos	178	87.891 <sup>e</sup>

<sup>a</sup>ótimo global encontrado na inicialização

<sup>b</sup>ótimo local US\$ 1,503,636,000.00 - corte de carga 39 MW

<sup>c</sup>ótimo local US\$ 1,497,412,000.00 - corte de carga 4 MW

<sup>d</sup>ótimo local US\$ 2,594,216,000.00 - corte de carga 2 MW

<sup>e</sup>ótimo local US\$ 2,667,638,000.00 - corte de carga 0 MW

Observa-se que para sistemas de pequeno porte que todos os arranjos de times propostos geraram configurações iniciais de boa qualidade, sendo que, para o time composto por *todos* os agentes e o time de agentes *ugsB* a solução ótima global foi encontrada durante a inicialização para os dois sistemas testes analisados.

Para o sistema Sul brasileiro com redespacho, as simulações do algoritmo genético inicializado com todos os arranjos de time convergiram para o ótimo global em poucas gerações, com o time de *todos* os agentes convergindo mais rápido que os demais. Para o sistema Sul brasileiro sem redespacho, o algoritmo genético só converge para o ótimo global quando inicializado com configurações zeradas e com o time assíncrono composto por *todos* os agentes.

Em sistemas de grande porte, a falta de diversidade nas configurações geradas com times compostos por uma heurística fez com que o algoritmo genético ficasse preso em regiões de ótimo local de baixa qualidade logo nas primeiras gerações ou simplesmente não

---

melhorasse a solução encontrada na inicialização. Para o sistema Norte-Nordeste 2008 o time de agentes *vgsB* encontrou uma solução melhor que o time de todos os agentes.

## 6 *Conclusões e Sugestões para Trabalhos Futuros*

Neste trabalho foram apresentados métodos heurísticos, implementados como agentes construtivos e aplicados a teoria de times assíncronos, com o objetivo de gerar soluções iniciais de boa qualidade para a inicialização das metaheurísticas aplicadas a resolução do problema do planejamento estático da expansão da transmissão de energia elétrica.

Os métodos heurísticos construtivos são algoritmos que, passo-a-passo, adicionam a uma configuração base os circuitos mais atrativos segundo um índice de desempenho inerente a cada método. Fundamentados no *modelo DC*, os métodos do *mínimo esforço* e do *mínimo corte de carga*, passaram por pequenas adaptações em seus índices de sensibilidade e foram implementados com sucesso para o *modelo híbrido linear*.

Baseado nos dois métodos citados acima mais o método de *Villasana-Garver-Salon*, foram implementados seis agentes construtores, dois de cada método heurístico construtivo e uma memória central que formaram o time assíncrono inicializador *inic\_refina*.

Com o uso de times assíncronos foi possível reunir as qualidades individuais de cada método heurístico, de maneira que, em cada configuração gerada estão presentes características de todos os métodos que compõem o time.

As configurações geradas com os diferentes arranjos de times assíncronos foram então aplicadas a metaheurística *algoritmo genético* no intuito de validar a qualidade das mesmas.

Com base nos resultados apresentados no capítulo 5 conclui-se que, quando o algoritmo genético é inicializado com as configurações geradas pelo time assíncrono composto por *todos* os agentes, o mesmo converge para as melhores soluções e em um menor tempo em relação a inicialização com configurações geradas por uma única heurística, mostrando a eficácia da aplicação dos times assíncronos na geração de soluções iniciais.

Para sistemas de pequeno porte os resultados obtidos foram melhores que o esperado,

pois os times assíncronos que deveriam apenas gerar configurações de boa qualidade, encontraram a solução ótima global durante a inicialização para a maioria dos arranjos de times analisados.

Em se tratando de sistemas de médio porte, no sistema Sul Brasileiro com redespacho, os arranjos de times assíncronos propostos se comportaram como o esperado, com o time de *todos* os agentes apresentando o melhor desempenho. Porém, para o sistema teste Sul Brasileiro sem redespacho, o algoritmo genético só encontra a solução ótima global quando inicializado com configurações zeradas e com o time assíncrono composto por *todos* os agentes. Isto se deve ao fato de que as configurações geradas com times de uma única heurística são muito homogêneas, impedindo que o algoritmo genético saia de regiões de ótimo local.

Em sistemas de grande porte, no sistema Norte-Nosteste 2002 o algoritmo genético com configurações iniciais geradas pelo time assíncrono composto por *todos* os agentes encontrou uma solução melhor que a solução de referência, mostrando o bom desempenho dos agentes implementados com o modelo *híbrido linear*. Os resultados não foram os esperados para o sistema Norte-Nordeste 2008, onde o time de agentes *vgsB* encontrou uma solução melhor que o time de *todos* os agentes.

De um modo geral, mesmo as configurações geradas com os times assíncronos de uma única heurística apresentam valores de função objetivo bem próximos do ótimo global ou das melhores soluções existentes. Desta maneira, fica como sugestão para trabalhos futuros a aplicação destas configurações em metaheurísticas que explorem melhor os espaços de busca restritos, como por exemplo, a metaheurística Busca Tabu.

Outra proposta de trabalho futuro é a implementação de agentes baseados em outros modelos matemáticos, como por exemplo, os modelos de *transportes* e *DC*, inserindo desta forma uma maior diversificação nas configurações geradas.

## *Referências*

- AVILA, S. L. *Algoritmos Genéticos Aplicados na Otimização de Antenas Refletoras*. 2002. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal de Santa Catarina, Florianópolis, 2002.
- BENDERS, J. F. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, [s.l.], v. 4, p. 238–252, 1962.
- BINATO, S. *Expansão ótima de sistemas de transmissão através de decomposição de Benders e técnicas de planos cortantes*. 2000. Tese (Doutorado em Engenharia Elétrica) — COPPE/UFRJ, Rio de Janeiro, 2000.
- CENAPAD. Curso de MPI. 2001. Disponível em: <<ftp://ftp.cenapadne.br/pub/cenapad/mpi>>. Acesso em: 05 de maio de 2006.
- CHOI, J.; EL-KEIB, A. A.; TRAN, T. A fuzzy branch and bound based transmission system expansion planning for the highest satisfaction level of the decision maker. *IEEE Trans. Power Systems*, New York, v. 20, n. 1, p. 476–484, Feb. 2005.
- DE ALMEIDA, C. R. T. *Time assíncrono inicializador para o planejamento da expansão da transmissão*. 1998. 95 p. Dissertação (Mestrado em Engenharia Elétrica) — Faculdade de Engenharia Elétrica e de Computação - Universidade Estadual de Campinas, Campinas, 1998.
- DE OLIVEIRA, S. A. *Metaheurísticas aplicadas ao planejamento da expansão da transmissão de energia elétrica em ambiente de processamento distribuído*. 2004. Tese (Doutorado em Engenharia Elétrica) — Faculdade de Engenharia Elétrica e de Computação - Universidade Estadual de Campinas, Campinas, 2004.
- DE OLIVEIRA, S. A.; DE ALMEIDA, C. R. T.; MONTICELLI, A. Times assíncronos aplicados a métodos heurísticos construtivos de planejamento da expansão da transmissão. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA - CBA, 1995, Uberlândia. *Proceedings...* Uberlândia: SBA/UFU, 1998. p. 1029–1034.
- DE OLIVEIRA, S. A.; DE ALMEIDA, C. R. T.; MONTICELLI, A. Time assíncrono inicializador de métodos combinatoriais para planejamento da expansão da transmissão. In: SEMINÁRIO NACIONAL DE PRODUÇÃO E TRANSMISSÃO DE ENERGIA ELÉTRICA - SNPTEE, 1999, Foz do Iguaçu. *Anais...* Foz do Iguaçu: Cigré Brasil/Itaipu Binacional. Grupo VII-GLP/IT 04, 1999.
- GALLEGO, R. A. *Planejamento a longo prazo de sistemas de transmissão usando técnicas de otimização combinatorial*. 1997. 122 p. Tese (Doutorado em Engenharia Elétrica) — Faculdade de Engenharia Elétrica e de Computação - Universidade Estadual de Campinas, Campinas, 1997.

- GALLEGO, R. A.; DE OLIVEIRA, S. A. *GANORD*: modelo de algoritmos genéticos para o planejamento da transmissão de sistemas de grande porte - manual do usuário. Campinas, março 1997. 7 p. Relatório.
- GARVER, L. L. Transmission network estimation using linear programming. *IEEE Transactions on Power Apparatus and Systems*, New York, v. PAS-89, n. 7, p. 1688–1697, Sept./Oct. 1970.
- GLOVER, F. Tabu Search - Part I. *ORSA Journal Computing*, [s.l.], v. 1, n. 3, p. 190–206, Summer 1989.
- GLOVER, F. Tabu Search - Part II. *ORSA Journal Computing*, [s.l.], v. 2, n. 1, winter 1990.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. New York: Addison Wesley Longman, 1989.
- GRAMA, A.; KUMAR, V.; GUPTA, A.; KARYPIS, G. *Introduction to parallel computing: design and analysis of algorithms*. Redwood: Benjamin-Cummings Publishing, 1994. 498 p.
- GRANVILLE, S.; PEREIRA, M. V. F. *Analysis of the linearized power flow model in Benders decomposition*. Stanford University: EPRI, 1985. Report.
- GROPP, W.; LUSK, E. Why are PVM and MPI so different? In: EUROPEAN PVM/MPI USERS' GROUP MEETING ON RECENT ADVANCES IN PARALLEL VIRTUAL MACHINE AND MESSAGE PASSING INTERFACE, 1997, London. *Technical Report...* 1997. Disponível em: <<http://www-unix.mcs.anl.gov/mpi/>>. Acesso em: 8 mar. 2007.
- HAFFNER, S. L. *O Planejamento da Expansão dos Sistemas Elétricos no Contexto de Um Ambiente Competitivo*. 2000. Tese (Doutorado em Engenharia Elétrica) — Faculdade de Engenharia Elétrica e de Computação - Universidade Estadual de Campinas, Campinas, 2000.
- HAFFNER, S. L.; MONTICELLI, A.; GARCIA, A.; MANTOVANI, J.; ROMERO, R. Branch and bound algorithm for transmission system expansion planning using a transportation model. *IEE Proceedings of Generation, Transmission and Distribution*, London, v. 147, n. 3, p. 149–156, May 2000.
- HAFFNER, S. L.; MONTICELLI, A.; GARCIA, A. V.; ROMERO, R. Specialized branch and bound algorithm for transmission network expansion planning. *IEE Proceedings of Generation, Transmission and Distribution*, London, v. 148, n. 5, p. 482–488, Sep. 2001.
- HOLLAND, J. H. *Adaptation in Natural and Artificial Systems*. Michigan: University of Michigan Press, 1975.
- KIRKPATRICK, S.; GELATTI, C. D.; CECCHI, P. M. Optimization by simulated annealing. *Science*, New York, n. 22, p. 671, 1983.
- LAND, A. H.; DOIG, A. G. An automatic method for solving discrete programming problems. *Econometrica*, [s.l.], v. 28, p. 497–520, 1960.



LEE, S. T.; HOCKS, K. L.; HYNILICZA, E. Transmission expansion using branch-and-bound integer programming with optimal cost-capacity curves. *IEEE Trans. Power App. Systems*, New York, PAS-93, p. 1390–1400, July 1974.

MIASAKI, C. T. *Planejamento da Expansão de Sistemas de Transmissão Usando Transformadores Defasadores*. 2002. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Estadual Paulista - UNESP - Faculdade de Engenharia - Campus de Ilha Solteira, Ilha Solteira, 2002.

MONTICELLI, A. J. *Fluxo de Carga em Redes de Energia Elétrica*. São Paulo: Editora Edgar Blücher, 1983.

MONTICELLI, A. J.; SANTOS, A.; PEREIRA, M.; CUNHA, S.; PARKER, B.; PRACA, J. Interactive transmission network planning using a least-effort criterion. *IEEE Transaction on Power Apparatus and Systems*, New York, v. PAS-101, n. 10, p. 3919–3925, Oct. 1982.

MURTAGH, B. A.; SAUNDERS, M. A. *MINOS 5.1, User's guide*. Stanford: Department of Operations Research of Stanford University, 1987. 125 p. Technical Report SOL 83-20R.

ORTEGA, J. M.; RHEINBOLDT, W. C. Local and global convergence of generalized linear iterations. In: SIAM SYMPOSIUM ON NUMERICAL SOLUTION OF NONLINEAR PROBLEMS, 1968, Philadelphia. *Proceedings...* Philadelphia, 1970. p. 1029–1034.

PASSOS, C. A. S.; FONSECA, S. L. A. Uma arquitetura multiagente para a solução de problemas de seqüenciamento da produção. *INFOCOMP - Journal of Computer Science*, Dept. Computer Science - Federal University of Lavras, Lavras, v. 4, n. 2, June 2005. Disponível em: <<http://www.dcc.ufla.br/infocomp/artigos/v4.2/vol4.2.htm>>.

PEIXOTO, H. P.; DE SOUZA, P. S. *Times Assíncronos: Uma nova técnica para o flow shop problem*. Campinas: IMEC - UNICAMP, jul. 1994. 19 p.

PEIXOTO, H. P.; DE SOUZA, P. S. *Uma metodologia de especificação de times assíncronos*. Campinas: IMEC - UNICAMP, nov. 1994. 8 p.

PEREIRA, M. V. F.; PINTO, L. M. V. G. Application of sensitivity analysis of load supplying capability to interactive transmission expansion planning. *IEEE Transaction on Power Apparatus and Systems*, New York, v. PAS-104, n. 2, p. 381–389, Feb. 1985.

PEREIRA, M. V. F.; PINTO, L. M. V. G.; CUNHA, S. H. F.; OLIVEIRA, G. C. A decomposition approach to automated generation-transmission expansion planning. *IEEE Transaction on Power Apparatus and Systems*, New York, v. PAS-104, n. 11, Nov. 1985.

PINTO, L. M. V. G.; NUNES, A. A model for the optimal transmission expansion planning. In: POWER SYSTEM COMPUTING CONFERENCE, 1990, s.l. *Proceedings...* s.l.: s.n., 1990. p. 116–123.

PYO, S. S. *Asynchronous algorithms for distributed processing*. 1985. 191 p. Tese (Ph. D) — Carnegie-Mellon University, Pittsburgh, 1985.

REBONATO, M. T. Introdução a programação paralela com MPI em agregados de computadores (clusters). In: CONGRESSO BRASILEIRO DE CIÊNCIA DA COMPUTAÇÃO, 2004, Itajaí. *Anais...* Itajaí: SBC, 2004. p. 938–955.

ROMERO, R. *Planejamento a longo prazo da expansão de sistemas de transmissão de energia elétrica*. 1999. Tese (Livre-Docência em Engenharia Elétrica) — Universidade Estadual Paulista - UNESP - Faculdade de Engenharia - Campus de Ilha Solteira, Ilha Solteira, 1999.

ROMERO, R.; MONTICELLI, A. A hierarchical decomposition approach for transmission network expansion planning. *IEEE Transaction on Power Systems*, New York, v. 9, n. 1, p. 373–380, Feb. 1994.

TALUKDAR, S.; PYO, S.; GIRAS, T. Asynchronous procedures for parallel processing. *IEEE Transactions on Power Apparatus and Systems*, New York, PAS-102, n. 11, p. 3652–3659, 1983. ISSN 0018-9510.

TALUKDAR, S. N.; DE SOUZA, P. Asynchronous teams. In: SIAM SYMPOSIUM ON NUMERICAL SOLUTION OF NONLINEAR PROBLEMS, 1990, San Francisco. *Proceedings...* San Francisco: SIAM Publications, 1990.

TALUKDAR, S. N.; DE SOUZA, P. Scale efficient organizations. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS, 1992, New York. *Proceedings...* New York: IEEE, 1992. p. 1458–1463.

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

## *APÊNDICE A – Dados dos Sistemas Testes*

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

Tabela 23: 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 24: 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

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

Tabela 25: 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

continua na próxima página

Tabela 25: Dados de Barras - Sul Brasileiro (continuação)

Barra	Capacidade de Geração (MW)	Geração Atual (MW)	Carga (MW)
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
40	0,0	0,0	262,1

continua na próxima página

Tabela 25: Dados de Barras - Sul Brasileiro (continuação)

Barra	Capacidade de Geração (MW)	Geração Atual (MW)	Carga (MW)
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 26: Dados de Linhas - Sul Brasileiro

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 <sup>3</sup> 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

continua na próxima página

Tabela 26: Dados de Linhas - Sul Brasileiro (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 <sup>3</sup> US\$
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
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

continua na próxima página

Tabela 26: Dados de Linhas - Sul Brasileiro (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 <sup>3</sup> US\$
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
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



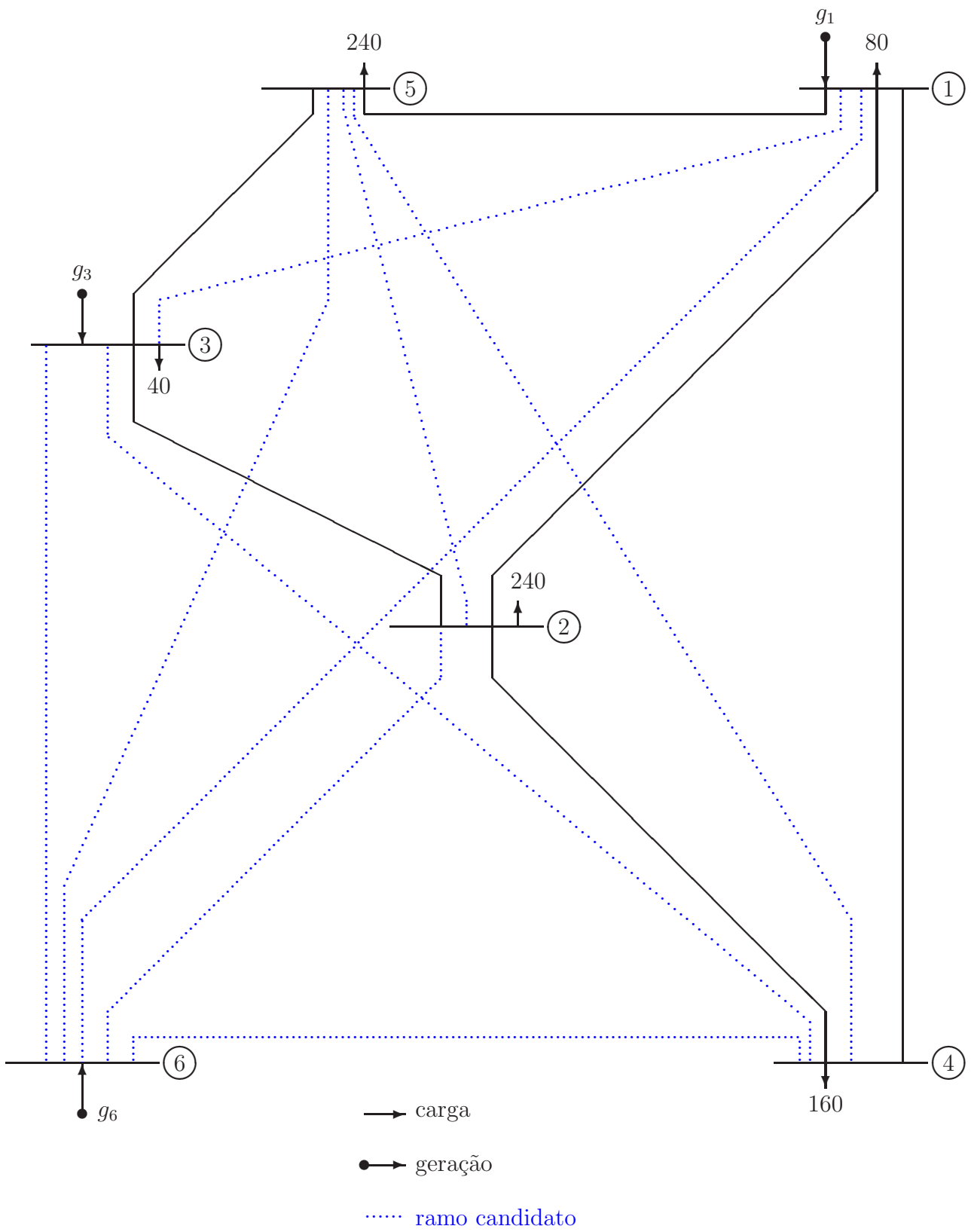


Figura 5: Sistema Garver - Configuração Inicial.

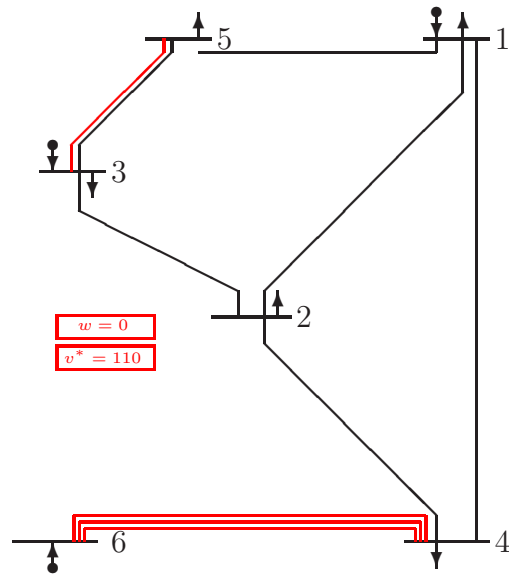


Figura 6: Sistema Garver - Configuração Ótima com Redespacho

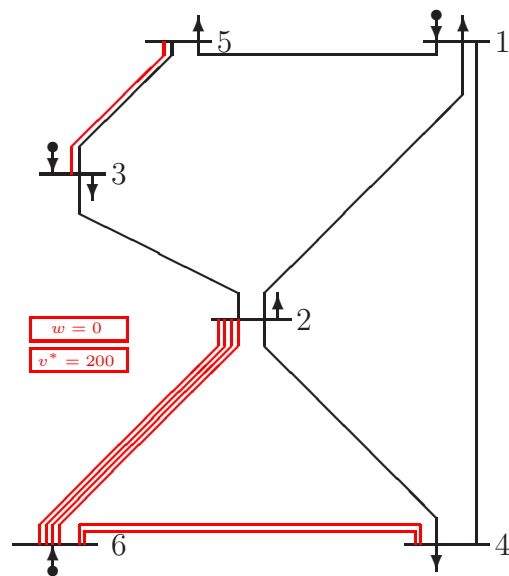


Figura 7: Sistema Garver - Configuração Ótima sem Redespacho

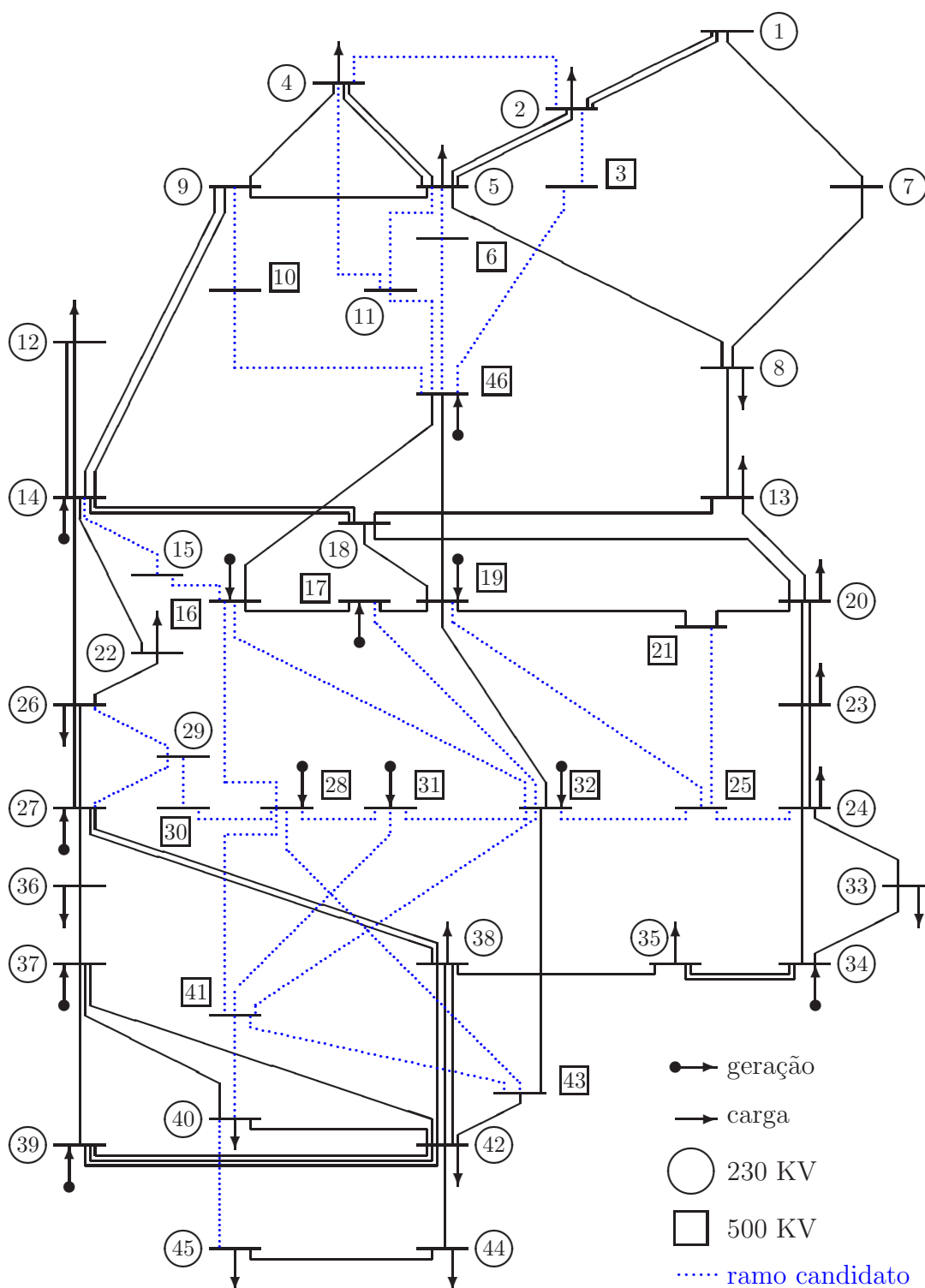


Figura 8: Sistema Sul brasileiro - Configuração Inicial.

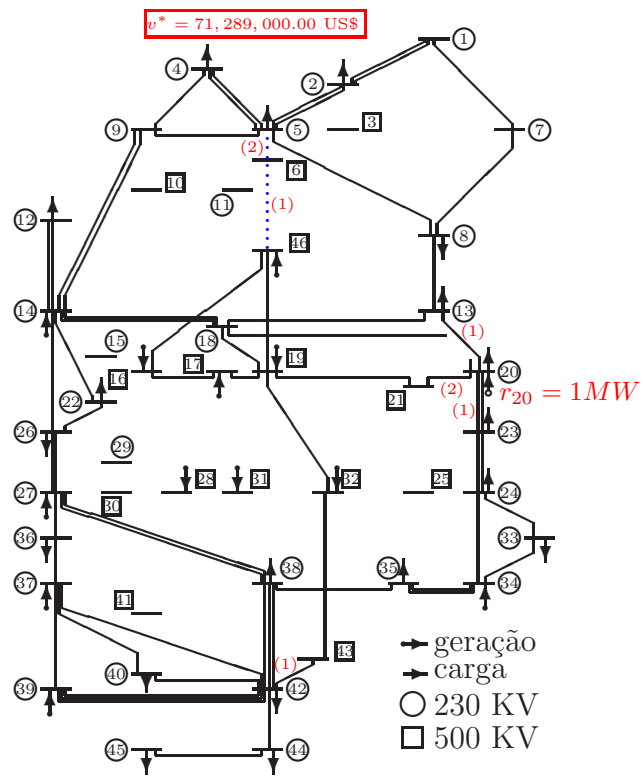


Figura 9: Sistema Sul Brasileiro - Config. Ótima com Redespacho.

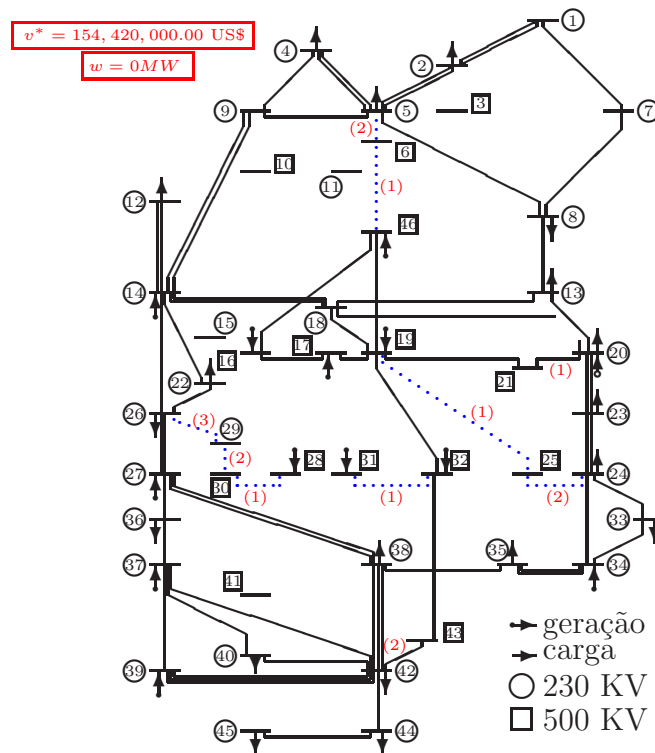


Figura 10: Sistema Sul Brasileiro - Config. Ótima sem Redespacho.

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

Tabela 27: 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

continua na próxima página

Tabela 27: Dados de Barras - Norte-Nordeste Brasileiro (continuação)

Barra	Geração em 2002 (MW)	Carga em 2002 (MW)	Geração em 2008 (MW)	Carga em 2008 (MW)
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
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

continua na próxima página

Tabela 27: Dados de Barras - Norte-Nordeste Brasileiro (continuação)

Barra	Geração em 2002 (MW)	Carga em 2002 (MW)	Geração em 2008 (MW)	Carga em 2008 (MW)
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
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 28: Dados de Linhas - Norte-Nordeste Brasileiro

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 <sup>3</sup> 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

continua na próxima página



Tabela 28: Dados de Linhas - Norte-Nordeste Brasileiro (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 <sup>3</sup> US\$
31	07-08	1	0,0234	1.000	29,048.00
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

continua na próxima página

Tabela 28: Dados de Linhas - Norte-Nordeste Brasileiro (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 <sup>3</sup> US\$
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
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

continua na próxima página

Tabela 28: Dados de Linhas - Norte-Nordeste Brasileiro (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 <sup>3</sup> US\$
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
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

continua na próxima página

Tabela 28: Dados de Linhas - Norte-Nordeste Brasileiro (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 <sup>3</sup> US\$
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

continua na próxima página

Tabela 28: Dados de Linhas - Norte-Nordeste Brasileiro (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 <sup>3</sup> US\$
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
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

## *APÊNDICE B – Uso do Ambiente LAM-MPI*

Abaixo, descrevem-se os passos para preparar uma sessão MPI utilizando LAM-MPI.

### **B.1 Começando com o LAM-MPI**

#### **B.1.1 O arquivo de Máquinas**

Em seu diretório de trabalho (onde seus arquivos binários MPI vão ficar), crie um arquivo com os nomes das máquinas onde deseja executar os seus programas MPI. O seu arquivo de máquinas pode ter qualquer nome, como por exemplo *maquinas* e deve conter uma lista de máquinas a serem incluídas na seção MPI. A seguir encontra-se um exemplo de um arquivo *maquinas*:

bumbassauro.dee.feis.unesp.br

leitao.dee.feis.unesp.br

robim.dee.feis.unesp.br

pigeotto.dee.feis.unesp.br

#### **B.1.2 Lamboot**

Toda sessão MPI deve ser inicializada com um e somente um comando *lamboot*. Para iniciar uma sessão MPI sob o LAM, no prompt do Linux, o comando usado para a montagem das máquinas é:

```
lamboot -v maquinas
```

A opção *-v* é para habilitar o modo *verbose*, de modo que se possa observar o que o LAM está fazendo e o *maquinas* é o arquivo onde estão os nomes das máquinas. Se o

comando funcionar corretamente, deve-se ver algo assim (dependendo do seu *hostfile*):

LAM 7.1.1/MPI 2 C++/ROMIO - Indiana University

```
n - 1 < 13956 > ssi:boot:base:linear: booting n0 (bumbassauro)
```

```
n - 1 < 13956 > ssi:boot:base:linear: booting n1 (leitao)
```

```
n - 1 < 13956 > ssi:boot:base:linear: booting n2 (robim)
```

```
n - 1 < 13956 > ssi:boot:base:linear: booting n3 (pigeotto)
```

```
n - 1 < 13956 > ssi:boot:base:linear: finished
```

### B.1.3 Tping N

Um modo simples de verificar o estado de cada máquina em uma sessão corrente MPI é utilizar o comando `tping N`. Tping pode ser de grande ajuda quando suspeitar-se que um programa MPI travou ou quando não se observa os resultados esperados. A opção `N` na linha de comando é para indicar ao LAM para testar todas as máquinas do arquivo *hostfile*. A não ser que se indique para tping o número específico de vezes para testar, o comando será executado até que as teclas `CTRL-c` sejam digitadas. Pode-se usar a opção `-c` para especificar o número de testes a serem executados:

```
tping -c3 N.
```

## B.2 Compilando Programas MPI

A maneira mais simples de compilar os programas MPI sendo executadas sob a implementação do LAM é modificar um *Makefile* existente. Neste arquivo deve conter o caminho completo para as bibliotecas do MPI. A seguir um exemplo de um arquivo *Makefile* para o programa “*inic\_refina*” em MPI.

```
#
# ARQUIVO MAKEFILE
#
# ABAIXO TEMOS O NOME DO EXECUTAVEL
FPO = inic_refina
#
# ABAIXO TEMOS O DIRETORIO DE DESTINO DE EXECUTAVEL
```

```

SDIR = $(HOME)/mpi_INIC/
#
#ABAIXO OS COMPILADORES QUE IRA USAR
F77    = mpif77
#
# ABAIXO TEMOS AS BIBLIOTECAS MPI
MPILIB = /usr/lib/lam/liblammpi++.a \
/usr/lib/lam/liblammpio.a \
        /usr/lib/lam/liblamf77mpi.a \
        /usr/lib/lam/liblam.a \
        /usr/lib/lam/libmpi.a
#
# ABAIXO TEMOS AS OPCOES DE COMPILACOES
FFLAGS = -c -C
#
# ABAIXO TEMOS OS ARQUIVOS INCLUDE (se existir)
EXTRASM = inic.o timerL.o
all: $(SDIR)/$(FPO)
$(SDIR)/$(FPO) : $(EXTRASM) make_i
$(F77) -o $(SDIR)/$(FPO) $(EXTRASM) $(FFLAGS) $(MPILIB)

```

Modifique esse *Makefile* de acordo com o seu ambiente.

Se tiver problemas com o comando *lamboot*, verifique o site LAM FAQ <http://www.lam-mpi.org/faq/> na seção “*Booting LAM*” que contém várias informações.

### B.2.1 Executando Programas no Ambiente LAM-MPI

Para processar seus arquivos MPI executáveis controlando em qual máquina eles devem ser executados, você deve criar um arquivo de partida contendo os “nós” MPI seguidos dos nomes do arquivos a ser executados nos respectivos “nós”. A seguir um exemplo de arquivo de partida chamado *partida*.

Conteúdo do arquivo:

```

n0  inic_refina
n1  vgsB
n2  minhB
n3  mcchB

```



Comando executar os programas do arquivo *partida*.

```
mpirun -v partida
```

Desta maneira, o arquivo *partida* fica “amarrado” ao arquivo *maquinas* determinando exatamente em qual máquina vai executar cada programa.

## B.3 Finalizando o LAM-MPI

Para matar todos os processos MPI e apagar todas as mensagens pendentes, use o comando *lamclean*:

```
lamclean -v ou apenas lamclean
```

É extremamente importante que cada sessão MPI seja finalizada com o comando do LAM para finalização, *lamhalt*. Para finalizar de modo correto uma sessão MPI, entre com o comando a seguir.

```
lamhalt
```

Classificação de Segurança <b>Livre</b>		Documento no.	
Data (mês e ano)		Projeto no.	
Título e subtítulo <b>TIMES ASSÍNCRONOS INICIALIZADORES PARA O PLANEJAMENTO DA EXPANSÃO DA TRANSMISSÃO DE ENERGIA ELÉTRICA BASEADOS NO MODELO HÍBRIDO LINEAR</b>			No. do volume No. da parte
Título do projeto			
Entidade Executora (autor coletivo)		Autor(es) <b>Fernando Rodrigo Lopes Sanchez</b>	
Entidade patrocinada (cliente ou destinatário principal) <b>PPGEE - UNESP Campus de Ilha Solteira</b>			
Resumo Neste trabalho foram implementados diversos agentes heurísticos construtivos, baseados no modelo híbrido linear, que fazem parte de um time assíncrono que tem como objetivo gerar configurações de boa qualidade para inicializar as metaheurísticas que resolvem o problema do planejamento da expansão da transmissão dos sistemas de energia elétrica. A teoria de times assíncronos foi aplicada para reunir as qualidades individuais dos métodos heurísticos, de uma maneira que, partindo de uma configuração base (sem adições) e utilizando um fluxo de dados cíclico, os agentes construtivos adicionassem circuitos a esta configuração de maneira sistemática e aleatória até que esta atenda as demandas de carga solicitadas pelo sistema elétrico em um horizonte futuro. Estas configurações foram então utilizadas por um algoritmo genético no intuito de validar a qualidade das mesmas. Os algoritmos foram implementados em Fortran, utilizando as rotinas de trocas de mensagens do LAM-MPI e simulados para sistemas teste de pequeno, médio e grande porte em ambiente de processamento distribuído. Os resultados comprovam que os times assíncronos de vários métodos heurísticos são mais eficazes comparados com uma única heurística.			
Palavras-chave <b>Times Assíncronos, Algoritmos Heurísticos Construtivos, Modelo Híbrido Linear, Planejamento da Expansão da Transmissão, Metaheurísticas, Processamento Paralelo.</b>			
No. da edição	No. de páginas <b>96</b>	ISSN (para relatórios publicados)	Classificação (CDC ou CDD)
Distribuidor		Número de exemplares <b>2 (dois)</b>	Preço <b>-</b>
Observações <b>Trabalho apresentado 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.</b>			

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)