



Universidade Estadual de Campinas

FEEC-Faculdade de Engenharia Elétrica e de Computação

DSEE-Departamento de Sistemas de Energia Elétrica

*Metaheurísticas Aplicadas ao
Planejamento da Expansão da Transmissão de Energia Elétrica
em Ambiente de Processamento Distribuído*

Sérgio Azevedo de Oliveira

Orientador: Prof. Dr. Rubén Augusto Romero Lázaro - FEIS/UNESP

Co-Orientador: Prof. Dr. André Luiz Morelato França - FEEC/UNICAMP

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação,
UNICAMP, em 01/10/2004, como parte dos requisitos exigidos para a
obtenção do título de Doutor em Engenharia Elétrica.

Banca Examinadora:

Prof. Dr. Rubén Augusto Romero Lázaro (Presidente) - FEIS/UNESP

Prof. Dr. Edson Luiz da Silva - EEL/UFSC

Prof. Dr. Antônio César Baleeiro Alves - EEE/UFG

Prof. Dr. Ariovaldo Verandio Garcia - FEEC/UNICAMP

Prof. Dr. Fujio Sato - FEEC/UNICAMP

Prof. Dr. Anésio dos Santos Júnior - FEEC/UNICAMP

Campinas-SP, outubro/2004

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

OL4m Oliveira, Sérgio Azevedo de
Metaheurísticas aplicadas ao planejamento da expansão da
transmissão de energia elétrica em ambiente de
processamento distribuído / Sérgio Azevedo de Oliveira.--
Campinas, SP: [s.n.], 2004.

Orientadores: Rubén Augusto Romero Lázaro, André Luiz
Morelato França.

Tese (Doutorado) - Universidade Estadual de Campinas,
Faculdade de Engenharia Elétrica e de Computação.

1. Sistemas de energia elétrica – Planejamento. 2. Energia
elétrica - Transmissão. 3. Otimização combinatória. 4.
Processamento Paralelo (Computadores). I. Romero Lázaro,
Rubén Augusto. II. França, André Luiz Morelato. III.
Universidade Estadual de Campinas. Faculdade de
Engenharia Elétrica e de Computação. IV. Título.

Resumo

Neste trabalho foram desenvolvidas diversas metaheurísticas combinatórias para a resolução do problema do planejamento da expansão da transmissão dos sistemas de energia elétrica analisado do ponto de vista estático e à longo prazo, dentre as quais uma versão paralela da metodologia “simulated annealing” e diversas versões paralelas de algoritmos genéticos; além de um time assíncrono cujos agentes são variantes destas metaheurísticas. Todas estas versões são inicializadas por um time assíncrono de algoritmos heurísticos construtivos e executadas em um ambiente de processamento distribuído composto por uma rede heterogênea de estações SUN, sistema operacional SunOS, com biblioteca para processamento paralelo PVM. Foram feitos diversos testes para os sistemas: Garver (6 barras/15 ramos), Sul brasileiro (46 barras/79 ramos), Norte-Nordeste brasileiro (87 barras/179 ramos) e sistema colombiano (93 barras/155 ramos), e os resultados comprovam a eficácia das metodologias propostas quando comparados com os resultados das versões seriais de cada metaheurística isoladamente, bem como mostram uma redução significativa nos tempos de processamento.

Abstract

In this work, several combinatorial metaheuristics are developed for solving the transmission expansion planning problem of electric power systems that is analysed considering the static and long-term approach, e.g. a parallel version of the simulated annealing methodology and several parallel versions of a genetic algorithms, besides an asynchronous team which agents are variants of these metaheuristics. All of these versions are initialized by an asynchronous team of constructive heuristic algorithms, executed in a distributed processing environment, composed of a heterogeneous network of SUN workstations, SunOS, with PVM parallel processing library. Several tests are effectuated for the systems: Garver (6 busses/15 branches); brazilian South (46 busses/79 branches), brazilian North/Northeast (87 busses/179 branches) and the colombian system (93 busses/155 branches). The results show the efficiency of the proposed methodologies when compared to the serial versions of each metaheuristic isolatedely, as well as a significative reduction on the processing times.

Agradecimentos

À Deus que me concedeu vida e saúde;

à minha esposa Tânia, e às minhas filhas Pérola e Rannay, pelos sacrifícios em minha ausência e pelos estímulos durante este trabalho;

ao Prof. Alcir José Monticelli pela orientação e incentivo;

aos amigos “Don Ramón” e Rubén pela valiosa colaboração nas discussões dos métodos estudados;

ao casal amigo Mantovani e Suely pelo inestimável apoio logístico;

ao amigo Fujio Sato, pessoa do mais alto valor com quem muito aprendi, pelo apoio e amizade;

ao Prof. Ariovaldo e esposa Arlete pela valiosa ajuda à minha família em Campinas;

aos demais professores do DSEE: Castro, Murari e Morelato, pela convivência e aprendizado recebidos;

a analista Míriam Von Zuben pelo apoio computacional;

à secretária do DSEE, senhorita Edna, pelos serviços prestados;

aos amigos e colegas de trabalho: Adriana, Asada, Baleeiro, Cláudio, Gandini, Haffner, Fernando Pereira, Fernando Diniz, Héder, José Carlos, Luciana, Madson, Nicola, Regina, Walmir, pela amizade, convivência e discussões;

às instituições CAPES e FUNDUNESP pelo apoio financeiro via bolsa de estudo e auxílio para apresentação de trabalhos em congressos, respectivamente.

Dedicatória

Dedico este trabalho:

à memória de minha mãe,

Isabel de Azevedo Oliveira

(05/07/33 - 11/10/95);

e de meu orientador,

Alcir José Monticelli

(16/11/46 - 05/08/01).

"Por causa do cansaço, às vezes não vemos a hora de determinada coisa acabar. Mas se nos lembramos de que é preciso viver o momento presente chegamos ao fim totalmente aliviados, porque, se nos concentramos no momento presente somos transportados e construímos nossos atos para a eternidade".

(Chiara Lubich)

Sumário

Lista de Figuras	xiv
Lista de Tabelas	xix
Lista de Abreviaturas, Variáveis e Símbolos	xxiv
1 Introdução Geral	1
2 O Problema de Planejamento Estático da Expansão da Transmissão	3
2.1 Introdução	3
2.2 Formulações do Problema	5
2.2.1 Modelo DC	5
2.2.2 Modelo de Transportes	7
2.2.3 Modelo Híbrido	8
2.3 Técnicas para a Resolução do Problema	8
2.3.1 Algoritmos Heurísticos Construtivos	8
2.3.1.1 Algoritmo heurístico de Garver	9
2.3.1.2 Método de Mínimo Esforço	10
2.3.1.3 Método de Mínimo Corte de Carga	11
2.3.2 Algoritmos de Otimização Clássica	11
2.3.3 Algoritmos de Otimização Combinatória	12
2.4 O Problema da Inicialização dos Algoritmos Combinatórios	13
3 As Metaheurísticas	15

3.1	Introdução	15
3.2	Conceitos Básicos	16
3.2.1	O Problema de Otimização Combinatória	16
3.2.2	Vizinhança	17
3.2.3	Mecanismo de Geração	17
3.2.4	Algoritmo de Busca Local	17
3.2.5	Otimidade Local	18
3.2.6	Estrutura de Vizinhança Exata	18
3.3	Algoritmos Genéticos	18
3.3.1	Descrição dos Algoritmos Genéticos	19
3.3.1.1	Dimensionamento linear	22
3.3.1.2	Dimensionamento de truncamento sigma	22
3.3.1.3	Seleção baseada na ordenação	22
3.3.1.4	Seleção “Breeding Pool”	23
3.3.1.5	Seleção “Roulette Whell”	23
3.3.2	Aplicações de Algoritmos Genéticos	24
3.4	“Simulated Annealing”	25
3.4.1	Descrição de “Simulated Annealing”	26
3.4.1.1	Critério de aceitação	26
3.4.1.2	Transição	27
3.4.2	Aplicações de “Simulated Annealing”	28
3.5	Busca Tabu	30
3.5.1	Descrição de Busca Tabu	30
3.5.1.1	Memória de curto prazo	32
3.5.1.2	Memória de longo prazo	32
3.5.1.3	Estratégia de intensificação	32
3.5.1.4	Estratégia de diversificação	34
3.5.2	Aplicações de Busca Tabu	36
3.6	Times Assíncronos	37

3.6.1	Conceitos Básicos de Times Assíncronos	37
3.6.2	Descrição de Times Assíncronos	38
3.6.2.1	O “Traveling Salesman Problem” - TSP	39
3.6.2.2	Protocolo para construção de times assíncronos aplicados ao TSP	40
3.6.2.3	Comportamento observado	44
3.6.3	Aplicações de Times Assíncronos	45
4	O Time Assíncrono Inicializador	47
4.1	Introdução	47
4.2	Time Assíncrono de Algoritmos Heurísticos Construtivos	47
4.2.1	Algoritmo Heurístico de Garver	48
4.2.1.1	O algoritmo do agente GarverA	48
4.2.1.2	O algoritmo do agente GarverB	49
4.2.1.3	O algoritmo do agente GarverC	49
4.2.2	Método de Mínimo Esforço	49
4.2.2.1	O algoritmo do agente minD	50
4.2.2.2	O algoritmo do agente minE	50
4.2.3	Método de Mínimo Corte de Carga	50
4.2.3.1	O algoritmo do agente mccF	51
4.3	Time Assíncrono Inicializador proposto	51
4.3.1	Algoritmo INIC	51
4.4	O uso do Time Assíncrono Inicializador com as Metaheurísticas Combinatórias	53
5	Metaheurísticas em Ambiente de Processamento Distribuído	57
5.1	Introdução	57
5.2	O Ambiente de Processamento Distribuído	57
5.3	As Metaheurísticas Sequenciais Utilizadas	59
5.3.1	O algoritmo SANORD	59
5.3.1.1	Estruturas básicas	59
5.3.1.2	Descrição do algoritmo	60

5.3.2	O algoritmo GANORD	61
5.3.2.1	Estruturas básicas	61
5.3.2.2	Descrição do algoritmo	63
5.4	O Algoritmo “Simulated Annealing” Paralelo	63
5.4.1	Descrição do Algoritmo SAPARNOR	65
5.5	Algoritmos Genéticos Paralelos Propostos	67
5.5.1	Avaliação paralela das funções objetivo da população	68
5.5.1.1	Algoritmo GAPAR1A	68
5.5.1.2	Algoritmo GAPAR1B	70
5.5.2	Alocação de subpopulação de configurações para cada processador	72
5.5.2.1	Algoritmo GAPAR2A	73
5.5.2.2	Algoritmo GAPAR2B	75
5.5.2.3	Algoritmo GAPAR3A	77
5.5.2.4	Algoritmo GAPAR3B	79
5.6	O Time Assíncrono de Metaheurísticas	82
5.6.1	Algoritmo MHEUR proposto	83
5.6.1.1	Descrição do algoritmo MHEUR	84
5.6.1.2	Descrição do agente SAM	85
5.6.1.3	Descrição do agente GAM	85
6	Resultados	87
6.1	Introdução	87
6.2	Sistema Garver	88
6.2.1	Sistema Garver (com redespacho)	88
6.2.1.1	Algoritmo SANORD	88
6.2.1.2	Algoritmo GANORD	89
6.2.1.3	Algoritmo SAPARNOR	90
6.2.1.4	Algoritmo GAPAR1B	91
6.2.1.5	Algoritmo GAPAR2A	92
6.2.1.6	Algoritmo GAPAR2B	93

6.2.1.7	Algoritmo GAPAR3A	94
6.2.1.8	Algoritmo MHEUR	95
6.2.2	Sistema Garver (sem redespacho)	96
6.2.2.1	Algoritmo SANORD	96
6.2.2.2	Algoritmo GANORD	97
6.2.2.3	Algoritmo SAPARNOR	98
6.2.2.4	Algoritmo GAPAR1B	99
6.2.2.5	Algoritmo GAPAR2A	100
6.2.2.6	Algoritmo GAPAR2B	101
6.2.2.7	Algoritmo GAPAR3A	102
6.2.2.8	Algoritmo MHEUR	103
6.3	Sistema Sul Brasileiro	104
6.3.1	Sistema Sul Brasileiro (com redespacho)	104
6.3.1.1	Algoritmo SANORD	104
6.3.1.2	Algoritmo GANORD	106
6.3.1.3	Algoritmo SAPARNOR	107
6.3.1.4	Algoritmo GAPAR1B	108
6.3.1.5	Algoritmo GAPAR2A	109
6.3.1.6	Algoritmo GAPAR2B	110
6.3.1.7	Algoritmo GAPAR3A	111
6.3.1.8	Algoritmo MHEUR	112
6.3.2	Sistema Sul Brasileiro (sem redespacho)	113
6.3.2.1	Algoritmo SANORD	113
6.3.2.2	Algoritmo GANORD	114
6.3.2.3	Algoritmo SAPARNOR	115
6.3.2.4	Algoritmo GAPAR1B	116
6.3.2.5	Algoritmo GAPAR2A	117
6.3.2.6	Algoritmo GAPAR2B	118
6.3.2.7	Algoritmo GAPAR3A	119

6.3.2.8	Algoritmo MHEUR	120
6.4	Sistema Norte-Nordeste Brasileiro	121
6.4.1	Sistema Norte-Nordeste Brasileiro - Plano 2002	121
6.4.1.1	Algoritmo SANORD	121
6.4.1.2	Algoritmo GANORD	121
6.4.1.3	Algoritmo SAPARNOR	122
6.4.1.4	Algoritmo GAPAR2A	123
6.4.1.5	Algoritmo GAPAR2B	124
6.4.1.6	Algoritmo GAPAR3A	127
6.4.1.7	Algoritmo MHEUR	129
6.4.2	Sistema Norte-Nordeste Brasileiro - Plano 2008	129
6.4.2.1	Algoritmo SANORD	129
6.4.2.2	Algoritmo GANORD	129
6.4.2.3	Algoritmo SAPARNOR	131
6.4.2.4	Algoritmo GAPAR1B	131
6.4.2.5	Algoritmo GAPAR2A	132
6.4.2.6	Algoritmo GAPAR2B	133
6.5	Sistema Colombiano - Plano 2012	134
6.5.1	Algoritmo SANORD	134
6.5.2	Algoritmo GANORD	135
6.5.3	Algoritmo SAPARNOR	136
6.5.4	Algoritmo GAPAR1B	137
6.5.5	Algoritmo GAPAR2A	138
6.5.6	Algoritmo GAPAR2B	139
6.5.7	Algoritmo GAPAR3A	141
6.5.8	Algoritmo MHEUR	142
6.6	Resumo dos Resultados	143

8	Referências Bibliográficas	153
9	Bibliografia	159
9.1	Gerais	159
9.2	Planejamento da Expansão da Transmissão	163
9.3	“Simulated Annealing”	171
9.4	Algoritmos Genéticos	182
9.5	Busca Tabu	196
9.6	Algoritmos Híbridos	201
9.7	Times Assíncronos	204
A	Dados dos Sistemas Testes	209
A.1	Sistema Garver (6 Barras/15 Ramos)	209
A.2	Sistema Sul Brasileiro (46 Barras/79 Ramos)	212
A.3	Sistema Norte-Nordeste Brasileiro (87 Barras/179 Ramos)	217
A.4	Sistema Colombiano (93 Barras/155 Ramos)	225
B	Executando o Time Assíncrono Inicializador	235
C	Artigos Publicados	241

Lista de Figuras

2.1	Diagrama de blocos de um processo típico de planejamento da transmissão.	4
3.1	Descrição de um algoritmo de busca local	18
3.2	Descrição do algoritmo “annealing”.	27
3.3	Esquema do processo tabu de avaliação.	33
3.4	Enfoque simples de intensificação na Busca Tabu.	34
3.5	Enfoque simples de diversificação na Busca Tabu.	35
3.6	Um super-agente	38
3.7	Um time assíncrono para resolver o problema do caixeiro viajante.	43
4.1	Exemplo de time assíncrono inicializador com três agentes.	52
4.2	Estrutura básica do programa <i>inic</i>	54
4.3	Estrutura básica dos agentes de construção.	55
5.1	O ambiente de processamento distribuído utilizado nas simulações.	58
5.2	Geração das N_k configurações candidatas.	64
5.3	SAPARNOR - Princípio de funcionamento do algoritmo.	66
5.4	GAPAR1A - Avaliação paralela das funções objetivo da população, uma em cada processador.	69
5.5	GAPAR1B - Avaliação paralela das funções objetivo da população, distribuídas para p processadores.	71
5.6	GAPAR2A - Alocação de subpopulações em cada processador, armazenando a incumbente no final de k gerações.	74
5.7	GAPAR2B - Alocação de subpopulações em cada processador, armazenando a incumbente após cada k1 gerações.	76

5.8	GAPAR3A - Alocação de subpopulações em cada processador, armazenando p con- figs. elites diversificadas no final de k1 gerações.	78
5.9	GAPAR3B - Alocação de subpopulações em cada processador, armazenando m con- figs. elites das elites após cada k1 gerações.	80
5.10	Exemplo 1 de time assíncrono de metaheurísticas.	82
5.11	Exemplo 2 de time assíncrono de metaheurísticas.	83
5.12	Esquema de funcionamento do time assíncrono de metaheurísticas.	84
A.1	Sistema Garver - Configuração Inicial.	210
A.2	Sistema Garver - Configuração Ótima com Redespacho	211
A.3	Sistema Garver - Configuração Ótima sem Redespacho	211
A.4	Sistema Sul brasileiro - Configuração Inicial.	215
A.5	Sistema Sul Brasileiro - Config. Ótima com Redespacho.	216
A.6	Sistema Sul Brasileiro - Config. Ótima sem Redespacho.	216
A.7	Sistema Colombiano - Rede Inicial	232
A.8	Sistema Colombiano - Configuração Ótima	233

Lista de Tabelas

2.1	Formulações do Problema	6
3.1	Uma população de cromossomos	20
3.2	Operadores de algoritmos genéticos.	21
3.3	Equivalência entre otimização combinatória e algoritmos genéticos.	23
3.4	Algumas áreas de aplicações de algoritmos genéticos.	24
3.5	Equivalência entre otimização combinatória e “simulated annealing”.	26
3.6	Levantamento de aplicações de algoritmos “simulated annealing”.	29
3.7	Principais Características da Busca Tabu.	31
3.8	Algumas áreas de aplicações da Busca Tabu.	36
3.9	Exemplos de algoritmos para o TSP.	41
3.10	Algumas áreas de aplicações de times assíncronos	45
6.1	Sistema Garver - algoritmo SANORD (sem inicialização)	88
6.2	Sistema Garver - algoritmo SANORD (com inicialização)	89
6.3	Sistema Garver - algoritmo GANORD (sem inicialização)	89
6.4	Sistema Garver - algoritmo GANORD (com inicialização)	90
6.5	Sistema Garver - algoritmo SAPARNOR (sem inicialização)	90
6.6	Sistema Garver - algoritmo SAPARNOR (com inicialização)	91
6.7	Sistema Garver - algoritmo GAPAR1B (sem inicialização)	91
6.8	Sistema Garver - algoritmo GAPAR1B (com inicialização)	92
6.9	Sistema Garver - algoritmo GAPAR2A (sem inicialização)	92
6.10	Sistema Garver - algoritmo GAPAR2A (com inicialização)	93

6.11 Sistema Garver - algoritmo GAPAR2B (sem inicialização)	93
6.12 Sistema Garver - algoritmo GAPAR2B (com inicialização)	94
6.13 Sistema Garver - algoritmo GAPAR3A (sem inicialização)	94
6.14 Sistema Garver - algoritmo GAPAR3A (com inicialização)	95
6.15 Sistema Garver - algoritmo MHEUR (com 3 agentes SAM) - sem inicialização	95
6.16 Sistema Garver - algoritmo MHEUR (com 3 agentes SAM) - com inicialização	96
6.17 Sistema Garver - algoritmo SANORD (sem inicialização)	96
6.18 Sistema Garver - algoritmo SANORD (com inicialização)	97
6.19 Sistema Garver - algoritmo GANORD (sem inicialização)	97
6.20 Sistema Garver - algoritmo GANORD (com inicialização)	98
6.21 Sistema Garver - algoritmo SAPARNOR (sem inicialização)	98
6.22 Sistema Garver - algoritmo SAPARNOR (com inicialização)	99
6.23 Sistema Garver - algoritmo GAPAR1B (sem inicialização)	99
6.24 Sistema Garver - algoritmo GAPAR1B (com inicialização)	100
6.25 Sistema Garver - algoritmo GAPAR2A (sem inicialização)	100
6.26 Sistema Garver - algoritmo GAPAR2A (com inicialização)	101
6.27 Sistema Garver - algoritmo GAPAR2B (sem inicialização)	101
6.28 Sistema Garver - algoritmo GAPAR2B (com inicialização)	102
6.29 Sistema Garver - algoritmo GAPAR3A (sem inicialização)	102
6.30 Sistema Garver - algoritmo GAPAR3A (com inicialização)	103
6.31 Sistema Garver - algoritmo MHEUR (com 4 agentes SAM) - sem inicialização	103
6.32 Sistema Garver - algoritmo MHEUR (com 4 agentes SAM) - com inicialização	104
6.33 Sistema Sul Brasileiro - algoritmo SANORD (sem inicialização)	105
6.34 Sistema Sul Brasileiro - algoritmo SANORD (com inicialização)	105
6.35 Sistema Sul Brasileiro - algoritmo GANORD (sem inicialização)	106
6.36 Sistema Sul Brasileiro - algoritmo GANORD (com inicialização)	106
6.37 Sistema Sul Brasileiro - algoritmo SAPARNOR (sem inicialização)	107
6.38 Sistema Sul Brasileiro - algoritmo SAPARNOR (com inicialização)	107
6.39 Sistema Sul Brasileiro - algoritmo GAPAR1B (sem inicialização)	108

6.40	Sistema Sul Brasileiro - algoritmo GAPAR1B (com inicialização)	108
6.41	Sistema Sul Brasileiro - algoritmo GAPAR2A (sem inicialização)	109
6.42	Sistema Sul Brasileiro - algoritmo GAPAR2A (com inicialização)	109
6.43	Sistema Sul Brasileiro - algoritmo GAPAR2B (sem inicialização)	110
6.44	Sistema Sul Brasileiro - algoritmo GAPAR2B (com inicialização)	110
6.45	Sistema Sul Brasileiro - algoritmo GAPAR3A (sem inicialização)	111
6.46	Sistema Sul Brasileiro - algoritmo GAPAR3A (com inicialização)	111
6.47	Sistema Sul Brasileiro - algoritmo MHEUR (4 agentes SAM) - sem inicialização	112
6.48	Sistema Sul Brasileiro - algoritmo MHEUR (4 agentes SAM) - com inicialização	112
6.49	Sistema Sul Brasileiro - algoritmo SANORD (sem inicialização)	113
6.50	Sistema Sul Brasileiro - algoritmo SANORD (com inicialização)	113
6.51	Sistema Sul Brasileiro - algoritmo GANORD (sem inicialização)	114
6.52	Sistema Sul Brasileiro - algoritmo GANORD (com inicialização)	114
6.53	Sistema Sul Brasileiro - algoritmo SAPARNOR (sem inicialização)	115
6.54	Sistema Sul Brasileiro - algoritmo SAPARNOR (com inicialização)	115
6.55	Sistema Sul Brasileiro - algoritmo GAPAR1B (sem inicialização)	116
6.56	Sistema Sul Brasileiro - algoritmo GAPAR1B (com inicialização)	116
6.57	Sistema Sul Brasileiro - algoritmo GAPAR2A (sem inicialização)	117
6.58	Sistema Sul Brasileiro - algoritmo GAPAR2A (com inicialização)	117
6.59	Sistema Sul Brasileiro - algoritmo GAPAR2B (sem inicialização)	118
6.60	Sistema Sul Brasileiro - algoritmo GAPAR2B (com inicialização)	118
6.61	Sistema Sul Brasileiro - algoritmo GAPAR3A (sem inicialização)	119
6.62	Sistema Sul Brasileiro - algoritmo GAPAR3A (com inicialização)	119
6.63	Sistema Sul Brasileiro - algoritmo MHEUR (3 agentes SAM) - sem inicialização	120
6.64	Sistema Sul Brasileiro - algoritmo MHEUR (4 agentes SAM) - com inicialização	120
6.65	Sistema Norte Nordeste (Plano 2002) - algoritmo SANORD	121
6.66	Sistema Norte Nordeste (Plano 2002) - algoritmo GANORD	122
6.67	Sistema Norte-Nordeste (Plano 2002) - algoritmo SAPARNOR	123
6.68	Sistema Norte-Nordeste (Plano 2002) - algoritmo GAPAR2A	124

6.69	Sistema Norte Nordeste (Plano 2002) - algoritmo GAPAR2B (resultado 1)	125
6.70	Sistema Norte Nordeste (Plano 2002) - algoritmo GAPAR2B (resultado 2)	125
6.71	Sistema Norte Nordeste (Plano 2002) - algoritmo GAPAR2B (resultado 3)	126
6.72	Sistema Norte Nordeste (Plano 2002) - algoritmo GAPAR2B (resultado 4)	126
6.73	Sistema Norte Nordeste (Plano 2002) - algoritmo GAPAR2B (resultado 5)	127
6.74	Sistema Norte Nordeste (Plano 2002) - algoritmo GAPAR3A (Resultado 1)	128
6.75	Sistema Norte Nordeste (Plano 2002) - algoritmo GAPAR3A (Resultado 2)	128
6.76	Sistema Norte Nordeste (Plano 2002) - algoritmo MHEUR (3 agentes SAM)	129
6.77	Sistema Norte Nordeste (Plano 2008) - algoritmo SANORD	130
6.78	Sistema Norte Nordeste (Plano 2008) - algoritmo GANORD	130
6.79	Sistema Norte_Nordeste (Plano 2008) - algoritmo SAPARNOR	131
6.80	Sistema Norte Nordeste (Plano 2008) - algoritmo GAPAR1B	132
6.81	Sistema Norte-Nordeste (Plano 2008) - algoritmo GAPAR2A	133
6.82	Sistema Norte Nordeste (Plano 2008) - algoritmo GAPAR2B	134
6.83	Sistema Colombiano (Plano 2012) - algoritmo SANORD (configuração inicial zerada)	135
6.84	Sistema Colombiano (Plano 2012) - algoritmo GANORD	136
6.85	Sistema Colombiano (Plano 2012) - algoritmo SAPARNOR (sem inicialização)	137
6.86	Sistema Colombiano (Plano 2012) - algoritmo GAPAR21B	138
6.87	Sistema Colombiano (Plano 2012) - algoritmo GAPAR2A	139
6.88	Sistema Colombiano (Plano 2012) - algoritmo GAPAR2B (resultado 1)	140
6.89	Sistema Colombiano (Plano 2012) - algoritmo GAPAR2B (resultado 2)	140
6.90	Sistema Colombiano (Plano 2012) - algoritmo GAPAR2B (resultado 3)	141
6.91	Sistema Colombiano (Plano 2012) - algoritmo GAPAR3A (com inicialização)	142
6.92	Sistema Colombiano (Plano 2012) - algoritmo MHEUR (4 agentes SAM)	143
6.93	Resultados gerais com algoritmo “Simulated Annealing” serial - SANORD	144
6.94	Resultados gerais com Algoritmo Genético serial - GANORD	144
6.95	Resultados Gerais com Algoritmo “Simulated Annealing” Paralelo - SAPARNOR	145
6.96	Resultados Gerais com Algoritmo Genético Paralelo - GAPAR1B	146
6.97	Resultados Gerais com Algoritmo Genético Paralelo - GAPAR2A	146

6.98 Resultados Gerais com Algoritmo Genético Paralelo - GAPAR2B	147
6.99 Resultados Gerais com Algoritmo Genético Paralelo - GAPAR3A	147
6.100 Resultados Gerais com Tímes Assíncronos - MHEUR	148
7.1 Desempenho dos algoritmos propostos	150
A.1 Dados de Barras - Garver	209
A.2 Dados de Linhas - Garver	209
A.3 Dados de Barras - Sul Brasileiro	212
A.4 Dados de Linhas - Sul Brasileiro	213
A.4 Dados de Linhas - Sul Brasileiro (continuação)	214
A.5 Dados de Barras - Norte-Nordeste Brasileiro	217
A.5 Dados de Barras - Norte-Nordeste Brasileiro (continuação)	218
A.5 Dados de Barras - Norte-Nordeste Brasileiro (continuação)	219
A.6 Dados de Linhas - Norte-Nordeste Brasileiro	219
A.6 Dados de Linhas - Norte-Nordeste Brasileiro (continuação)	220
A.6 Dados de Linhas - Norte-Nordeste Brasileiro (continuação)	221
A.6 Dados de Linhas - Norte-Nordeste Brasileiro (continuação)	222
A.6 Dados de Linhas - Norte-Nordeste Brasileiro (continuação)	223
A.6 Dados de Linhas - Norte-Nordeste Brasileiro (continuação)	224
A.7 Dados de Barras - Sistema Colombiano.	225
A.7 Dados de Barras - Sistema Colombiano (continuação)	226
A.7 Dados de Barras - Sistema Colombiano (continuação)	227
A.8 Dados de Linhas - Sistema Colombiano	227
A.8 Dados de Linhas - Colombiano (continuação)	228
A.8 Dados de Linhas - Colombiano (continuação)	229
A.8 Dados de Linhas - Colombiano (continuação)	230
A.8 Dados de Linhas - Colombiano (continuação)	231

Lista de Abreviaturas, Variáveis e Símbolos

α	parâmetro adequado de transformação de unidades;
γ_{ij}^o	Susceptância inicial no ramo $(i, j) \in \Omega$;
$\bar{\phi}_{ij}$	Definida pela relação: $\bar{\phi}_{ij} = \frac{\bar{f}_{ij}}{n_{ij}}$;
θ	ângulos das tensões nodais;
Ω	conjunto de todos os ramos definidos pelas linhas existentes e as alternativas de expansão;
Ω_1	Representa o conjunto de ramos onde já existem linhas na configuração básica;
Ω_2	Representa o conjunto de ramos candidatos em caminhos onde não existe linha na configuração inicial;
A-Team	“asynchronous team”;
alfat	taxa de resfriamento;
$B(.)$	matriz de susceptâncias;
B_1	matriz de susceptância (ramos $\in \Omega_1$);
bconst	taxa de variação no número de mutações;
c_{ij}	custo de um circuito no ramo (i, j) ;
cadeia	parâmetro que controla o incremento da cadeia ;
d	vetor de demandas;
f	vetor de fluxo de potência ativa nos ramos;
f_{ij}	fluxo de potência ativa através dos circuito no ramo ij ;
$\bar{f}_{ij}$	fluxo máximo de potência ativa de um circuito no ramo (i, j) ;
g	vetor de gerações;
$\bar{g}$	Vetor de limites de geração;
GA	“genetic algorithms”;
iterm	máximo número de iteracoes permitidas;
ix	semente inicial para as sub-rotinas <i>random</i> e ; <i>randu</i>
jgera	número máximo de configurações;
kfator	comprimento da cadeia, parâmetro para o n. de iterações;
kincumb	valor da incumbente;
kk1	tamanho do bloco de gerações, para troca de informacoes entre processadores;
kseed	semente inicial;
kselect	fator para escolha da temperatura inicial;
limcong	limite de variações permitida no congelamento;

malea	número de proposta aleatória;
mcarg	número de proposta de carga;
mcor	limite de corte de carga na mutação;
md1	cte. p/ diversificar por adição e retirada;
md2	cte. p/ diversificar por residência;
melhor	melhores configurações armazenadas;
mgarv	número de proposta de Garver;
mgera	número de proposta de geração;
mmin1	número de proposta de mínimo esforço por sobrecarga;
mmin2	proposta de mínimo esforço por índice;
msens	número de proposta de sensibilidade;
mtrono	número de trocas (que não melhoraram no passado);
mtrosi	número de trocas (que melhoraram no passado);
n_{ij}	número de linhas adicionadas, $n_{ij} = \frac{x_{ij}}{\gamma_{ij}}$, sendo γ_{ij} a susceptância nominal de uma linha entre as barras i - j ;
$\bar{n}_{ij}$	Vetor de número máximo de adições no ramo (i, j) ;
ntrocas	máximo número de trocas permitidas;
nactmax2	limitante do número de filhos;
nafa	custo do corte de carga por MW;
nanaliza	número máximo de ocasiões em que se analisa com diversificação;
ncl	número de caminhos a serem gerados;
ncorte	corte de carga permitido;
ncosto	número de proposta por custo para sair;
ndes	número máximo de descendentes gerados (em %);
ndiver1	número máximo de diversificações;
ndiver2	número de elementos candidatos a diversificar;
nduramin	número mínimo de iterações tabu;
nduravar	número variável de iterações tabu;
minli	custo minimo de uma linha;
nintens	número máximo de intensificação sem melhoria;
nmax	número máximo de linhas permitido;
nmutar	número máximo de mutações por cromossomo;
npara	número de configurações iniciais geradas com INIC;
nplmax	número máximo de PL's;
npop	tamanho da população;
nsaen	número de proposta aleatória para sair;
nsem	número de sementes (I etapa do processo);
nsem2	número de sementes (II etapa do processo);
nut	comprimento da cadeia;
nwcore	parâmetro para o MINOS;
nwkmax	limite para o corte de carga;
phi	parâmetro que controla o decremento da temperatura;
PL	programação linear;
PPL	problema de programação linear;
r	vetor de geradores fictícios ou artificiais;

ro	Parametro que controla o incremento da cadeia;
refcor	valor de referência do corte de carga;
S	matriz de incidência nó-ramo (ramos $\in \Omega_1$);
S_2	matriz de incidência nó-ramo (ramos $\in \Omega_2$);
SA	“simulated annealing”;
temp	valor inicial da temperatura;
temperat	parâmetro que controla o decremento da temperatura;
TS	“tabu search”;
v	investimento devido a adição de circuitos;
x_{ij}	nova susceptância a ser instalada no ramo $(i, j) \in \Omega$;
w	corte de carga;
ycross	taxa de “crossover”;

Capítulo 1

Introdução Geral

Nos tempos atuais passamos, em nosso país, por um racionamento de grande impacto, principalmente por falta de uma política de planejamento efetiva e investimentos no setor elétrico. São imprescindíveis novas fontes de geração de energia elétrica, seja de forma hidráulica, nuclear, a gás, solar, eólica, etc., mas igualmente importante são os investimentos em novas linhas de transmissão e/ou transformadores, para o perfeito escoamento dessa geração de energia, principalmente pela predominância de geração de energia elétrica pelo aproveitamento de recursos hidráulicos, e ainda pela consideração que quase sempre os centros geradores estão distantes dos centros consumidores, devido as dimensões continentais de nosso país.

O planejamento de sistemas elétricos de potência, envolve os planejamentos de curto e longo prazos. No contexto do planejamento de curto prazo estão envolvidos o planejamento da operação e as medidas corretivas de curto prazo; dentre as quais destacam-se a alocação de fontes e dispositivos de controle de reativos, que permitem manter o sistema em operação atendendo as condições de qualidade do fornecimento de energia elétrica e confiabilidade sob diferentes condições de carregamento e contingências. Considerando a projeção de crescimento de cargas, num horizonte de crescimento preestabelecido, existe um certo patamar deste crescimento que o sistema existente não é mais capaz de atender a demanda com confiabilidade e qualidade de fornecimento. Nessas condições há a necessidade de se proceder ao planejamento de longo prazo, incluindo-se no contexto deste a construção de novas unidades geradoras e linhas de transmissão com toda a infraestrutura necessária para a operação segura, confiável e economicamente viável do sistema elétrico.

Esta tese trata particularmente do problema do planejamento estático da expansão da transmissão de energia elétrica num horizonte de longo prazo.

A formulação do problema do planejamento estático da expansão da transmissão de energia elétrica, num horizonte de longo prazo, é um problema complexo, de grande porte e de natureza combinatória, existindo inúmeras técnicas para a sua resolução. Dentre elas, destacam-se os métodos combinatórios como “Simulated Annealing”, Algoritmos Genéticos e Busca Tabu, que normalmente necessitam da determinação prévia de blocos construtivos atraentes (conjuntos coerentes de linhas e transformadores) para a sua solução. Ou seja, exige-se uma inicialização adequada dos referidos métodos.

Como um primeiro passo, foi implementada uma versão de Times Assíncronos, tendo como agentes métodos heurísticos construtivos, para resolver o problema de inicialização do problema de planejamento estático.

Além disso, a parte principal dos algoritmos combinatórios é geralmente paralelizável e uma paralelização via “Simulated Annealing” e Algoritmos Genéticos, como também via Times Assíncronos, pode ser altamente eficaz. Assim, neste trabalho, caminhou-se nessa direção desenvolvendo algoritmos paralelos com o uso de “simulated annealing” e algoritmos genéticos, que trabalham como agentes de um Time Assíncrono de métodos combinatórios.

Neste trabalho são propostas, portanto, novas técnicas para a resolução do problema do planejamento estático da expansão da transmissão, tendo como proposta básica a utilização de algoritmos que podem ser processados em uma rede de computadores, com metodologias que tentam tirar proveito da computação concorrente dos mesmos.

O Capítulo 2 trata do problema do planejamento da expansão da transmissão, onde são apresentadas as modelagens do problema e as principais técnicas para a resolução do mesmo, bem como discutido os aspectos relevantes quanto a inicialização destas técnicas. No Capítulo 3, é feita uma revisão geral das metaheurísticas combinatórias: Algoritmos Genéticos, “Simulated Annealing” e Busca Tabu, e são tratados também os conceitos básicos e a teoria de Times Assíncronos. No Capítulo 4 é detalhada a implementação de um Time Assíncrono inicializador para a inicialização dos algoritmos combinatórios. Na sequência, no Capítulo 5, são detalhados inicialmente as metaheurísticas sequenciais utilizadas como referência, bem como são detalhadas todas as metaheurísticas paralelas propostas para um ambiente de processamento distribuído. No Capítulo 6, são apresentados inúmeros resultados de todos os algoritmos implementados. E finalmente, no Capítulo 7, as conclusões e sugestões para futuros trabalhos são apresentadas, seguidas das referências bibliográficas (Capítulo 8) e de uma extensa bibliografia separada por tópicos, dos assuntos abordados neste trabalho (Capítulo 9). Nos Apêndices A, B e C, são apresentados os dados dos sistemas testes utilizados, procedimentos para execução de Times Assíncronos e os trabalhos publicados, respectivamente.

Capítulo 2

O Problema de Planejamento Estático da Expansão da Transmissão

2.1 Introdução

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

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

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

A Figura 2.1 a seguir, descreve um processo típico de planejamento da transmissão em que o objetivo principal é minimizar os investimentos a longo prazo e custos de operação envolvidos nos níveis de provisão e adequação da confiabilidade do sistema, com as devidas considerações de impacto ambiental e outros temas relacionados. O ponto de partida do procedimento de planejamento é

desenvolver previsões de carga em termos dos picos anuais de demanda do sistema como um todo, bem como considerar as gerações atuais e futuras, e então determinar alternativas específicas que satisfaçam as novas condições de carga. O planejador do sistema pode também utilizar ferramentas computacionais tais como os programas de fluxo de carga, estabilidade e curto-circuito, para analisar o desempenho das alternativas específicas do sistema de transmissão.

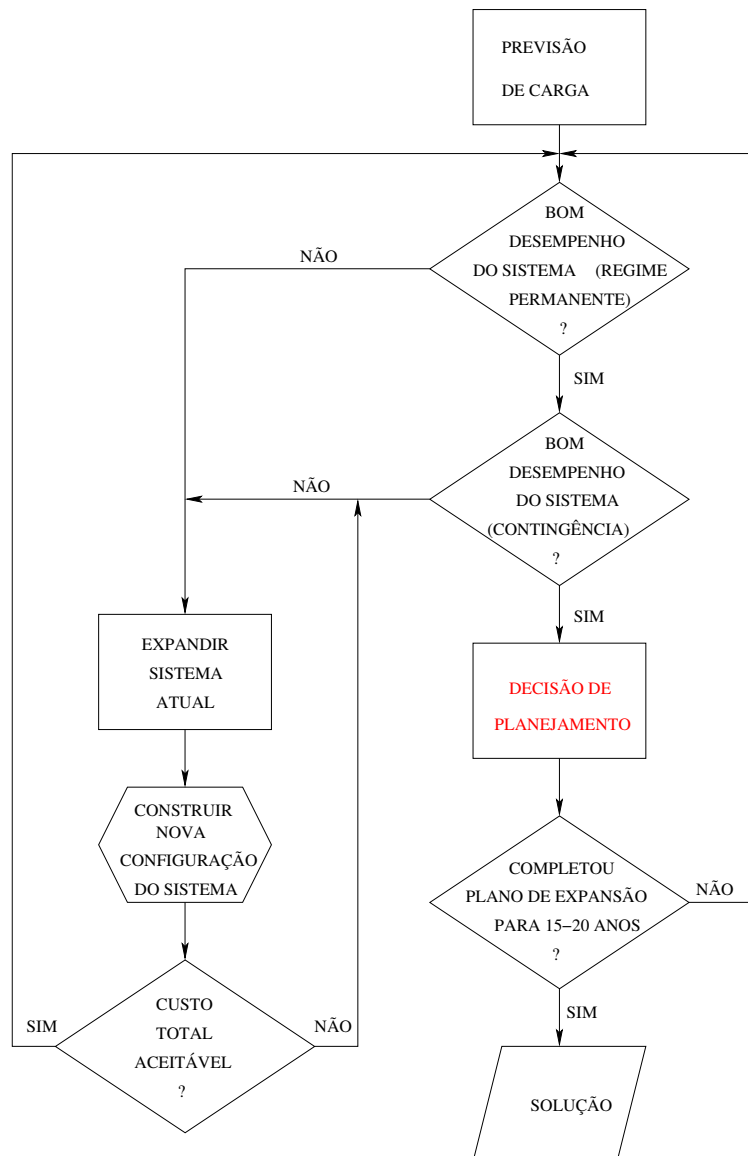


Figura 2.1: Diagrama de blocos de um processo típico de planejamento da transmissão.

Nas décadas passadas, quando os recursos computacionais eram mais limitados, foram de-

envolvidos métodos heurísticos (os chamados *métodos construtivos*) para a resolução do problema, como por exemplo os métodos de Garver, Mínimo Esforço e Mínimo Corte de Carga. Essas metodologias ainda são utilizadas por concessionárias como parte de procedimento iterativos que exigem uma participação ativa dos planejadores. Posteriormente, foram desenvolvidas metodologias baseadas na decomposição de Benders (os chamados *métodos clássicos*), que exploram a decomposição natural do problema em dois subproblemas: *de investimento* e *de operação*. Sendo que mais recentemente, para resolver aqueles problemas de grande porte não resolvidos pelas metodologias anteriores, aparecem os chamados *métodos de otimização* cujas características fundamentais são as de serem aplicáveis a sistemas de grande porte, chegar a soluções próximas ao ótimo global e obter soluções em razoáveis tempos de processamento.

Tem-se então, uma nova classe de heurísticas chamadas *metaheurísticas*; classe de métodos que são projetados para atacar problemas “hard” de otimização combinatória onde heurísticas clássicas têm falhado em eficiência. Dentre elas se destacam: “Simulated Annealing”, Algoritmos Genéticos e Busca Tabu.

Na seqüência, neste capítulo, procura-se dar um visão geral da formulação do problema do planejamento estático da expansão da transmissão, apresentando uma síntese de sua modelagem e das diferentes técnicas de resolução do problema.

No final do capítulo, são discutidos aspectos relevantes quanto a inicialização dos algoritmos acima mencionados.

2.2 Formulações do Problema

Na formulação do problema do planejamento estático da expansão dos sistemas de transmissão num horizonte de longo prazo como um problema de otimização, pode-se utilizar diferentes modelos de representação da rede de transmissão, entre os quais destacam-se: *modelo DC*, *modelo de transportes* e *modelo híbrido*, cujas formulações matemáticas e as descrições de suas variáveis, são apresentadas na Tabela 2.1 a seguir.

2.2.1 Modelo DC

O modelo DC é considerado como o modelo ideal para ser usado em planejamento de sistemas de transmissão, pois leva em conta as duas leis de Kirchhoff para o sistema elétrico, ou seja, todas as barras do sistema e todos os laços existentes devem satisfazer respectivamente, a primeira e a segunda leis de Kirchhoff. Nesta formulação têm-se variáveis contínuas de operação tais como os fluxos nas linhas, os níveis de geração e demanda e a diferença angular nas barras; e variáveis de investimento inteiras, como são os circuitos candidatos à adição.

O modelo apresentado está ligeiramente modificado em relação ao modelo DC convencional pois foi adicionado o termo $\sum_i \alpha_i r_i$ na função objetivo para facilitar o processo de resolução. Quando é escolhido um valor de α relativamente grande, na solução final todos os r_i são iguais a zero. E nestas condições a solução obtida é exatamente igual para a formulação original e para a modificada.

Tabela 2.1: Formulações do Problema

Modelos	Descrição das variáveis
MODELO DC *	
$\text{Min } v = \sum_{(i,j) \in \Omega} c_{ij} n_{ij} + \sum_i \alpha_i r_i$ $\text{s.a. } B(n_{ij})\theta + g + r = d$ $(n_{ij} + n_{ij}^o) \theta_i - \theta_j \leq (n_{ij} + n_{ij}^o)\bar{\phi}_{ij}$ $0 \leq g \leq \bar{g}$ $0 \leq r \leq d$ $0 \leq n_{ij} \leq \bar{n}_{ij}$ $\forall (i, j) \in \Omega$	v investimento devido a adição de circuitos. c_{ij} custo de um circuito no ramo (i, j) . n_{ij} número de linhas adicionadas, $n_{ij} = \frac{x_{ij}}{\gamma_{ij}}$, sendo γ_{ij} a susceptância nominal de uma linha entre as barras i - j . α parâmetro adequado de transformação de unidades. $B(\cdot)$ matriz de susceptâncias. θ ângulos das tensões nodais. g vetor de gerações. r vetor de geradores fictícios ou artificiais. d vetor de demandas.
MODELO DE TRANSPORTES †	
$\text{Min } v = \sum_{(i,j) \in \Omega} c_{ij} n_{ij} + \sum_i \alpha_i r_i$ $\text{s.a. } Sf + g + r = d$ $ f_{ij} - (n_{ij} + n_{ij}^o)\bar{\phi}_{ij} \leq 0$ $0 \leq g \leq \bar{g}$ $0 \leq r \leq d$ $0 \leq n_{ij} \leq \bar{n}_{ij}$ $\forall (i, j) \in \Omega$	x_{ij} nova susceptância a ser instalada no ramo $(i, j) \in \Omega$. n_{ij}^o número de linhas iniciais no ramo $(i, j) \in \Omega$. $\bar{\phi}_{ij}$ definida pela relação: $\bar{\phi}_{ij} = \frac{\bar{f}_{ij}}{n_{ij}}$. $\bar{f}_{ij}$ fluxo máximo de potência ativa no ramo (i, j) . $\bar{g}$ vetor de limites de geração. $\bar{n}_{ij}$ vetor de número máximo de adições no ramo (i, j) . Ω conjunto de todos os ramos definidos pelas linhas existentes e as alternativas de expansão. S matriz de incidência nó-ramo (ramos $\in \Omega_1$). f vetor de fluxo de potência ativa nos ramos.
MODELO HÍBRIDO ‡	
$\text{Min } v = \sum_{(i,j) \in \Omega} c_{ij} n_{ij} + \sum_i \alpha_i r_i$ $\text{s.a. } B_1\theta + S_2f + g + r = d$ $ \theta_i - \theta_j \leq \bar{\phi}_{ij} \quad \forall (i, j) \in \Omega_1$ $ f_{ij} \leq (n_{ij} + n_{ij}^o)\bar{\phi}_{ij} \quad \forall (i, j) \in \Omega_2$ $0 \leq g \leq \bar{g}$ $0 \leq r \leq d$ $0 \leq n_{ij} \leq \bar{n}_{ij}$	f_{ij} fluxo de potência ativa no ramo (ij) . B_1 matriz de susceptância (ramos $\in \Omega_1$). S_2 matriz de incidência nó-ramo (ramos $\in \Omega_2$). Ω_1 conjunto de ramos onde já existem linhas na configuração básica. Ω_2 conjunto de ramos candidatos em caminhos onde não existe linha na configuração inicial.
* [vários autores]; † [Garver - 1970]; ‡ [Villasana, et al. - 1985]	

Com isso, o problema torna-se sempre factível, pois as sobrecargas no sistema, que eventualmente apareceriam no processo de resolução, são compensadas pelos geradores artificiais alocados nas barras de carga.

Para uma determinada proposta de investimento n^k (uma dada configuração) o problema torna-se num problema de programação linear (PPL). Se sua solução apresenta corte de carga ($r \neq 0$), então a proposta de investimento é infactível para a formulação original, mas é factível para a formulação apresentada. Portanto, um algoritmo eficiente que resolva o modelo deve apresentar uma solução final sem corte de carga para que essa solução seja equivalente à formulação original. O problema geral para o planejamento da transmissão em sistemas de energia elétrica, usando o modelo DC, pode ser escrito como:

$$\begin{aligned}
 \text{Min } w &= \sum_i \alpha_i r_i \\
 \text{s.a.} \quad &B(n_{ij})\theta + g + r = d \\
 &(n_{ij}^k + n_{ij}^o)|\theta_i - \theta_j| \leq (n_{ij}^k + n_{ij}^o)\bar{\phi}_{ij} \\
 &0 \leq g \leq \bar{g} \\
 &0 \leq r \leq d
 \end{aligned} \tag{2.1}$$

O sistema (2.1) é um PPL com valores de n_{ij}^k conhecidos ou determinados para uma alternativa de investimento. Portanto, para um n^k escolhido, a solução obtida por um algoritmo de PL aplicado ao sistema indicará simplesmente se o sistema apresenta corte de carga (solução infactível), ou não (solução factível), em relação a formulação original.

2.2.2 Modelo de Transportes

O modelo de transportes foi formulado por Garver [Garver - 1970], sendo a primeira proposta para planejamento de redes de transmissão que usou programação linear. Esta metodologia consiste basicamente em resolver de maneira aproximada uma versão relaxada do modelo DC. No modelo de Garver, conhecido como *modelo de transportes*, somente se leva em conta a primeira lei de Kirchhoff. Ou seja, não se leva em conta o conjunto de restrições correspondentes a segunda lei de Kirchhoff.

Assim, o modelo de transportes assume a forma apresentada na Tabela 2.1, em que S é a matriz de incidência nó-ramo e f_{ij} o fluxo de potência no ramo ij . O conjunto de restrições $Sf + g + r = d$ representa as equações correspondentes a primeira lei de Kirchhoff; as restrições $|f_{ij}| - (n_{ij}^k + n_{ij}^o)\bar{\phi}_{ij} \leq 0$ representam as restrições de capacidade de transmissão dos circuitos (linhas e/ou transformadores) e as demais restrições são restrições de limites de geração, de cargas e de circuitos adicionados em cada ramo ij .

2.2.3 Modelo Híbrido

Este modelo é uma combinação do modelo de transportes e o modelo DC. Neste, trata-se de contornar as desvantagens na solução do modelo de transportes. Adicionou-se ao mesmo somente uma parcela das restrições correspondentes a segunda lei de Kirchhoff, isto é, são consideradas como parte da formulação do problema aquelas restrições da segunda lei de Kirchhoff correspondentes às linhas existentes e eliminadas as restrições correspondentes aos novos caminhos. Logo, no *modelo híbrido*, deve-se satisfazer a primeira lei de Kirchhoff em todas as barras do sistema e a segunda lei de Kirchhoff somente nos laços já existentes na configuração base.

A formulação, assim definida, foi introduzida por Garver em [Villasana, *et al.* - 1985]. A Tabela 2.1 apresenta a formulação correspondente a este modelo. O conjunto de restrições $B_1\theta + S_2f + g + r = d$ representa as equações da primeira e da segunda lei de Kirchhoff para todas as barras do sistema e para os laços existentes na topologia base, em que S_2 é a matriz de incidência nó-ramo dos circuitos que não têm linhas na configuração base. B_1 é a matriz de susceptâncias dos circuitos que apresentam linhas na configuração base. As inequações $|\theta_i - \theta_j| \leq \bar{\phi}_{ij}$ e $|f_{ij}| \leq (n_{ij} + n_{ij}^o)\bar{\phi}_{ij}$ representam restrições de fluxo máximo para os circuitos adicionados a configuração base e aos circuitos adicionados na rede fictícia, respectivamente.

2.3 Técnicas para a Resolução do Problema

Na literatura especializada são apresentados diversos métodos aproximados que determinam planos de expansão de redes de transmissão à longo prazo. São os chamados *algoritmos heurísticos construtivos*, descritos a seguir.

2.3.1 Algoritmos Heurísticos Construtivos

Os algoritmos heurísticos construtivos ([Garver - 1970], [Monticelli, *et al.* - 1982], [Villasana, *et al.* - 1985], [Pereira, Pinto - 1985], entre outros) se caracterizam como um procedimento passo-a-passo que, a partir da configuração base, adiciona em cada passo um ou mais circuitos até se conseguir uma adequada operação do sistema elétrico. A diferença básica entre um algoritmo e outro está no índice de sensibilidade utilizado, ou seja, em cada passo o circuito escolhido para ser adicionado é identificado por um índice de sensibilidade preestabelecido.

Estes métodos geralmente são de fácil implementação e requerem pouco esforço computacional. Com relação a qualidade da resposta, os valores obtidos ficam geralmente afastados da resposta ótima quando são resolvidos sistemas de médio e grande porte. Estas metodologias ainda hoje são empregadas no planejamento da expansão dos sistemas de transmissão pelas empresas de energia elétrica.

2.3.1.1 Algoritmo heurístico de Garver

O modelo formulado em [Garver - 1970], também conhecido como *modelo de transportes*, foi uma das primeiras propostas de modelagem matemática para planejamento de redes de transmissão que usou programação linear. A outra proposta foi a de [Kaltenbach, et al. - 1970], que utilizou o modelo de fluxo de potência DC. A modelagem proposta por Garver é uma versão relaxada do modelo DC. Para resolver o modelo de transportes, Garver apresentou um algoritmo heurístico construtivo cujo indicador de sensibilidade é obtido resolvendo um PPL.

Na metodologia de Garver, todo fluxo que não puder ser transportado pelas ligações normais, fluirão pelas ligações de sobrecarga, pois estas têm capacidades ilimitadas, e só passarão através das ligações de sobrecarga quando for impossível transportá-los pelas ligações normais, já que estas têm custos muito inferiores. Em cada estágio do processo de planejamento, deve-se resolver um PPL e assim adicionar um circuito na trajetória de maior sobrecarga. O processo é repetido até eliminar todas as sobrecargas.

O algoritmo de Garver apresenta a seguinte estrutura básica:

1. Tomar a configuração base como configuração corrente.
2. Resolver um PPL de fluxo de rede com custo mínimo para a configuração corrente. Se não existirem mais novos caminhos a serem inseridos, então pare. Caso contrário, ir para o passo 3.
3. Calcular os fluxos através de todos os novos circuitos adicionados pelo PPL. Atualizar a configuração corrente adicionando um circuito para o novo caminho que apresentar o maior valor de fluxo. Volte ao passo 2.

A vantagem da metodologia de Garver é a simplicidade na implementação do algoritmo pois ela exige somente soluções sucessivas de programação linear. A principal limitação da metodologia é que ela não garante a obtenção da solução ótima do sistema planejado.

A metodologia proposta por [Villasana, et al. - 1985] é uma extensão da metodologia de Garver, na qual se adiciona a segunda lei de Kirchhoff para a rede existente. Com isto, gera-se a formulação apresentada como *modelo híbrido* que é resolvida usando uma metodologia muito parecida com a metodologia de Garver. Aqui também não se garante a otimalidade da solução obtida. Nesta metodologia, mantêm-se os conceitos de linhas de sobrecarga e, usa-se programação linear para determinar o circuito mais sobrecarregado e, portanto, candidato a adição de um novo circuito.

A estrutura básica do algoritmo de Villasana-Garver é dada por:

Fase I:

1. Tomar a configuração base como configuração corrente.
2. Resolver um PPL para a configuração corrente. Se não existirem mais novos caminhos a serem inseridos, então pare. Caso contrário, ir para o passo 3.

3. Calcular os fluxos através de todos os novos circuitos adicionados pelo PPL. Atualizar a configuração corrente adicionando um circuito para o novo caminho que apresentar o maior valor de fluxo. Volte ao passo 2.

Fase II:

1. Ordenar os circuitos adicionados em ordem decrescente de seus custos e eliminar aqueles cuja saída não produzem cortes de carga no sistema. FIM.

2.3.1.2 Método de Mínimo Esforço

O método de Mínimo Esforço [Monticelli, et al. - 1982] baseia-se no fato de que a distribuição dos fluxos em um rede segue uma *lei de mínimo esforço* que minimiza o produto das reatâncias (p.u.) de cada ramo pelo quadrado do respectivo fluxo. Esta função de mínimo esforço, é utilizada como um índice de sensibilidade para ordenar as adições de novos circuitos ao sistema.

$$IS_{me} = \Delta Z_{ij} = -\frac{1}{2}(\theta_i - \theta_j)^2 \Delta \gamma_{ij} \quad (2.2)$$

em que $\theta_i - \theta_j$ é a diferença angular do ramo ij antes da adição, e $\Delta \gamma_{ij}$ é a variação da susceptância de um circuito no ramo ij .

Em cada passo do processo de planejamento é adicionado ao sistema aquele circuito que produz o maior impacto na distribuição de fluxos na rede, isto é, aquele que apresenta o maior valor de $|\Delta Z_{ij}|$.

A estrutura básica do algoritmo de Mínimo Esforço é dada por:

Fase I:

1. Tomar a configuração base como configuração corrente.
2. Resolver uma análise DC para a configuração corrente. Se não existirem sobrecargas, então ir para a Fase II. Caso contrário, calcular os IS_{me} e ordenar os circuitos candidatos iniciando pelo circuito que apresentar maior valor absoluto do índice. Ir para o passo 3.
3. Adicionar à configuração corrente o primeiro circuito da lista anterior. Volte ao passo 2.

Fase II:

1. Ordenar os circuitos adicionados em ordem decrescente de seus custos e eliminar aqueles cuja saída não produzem cortes de carga no sistema. FIM.

2.3.1.3 Método de Mínimo Corte de Carga

O algoritmo de Mínimo Corte de Carga [Pereira, Pinto - 1985], usado no planejamento de sistemas de transmissão, é um algoritmo heurístico de tipo construtivo que em cada passo do algoritmo produz a adição de um circuito na configuração base até que o sistema opere adequadamente, isto é, sem corte de carga.

Em cada iteração do algoritmo de mínimo corte de carga é realizada a adição de um circuito e esse circuito é selecionado de acordo com um índice de desempenho ou índice de sensibilidade. O índice de sensibilidade é um indicador do impacto que produziria a adição de um circuito no corte de carga de um sistema se o circuito fosse adicionado ao sistema elétrico. Assim, aquele circuito que possui o maior valor do índice de sensibilidade deve ser adicionado à configuração base pois é o melhor candidato para produzir uma maior diminuição no corte de carga do sistema.

A estrutura básica do algoritmo de Mínimo Corte de Carga é dada por:

Fase I:

1. Tomar a configuração base como configuração corrente.
2. Resolver um PPL para corte mínimo de carga para a configuração corrente. Se não existirem sobrecargas, então ir para a Fase II. Caso contrário, calcular os IS_{mcc} e ordenar os circuitos candidatos iniciando pelo circuito que apresentar maior valor absoluto do índice. Ir para o passo 3.
3. Adicionar à configuração corrente o primeiro circuito da lista anterior. Volte ao passo 2.

Fase II:

1. Ordenar os circuitos adicionados em ordem decrescente de seus custos e eliminar aqueles, cuja saída, não produzem cortes de carga no sistema. FIM.

O índice de sensibilidade que permite encontrar o circuito mais atrativo para adição é determinado pela seguinte relação:

$$IS_{mcc} = (\pi_i - \pi_j)(\theta_j - \theta_i) \quad (2.3)$$

em que π_j é o multiplicador de Lagrange da j -ésima restrição do sistema $B\theta + g + r = d$ e os θ_j são os ângulos de tensão de barra obtidos ao resolver o *modelo DC* para a configuração corrente usando um algoritmo de PL.

2.3.2 Algoritmos de Otimização Clássica

Em meados da década de 80, iniciou-se uma nova fase na tentativa de resolver o problema do planejamento da expansão da transmissão de maneira ótima e a principal ferramenta matemática

encontrada foram as técnicas de decomposição matemática. Nesta perspectiva, a metodologia mais usada foi a *técnica de decomposição de Benders* [Benders - 1962], a qual explora a decomposição natural do problema de planejamento da expansão de sistemas de transmissão em duas partes, ou seja:

- **Um subproblema de investimento** em que se escolhe um plano de expansão candidato e, calculam-se os custos de investimento associados ao mesmo.
- **Um subproblema de operação** onde é testado o plano de expansão candidato em termos do adequado atendimento da carga.

A busca por um ótimo global é feita através de uma resolução iterativa das resoluções separadas dos subproblemas de operação e investimento.

Em [Pereira, et al. - 1985], o uso desta técnica é utilizado com a rede de transmissão representada pelo fluxo de potência linearizado e pelo modelo de transportes combinados com programação linear. Em [Santos, et al. - 1989] é proposta uma formulação através de um modelo de otimização não linear inteiro misto que é resolvido através de técnicas de relaxação/projeção. Em [Romero, Monticelli - 1994a] o planejamento estático da expansão da transmissão é formulado como um problema de otimização e são utilizados os modelos: de transportes, modelo DC e modelo híbrido. Em [Romero, Monticelli - 1994b] são propostas formulações através de modelos de otimização de grande porte resolvidos através de técnicas de decomposição. Estes modelos representam o problema de decisão sobre circuitos que devem ser incorporados à rede de transmissão.

[Gallego - 1997] relata que com as metodologias de decomposição foram obtidas as soluções globais em sistemas de pequeno e médio porte, como é o caso do sistema Garver de 6 nós e 15 linhas e o sistema Sul brasileiro de 46 nós e 79 linhas. Já para o sistema Norte-Nordeste de 87 nós e 179 linhas, sistema de grande porte, a metodologia de decomposição não obteve convergência devido ao número incalculável de alternativas que se apresentam na última fase do processo onde se tem que resolver um problema de programação inteira.

Ainda em [Gallego - 1997] foram iniciadas novas pesquisas relacionadas com os métodos de otimização combinatórias, cujas características fundamentais são as de resolver sistemas de grande porte, chegar a soluções próximas ao ótimo global e obter soluções em tempos de computação razoáveis.

2.3.3 Algoritmos de Otimização Combinatória

Nas últimas décadas uma grande variedade de problemas surgiram em diversas áreas, tais como: ciências matemáticas, ciências da computação, engenharia, etc. Entre todos os problemas de otimização combinatória, o problema do caixeiro viajante é provavelmente o mais conhecido. Neste problema o caixeiro inicia o percurso em uma cidade, visita cada uma das cidades prescritas na lista e retorna à cidade de partida, de tal maneira que o comprimento da rota seja o mínimo possível. A importância deste problema está no aspecto que combina as características típicas de grandes problemas de otimização combinatória, com suas complexidades (problemas *NP-completo*).

Uma consequência direta da propriedade de *NP-completo* é que a solução ótima não pode ser obtida em razoáveis tempos de computação. Grandes problemas *NP-completo* podem ser resolvidos com algoritmos apropriados; estes, por sua vez, podem ser selecionados entre duas opções: uma que obtém o ótimo global com a desvantagem de requerer tempos de computação impraticáveis, e a outra que obtém soluções sub-ótimas com menor tempo de computação.

A primeira opção, constitui a classe de algoritmos de otimização. Exemplos bem conhecidos são métodos de enumeração usando planos de corte (“*branch and bound*”, entre outros) ou *técnicas de programação dinâmica*. A segunda opção, constitui a classe de algoritmos de aproximação, também freqüentemente chamados algoritmos heurísticos, tais como: *algoritmos de busca local* e *algoritmos randômicos*. A divisão entre estas duas classes de algoritmos não é muito restrita. Alguns tipos de algoritmos podem ser empregados com ambos propósitos; por exemplo, regras de limitação heurística de “branch and bound” podem ser facilmente transferidas a um algoritmo aproximado.

Além disso, distinguem-se em ambas classes: *algoritmos gerais* e *algoritmos especializados*. Os primeiros, aplicáveis a uma grande variedade de problemas e portanto podem ser chamados problemas independentes. Os segundos, empregam informação específica e sua aplicação é por conseguinte limitada a um conjunto restrito de problemas. Nos algoritmos especializados, em cada tipo de problema de otimização combinatória um novo algoritmo é construído.

Algoritmos tais como: “simulated annealing”, genéticos e busca tabu são algoritmos gerais de alta qualidade, que estão sendo empregados na solução de sistemas de grande porte como é o caso do problema de planejamento da expansão da transmissão. O algoritmo “simulated annealing” faz uso de conceitos descritos na Mecânica Estatística. No processo físico o material é esquentado a uma temperatura elevada para logo ser resfriado de maneira lenta de tal forma que a estrutura do material ao final do processo fique perfeita. O algoritmo genético é uma técnica de busca baseada na mecânica da seleção natural das espécies. Busca tabu nasce de regras geradas na inteligência artificial; neste, se conta com um esquema de busca local para explorar o espaço de soluções além do ótimo local.

No próximo capítulo as metodologias de otimização combinatória mencionadas anteriormente são melhor detalhadas.

2.4 O Problema da Inicialização dos Algoritmos Combinatórios

No processo de obtenção de novas configurações nos algoritmos combinatórios, sempre se parte de uma determinada configuração inicial ou de uma população inicial para se gerar uma nova configuração ou população. Em sistemas de pequeno e médio porte, independentemente das configurações iniciais os algoritmos combinatórios normalmente conseguem determinar sempre novas configurações que representam melhores soluções que as anteriores. O mesmo não ocorre para sistemas de grande porte, em que dependendo da qualidade das configurações iniciais estes algoritmos ficam presos a ótimos locais ou mesmo não apresentam boas soluções ao término de seu processamento. É importante, portanto, construir configurações iniciais de boa qualidade para que o desempenho dos algoritmos combinatórios se mostrem satisfatórios.

Particularmente, para o problema do planejamento da expansão da transmissão, a obtenção de configurações iniciais factíveis pode ser feita de duas maneiras: por inicialização aleatória ou inicialização através de métodos aproximados.

A inicialização aleatória, como o próprio nome estabelece, determina aleatoriamente o número de linhas que comporão o ramo candidato (número limitado por um valor máximo de linhas por ramo). Esta inicialização é utilizada para sistemas de pequeno e médio porte. Para sistemas de grande porte, é utilizada a inicialização através dos métodos aproximados descritos anteriormente, ou seja, Garver, Villasana-Garver, mínimo esforço e mínimo corte de carga.

Para se obter uma melhoria de desempenho dos algoritmos combinatórios pode-se, entre outras coisas, compor a população inicial com configurações obtidas através de diferentes métodos, de maneira que as características próprias de cada uma delas estabeleçam uma população inicial altamente diversificada, ampliando a região de busca no espaço das soluções do problema.

Neste trabalho, especificamente no Capítulo 4, página 47, é proposto um time assíncrono inicializador formado por diferentes métodos aproximados, para a inicialização dos algoritmos combinatórios.

Na seqüência, são apresentados mais detalhes da teoria que envolve os Algoritmos Genéticos, “Simulated Annealing”, Busca Tabu e Times Assíncronos.

Capítulo 3

As Metaheurísticas

3.1 Introdução

Otimização Linear e Não-Linear, a busca por um ótimo de uma função de variáveis contínuas, foi o principal desafio nas décadas de 50 e 60, tendo como principal exemplo o *algoritmo simplex* para resolver problemas de programação linear.

Por outro lado, os principais resultados na *otimização combinatória*, a busca por ótimos de funções de variáveis discretas, foram obtidas predominantemente na década de 70, sendo o *TSP* - “*Travelling Salesman Problem*”, o problema mais conhecido.

Levantamentos dos desenvolvimentos históricos na construção e investigação de métodos para resolver de maneira ótima ou aproximada problemas de otimização combinatória, são apresentados em diversos trabalhos nas últimas décadas, envolvendo programação inteira, linear e não-linear, bem como programação dinâmica. Entre esses trabalhos destacam-se: [Ford, Fulkerson - 1962], [Dantzig, et al. - 1963], [Simonnard - 1966], [Hu - 1970], [Aho, et al. - 1974], [Lawler, et al. - 1985a], [Lawler, et al. - 1985b] e [Schrijver - 1986], entre outros. (*Ver Bibliografia: Gerais, pág. 159*)

Existe uma conjectura, obtida no final da década de 60, que distingue classes “*easy*” e “*hard*” de problemas de otimização combinatória. Esta, foi formalizada durante a década de 70, resultando na teoria de *NP-completeness* ([Cook - 1971], [Miller, Thatcher - 1972]) que aumentou grandemente o sentimento na relação entre os problemas “*hard*”. Nos anos que se seguiram, tem sido mostrado que muitos problemas de otimização combinatória teóricos e práticos pertencem a classe dos problemas “*NP-complete (non-deterministic polynomial-time complete)*” [Garey, Johnson - 1979].

Na resolução dos problemas “*NP-complete*” de grande porte, tem-se os *algoritmos de otimização*, que buscam o ótimo demandando grande soma de tempo de computação, e os *algoritmos de aproximação*, com soluções sub-ótimas obtidas rapidamente. Dentre os algoritmos de otimização destacam-se os *métodos de enumeração* usando plano de corte, “*branch and bound*” ou técnicas de programação dinâmica [Papadimitriou, Steiglitz - 1982]. Exemplos de algoritmos de aproximação, também conhecidos por *algoritmos heurísticos*, são os algoritmos de *busca local* e de “*randomization*”. Pode-se também distinguir nestas duas classes os algoritmos *gerais*, aplicáveis independentes

do problema, e os *feitos sob medida*, que usam informações específicas do problema.

[Nicholson - 1971] define uma *heurística* como um procedimento “para resolver problemas por uma proposição na qual o problema pode ser interpretado e explorado inteligentemente para se obter uma solução razoável”. E [Reeves, Beasley - 1993] a define como: “uma técnica que busca soluções boas (ou seja, próximas do ótimo) para um custo computacional razoável sem estar apta a garantir tanto factibilidade ou otimalidade, ou ainda em muitos casos estabelecer o quanto uma solução particular está próxima à otimalidade”.

Durante as últimas décadas, *heurísticas clássicas* foram desenvolvidas em diferentes estágios, sendo que na década de 80 tornaram-se as proposições mais populares para solução de problemas práticos. [Zanakis, et al. - 1989], apresentam um extenso levantamento das mesmas cobrindo diversas áreas de diferentes aplicações, classificando-as em seis categorias: “*construction*”, “*improvement*”, “*mathematical programming*”, “*decomposition*”, “*partitioning*”, e “*solution space restriction and relaxation*”. Outros trabalhos também apresentam discussões sobre estas heurísticas ([Eglese - 1986], [Silver, et al. - 1980]).

Devido ao grande avanço tecnológico dos últimos anos, as heurísticas clássicas foram revistas no sentido de aperfeiçoar seus desempenhos, resultando na nova classe de heurísticas chamadas *metaheurísticas*; classe de métodos aproximados desenvolvidos a partir de meados da década de 80, que são projetados para atacar problemas “hard” de otimização combinatória onde heurísticas clássicas tem falhado em eficiência.

[Osman - 1996] define *metaheurísticas* como sendo “um processo de geração iterativo o qual guia uma heurística subordinada, combinando inteligentemente diferentes conceitos de fenômenos naturais para exploração, e explorando os espaços de busca usando estratégias de aprendizagem para estruturar informação no sentido de encontrar eficientemente soluções próximas do ótimo”. Esta definição enfatiza as principais características que são encontradas nas melhores implementações metaheurísticas: combinações híbridas de várias idéias, exploração de informação disponível armazenada em estruturas flexíveis de memória para buscar e explorar os espaços inteligentemente, eficiência para refletir solicitação computacional razoável e “próxima-otimalidade” para enfatizar a alta qualidade das soluções obtidas sem garantir a otimalidade. Estas características são o que diferencia as metaheurísticas dos seus princípios originais, e sem as quais não teriam sucesso.

A seguir, são apresentados alguns conceitos básicos, utilizados neste trabalho, baseados principalmente nas definições de [Aarts, Korst - 1989].

3.2 Conceitos Básicos

3.2.1 O Problema de Otimização Combinatória

[Lawler - 1976] define a otimização combinatória como sendo “o estudo matemático de encontrar arranjo, agrupamento, ordenação ou seleção ótimos de objetos discretos normalmente finitos em números”.

Um *problema de otimização combinatória* é tanto um *problema de minimização* quanto um

problema de maximização e é especificado por um conjunto de instâncias do problema. Sendo que uma *instância* de um problema de otimização combinatória pode ser formalizado como um par $(\mathcal{S}, f)$, onde o *espaço de solução* $\mathcal{S}$ denota um conjunto finito de todas as soluções possíveis e a *função custo* f é um mapeamento definido como

$$f : \mathcal{S} \rightarrow \mathbb{R}. \quad (3.1)$$

No caso de minimização, o problema é encontrar um solução $i_{opt} \in \mathcal{S}$ que satisfaz

$$f(i_{opt}) \leq f(i), \text{ para todo } i \in \mathcal{S}. \quad (3.2)$$

No caso de maximização, i_{opt} satisfaz

$$f(i_{opt}) \geq f(i), \text{ para todo } i \in \mathcal{S}. \quad (3.3)$$

Tal solução i_{opt} é chamada uma *solução globalmente ótima*, ou simplesmente um *ótimo*; $f_{opt} = f(i_{opt})$ denota o custo ótimo, e $\mathcal{S}_{opt}$ o conjunto de soluções ótimas.

3.2.2 Vizinhaça

Seja $(\mathcal{S}, f)$ uma instância de um problema de otimização combinatória. Então a *estrutura de vizinhaça* é um mapeamento

$$\mathcal{N} : \mathcal{S} \rightarrow 2^{\mathcal{S}}, \quad (3.4)$$

o qual define para cada $i \in \mathcal{S}$ um conjunto $\mathcal{S}_i \subset \mathcal{S}$ de soluções que são “próximas” a i . O conjunto $\mathcal{S}_i$ é chamado a *vizinhaça* de solução i , e cada $j \in \mathcal{S}_i$ é chamada uma *solução vizinha* ou *vizinho* de i . Dessa forma, assume-se que $j \in \mathcal{S}_i \Leftrightarrow i \in \mathcal{S}_j$.

3.2.3 Mecanismo de Geração

Seja $(\mathcal{S}, f)$ uma instância de um problema de otimização combinatória e $\mathcal{N}$ uma estrutura de vizinhaça. Então um *mecanismo de geração* é uma maneira de selecionar uma solução j da vizinhaça $\mathcal{S}_i$ de uma solução i .

3.2.4 Algoritmo de Busca Local

Dada uma instância de um problema de otimização combinatória e uma estrutura de vizinhaça, o *algoritmo de busca local* é um algoritmo que itera sobre um número de soluções. Ele parte com uma solução inicial dada, o qual é frequentemente escolhida aleatoriamente. A seguir, pela aplicação de um mecanismo de geração, ele tenta encontrar a melhor solução pela busca da vizinhaça da solução atual para uma solução com custo mais baixo. Se tal solução é encontrada, a solução atual é trocada por esta solução; caso contrário, o algoritmo continua com a solução atual. O algoritmo termina quando nenhuma melhoria pode ser obtida.

A Figura 3.1, a seguir, ilustra um programa de um algoritmo de busca local, em pseudo-Pascal.

```

procedure LOCAL_SEARCH;
begin
  INITIALIZE ( $i_{start}$ );
   $i := i_{start}$  ;
  repeat
    GENERATE ( $j$  from  $S_i$ );
    if  $f(j) < f(i)$  then  $i := j$ ;
  until  $f(j) \geq f(i)$ , for all  $j \in S_i$ ;
end;

```

Fonte: [Aarts, Korst - 1989]

Figura 3.1: Descrição de um algoritmo de busca local

3.2.5 Otimalidade Local

Seja $(\mathcal{S}, f)$ uma instância de um problema de otimização combinatória e $\mathcal{N}$ uma estrutura de vizinhança, então $\hat{i} \in \mathcal{S}$ é chamado uma *solução ótima localmente* ou simplesmente um *ótimo local* com relação a $\mathcal{N}$ se $\hat{i}$ é melhor ou igual a todas suas soluções de vizinhança com relação a seu custo. Mais especificamente, se no caso de minimização, $\hat{i}$ é chamado uma *solução mínima localmente* ou simplesmente um *mínimo local* se

$$f(\hat{i}) \leq f(j) \quad , \text{ para todo } j \in \mathcal{S}_{\hat{i}}, \quad (3.5)$$

e no caso de maximização, $\hat{i}$ é chamado uma *solução máxima localmente* ou simplesmente um *máximo local* se

$$f(\hat{i}) \geq f(j), \text{ para todo } j \in \mathcal{S}_{\hat{i}}. \quad (3.6)$$

3.2.6 Estrutura de Vizinhança Exata

Seja $(\mathcal{S}, f)$ uma instância de um problema de otimização combinatória e $\mathcal{N}$ uma estrutura de vizinhança. Então $\mathcal{N}$ é chamada *exata* se, para cada $\hat{i} \in \mathcal{S}$ que é ótimo localmente com relação a $\mathcal{N}$, $\hat{i}$ é também ótimo globalmente.

Neste trabalho procurou-se investigar os princípios básicos de algumas metaheurísticas aplicáveis ao problema de otimização combinatória: *algoritmos genéticos*, “*simulated annealing*” e *busca tabu*, bem como, *times assíncronos*.

3.3 Algoritmos Genéticos

Os algoritmos genéticos (“Genetic Algorithms” - **GA**) são técnicas de busca de caráter geral baseadas nos princípios dos mecanismos genéticos e de evolução, observados nas existências de vida

dos sistemas da natureza e populações. Seu princípio básico é a manutenção de uma população de soluções (genótipos) para um problema na forma de informações individuais codificadas que desenvolve-se no tempo. A evolução é baseada em leis de seleção natural (sobrevivência do mais preparado) e recombinação de informação genética dentro da população. O desenvolvimento da população testa o espaço de busca e acumula conhecimentos sobre áreas de boa ou má qualidade, recombina este conhecimento para formar soluções com ótimo desempenho para problemas específicos.

No princípio, uma população de M soluções é gerada aleatoriamente, codificada em cadeias de símbolos (preferencialmente cadeias binárias) representando cromossomos naturais. Cada membro da população é então codificado para uma solução do problema real e um *valor de "aptidão"* ("fitness") é atribuído para o mesmo por uma função qualidade que dá a medida da qualidade da solução.

Quando a evolução é completada, indivíduos da população são selecionados em pares para reproduzir e formar indivíduos "descendentes" (ou seja, novas soluções do problema). A seleção é desenvolvida probabilisticamente de modo que uma probabilidade de seleção do indivíduo é proporcional à aptidão do mesmo. Isto assegura que soluções de alta qualidade serão selecionadas muitas vezes e tornam-se *pais* de muitas novas soluções enquanto soluções de baixa qualidade contribuirão menos para a nova população com a probabilidade de não serem escolhidas para reprodução.

Quando dois pais são selecionados suas cadeias de símbolos são combinadas para produzirem uma solução "de descendência" usando genética como operadores. Os principais operadores utilizados são: "*Crossover*" (Recombinação), *Mutação* e *Inversão*.

"Crossover" - simplesmente combina as cadeias de símbolos dos pais, formando uma nova cadeia de cromossomos, que herdará características de solução de ambos. Apesar do "crossover" ser o operador principal de busca, não produz informação que já não exista dentro da população.

Mutação - cobre a necessidade anterior, injetando novas informações na cadeia produzida. A injeção é feita alterando aleatoriamente símbolos dos novos cromossomos. Geralmente mutação é considerada como um operador secundário, mas não inútil, que dá uma probabilidade não nula de cada solução ser considerada e avaliada.

Inversão - dois pontos de cortes são escolhidos aleatoriamente e a ordem dos símbolos são trocadas. O objetivo é reduzir o comprimento de *esquemas* bons e aumentar sua ligação de combinações de genes.

Quando M novas soluções são produzidas, elas são consideradas como uma nova geração e elas trocam totalmente os pais no sentido de continuar a evolução. Muitas gerações são necessárias para a população convergir para a solução ótima ou próxima da ótima, o número aumenta de acordo com a dificuldade do problema.

3.3.1 Descrição dos Algoritmos Genéticos

[Davis - 1991] descreve de uma maneira generalizada, um algoritmo genético:

PASSO 1 - inicie a população de soluções (cromossomos). Isto pode ser feito por uma heurística ou por um gerador randômico.

PASSO 2 - avalie cada cromossomo na população usando a função aptidão ("fitness").

PASSO 3 - selecione cromossomos da população atual e aplique operadores de recombinação e de mutação para gerar novos cromossomos. Normalmente, cromossomos com os melhores valores de aptidão têm a maior probabilidade de serem selecionados por recombinação.

PASSO 4 - selecione e delete alguns cromossomos da população atual para abrir espaços para novos cromossomos. Cromossomos com os piores valores de aptidão têm a maior probabilidade de serem deletados.

PASSO 5 - avalie os novos cromossomos pela função aptidão e insira-os na população.

PASSO 6 - se um critério de convergência é satisfeito, então retorne os melhores cromossomos e PARE o processo. Caso contrário, vá para o PASSO 3.

Considere a Tabela 3.1 a seguir, que apresenta uma população de 8 cromossomos gerados aleatoriamente no sentido de maximizar a função $g(x) = x$ sobre o conjunto de inteiros de 0 a 63. Para usar **GA**, a variável x deve ser codificada como uma cadeia de bit de comprimento 6, de modo que 0 (000000) e 63 (111111) possam ser obtidos. Isto é feito codificando x pela identidade: $x = x_1 + 2x_2 + 4x_3 + 8x_4 + 16x_5 + 32x_6$, em que $x_i \in \{0, 1\}$. O valor de aptidão de cada cromossomo é $g(x)$ codificado no cromossomo.

Tabela 3.1: Uma população de cromossomos

Cromossomo 1:	110001	Aptidão:	35
Cromossomo 2:	010101	Aptidão:	42
Cromossomo 3:	110101	Aptidão:	43
Cromossomo 4:	100101	Aptidão:	41
Cromossomo 5:	000011	Aptidão:	48
Cromossomo 6:	010011	Aptidão:	50
Cromossomo 7:	001100	Aptidão:	12
Cromossomo 8:	101010	Aptidão:	21

Algumas definições são apresentadas aqui, para facilitar o entendimento.

esquema - descreve um subconjunto de cadeias com similaridades para certas posições. É composto por 0's e 1's como os cromossomos mas um símbolo genérico permite 0 ou 1, ou seja, um esquema é uma cadeia de $(x_1, \dots, x_n)$ onde $x_i \in \{0, 1, *\}$.

instância - representa subconjuntos de cromossomos, ou seja, cada cadeia representada por um esquema.

O esquema 110*01 permite os cromossomos 1 e 3, da Tabela 3.1, enquanto o esquema 0*0011 permite os cromossomos 5 e 6.

Existe duas características de um esquema:

ordem - é o número de posições fixas de um esquema. Por exemplo, 11**** tem ordem 2, enquanto 110*00 tem ordem 5)

comprimento de definição - é a distância entre a primeira e última posições fixas. (11**** tem comprimento 1 e 110*00 tem comprimento 5).

bloco de construção - é um esquema de ordem baixa, comprimento de definição pequeno e com valor de aptidão acima do valor médio de seus membros na população.

A hipótese básica é que pode ser encontrada a melhor descendência combinando as melhores características de dois bons cromossomos. Na Tabela 3.1, os dois blocos de construção acima da média, 11**** e ****11, são combinados sobre um único cromossomo para criar uma descendência com uma aptidão maior. A essência do teorema fundamental dos **GA** estabelece que esquema com valores de aptidão altos e comprimentos de definição baixos, aumentam exponencialmente com o tempo.

Em um ciclo de geração o **GA** processa somente M cadeias (M é o tamanho da população), mas implicitamente ele avalia aproximadamente M^3 esquemas [Goldberg - 1989]. Esta capacidade dos **GA** de processar simultaneamente um grande número de esquemas, chamada *paralelismo implícito*, vem do fato que uma cadeia representa simultaneamente 2^n diferentes esquemas.

O tamanho da população é um parâmetro importante a se considerar, pois seu valor permanece constante durante a busca e é igual ao da população temporária. Se o tamanho da população é muito pequeno, pode ocorrer uma convergência prematura de toda a população dentro de um ótimo local.

A Tabela 3.2 a seguir, dá uma idéia dos diferentes operadores genéticos utilizados. É mostrado um “crossover”, com dois pontos de corte, operando sobre os cromossomos 1 e 6 para produzir as descendências 1 e 2. A descendência 3 é obtida da descendência 1 variando a quarta posição de 0 a 1 aleatoriamente por uma operação de mutação. E finalmente, uma inversão é obtida entre as posições 2 e 5 no cromossomo 1 fornecendo a descendência 4 na tabela.

Tabela 3.2: Operadores de algoritmos genéticos.

Cromossomo 1:	11 <u>000</u> 1	Aptidão:	35
Cromossomo 6:	01 <u>001</u> 1	Aptidão:	50
Descendência 1:	11 <u>001</u> 1	Aptidão:	51
Descendência 2:	01 <u>000</u> 1	Aptidão:	34
Descendência 3:	110 <u>1</u> 11	Aptidão:	59
Descendência 4:	1 <u>0001</u> 1	Aptidão:	49

Algoritmos Genéticos fazem uso de gerações de populações que são criadas com o tempo. A população velha é totalmente ou parcialmente substituída por novos cromossomos antes de iniciar a fase de avaliação.

Para se evitar os problemas: de que nas gerações iniciais o valor médio de aptidão da população seja baixo e os poucos valores altos levem a alocação de um grande número de descendência (causando convergência prematura), e de que nos últimos estágios a variância dos valores de aptidão seja pequena (causando uma alocação de igual número de descendência e consequentemente diminuindo o efeito de promoção das melhores cadeias); são propostos *mecanismos de dimensionamento e de seleção* no sentido de se superar estes problemas.

3.3.1.1 Dimensionamento linear

Envolve o reajustamento dos valores de aptidão das cadeias, como segue.

$$f' = af + b \quad (3.7)$$

em que:

- $f =$ valor de aptidão;
- $f' =$ valor de aptidão redimensionado;
- $a, b =$ constantes escolhidas adequadamente.

3.3.1.2 Dimensionamento de truncamento sigma

É feito considerando o *desvio padrão* dos valores de aptidão, como segue.

$$f' = f - (\bar{f} - c\sigma) \quad (3.8)$$

em que:

- $\bar{f} =$ valor de aptidão médio;
- $\sigma =$ desvio padrão dos valores de aptidão da população;
- $c =$ constante variando de 1 a 3.
- $f < c\sigma$ são descartados.

3.3.1.3 Seleção baseada na ordenação

A população é ordenada de acordo com os valores de aptidão de seus membros. A seleção é estabelecida seguindo uma função probabilidade de distribuição pré-definida. [Whitley - 1989] desenvolveu um **GA** sequencial que usa seleção ordenada em um algoritmo *em regime permanente* no qual descendências são introduzidas na população uma por vez na posição correta de ordenação, ao invés da população inteira por vez.

3.3.1.4 Seleção “Breeding Pool”

A aptidão de cada cromossomo é calculada e o *valor relativo* de cada membro é calculado como segue.

$$fr_c = \frac{f_{max} - f(i_c)}{\sum_c (f_{max} - f(i_c))} \quad (3.9)$$

f_{max} é o melhor valor objetivo observado até o momento em um problema de maximização.

O número esperado de descendência para cada cromossomo é então calculado tomando o inteiro mais próximo do valor de $desc_c = fr_c \times n$. Cada cromossomo da população tem $desc_c$ cópias de si mesmo colocado no grupo de acasalamento. Pares para acasalamento são então selecionados aleatoriamente deste grupo para procriarem. Após cada geração completada, a população inteira ou alguma parte pré-definida é trocada. Na prática, uma estratégia *elitista* é usada trocando a população inteira exceto os indivíduos melhores preparados.

3.3.1.5 Seleção “Roulette Whell”

O valor de fr_c determina o tamanho do setor de uma roleta. Indivíduos são escolhidos para seleção pela geração de um número aleatório entre 0 e 1 e então movimentando através da população um indivíduo por vez, até que a aptidão relativa acumulativa seja maior do que o número aleatório. Neste caso, o indivíduo associado com o setor é selecionado.

A equivalência de **GA** aos problemas de otimização combinatória pode ser entendida pela verificação da Tabela 3.3.

Tabela 3.3: Equivalência entre otimização combinatória e algoritmos genéticos.

Otimização Combinatória	Algoritmos Genéticos
Solução codificada	Cromossomo
Solução	Cromossomo decodificado
Conjunto de soluções	População
Vizinhos	Descendência
Função objetivo	Função aptidão

Uma boa característica de **GA** é a facilidade de executá-los em paralelo. Uma arquitetura proposta por [Grefenstette, *et al.* - 1985] é o *GA concorrente assíncrono distribuído*. Nesta arquitetura, N processadores acessa uma memória compartilhada comum. Cada processador faz avaliação da função aptidão, bem como recombinações. O sistema é completamente assíncrono, exceto pela restrição de que não mais do que um processador possa acessar a mesma posição de memória ao mesmo tempo.

As implementações de **GA** de maior sucesso usam combinações híbridas de **GA** com outras técnicas, como: “*simulated annealing*” [Lin, *et al.* - 1993], [Thangiah, *et al.* - 1994] (Ver Bibliogra-

fia: “Simulated Annealing”), página 171), *busca tabu* [Thangiah, et al. - 1994], [Moscato - 1993], [Glover, et al. - 1995] (Ver Bibliografia: *Busca Tabu*, página 196), entre outras.

3.3.2 Aplicações de Algoritmos Genéticos

A Tabela 3.4 a seguir, apresenta algumas áreas de aplicação de Algoritmos Genéticos.

Tabela 3.4: Algumas áreas de aplicações de algoritmos genéticos.

Aplicações de Algoritmos Genéticos
“Assignment Problems” e variantes
Problemas de Otimização Combinatória “Bin Packing”, “Grouping”, “Ising and Spin-Glass”, “Knapsack”, Programação Linear, “Satisfiability”, “Set Covering and Partitioning”
“Design Problems” Bases de Dados, Engenharia, “Floorplan”
Problemas de Grafos Desenho e Coloração de Grafos, “Graph Partitioning”, “K-mean Partitioning”, “Steiner and Decisions Trees”
Otimização de Função
Problemas de “Layout” “Quadratic Assignment Problem”, “Networks”, “Facility”, “VLSI Placement”
Problemas Industriais “Cell and Parts Formation”, “Line Balancing”, Carregamento e Empacotamento, Planejamento da Manutenção, Planejamento de Processo e Produção, “Reliability Improvement”
Problemas de Otimização Multi-Objetivos
Redes e Telecomunicações
Trajетórias e Problemas de Transporte Trajetória de Ônibus Escolar, Transporte, Caixeiro Viajante, Trajetória de Veículos, Trajetória de Veículos com Restrições de Tempo
Problemas de Escalonamento e Seqüenciamento “Flow-shop”, Escalonamento em geral, Inspeção e Teste, “Job-shop”, Multiprocessadores, Projeto, Seqüenciamento, Máquinas Simples e Paralelas, Tabela de Horários
Problemas Estatísticos Estimação Econométrica, Regressão, Componentes Principais, Otimização da Simulação
Aplicações Diversas Química, Finanças, outras

3.4 “Simulated Annealing”

Em sua forma original ([Kirkpatrick, et al. - 1983], [Černý - 1985]) o algoritmo de “simulated annealing” (**SA**) é baseado na analogia entre a simulação de recozimento de sólidos e a resolução de problemas de otimização combinatória de grande porte. Por esta razão que o mesmo tornou-se conhecido como “*simulated annealing*”. Na física da matéria condensada, “annealing” representa um processo físico no qual um sólido em um banho quente é aquecido pelo aumento de temperatura deste banho até um valor máximo no qual todas as partículas do sólido se arranjam aleatoriamente na fase líquida, seguido por um resfriamento através de uma lenta diminuição da temperatura do banho. Dessa maneira, todas as partículas se arrumam por si só num estado de menor energia de sua estrutura cristalina (“*ground state*”), desde que a máxima temperatura seja suficientemente alta e o resfriamento seja realizado de maneira suficientemente lenta. Tendo como ponto de partida o valor máximo de temperatura, a fase de resfriamento do processo (“annealing”) pode ser descrita como segue.

Para cada valor de temperatura T , o sólido acha um *equilíbrio térmico*, caracterizado pela probabilidade de estar em um estado i com energia E_i dado pela *distribuição de Boltzmann* [Toda, et al. - 1983]:

$$P_T\{\mathbf{X} = i\} = \frac{1}{Z(T)} \exp\left(-\frac{E_i}{k_B T}\right), \quad (3.10)$$

em que:

$Z(T) = \sum_j \exp\left(-\frac{E_j}{k_B T}\right)$ é um fator de normalização, conhecido como *função partição*;

k_B é a *constante de Boltzmann*; e

$\exp\left(-\frac{E}{k_B T}\right)$ é conhecido como o *fator de Boltzmann*.

Como a temperatura decresce, a distribuição de Boltzmann concentra sobre os estados com energia mais baixa e quando a temperatura aproxima-se de zero, somente os estados de energia mínima têm uma probabilidade de ocorrência não nula. Porém, se o resfriamento é muito rápido, isto é, se não se permite que o sólido ache o equilíbrio térmico para cada valor de temperatura, defeitos podem ser *congelados* dentro do mesmo e estruturas amorfas podem aparecer em lugar de uma estrutura cristalina de baixa energia.

Para simular a evolução do equilíbrio térmico de um sólido para um valor fixo da temperatura T , [Metropolis, et al. - 1953] propôs um *método Monte Carlo*, o qual gera uma seqüência de estados no sólido como segue.

Algoritmo de Metropolis

Dado um estado atual i do sólido com *energia* E_i , então um estado subsequente j é gerado pela aplicação de um mecanismo de perturbação que transforma o estado atual em um próximo

estado por uma pequena perturbação, por exemplo, pelo deslocamento de uma partícula. A energia do próximo estado é E_j . Se a *diferença de energia*, $E_j - E_i$, for maior ou igual a 0, o estado j é aceito como o estado atual. Se $E_j - E_i > 0$, o estado j é aceito com uma certa probabilidade dada pelo fator de Boltzman. Esta regra é conhecida como *critério de Metropolis* e o algoritmo que vem com ela é conhecido como *algoritmo de Metropolis*.

Na seqüência, o algoritmo “simulated annealing” é descrito com mais detalhes.

3.4.1 Descrição de “Simulated Annealing”

O algoritmo de Metropolis pode ser usado para gerar uma seqüência de configurações em um algoritmo “simulated annealing”. Neste caso, as configurações assumem o papel dos estados de um sólido, sendo que a função custo assume o papel da energia e pode-se introduzir um *parâmetro de controle* c para representar o papel da temperatura. Dessa forma, o algoritmo “simulated annealing” pode ser visto como uma iteração do algoritmo de Metropolis, avaliado para valores decrescentes do parâmetro de controle.

A analogia entre sistemas físicos de muitas partículas e um problema de otimização combinatória, baseia-se nas equivalências da Tabela 3.5 que se segue.

Tabela 3.5: Equivalência entre otimização combinatória e “simulated annealing”.

Otimização Combinatória	“Simulated Annealing”
Solução factível	<i>Estado</i> do sólido
Custo de uma solução	<i>Energia</i> de um estado
Solução ótima	<i>Energia</i> (“ground state”) mínima
Busca local descendente	“ <i>Rapid quenching</i> ”
Parâmetro de controle	Temperatura

Admitindo a existência de uma estrutura de vizinhança e um mecanismo de geração, como nos algoritmos de busca local, pode-se definir um *critério de aceitação* como segue.

3.4.1.1 Critério de aceitação

Seja $(\mathcal{S}, f)$ uma instância de um problema de otimização combinatória e i e j duas soluções com custos $f(i)$ e $f(j)$, respectivamente. Então o *critério de aceitação* determina se j é aceita para substituir i pela aplicação da seguinte *probabilidade de aceitação*:

$$P_c(j \text{ aceita}) = \begin{cases} 1 & \text{se } f(j) \leq f(i) \\ \exp\left(\frac{f(i)-f(j)}{c}\right) & \text{se } f(j) > f(i), \end{cases} \quad (3.11)$$

em que $c \in \mathbb{R}^+$ representa o parâmetro de controle.

Definindo *transição* como segue.

3.4.1.2 Transição

Uma *transição* é uma ação combinada resultando na transformação de uma solução candidata em uma subsequente. A ação consiste dos seguintes passos:

- (i) aplicação do mecanismo de geração;
- (ii) aplicação do critério de aceitação.

E sendo c_k o valor do parâmetro de controle e L_k o número de transições geradas para a $k^{\text{ésima}}$ iteração do algoritmo de Metropolis; então o algoritmo **SA** pode ser descrito conforme mostrado na Figura 3.2, a seguir.

```

procedure SIMULATED ANNEALING;
begin
  INITIALIZE ( $i_{start}, c_0, L_0$ );
   $k := 0$ ;
   $i := i_{start}$ ;
  repeat
    for  $l := 1$  to  $L_k$  do
      begin
        GENERATE ( $j$  from  $S_i$ );
        if  $f(j) \leq f(i)$  then  $i := j$ ;
        else
          if  $\exp\left(\frac{f(i)-f(j)}{c_k}\right) > \text{random}[0, 1)$  then  $i := j$ ;
        end;
         $k := k + 1$ ;
        CALCULATE_LENGTH ( $L_k$ );
        CALCULATE_CONTROL ( $c_k$ );
      until stopcriterion
    end;

```

Fonte: [Aarts, Korst - 1989]

Figura 3.2: Descrição do algoritmo “annealing”.

De acordo com [De Souza - 1993] pode-se também resumir os algoritmos de **SA** nos passos que se seguem.

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

PASSO 2 - Gere uma solução inicial aleatória;

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

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

PASSO 5 - Aplique uma perturbação aleatória no estado atual, criando um novo estado; avalie ΔE ;

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

PASSO 7 - Diminua a temperatura;

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

O algoritmo **SA** pode ser visto como uma versão “aleatorizada” da busca local. Ele usa a mesma representação de um algoritmo de busca local adicionado de um *plano de resfriamento*, que especifica:

- O valor inicial de partida do parâmetro de temperatura T .
- A taxa de resfriamento e a regra de atualização da temperatura.
- O número de iterações a ser executado para cada temperatura.
- O término do algoritmo (critério de parada).

O desempenho do algoritmo **SA** depende fortemente do plano de resfriamento escolhido. Com um plano apropriado, soluções próximas do ótimo podem ser obtidas para muitos problemas de otimização combinatória. Na literatura, encontra-se uma grande variedade de planos de resfriamento práticos e teóricos, e os mesmos são classificados em três categorias [Osman, Christofides - 1994]:

redução em degraus - um número de iterações são executadas antes que a temperatura seja reduzida [Kirkpatrick, et al. - 1983].

redução contínua - a temperatura é atualizada para cada iteração [Lundy, Mees - 1986].

redução não-monotônica - a temperatura é reduzida a cada iteração com ocasional aumento na temperatura [Lundy, Mees - 1986], [Connolly - 1990], [Osman, Christofides - 1994], [Dowsland - 1995]. (Ver Bibliografia: “Simulated Annealing”, página 171)

Os algoritmos de “Simulated Annealing” têm sido combinados com *algoritmos genéticos*, *busca tabu*, entre outros, [Stewart, et al. - 1994], [Osman, Laporte - 1996]).

3.4.2 Aplicações de “Simulated Annealing”

Na Tabela 3.6 a seguir, são apresentadas a título de ilustração, algumas aplicações de algoritmos de **SA**, correspondentes ao período de 1983 a 1987, conforme levantamento realizado por [Aarts, Korst - 1989]. Nesse trabalho os autores mencionam inúmeros livros e revisões bibliográficas de **SA**, entre as quais destacam-se: [Van Laarhoven, Aarts - 1987], [Wong, et al. - 1988], [Collins, et. al. - 1987], [Wille - 1986a], [Aarts, Van Laarhoven - 1987], [Hajek - 1985]. (Ver Bibliografia: “Simulated Annealing”, página 171)

Tabela 3.6: Levantamento de aplicações de algoritmos “simulated annealing”.

Problemas Básicos	Trabalhos
“ <i>Travelling Salesman</i> ”	Aarts & Van Laarhoven (1988), Aarts, Korst & Van Laarhoven (1988), Bonomi & Lutton (1984), Černý (1985), Felten, Karlin & Otto (1985), Golden & Skiscim (1986), Johnson <i>et al.</i> (1988), Kirkpatrick (1984), Kirkpatrick & Toulouse (1985), Van Laarhoven (1988), Randelman & Grest (1986), Rossier, Troyon & Liebling (1986), Skiscim & Golden (1983), Sourlas (1988).
“ <i>Graph Partitioning</i> ”	Aarts & Van Laarhoven (1985), Johnson <i>et al.</i> (1987), Kirkpatrick (1984), Van Laarhoven (1988), Sheild (1987).
“ <i>Matching</i> ”	Lutton & Bonomi (1986), Sasaki & Hajek (1988), Telley, Liebling & Mocellin (1987), Weber & Liebling (1986).
“ <i>Quadratic Assignment</i> ”	Bonomi & Lutton (1986), Burkard & Rendl (1984), Wilhelm & Ward (1987).
“ <i>Linear Arrangement</i> ”	Bhasker & Sahni (1987), Nahar, Sahni & Shragowitz (1985).
“ <i>Graph Colouring</i> ”	Chams, Hertz & De Werra (1987), Johnson <i>et al.</i> (1988), Morgenstern & Shapiro (1986).
“ <i>Scheduling</i> ”	Abramson (1987), Chen, Wong & Pin (1987), Eglese & Rand (1987), Van Laarhoven, Aarts & Lenstra (1988), Perush (1987).
Problemas de Engenharia	Trabalhos
“ <i>VLSI Design</i> ” “ <i>Placement</i> ”	Banerjee & Jones (1986), De Bont <i>et al.</i> (1988), Casotto, Romeo & Sangiovanni-Vicentelli (1987a), Casotto, Romeo & Sangiovanni-Vicentelli (1987b), Darema-Rogers, Kirkpatrick & Norton (1987), Devadas & Newton (1986), Grover (1986), Jepsen & Gelatt (1983), Kravitz & Rutenbar (1987), Otten & Van Ginneken (1984), Rose <i>et al.</i> (1986), Sechen & Sangiovanni-Vicentelli (1985), Siarry, Bergonzi & Dreyfus (1987), Storer, Becker & Nicas (1985), Wong & Liu (1986).
“ <i>Routing</i> ”	Leong (1986), Leong & Liu (1985), Leong, Wong & Liu (1985), Vecchi & Kirkpatrick (1983).
“ <i>Array Logic Minimization</i> ”	Devadas & Newton (1986), Fleisher <i>et al.</i> (1985), Gonsalves (1986), Lam & Delosme (1986), Leong (1986), Rowen & Henessy (1985), Wong, Leong & Liu (1988).
“ <i>Testing</i> ”	Distante & Piuri (1986), Lighthart, Aarts & Beenker (1986).
“ <i>Facilities Layout</i> ”	Sharpe & Marksjo (1985), Sharpe <i>et al.</i> (1985).
“ <i>Image Processing</i> ”	Carnevalli, Coletti & Patarnello (1985), Geman (1987), Geman & Geman - 1986, Murray & Buxton (1987), Paxman, Smith & Barret (1984), Ripley (1986), Smith, Barret & Paxman (1983), Sontag & Sussmann (1986), Wolberg & Pavlidis (1985).
“ <i>Code Design</i> ”	Beenker, Claasen & Hermes (1985), Bernasconi (1987), El Gamal <i>et al.</i> (1987), Van Laarhoven <i>et al.</i> (1988), Wille (1987).
“ <i>Biology</i> ”	Dress & Kruger (1987), Goldstein & Waterman (1987), Lundy (1985).
“ <i>Physics</i> ”	Lyberatos, Wohlfarth & Chantrell (1985), Nicholson, Chowdhary & Schwartz (1984), Rothman (1985), Semenovskaya, Khachatryan & Khachatryan (1985), Telley, Liebling & Mocellin (1987), Wooten, Winer & Weaire (1985), Wille (1986b), Wille & Vennik (1985).
“ <i>Optimization</i> ”	Bohachevski, Johnson & Stein (1986), Corana <i>et al.</i> (1987), Geman & Hwang (1986), Khachatryan (1986), Vanderbilt & Louie (1984), Wille (1986a).

Fonte: [Aarts, Korst - 1989] - Ver Bibliografia: “Simulated Annealing”, página 171.

3.5 Busca Tabu

Busca Tabu (“Tabu Search” - **TS**) é um procedimento metaheurístico que guia um algoritmo heurístico de busca local para explorar o espaço de soluções além da simples otimalidade local. O procedimento local é uma busca que usa uma operação chamada *movimento* (“move”) para definir uma transição na vizinhança de qualquer solução dada. Se baseia na premissa de que para qualificar o processo de busca da solução de um problema como inteligente, deve incorporar *memória adaptativa* e *exploração sensível*. Uma analogia que se faz é com um alpinista, onde o mesmo deve lembrar seletivamente (memória adaptativa) de elementos importantes do caminho já percorrido e deve estar apto a fazer escolhas estratégicas (exploração sensível) ao longo do caminho.

Um dos principais componentes da Busca Tabu é seu uso de *memória adaptativa*, o qual cria um comportamento de busca mais flexível, ou seja, permite a implementação de procedimentos que são capazes de fazer uma busca em uma grande parte do espaço de solução com pouca solicitação de memória. Visto que escolhas locais são guiadas pela informação coletada durante a busca, a Busca Tabu contrasta com procedimentos sem memória que dependem grandemente de processos semi-aleatórios, bem como, com procedimentos de memória rígida típicos de estratégias “branch and bound”.

Por outro lado, a conjectura de *exploração sensível* na Busca Tabu é que uma escolha estratégica ruim pode fornecer mais informação do que uma escolha aleatória boa. Com o uso da memória, uma escolha estratégica ruim para um ponto particular na busca pode tornar-se em uma decisão boa para um ponto posterior no processo. Embora isto pode ser visto como uma forma primitiva de aprendizagem, a exploração sensível integra os princípios básicos da busca inteligente (exploração de características de solução boa enquanto explorando novas regiões promissoras).

A Tabela 3.7 a seguir, resume as principais características da Busca Tabu.

A finalidade da Busca Tabu é encontrar novas formas e mais eficazes para tomar vantagem dos conceitos apresentados na Tabela 3.7, e identificar princípios associados que podem empregar os fundamentos da busca inteligente. Neste sentido, pode-se dizer que a Busca Tabu é baseada em conceitos selecionados do campo da Inteligência Artificial.

Na seção que se segue, são descritos os fundamentos da Busca Tabu

3.5.1 Descrição de Busca Tabu

Busca Tabu opera sobre uma função $f(x)$ a ser otimizada sobre um conjunto X , onde $f(x)$ pode ser linear ou não-linear, e o conjunto X contém restrições sobre o vetor x . Estas restrições podem incluir desigualdades lineares ou não-lineares, e podem obrigar todos ou alguns elementos de x a receberem valores discretos. Mas, nem sempre é fácil formular o problema de interesse como uma função objetivo sujeita a um conjunto de restrições.

Tabela 3.7: Principais Características da Busca Tabu.

Memória Adaptativa
Seletividade (incluindo esquecimento estratégico)
Abstração e decomposição (através de memória explícita e por atributos)
Tempo:
atualidade de eventos
frequência de eventos
diferenciação entre curto e longo prazo
Qualidade e Impacto:
atração relativa de escolhas alternativas
tamanho das mudanças em relações de estrutura ou restrições
Contexto:
interdependência regional
interdependência estrutural
interdependência seqüencial
Exploração sensível
Imposição estratégica de limitações e induções
(<i>condições tabu e níveis de aspiração</i>)
Enfoque concentrado em boas regiões e boas características das soluções
(<i>processos de intensificação</i>)
Caracterização e exploração de novas regiões promissoras
(<i>processo de diversificação</i>)
Padrões de busca não monótonos
(<i>oscilação estratégica</i>)
Integração e extensão de soluções
(<i>reencadeamento de trajetórias</i>)

A Busca Tabu inicia da mesma maneira que a busca local ou de vizinhança, procedendo iterativamente de um ponto (solução) para outro até que seja satisfeito um critério de parada escolhido. Cada $x \in X$ tem uma vizinhança $N(x) \subset X$, e cada solução $x' \in N(x)$ é obtida de x por um operador chamado *movimento*. Normalmente assume-se que vizinhanças são simétricas, ou seja, x' é um vizinho de x se e somente se x é um vizinho de x' .

A memória usada pode ser *explícita*, que conserva soluções completas, e consiste tipicamente em uma elite de soluções visitadas durante a busca; ou *baseada em atributos*, a qual guarda informação sobre atributos das soluções que mudam ao mover-se de uma solução à outra.

Uma importante distinção surge pela diferenciação entre *memórias de curto e longo prazo*. Cada tipo de memória é acompanhado por suas próprias estratégias especiais. Contudo, o efeito de ambos tipos de memória pode ser visto como uma redução ou modificação da vizinhança $N(x)$ da solução atual x . A vizinhança modificada $N^*(x)$ é resultado da manutenção de um histórico seletivo dos estados encontrados durante a busca.

3.5.1.1 Memória de curto prazo

Nas estratégias baseadas em considerações de *curto prazo*, $N^*(x)$ caracteristicamente é um subconjunto de $N(x)$, e a classificação tabu serve para identificar elementos de $N(x)$ excluídos de $N^*(x)$. Desta forma, a Busca Tabu pode ser vista como um metodologia de vizinhança dinâmica. Este processo permite que uma solução x pode ser visitada mais de uma vez, contudo é provável que a correspondente vizinhança reduzida $N^*(x)$ será diferente a cada vez. As funções de memória de curto prazo estão normalmente no núcleo da metodologia Busca Tabu. A memória *baseada em eventos recentes*, a mais comumente utilizada, leva em conta os atributos de soluções que têm sido mudados em passado recente, atributos estes designados *tabu-ativos*. O uso de *avaliações tabu*, que atribuem penalizações grandes aos conjuntos apropriados de atributos tabu-ativos, permitem que o “status” atual do tabu sejam identificado, sendo que a duração que um atributo permanece tabu-ativo é denominada *ocupação tabu*. Esta variável ocupação pode variar para diferentes tipos de combinações de atributos como também para diferentes intervalos de tempo ou etapas de busca.

A Figura 3.3 a seguir, apresenta o esquema do processo tabu de avaliação para memória de curto prazo.

3.5.1.2 Memória de longo prazo

Em algumas aplicações, as componentes da memória de curto prazo são suficientes para produzir soluções de boa qualidade, mas, em geral, a Busca Tabu se torna significativamente mais potente incluindo-se *memória de longo prazo* e suas estratégias associadas, entre as quais a *memória baseada na frequência*, que opera introduzindo penalizações e incentivos determinados pelo intervalo relativo de tempo em que os atributos tenham permanecidos em soluções visitadas durante a busca. As *medidas de transição* mantêm um registro de com que frequência mudam os atributos, enquanto as *medidas de residência* mantêm o registro das durações relativas dos atributos nas soluções geradas.

3.5.1.3 Estratégia de intensificação

As estratégias de *intensificação* e *diversificação* são dois componentes importantes nas considerações de longo prazo. As estratégias de *intensificação* estão baseadas na modificação das regras de seleção de maneira a favorecer combinações de movimentos e características de solução que historicamente tenham sido boas. Podem também, iniciar um regresso às regiões atrativas para fazer, nelas, uma busca mais intensa. A Figura 3.4, mostra um exemplo simples deste último tipo de estratégia de intensificação.

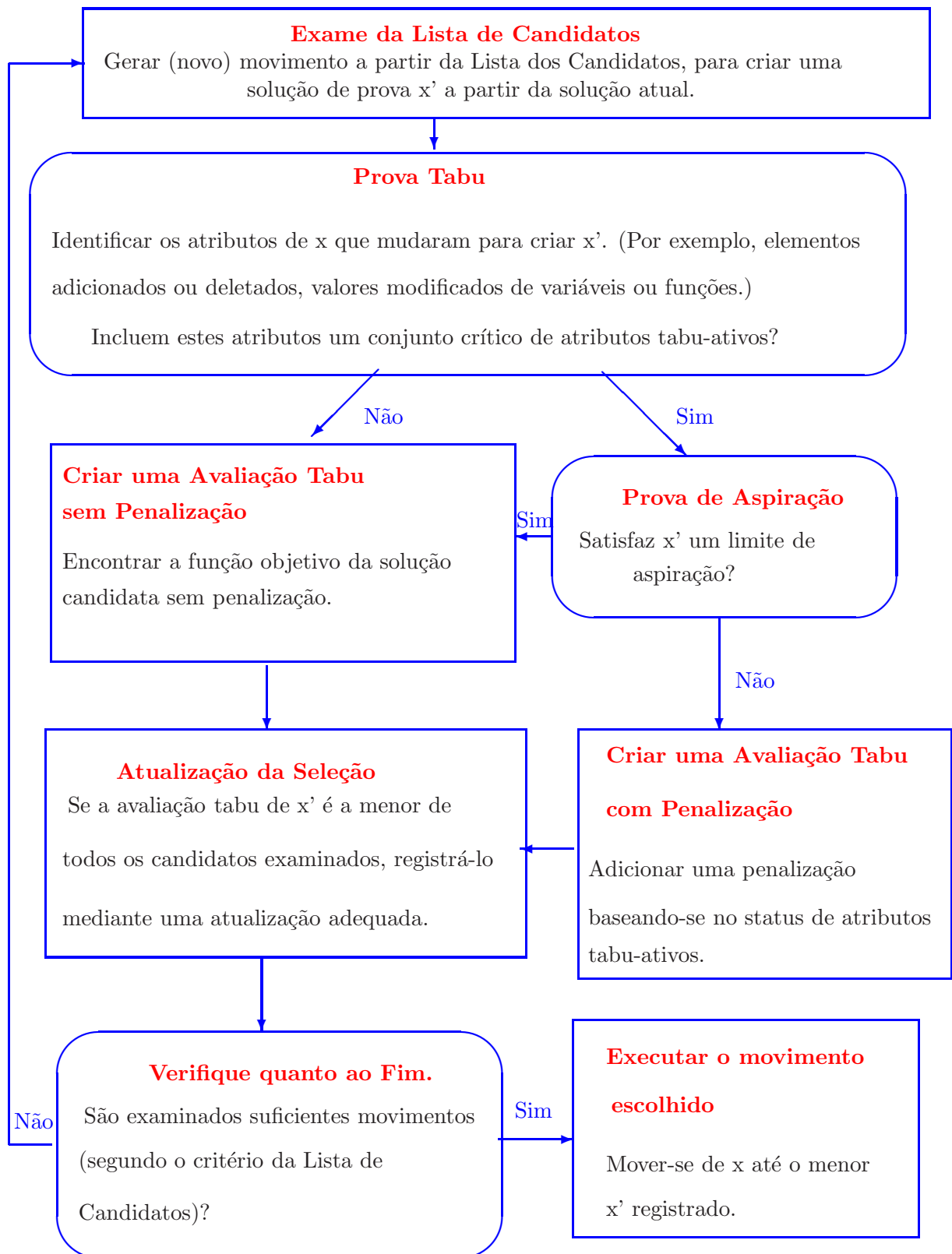


Figura 3.3: Esquema do processo tabu de avaliação.

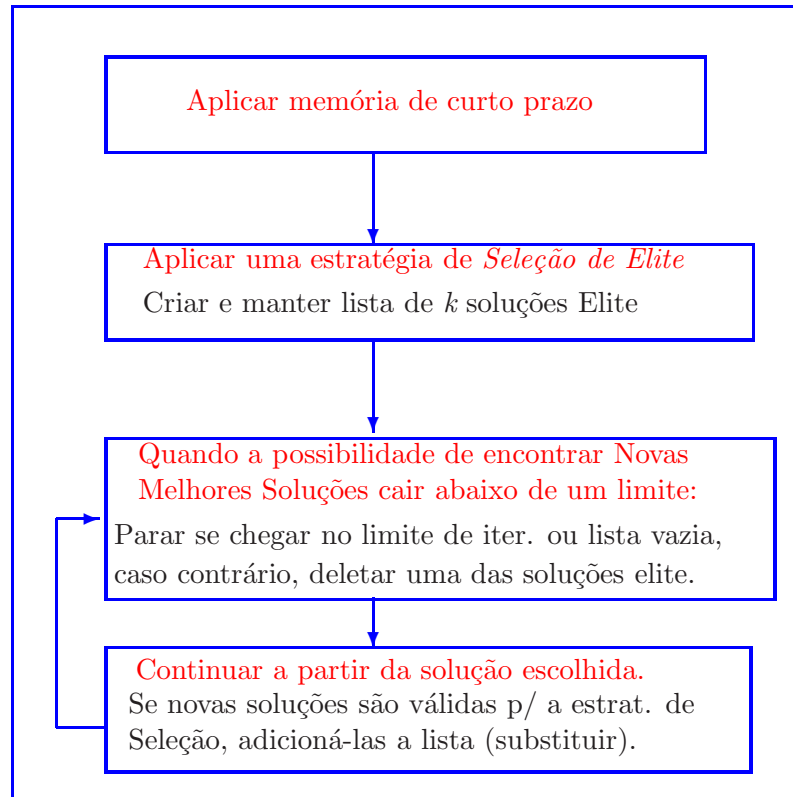


Figura 3.4: Enfoque simples de intensificação na Busca Tabu.

3.5.1.4 Estratégia de diversificação

As estratégias de *diversificação* são projetadas para conduzir as buscas para novas regiões. Estão baseadas, frequentemente, em modificar as regras de seleção para levar à solução atributos que não tenham sido utilizados com frequência. Podem também introduzir estes atributos ao reiniciar parcial ou completamente o processo de solução. A Figura 3.5, a seguir, dá um enfoque simples de diversificação que mantém uma memória baseada na frequência sobre todas as soluções geradas previamente.

Outras estratégias são propostas em diferentes algoritmos, para diferentes aplicações, bem como, existe uma variedade de composições de estratégias dentro do contexto da metaheurística Busca Tabu, tanto para considerações de curto quanto de longo prazo [Battiti, Tecchiolli - 1992], [Wooddruff, Zemel - 1993], [Osman - 1993], [Glass, et al. - 1994], [Battiti, Tecchiolli - 1994], [Lokketangen, Glover - 1995], [Osman - 1995], [Battiti, Tecchiolli - 1996], entre outras. (*Ver Bibliografia: Busca Tabu, página 196.*)

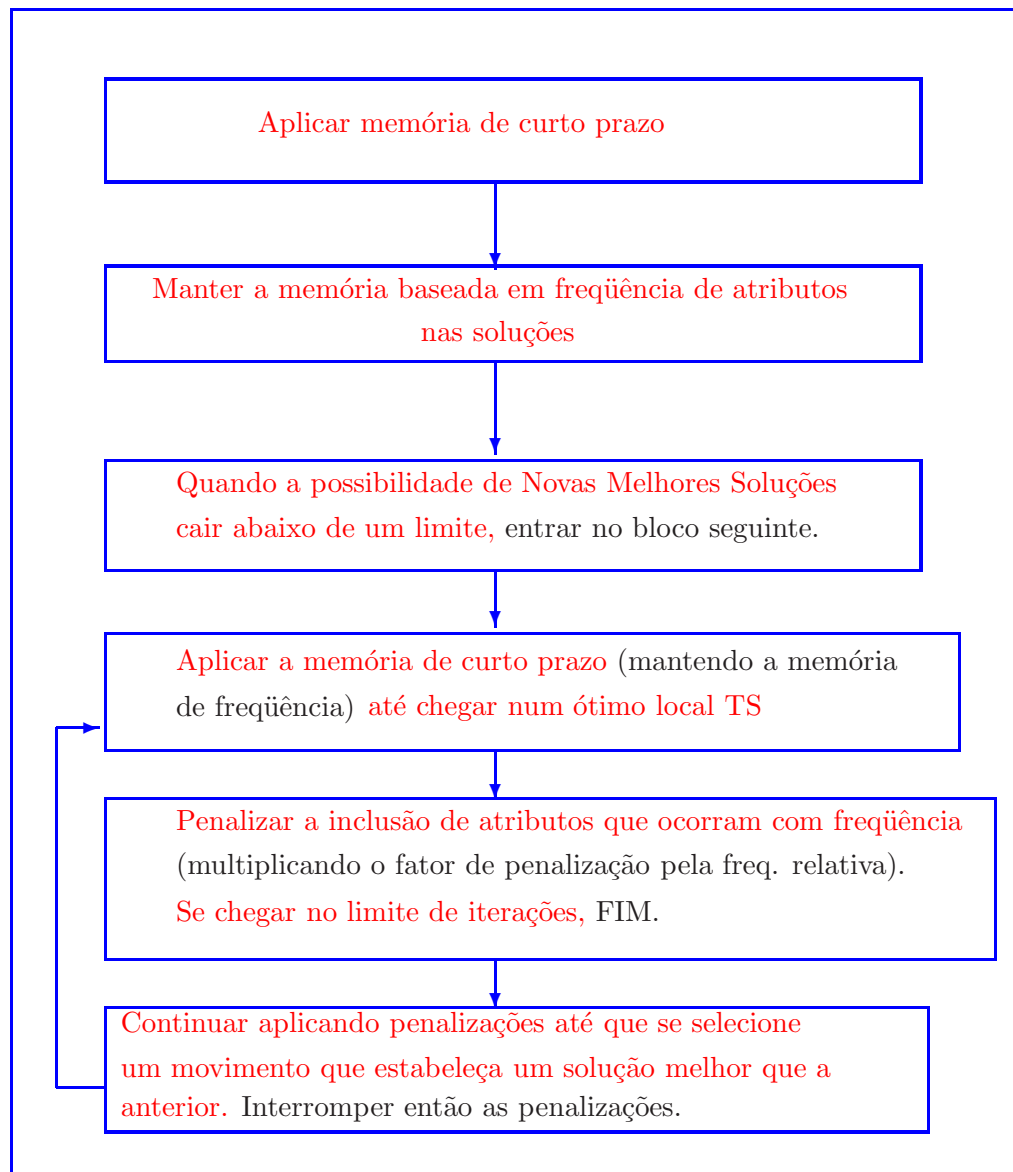


Figura 3.5: Enfoque simples de diversificação na Busca Tabu.

3.5.2 Aplicações de Busca Tabu

A Tabela 3.8, a seguir, mostra a diversidade das áreas de aplicação de Busca Tabu.

Tabela 3.8: Algumas áreas de aplicações da Busca Tabu.

Seqüenciamento “Flow-time Cell Manufacturing” - Processadores heterogêneos - Escalonamento - Planificação da força de trabalho - Horários escolares Desenho - CAD - Redes tolerantes a falhas - Desenhos de redes de transporte - Planificação do espaço em arquitetura - Diagrama de coerência - Desenho de redes de carga fixa - Problemas de corte irregular - Distribuição em plantas (“layout”) Localização “Multicommodity” - Atribuição quadrática - Semi-atribuição quadrática - Atribuição Generalizada Multi-nível Lógica e Inteligência Artificial - Satisfatibilidade máxima - Lógica Probabilística “Clustering” - Reconhecimento/Classificação de Padrões - Integridade dos dados - Treinamento de Redes Neurais - Desenho de redes neurais Tecnologia - Inversão sísmica - Distribuição de energia elétrica - Desenho estrutural de engenharia - Mínimo volume de elipsóides - Construção de estações espaciais - Colocação de células de circuitos - Exploração de petróleo “off-shore”	Telecomunicações - Trajetórias de camadas “Bandwidth Packing” - Atribuição de caminhos - Desenhos de redes para serviços - Planificação de descontos a clientes - Arquitetura imune a falhas - Redes óticas síncronas Produção, Inventários e Inversão - Fabricação flexível - JIT - MRP com restrições de capacidade - Seleção de componentes - Planificação de Inventários “Fixed Mix Investment” Trajetórias - Trajetórias de veículos “Time Window Routing” - Trajetórias multi-modal - Trajetórias com frota fixa - TSP “Traveling Purchaser Problem” Grafos - Partição de um grafo - Coloração de um grafo - Partição de “clicas” - Problemas de “clicas” máximas - Grafos planares máximos - Problemas de P-Mediana Otimização combinatória em geral - Programação 0-1 - Otimização de carga fixa - Programação Não-Linear Não-Convexa - Redes Tudo-Nada - Otimização Inteira Mista
---	---

3.6 Times Assíncronos

Um Time Assíncrono (“Asynchronous Team” - **A-Team**) é uma combinação das melhores características de sistema *natural* e *sintético*, incluindo: sociedades de insetos, comunidades celulares, algoritmos genéticos, “blackboards”, “simulated annealing” e busca tabu. É eficiente em resolver aqueles problemas para os quais estão disponíveis muitos algoritmos, mas nenhum inteiramente satisfatório. Um **A-Team** combina tais algoritmos de maneira que os mesmos cooperam para produzir resultados melhores e mais rápidos do que se cada um deles trabalhasse sozinho.

Segundo [Talukdar, De Souza - 1993], um **A-Team** é uma rede computacional *fortemente cíclica* composta por *agentes* e *memórias compartilhadas*. Os resultados circulam através dessa rede via um número de *agentes* (algoritmos) distribuídos que cooperam trabalhando um com os resultados dos outros, de maneira totalmente *autônoma*. Resultados que não estão sendo trabalhados são acumulados em *memórias compartilhadas* formando *populações*. Através da *aleatoriedade* (os efeitos de variação) e da *destruição* (eliminação de resultados indesejados) estas populações são atualizadas. Todos os agentes trabalham, todo o tempo, em paralelo.

Os Times Assíncronos têm inúmeras aplicações, como: problemas do caixeiro viajante, análise de estrutura de proteínas, planejamento de rotas de robôs, controle de tráfego de trens, controle de linha de produção, entre outras. Neste trabalho, analisa-se o comportamento do mesmo para o problema do planejamento da expansão da transmissão de energia elétrica.

3.6.1 Conceitos Básicos de Times Assíncronos

Para se tratar de *Times Assíncronos*, algumas definições iniciais são importantes.

Objetos - são entidades que englobam tanto *dados* como *procedimentos*.

Objetos de dados - quando a proposta principal do objeto é armazenar dados. Caso contrário, é um *agente*.

Fluxo de Dados é um hipergráfico direcionado. Representa a seqüência de transformações de objetos de um conjunto de objetos de dados (o conjunto do qual o agente lê), em outros (o conjunto para o qual o agente escreve).

Nós em um fluxo de dados são *diagramas de Venn* representando complexos de memórias sobrepostas, ou mais precisamente, os objetos que esta memória pode conter.

Arcos em um fluxo de dados representam agentes que lêem das memórias (rabo das flechas) e escrevem nas memórias (pontas das flechas).

Agentes Autônomos é quando um agente faz sua própria escolha sobre sua seleção de entrada, escalonamento e política de alocação de recursos.

Comunicações Assíncronas - agentes podem ler e escrever informações nas memórias compartilhadas sem qualquer sincronização entre elas. A autonomia dos agentes adicionada às comunicações assíncronas permitem que os agentes trabalhem em paralelo todo o tempo.

Fluxo de Dados Cíclico - agentes alimentam, modificam e armazenam informações continuamente nas memórias compartilhadas. Permitem que um fluxo contínuo de modificações possam ser realizados pelos agentes, de maneira a tornar possível iterações e “feedback” entre os mesmos.

Com base nestas definições, [Talukdar, De Souza - 1993] definem um **A-Team** como qualquer super-agente (metaheurística) cujos agentes (heurísticas) são autônomos, cujas comunicações são assíncronas e cujo fluxo de dados é cíclico. Em outras palavras, **A-Team** é uma seqüência de desenvolvimento de fluxos de dados fortemente cíclicos. São obtidos pela formação de operadores dentro de agentes autônomos e agentes dentro de fluxos de dados cíclicos. Cada **A-Team** é dedicado a um problema e cada fluxo de dados é um esquema distinto para resolver instâncias deste problema.

A Figura 3.6 dá uma idéia de um super-agente, onde os objetos de dados são representados por retângulos e agentes por flechas.

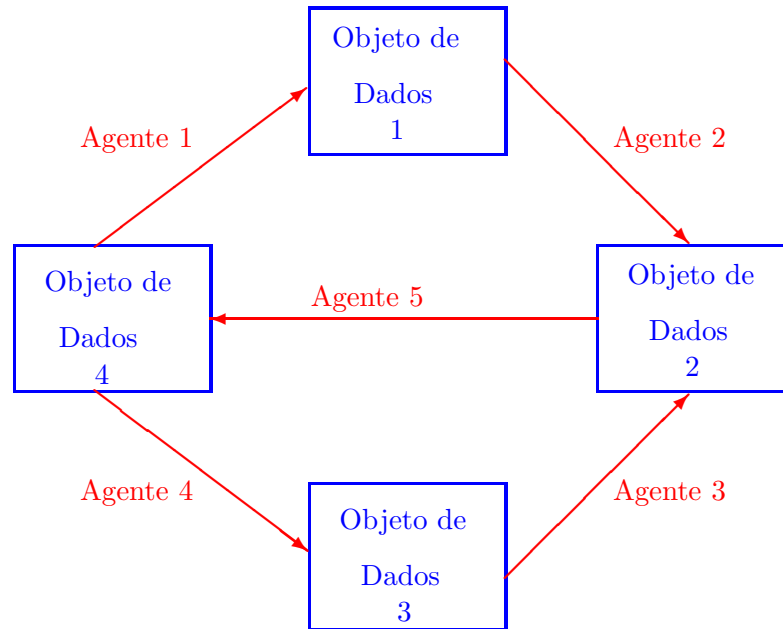


Figura 3.6: Um super-agente

Em um **A-Team** cada algoritmo tem a oportunidade de apresentar suas melhores soluções que, por sua vez, pode ser reutilizado como entrada por outro algoritmo no time. Um **A-Team** permite cooperação entre os algoritmos, aumentando a chance de gerar soluções melhores do que cada algoritmo isoladamente. Não existe a necessidade de processos gerentes para controlar como os algoritmos cooperam, ou seja, membros de um **A-Team** podem ser incluídos ou excluídos a qualquer momento, e quando executando os mesmos são independentes.

3.6.2 Descrição de Times Assíncronos

Pode-se resumir algumas observações importantes sobre **A-Teams**, conforme a seguir.

1. Resultados (soluções-teste) acumulam nas memórias do fluxo de dados (como nos “blackboards”, [Nii - 1986]) para formar populações (como nos algoritmos genéticos).
2. Estas populações são continuamente modificadas por dois tipos de agentes:
 - *Agentes de Construção* (que adiciona membros às populações); e
 - *Agentes de Destruição* (que elimina membros das populações).
 Este último trabalha a partir de listas de soluções a serem evitadas (como as listas usadas nas buscas tabu).
3. As habilidades de resolver problemas de um fluxo de dados pode ser arbitrariamente repartida entre construção e destruição.
4. Todos os agentes em cada fluxo de dados de um **A-Team** são autônomos. Um agente autônomo decide por si só o que é para ser feito e quando. Não existe um controle centralizado, mas novos agentes autônomos pode ser facilmente adicionados.
5. Os agentes cooperam trabalhando um sobre os resultados dos outros, de maneira assíncrona. Todos os agentes podem, se estabelecido previamente, trabalhar em paralelo todo o tempo.
6. Os fluxos de dados destes **A-Teams** não somente produzem boas soluções, mas eles parecem ser “scale-effective”, isto é, os mesmos podem ser montados para produzirem sempre as melhores soluções pela adição de certos agentes e memórias.

Para exemplificar a construção de um **A-Team**, formaliza-se a seguir um problema “NP_Hard” bastante conhecido: o “Traveling Salesman Problem - TSP”. Por suas características, não existem algoritmos de tempo polinomial que encontre rotas ótimas, mas existem muitos destes algoritmos que podem encontrar soluções próximas do ótimo.

3.6.2.1 O “Traveling Salesman Problem” - TSP

É dado n cidades e uma matrix $n \times n$, cujos elementos d_{pq} são a distância entre os pares p, q das n cidades. Uma *rota* é definida como um caminho fechado visitando uma cidade somente uma vez. O problema consiste em encontrar uma rota de comprimento mínimo.

Para este problema uma solução é dada por uma permutação cíclica $\pi = (\pi(1), \dots, \pi(n))$, onde $\pi(k)$ denota a cidade sucessora da cidade k , com $\pi^l(k) \neq k, l = 1, \dots, n - 1$ e $\pi^n(k) = k$, para todo k . Cada solução corresponde a uma única rota.

O espaço de solução é dado por:

$$\mathcal{S} = \{\text{todas as } \pi \text{ permutações cíclicas sobre as } n \text{ cidades}\}. \quad (3.12)$$

A função custo é definida como

$$f(\pi) = \sum_{i=1}^n d_{i, \pi(i)}, \quad (3.13)$$

isto é, $f(\pi)$ dá o comprimento da rota correspondente a π . Dessa forma, tem-se $|\mathcal{S}| = (n - 1)!$.

Na seqüência, apresenta-se de uma forma genérica a construção de um **A-Team** aplicado ao **TSP** com o objetivo de ilustrar e descrever seus parâmetros mais importantes.

3.6.2.2 Protocolo para construção de times assíncronos aplicados ao TSP

[Talukdar, et al. - 1996] apresentam um roteiro para a construção de **A-Teams** para o problema TSP (“Traveling Salesman Problem”), dado a seguir.

1. Escolha um problema

O TSP é um problema de otimização combinatória;

Forma Básica: Dado n cidades e suas distâncias, encontrar a rota mais curta passando por todas elas.

2. Decomponha o problema em subproblemas

A decomposição não necessita ser hierárquica;

Os subproblemas não precisam ser disjuntos, nem sempre distintos; eles podem ter pequenos “overlaps” com o problema original e pode ser mais complexo que ele;

Subproblemas devem ser escolhidos de maneira que eles possam ser conectados por agentes em “loops” fechados, assim as soluções de cada subproblema podem ser usadas para construir soluções para o próximo subproblema no “loop”;

Exemplos de escolha são os quatro subproblemas:

1. encontrar uma rota ótima;
2. encontrar rotas boas;
3. encontrar rotas parciais boas;
4. encontrar “1-trees” boas.

“boa” significa: contendo muito dos arcos de uma rota ótima;

“1-tree”: conecta todas as cidades mas nem sempre forma uma rota (as “1-trees” formam um superconjunto das rotas).

3. Atribua uma ou mais memórias para cada subproblema

A proposta de cada memória é manter uma população de soluções-teste para seu subproblema;

Populações moderadas trabalham tão bem quanto grandes populações;

Mesmo populações moderadas podem ocupar bastante espaço de armazenagem, especialmente quando cada solução-teste é um objeto complexo, tais como um “building-design”. Em tais casos, a representação deverá ser um compromisso cuidadosamente escolhido entre “compactness” e “clarity”;

Para o TSP, foi escolhido n como o tamanho da população e listas ordenadas de cidades como a representação para as rotas completa e parcial.

4. Selecione um conjunto de algoritmos ou operadores para cada subproblema

Quanto maior a faixa de especialidades destes algoritmos, melhor serão as soluções encontradas;

Os algoritmos não precisam ser uniformes em tamanho e desempenho: uns podem ser grandes, outros pequenos, uns gerais, outros especializados;

Deve-se usar os melhores e mais poderosos algoritmos disponíveis;

Algoritmos fracos, tais como “crossover” e “mutation”, quando usados exclusivamente, ou algoritmos poderosos, tais como um algoritmo “branch-and-bound” estado da arte, quando usado sozinho, ou qualquer mistura de baixa variedade, invariavelmente fará menos bem do que uma mistura de grande variedade e faixa.

A Tabela 3.9, apresenta exemplos de algoritmos utilizados para resolver o TSP.

Tabela 3.9: Exemplos de algoritmos para o TSP.

LK	Lin-Kernighan, um dos maiores e mais poderosos algoritmos disponíveis. [Lin, Kernighan - 1993]
CLK	Uma versão mais curta e mais simples do LK. [De Souza - 1993]
OR	Or-Opt, um algoritmo moderadamente complicado. [Lawler, et al. - 1985a]
AI	“Arbitrary Insertion”, um algoritmo muito curto e simples. [Lawler, et al. - 1985a]
HK	Held-Karp, um algoritmo para converter rotas em “1-Trees”. [Held, Karp - 1970]
Dec	Um decompositor que produz uma rota parcial de caminhos comuns de duas rotas completas. [De Souza - 1993]
MI	um alg. de mistura que combina duas rotas em uma. [Whitley - 1989]
TM	um algoritmo de mistura que combina uma rota com uma “1-tree” para obter uma nova rota. [De Souza - 1993]

Ver referências bibliográficas em *Bibliografia: Gerais e Times Assíncronos*, páginas 159 e 204.

5. Forme cada algoritmo dentro de um agente autônomo

Agente autônomo: um agente é autônomo se seu sistema de controle é completamente auto-contido, isto é, se ele não aceita instruções de seleção ou escalonamento de outros agentes;

Componentes de um agente:

um *operador*;

um *sistema de comunicação* (conecta o agente às suas memórias de entrada e saída designadas);

um *sistema de controle* (consiste de seletores e escalonadores).

O sistema de comunicação escolhe objetos da memória de entrada para o operador trabalhar, e o sistema de controle determina quando o operador trabalhará e quais dos computadores disponíveis ele usará;

A chave para a cooperação efetiva entre agentes autônomos está no projeto de suas estratégias.

6. Forme destruidores autônomos

Os agentes podem ser:

construtores (que adicionam soluções-teste a suas memórias de saída);

destruidores (que apagam soluções-teste de suas memórias de saída).

7. Forme os agentes e memórias dentro de um fluxo de dados fortemente cíclico.

8. Teste e modifique o fluxo de dados implementando-o numa rede de computadores.

[Talukdar, De Souza - 1993] propõem, entre outras, a configuração mostrada na Figura 3.7 de um **A-Team** para resolver o problema do caixeiro viajante.

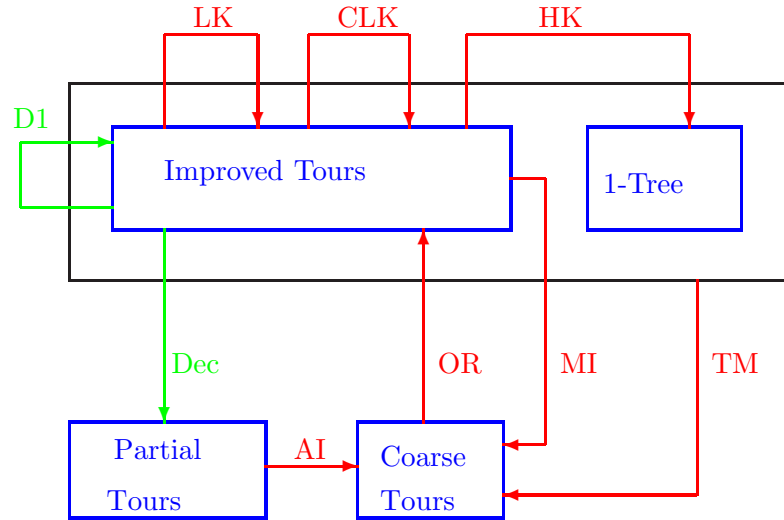


Figura 3.7: Um time assíncrono para resolver o problema do caixeiro viajante.

Na Figura 3.7, tem-se o que se segue:

O espaço de objetos de dados “Improved Tours” mantém as rotas, em ordem de comprimento, providas pelos agentes que escrevem nele. **LK** e **CLK** lêem e escrevem neste espaço, sendo que **LK** sempre toma a melhor rota ainda não checada. Todos os outros agentes selecionam rotas com distribuição uniforme aleatória sobre as rotas disponíveis.

HK usa o comprimento da melhor rota em “Improved Tours” como um limite superior para gerar as “1-trees” que são armazenadas no espaço de dados “1-Tree”.

TM toma uma rota de “Improved Tours” e a última versão “1-tree” do espaço “1-Tree”, armazenando sua descendência dentro de “Coarse Tours”. “Coarse Tours” e “Partial Tours” mantém seus dados no modo “FIFO - First-In-First-Out” (o primeiro que entra é o primeiro que sai).

MI escolhe duas rotas de “Improved Tours” e alimenta “Coarse Tours” com as mesmas.

O agente destruidor **Dec** quebra rotas completas em rotas parciais, tomando duas rotas de “Improved Tours” e mantendo somente os elementos que pertencem a ambas. Daí, conecta estes elementos criando uma rota parcial fechada e armazena-a em “Partial Tours”.

AI lê de “Partial Tours” e incorpora as cidades omitidas à rota parcial lida, armazenando a nova rota possível em “Course Tours”.

OR melhora as rotas de “Course Tours” e escreve seus resultados em “Improved Tours”.

D1 é um destruidor que mantém somente um número arbitrário de rotas em “Improved Tours” (normalmente o tamanho do problema que está sendo resolvido). Ele descarta imediatamente qualquer rota que chega que tem comprimento maior do que o comprimento da pior rota em “Improved Tours”. Caso contrário, ele seleciona uma rota de “Improved Tours” com uma distribuição de probabilidade linear e a substitui pela rota que chega.

Todos os algoritmos continuam fazendo suas funções de maneira assíncrona, em “loops” permanentes.

É importante observar que após a fase de inicialização, cada algoritmo roda em seu próprio ritmo usando os dados disponíveis nos objetos de dados. Isto mostra o caráter assíncrono do **A-Team** que pode facilmente ser executado em paralelo; cada agente e objeto de dados pode ser alocado para diferentes máquinas em uma rede de computadores.

A seguir, são enumeradas observações sobre o comportamento do **A-Team** construído segundo o protocolo descrito anteriormente.

3.6.2.3 Comportamento observado

1. *Diversidade* - aumento da qualidade da solução com a faixa de habilidades dos agentes de construção;
2. *Escala* - Existe pouca, se existir, penalidade para um excesso de habilidades de construção. “Scale-effectiveness” parece ser comum; a qualidade da solução pode invariavelmente ser aperfeiçoada pela adição de agentes construtores e destruidores;
3. *Expansão* - adicionar agentes autônomos aos fluxos de dados fortemente cíclicos é relativamente fácil, sem considerar as conseqüências de se os agentes são grandes ou pequenos, gerais ou especializados;
4. *Dualidade* - destruição apta pode compensar construção inapta, e vice-versa;
5. *Tamanho da população* - a qualidade da solução se beneficia do aumento dos tamanhos das populações de solução, mas estes benefícios estão propensos à saturação;
6. *Paralelismo* - a velocidade da solução aumenta quando computadores são adicionados até que existam bastante computadores para todos os agentes trabalharem em paralelo todo o tempo. Frequentemente, o aumento de velocidade é não-linear.

Como visto, a construção adequada de um **A-Team**, combinando diferentes algoritmos que agem de maneira assíncrona e iterativa, pode ajudar a resolver instâncias de grande porte de problemas combinatórios complexos. Diversos trabalhos têm sido apresentados em anos recentes utilizando

esta metaheurística ([Barán, et al. - 1993], [Chen S., *et. al.* - 1993], [Ramesh - 1994], [Talukdar, Ramesh - 1994], [Cavalcante, De Souza - 1995], [Rodrigues, De Souza - 1995], [Kao, et al. - 1995], [Rodrigues, De Souza - 1996], [Avila-Abascal, Talukdar - 1996], entre outros).

3.6.3 Aplicações de Times Assíncronos

A Tabela 3.10 a seguir, apresenta algumas áreas de aplicações de **A-Teams**.

Tabela 3.10: Algumas áreas de aplicações de times assíncronos

Aplicações de A-Teams
resolução de equações não-lineares problemas do caixeiro-viajante “high-rise building design” projeto reconfigurável de robôs diagnóstico de faltas em redes elétricas controle de redes elétricas escalonamento “job-shop” escalonamento de siderúrgicas escalonamento de fábricas de papel escalonamento de trens satisfação de restrição problemas de funções multi-objetivos

Neste capítulo foram estabelecidos, portanto, os conceitos fundamentais que envolvem as metaheurísticas combinatórias e a metaheurística times assíncronos, que compõem o núcleo dos algoritmos implementados nesta tese. Na sequência, serão retomados os conceitos relativos aos algoritmos heurísticos construtivos, (seção 2.3.1, página 8) juntamente como os conceitos de times assíncronos acima mencionados, para a descrição de um time assíncrono inicializador utilizado na inicialização do problema do planejamento da expansão da transmissão de energia elétrica, conforme estabelecido na seção 2.4, página 13.

Capítulo 4

O Time Assíncrono Inicializador

4.1 Introdução

Para o problema do planejamento da expansão da transmissão, especialmente para sistemas de grande porte, o desempenho dos algoritmos combinatórios é muito dependente das configurações iniciais utilizadas. Ou seja, uma população com configurações altamente diversificadas faz com que os algoritmos combinatórios iniciem sua busca em diferentes regiões do espaço de solução do problema.

Para se obter uma diversidade das configurações, é importante que as mesmas sejam obtidas através de diferentes metodologias ou heurísticas, pois assim as características peculiares à cada método estariam presentes na população inicial.

Como parte inicial desta tese, procurou-se atingir este objetivo com a proposição de um algoritmo inicializador para as metaheurísticas combinatórias, através de um **A-Team** de algoritmos heurísticos construtivos ([De Oliveira, et al. - 1998] e [De Oliveira, *et al.* - 1999]) para o problema do planejamento da expansão da transmissão. (Ver artigos no Apêndice C, página 241)

Nas seções que se seguem são detalhados os agentes, bem como o funcionamento do time assíncrono inicializador como um todo; complementado pelo Apêndice B, página 235, com informações de como executá-lo na prática.

4.2 Time Assíncrono de Algoritmos Heurísticos Construtivos

Com base nos algoritmos heurísticos construtivos apresentados no Capítulo 2, seção 2.3.1, página 8, e nos conceitos de times assíncronos apresentados na Capítulo 3, seção 3.6, página 37, construiu-se diferentes versões de algoritmos (doravante designados *agentes*) que em conjunto compõem um **A-Team** inicializador. A função deste último é obter configurações iniciais factíveis que serão posteriormente utilizadas pelas metaheurísticas combinatórias na resolução do problema do planejamento da expansão da transmissão.

Para a metodologia de Garver foram construídos três agentes: *GarverA*, *GarverB* e *GarverC*; para o método de mínimo esforço, dois agentes: *minD* e *minE* e finalmente, para o método de mínimo corte de carga, um agente: *mccF*.

A seguir são apresentadas as metodologias utilizadas e os respectivos agentes.

4.2.1 Algoritmo Heurístico de Garver

O modelo formulado por [Garver - 1970], também conhecido como *modelo de transporte* foi a primeira proposta para planejamento de redes de transmissão que usou programação linear. Esta metodologia consiste basicamente em resolver de maneira aproximada uma versão relaxada do modelo DC. Neste modelo somente se leva em conta a Primeira Lei de Kirchhoff e a capacidade de transmissão das linhas, não sendo levado em conta, portanto, a Segunda Lei de Kirchhoff.

Na metodologia de Garver, todo fluxo que não puder ser transportado pelas ligações normais, fluirão pelas ligações de sobrecarga, pois estas têm capacidades ilimitadas, e só passarão através das ligações de sobrecarga quando for impossível transportá-los pelas ligações normais, já que estas tem custos muito inferiores. Em cada estágio do processo de planejamento, deve-se resolver um problema de programação linear e logo adicionar um circuito na trajetória de maior sobrecarga. O processo é repetido até eliminar todas as sobrecargas.

A vantagem da metodologia de Garver é a simplicidade na implementação do algoritmo pois ela exige somente soluções sucessivas de programação linear. A maior limitação da metodologia é que ela não garante a obtenção da solução ótima do sistema planejado. Portanto, esta metodologia, em essência, é de natureza heurística.

A metodologia acima exposta, foi usada em três versões de agentes: *GarverA*, *GarverB* e *GarverC*. E seus respectivos algoritmos são descritos a seguir:

4.2.1.1 O algoritmo do agente GarverA

O agente *GarverA*, apresenta o seguinte algoritmo:

- (1) Faz-se a análise DC (MINOS - [Murtagh, Saunders - 1987]) do sistema atual (o banco de dados da rede já considera a presença das barras e linhas fictícias).
- (2) Com os resultados de (1), calcula-se o fluxo para cada ramo do sistema.
- (3) Ordena-se os ramos pelo valor dos respectivos fluxos, em ordem decrescente.
- (4) Se o fluxo do ramo mais sobrecarregado for menor que um limite preestabelecido no início do programa, então vá para (6). Caso contrário, o programa oferece duas opções de adição de linhas, dependendo do sistema em questão. Na primeira, pode-se simplesmente adicionar uma linha ao ramo mais sobrecarregado. Na segunda, é feita uma identificação da necessidade da criação de um caminho completo ou somente um ramo, a partir da adição do ramo mais sobrecarregado, a fim de evitar a presença de elementos desconexos no sistema.

- (5) Volte para (1).
- (6) Efetua-se o *refinamento* do sistema, que consiste na verificação da possibilidade de se remover linhas adicionadas desnecessariamente, e PARE.

4.2.1.2 O algoritmo do agente GarverB

O algoritmo *GarverB* é semelhante ao *GarverA*, com a diferença que agora a ordenação é realizada baseada no número de linhas requeridas em cada ramo e não no fluxo. Da mesma forma, o critério de parada considera agora uma parcela mínima de linhas permitida.

4.2.1.3 O algoritmo do agente GarverC

O algoritmo *GarverC* apresenta a mesma estrutura básica que o *GarverB*, exceto pela introdução das seguintes correções:

- correção de caminhos conectando linhas isoladas eletricamente;
- correção de caminhos entre barras fictícias, para que a capacidade de transmissão das linhas em série seja suficiente;
- adição de linhas cuja fração seja superior a um valor de entrada.

4.2.2 Método de Mínimo Esforço

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

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

em que:

$\theta_i - \theta_j$ é a diferença angular do ramo ij antes da adição,

$\Delta \gamma_i$ é a variação da susceptância do circuito adicionado.

Esta metodologia foi usada em duas versões de agentes: *minD* e *minE*, cujos algoritmos são descritos a seguir.

4.2.2.1 O algoritmo do agente minD

O agente *minD* possui algoritmo semelhante aos anteriores com diferenças, relacionadas ao novo critério, como segue:

- (1) Faz-se a análise DC do sistema (via MINOS), para a rede atual.
- (2) Calcula-se o somatório das sobrecargas do sistema e o índice de mínimo esforço para cada ramo do sistema.
- (3) Ordena-se os ramos pelo valor dos respectivos índices de mínimo esforço.
- (4) Se o somatório das sobrecargas do sistema for menor que um limite preestabelecido no início do programa, então vá para (6). Caso contrário, escolhe-se aleatoriamente um dos três primeiros ramos da lista ordenada para a adição de uma linha ou de um caminho, com as mesmas considerações anteriores.
- (5) Volte para (1).
- (6) Efetua-se o refinamento do sistema e PARE.

4.2.2.2 O algoritmo do agente minE

A única diferença deste agente em relação ao *minD* consiste na introdução de uma normalização pelo custo de cada ramo, quando do cálculo do índice de mínimo esforço.

4.2.3 Método de Mínimo Corte de Carga

De maneira semelhante ao Método de Mínimo Esforço, este método, [Pereira, Pinto - 1985], realiza a adição de linhas selecionadas de acordo com um índice de sensibilidade que permite encontrar linhas mais atrativas. O índice utilizado é:

$$IS_{mcc} = (\pi_i - \pi_j)(\theta_j - \theta_i) \quad (4.2)$$

em que:

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

θ_j é o ângulo de tensão de barra; obtidos na análise DC do sistema para a configuração atual.

Com este método elaborou-se uma única versão de agente: *mccF*, cujo algoritmo é exposto na seqüência.

4.2.3.1 O algoritmo do agente *mccF*

O agente *mccF* também mantém a mesma estrutura dos anteriores com diferenças, associadas ao novo índice, como segue:

- (1) Faz-se a análise DC (via MINOS) do sistema, para a rede atual.
- (2) Calcula-se o corte de carga do sistema (via MINOS), e o índice de mínimo corte de carga, já normalizado pelo custo, para cada ramo do sistema.
- (3) Ordena-se os ramos pelo valor dos respectivos índices de corte de carga.
- (4) Se o corte de carga do sistema for menor que um limite preestabelecido no início do programa, então vá para (6). Caso contrário, escolhe-se aleatoriamente um dos três primeiros ramos da lista ordenada para a adição de uma linha ou de um caminho, com as mesmas considerações anteriores.
- (5) Volte para 1.
- (6) Efetua-se o refinamento do sistema e PARE.

Uma vez detalhados os agentes, é descrito a seguir o funcionamento do time assíncrono inicializador.

4.3 Time Assíncrono Inicializador proposto

Em [De Oliveira, et al. - 1998] e [De Oliveira, *et al.* - 1999] é proposto um modelo de **A-Team** para servir como inicializador dos algoritmos combinatórios utilizados para a resolução do problema do planejamento da expansão da transmissão. Denominado INIC, o mesmo é composto por agentes baseados nos algoritmos heurísticos construtivos e tem sua estrutura apresentada a seguir.

4.3.1 Algoritmo INIC

A estrutura do **A-Team** implementada é composta pelos seguintes objetos:

- *memória central* - memória compartilhada pelos agentes, a qual é formada por uma população de configurações. O tamanho desta população é constante e determinado previamente pelo usuário.
- *agentes de construção* - são os diversos algoritmos heurísticos construtivos implementados (*GarverA*, *GarverB*, etc.)
- *agente destruidor* - mantém constante o tamanho da população.

Um exemplo de **A-Team**, dentro da estrutura elaborada, é mostrado na Figura 4.1. Os agentes são representados por setas que indicam um fluxo cíclico em relação à memória central, ou seja, os mesmos lêem os dados da memória, alteram esses dados e os reescrevem na mesma. A memória central, representada por um retângulo, contém a população de configurações. Os algoritmos (ou seja, os agentes) implementados na elaboração do time assíncrono são representados por linhas tracejadas, definindo processos distintos. Daí, observa-se que o agente destruidor e a memória central compõem o processo *inic*. Note que o agente destruidor não é um processo independente como os outros agentes, mas sim uma das funções do processo *inic* que age para manter constante o tamanho da população.

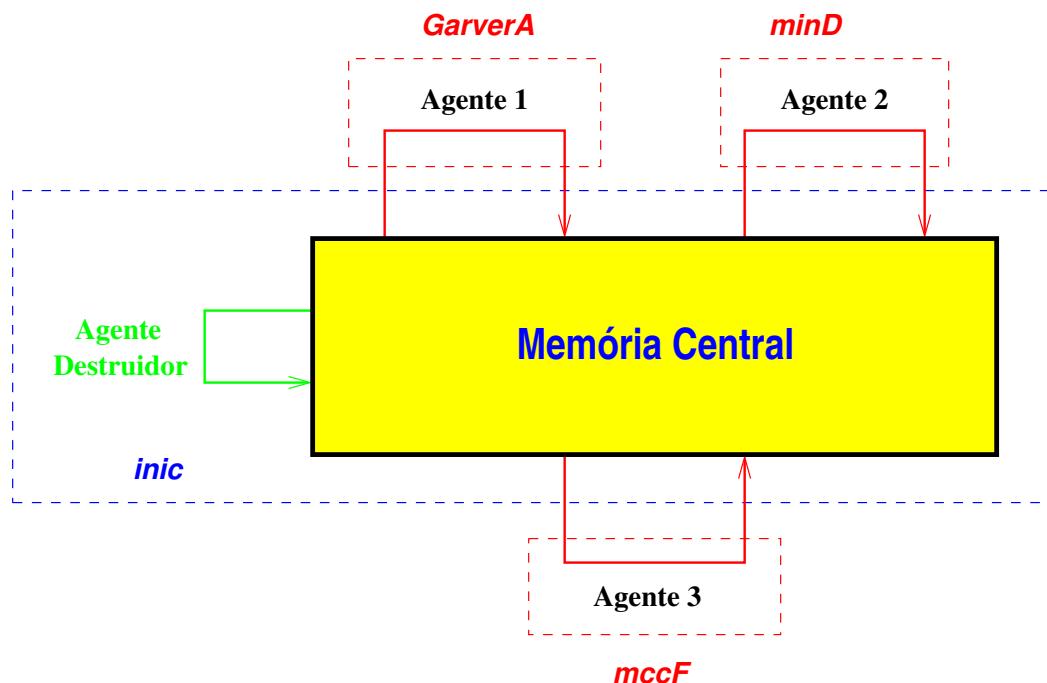


Figura 4.1: Exemplo de time assíncrono inicializador com três agentes.

A implementação paralela usa o modelo *Mestre/Escravo*, pois o *inic* sempre funciona como partida do processamento do time, adicionando, a seguir, um grupo de agentes predeterminados pelo usuário. Porém, é possível a adição/remoção dos agentes de forma manual, o que reforça a característica de autonomia dos mesmos (agentes entram ou saem do time sem interferir no funcionamento uns dos outros). A remoção manual dos agentes é feita por um programa auxiliar, enquanto que a adição dos mesmos é feita executando-os diretamente no “prompt” do sistema operacional.

O programa *inic* simula uma memória compartilhada utilizando semáforos para evitar conflitos ao se tentar acessá-la. Dessa forma, dois agentes não podem ler ou escrever ao mesmo tempo. Sendo assim, quando um agente estiver acessando a memória central, os outros necessariamente esperam numa fila do tipo FIFO (“First-In-First-Out”). Outra característica importante é o fato

de que, a cada instante do processamento do **A-Team**, cada configuração (elemento da população) é atualizada somente por um agente, isto é, dois agentes não podem trabalhar sobre os mesmos dados simultaneamente. Dessa forma, o número de membros do time nunca deve exceder o número de configurações da população. Se assim o for, o *inic* automaticamente se encarrega de eliminar gradativamente os agentes excedentes. A Figura 4.2 dá uma idéia geral do funcionamento do processo *inic*.

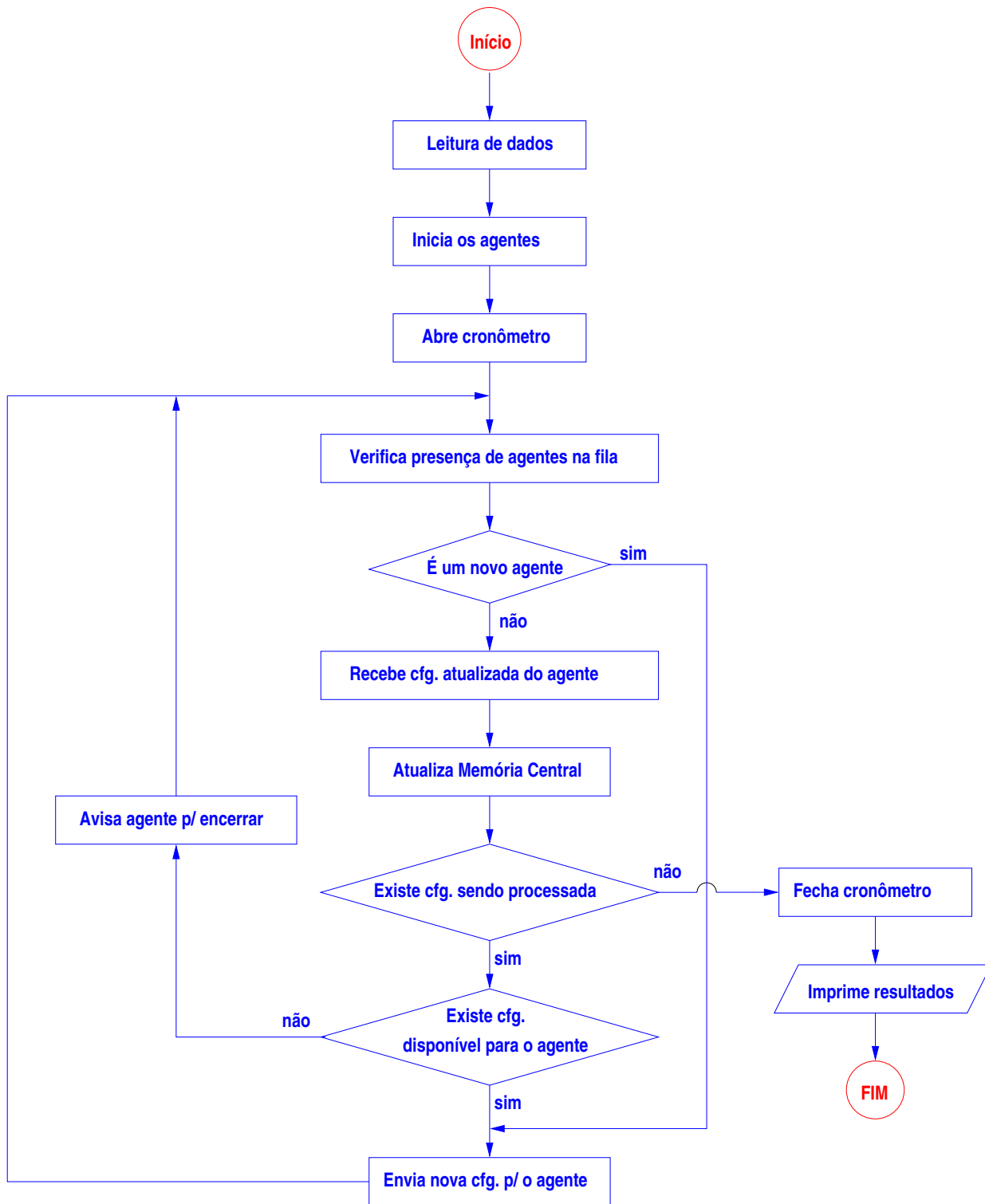
Quanto aos agentes de construção, de uma maneira geral, servem para efetuar os cálculos sobre os dados da memória central, segundo uma heurística própria, atualizando-os até satisfazer um determinado critério de parada. A Figura 4.3 apresenta a estrutura de funcionamento dos agentes em geral. Eles trabalham sinergicamente para obter configurações iniciais diversificadas em menor tempo, não visando necessariamente a obtenção de um ótimo global, função esta do problema do planejamento como um todo. Os agentes podem ser combinados em quantidades diferentes entre as máquinas da rede.

4.4 O uso do Time Assíncrono Inicializador com as Metaheurísticas Combinatórias

A utilização do **A-Team** inicializador antecede a execução de quaisquer das metaheurísticas combinatórias, ou seja, de maneira independente, faz-se uso do mesmo para se obter uma população de configurações a mais diversificada possível.

Na prática, executa-se o **A-Team** escolhendo previamente os agentes e estabelecendo-se a quantidade de configurações factíveis que deve ser gerada. O programa gera um arquivo de saída *mpop.dat* (contendo as configurações) que é utilizado posteriormente para gerar um outro arquivo (*mbloco.dat*), que obtém para cada ramo o número do bloco correspondente. Estes dois arquivos então são transferidos para o diretório de trabalho da metaheurística que se deseja processar.

Exemplo completo de como compilar, configurar e executar um **A-Team** com agentes de algoritmos heurísticos construtivos, é apresentado no Apêndice B.

Figura 4.2: Estrutura básica do programa *inic*.

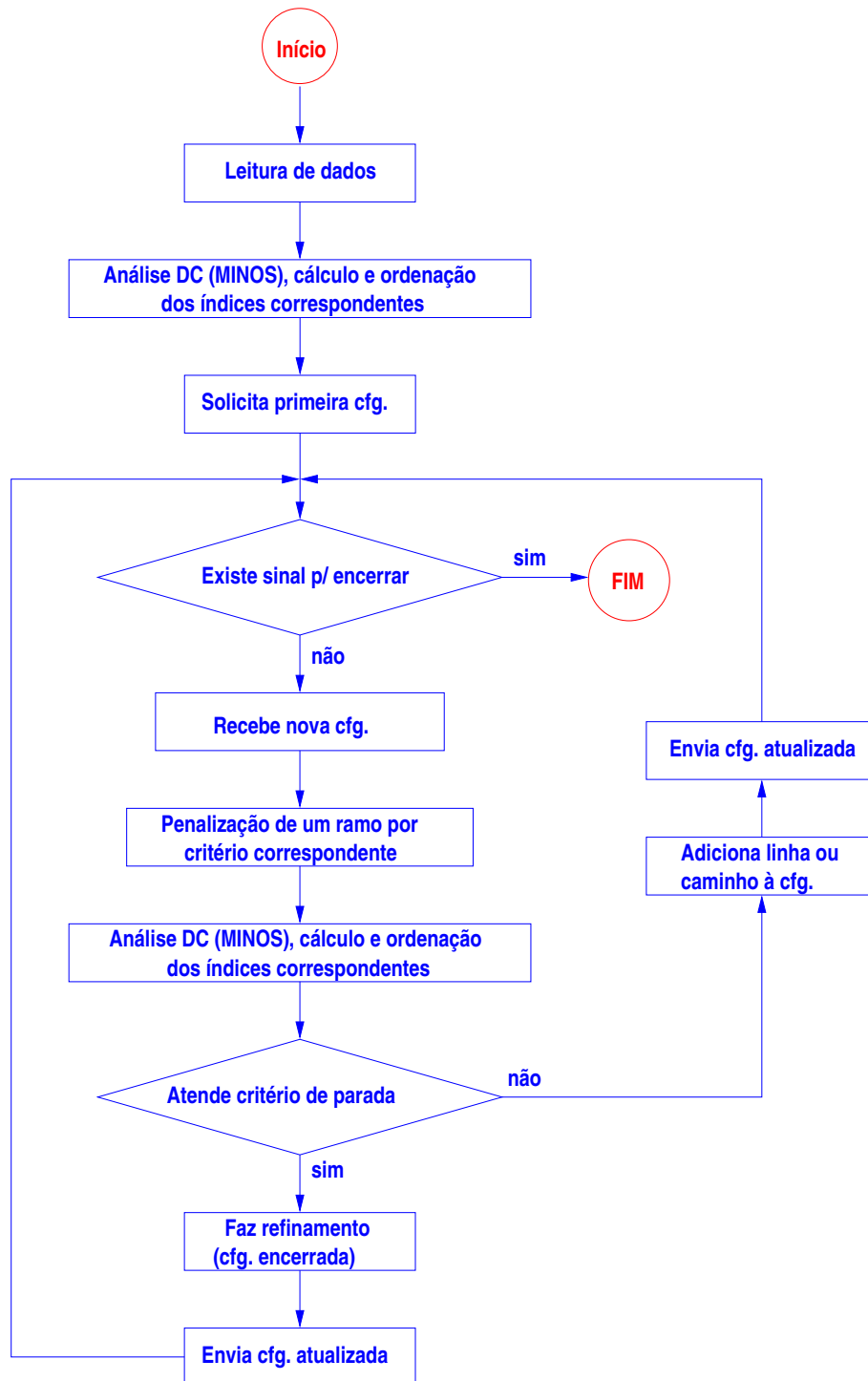


Figura 4.3: Estrutura básica dos agentes de construção.

Capítulo 5

Metaheurísticas em Ambiente de Processamento Distribuído

5.1 Introdução

Dando seqüência à descrição dos algoritmos combinatórios apresentados no Capítulo 3 e do Time Assíncrono Inicializador (Capítulo 4), neste capítulo são apresentados os algoritmos básicos para a montagem de um Time Assíncrono de metaheurísticas para a resolução do problema do planejamento estático da expansão da transmissão. Inicialmente é apresentado o ambiente de processamento distribuído utilizado para as simulações de todos os algoritmos implementados. Em seguida, são apresentadas as versões seqüenciais das metaheurísticas “simulated annealing” e algoritmos genéticos, cujos resultados serão usados para comparação com os resultados das versões paralelas propostas, que também são detalhadas neste capítulo.

5.2 O Ambiente de Processamento Distribuído

Em todas as simulações apresentadas neste trabalho, utilizou-se a rede de estações de trabalho do LSEE - Laboratório de Sistemas de Energia Elétrica do Departamento de Sistemas de Energia Elétrica da Faculdade de Engenharia Elétrica e de Computação da UNICAMP, que é constituída basicamente por estações SUN Sparc Ultra 1, conforme mostrado na Figura 5.1 a seguir.

Neste ambiente computacional, utilizou-se a biblioteca para processamento paralelo PVM ([Geist, *et al.* - 1994]), para o controle de processos e troca de mensagens entre os mesmos, quando do uso dos algoritmos das metaheurísticas paralelas desenvolvidas.

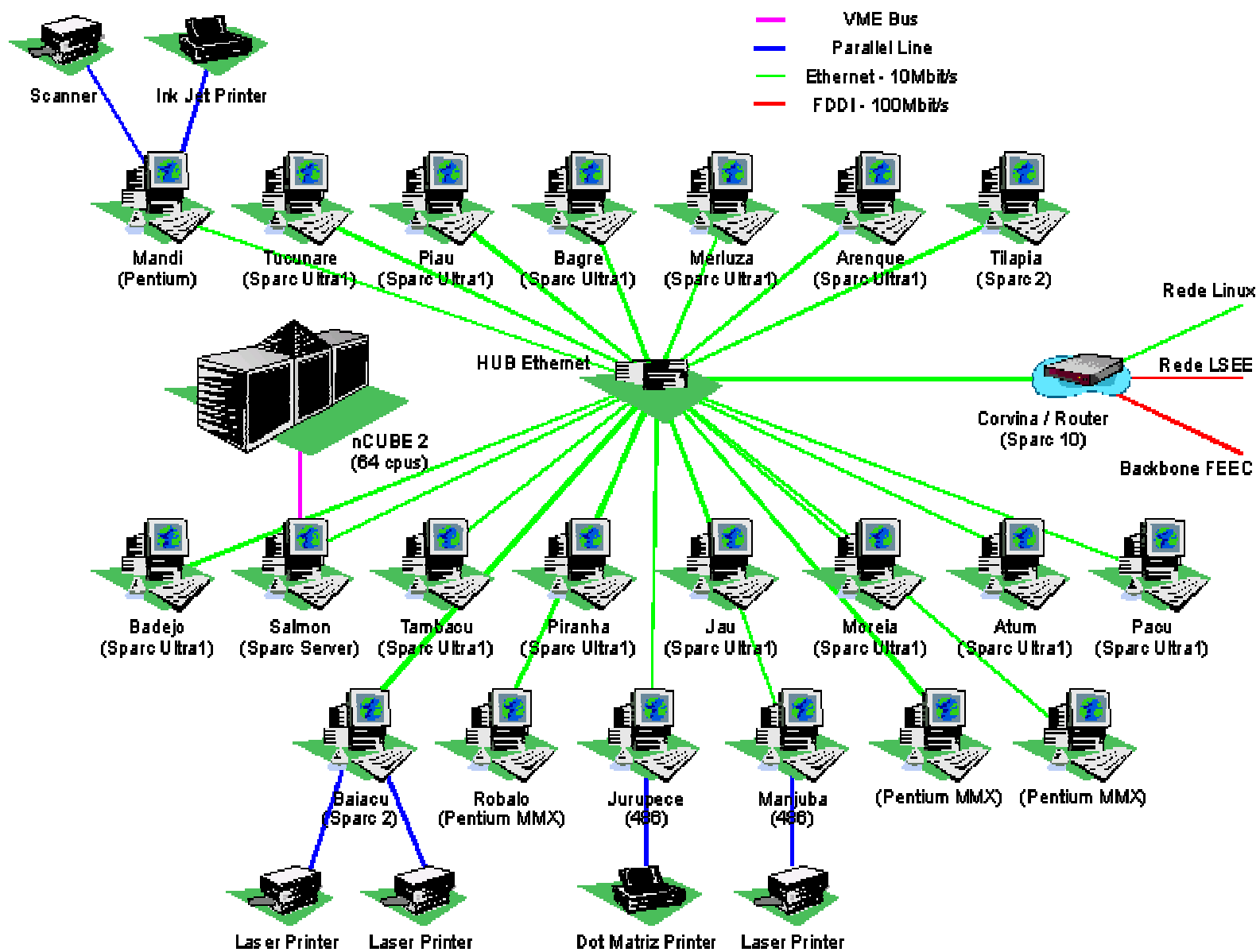


Figura 5.1: O ambiente de processamento distribuído utilizado nas simulações.

5.3 As Metaheurísticas Sequenciais Utilizadas

5.3.1 O algoritmo SANORD

Neste trabalho utilizou-se como referência, para um algoritmo serial de “simulated annealing”, o algoritmo SANORD ([Gallego, De Oliveira - 1997a] detalhado em [Gallego - 1997]), com alterações efetuadas no código fonte do mesmo para permitir traçar os gráficos que são apresentados no Capítulo 6. Estes gráficos permitirão uma análise comparativa com os resultados das metaheurísticas propostas para o ambiente de processamento distribuído.

Na sequência, apresenta-se de forma resumida os principais aspectos do algoritmo utilizado.

5.3.1.1 Estruturas básicas

De acordo com os conceitos apresentados no Capítulo 3, Seção 3.4, páginas 25 a 28, no algoritmo SANORD é utilizado o que segue:

critério de aceitação:

$$Pa = \begin{cases} 1 & \text{se } \Delta v < 0 \\ \exp(-\frac{\Delta v}{T}) & \text{se } \Delta v \geq 0 \end{cases} \quad (5.1)$$

em que: T é o valor do parâmetro de controle temperatura e Δv é a diferença da função custo entre a configuração candidata e a configuração corrente.

programa de resfriamento:

1. Expressão para o cálculo da *temperatura inicial* T_0 :

$$T_0 = \frac{\Delta V^+}{\ln(\frac{m_2}{m_2 \cdot X - m_1(1-X)})} \quad (5.2)$$

em que:

X é a taxa de aceitação;

m_1 é o número de transições propostas de i a j , para as quais $F(j) \leq F(i)$;

m_2 é o número de transições propostas, para as quais $F(j) > F(i)$;

ΔV^+ é o incremento médio no custo das m_2 transições, como segue:

$$\Delta V^+ = \frac{\sum_{i=1}^{m_2} \Delta F_i}{m_2} \quad (5.3)$$

$F(i)$ é o valor da função objetivo na alternativa i ;

2. Expressão para o *número de tentativas* a cada nível de temperatura:

$$N_{k+1} = \rho N_k \quad (5.4)$$

em que ρ (*cadeia*) é o parâmetro que controla o comprimento da cadeia de Markov.

3. Expressão para a *taxa de diminuição de temperatura*:

$$T_{k+1} = \beta T_k \quad (5.5)$$

em que β (*temperat*) é a taxa de resfriamento.

critério de parada:

- o processo pára se o número de iterações for maior do que *iterm* (número máximo de iterações), ou
- se durante o último nível de temperatura não forem aceitas até *limcong* novas propostas.

5.3.1.2 Descrição do algoritmo

O algoritmo SANORD é descrito como segue:

1. Determinar a temperatura inicial: T_o . (Na prática, esta informação é passada ao programa pela variável *temp*, calculada previamente de acordo com o sistema teste em análise.)
2. Escolher ou calcular os seguintes parâmetros:
 - β (*temperat*) taxa de resfriamento;
 - ρ (*cadeia*) controla o comprimento da cadeia de Markov;
 - μ (*nut*) para definir o número de iterações na temperatura inicial; ($N_o = \mu \cdot N_t$, em que N_t é o número de linhas candidatas);
 - w_o (*refcor*) define o corte de carga máximo aceitável para a seqüência *adição-troca-retirada* de linhas ou *retirada-troca-adição*.
3. Incrementar o contador de iterações n_{con} .
Se $n_{con} \geq N_k$, ir ao passo 7; caso contrário, ir ao passo 4.
4. Se o corte de carga corrente w_o^k for menor que w_o , ir ao passo 5; caso contrário, ir ao passo 6.
5. Seqüência *retirada-troca-adição*:
 - 5.1 Obter uma configuração candidata simulando a *retirada* de uma linha i aleatoriamente escolhida; processar um PPL para esta configuração e testar sua aceitação usando a *equação 5.1*. Se for aceita, atualize a configuração corrente e as outras grandezas do processo e volte ao passo 3; caso contrário, ir ao passo 5.2.
 - 5.2 Obter uma configuração candidata simulando a *troca* da linha i com uma linha j aleatoriamente escolhida; processar um PPL para a configuração candidata e testar sua aceitação usando a *equação 5.1*. Se for aceita, atualize a configuração corrente e as outras grandezas do processo e volte ao passo 3; caso contrário, ir ao passo 5.3.

- 5.3** Obter uma configuração candidata simulando a *adição* da linha j aleatoriamente escolhida; processar um PPL para a configuração candidata e testar sua aceitação usando a *equação 5.1*. Se for aceita, atualize a configuração corrente e as outras grandezas do processo. Volte ao passo **3**.
- 6.** Inverta a seqüência do passo **5** para *adição-troca-retirada*:
- 6.1** Simular a *adição* de uma linha. Se a configuração for aceita, volte ao passo **3**; caso contrário, ir a **6.2**.
 - 6.2** Simular a *troca* de linhas. Se a configuração for aceita, volte ao passo **3**; caso contrário, ir a **6.3**.
 - 6.3** Simular a *retirada* de uma linha. Volte ao passo **3**.
- 7.** Se o critério de parada for satisfeito, ir ao passo **9**; caso contrário, ir ao passo **8**.
- 8.** Atualizar os parâmetros: N_{k+1} e T_{k+1} e voltar ao passo **3**.
- 9.** Processo concluído, entrar na fase de melhoramento iterativo (busca local). PARE.

5.3.2 O algoritmo GANORD

Com o mesmo objetivo em relação ao algoritmo seqüencial anterior, neste trabalho utilizou-se como referência, para um algoritmo genético serial, o algoritmo GANORD ([Gallego, De Oliveira - 1997b] detalhado em [Gallego - 1997]), com alterações efetuadas no código fonte do mesmo para permitir também traçar os gráficos que são apresentados no Capítulo 6, e com isso comparar os resultados com as metaheurísticas propostas para o ambiente de processamento distribuído.

Na seqüência, apresenta-se de forma resumida os principais aspectos do algoritmo utilizado.

5.3.2.1 Estruturas básicas

De acordo com os conceitos apresentados no Capítulo 3, seção 3.3.1, páginas 19 a 23, no algoritmo GANORD são utilizadas as seguintes estruturas básicas:

população inicial:

- gerada com o auxílio do inicializador *INIC* para os diferentes sistemas testes.

representação e codificação do problema:

- na avaliação do nível de “fitness”, usa-se a normalização:

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

em que K é uma constante variável, correspondente à função objetivo de maior valor a cada geração; ou

$$\min v \iff \max [1/v]$$

- uso da codificação decimal.

parâmetros de controle:

- *npop* - tamanho da população;
- *ycross* - taxa de recombinação (“crossover”);
- *ndes* - número máximo de descendentes, em porcentagem;
- *nmutar* - taxa de mutação;
- *temp* - temperatura inicial do processo **SA**.

Identificação de “building blocks”:

- O algoritmo identifica o número e os elementos dos blocos construtivos correspondentes aos novos caminhos e, tanto no processo de recombinação como no de mutação, os mesmos são preservados.

seleção limitada:

- para evitar a convergência prematura num ótimo local no início da busca, pela maior quantidade de descendentes correspondentes àquelas configurações com função objetivo de boa qualidade, limita-se o número máximo de descendentes para cada configuração independentemente da qualidade de sua função objetivo. Ou seja, o parâmetro *ndes* (número máximo de descendentes) representa um limite, em porcentagem da população total.

recombinação:

- foi empregada a recombinação dupla, em que são selecionados de maneira aleatória dois pontos de recombinação e trocado o material genético entre estes dois pontos. Não se procede a recombinação de todas as configurações, mas através de uma taxa de recombinação (*ycross*) determina-se probabilisticamente a porcentagem de configurações que serão efetivamente re combinadas.

mutação variável controlada por **SA**

- *Critério de mutação:*

$$P(\Delta v) = \exp(-\Delta v/T)$$

em que Δv é a variação da função objetivo e T (*temp*) é o parâmetro de controle temperatura do processo “simulated annealing”.

Dada uma configuração candidata a mutação, determina-se Δv , simulando a introdução da mutação e resolvendo um PPL. Se Δv é menor que zero, então aceita-se a mutação, caso contrário, a mutação é aceita de forma probabilística quando $Pr(\Delta v) > P(0, 1)$; isto é, uma mutação de piora na função objetivo é aceita se $Pr(.)$ é maior que um número aleatoriamente gerado no intervalo $[0, 1]$.

A mutação pode ser simples ou dupla na representação decimal adotada; isto é, adiciona-se ou retira-se uma ou duas linhas na configuração do sistema. Isto é feito através de duas variantes: simples ou com a avaliação do corte de carga da configuração candidata. Nesta última, com adição de linhas em caso de existir corte de carga e retirada de linha caso contrário.

critério de parada

- o algoritmo pára após atingir um número máximo de gerações preestabelecido.

5.3.2.2 Descrição do algoritmo

- **Passo 1:** Ler o tamanho da população $npop$ e o número de configurações iniciais $npara$ (determinado previamente com o auxílio do programa inicializador *INIC*). Ler $kgera$ (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*).
Atualizar, se possível, a incumbente. k_2 conta o número de gerações.
- **Passo 3:** Se $k_2 > kgera$ 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.

5.4 O Algoritmo “Simulated Annealing” Paralelo

No algoritmo de “simulated annealing” o maior consumo de tempo computacional está na etapa de geração de tentativas para a formação de uma cadeia de Markov, ou seja, no problema de

planejamento da expansão da transmissão este maior consumo de tempo está na geração das N_k configurações candidatas para o nível de temperatura T_k . A Figura 5.2, a seguir, mostra as etapas para a avaliação de cada uma destas N_k configurações candidatas.

Observa-se que os três primeiros passos são independentes e podem ser implementados em paralelo, enquanto o passo 4 é estritamente seqüencial. O número de vezes em que são executados os passos de 1 a 3 praticamente não varia para os diferentes níveis de temperatura, sendo que em relação ao passo 4, o mesmo é mais freqüente para valores altos de temperatura e menos freqüente no final do processo, com a diminuição do valor da temperatura. Outra observação é que para o problema de planejamento da expansão da transmissão, o passo 2 apresenta um grande esforço computacional quando comparado com os outros passos, visto que neste passo deve-se resolver um PPL para a determinação do valor da função objetivo da configuração candidata.

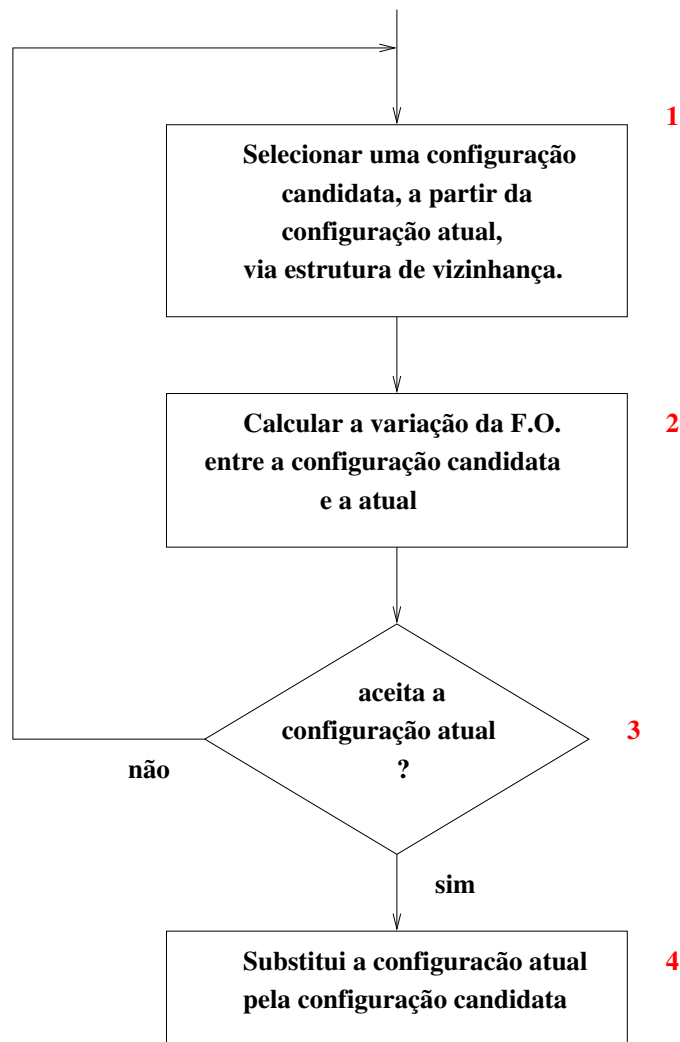


Figura 5.2: Geração das N_k configurações candidatas.

Com base nestas observações e a partir da versão do algoritmo paralelo da divisão proposto

em [Gallego, et al. - 1997], desenvolvido originalmente para uma máquina paralela nCUBE, foi desenvolvida uma nova versão do algoritmo, agora implementada para ambiente de processamento distribuído, com o auxílio da biblioteca PVM [Geist, et al. - 1994]; ou seja, utiliza-se uma rede de estações de trabalho com PVM. Esta nova versão, denominada SAPARNOR, tem seu princípio de funcionamento apresentado na Figura 5.3, a seguir. Foi acrescentado também ao algoritmo recursos para criação de arquivos de saída de dados que são utilizados para a confecção de gráficos que mostram o desempenho da incumbente do processo em cada processador. Estes gráficos são mostrados no Capítulo 6 para comparações de resultados.

No algoritmo, se N_k é o número de tentativas de busca de novas configurações realizadas na temperatura T_k e p o número de processadores, então, em cada nível de temperatura, cada processador realiza N_k/p tentativas, sendo que a tarefa de busca de novas configurações é dividida entre todos os processadores para cada nível de temperatura. Nesta versão, existem comunicações entre os processadores toda vez que há troca de temperatura, e então escolhe-se a melhor configuração das p configurações disponíveis, que por sua vez será usada como configuração inicial no nível de temperatura seguinte.

Na seqüência, o detalhamento do algoritmo é apresentado.

5.4.1 Descrição do Algoritmo SAPARNOR

1. Todos os p processadores alocados executam o fluxo DC inicial; com isto, o *mestre* (nó 0), bem como os *escravos* (demais processadores), resolvem um PPL e obtêm uma solução inicial do subproblema de operação.
2. Seleciona-se a temperatura inicial T_0 (*temp*), e determina-se o tamanho da cadeia de Markov inicial, $N_0 = nut.nl$. (*nut* é um parâmetro de controle que assume valores 2 ou 3 para sistemas de pequeno porte, 3 ou 4 para sistema de médio porte e 5 ou 6 para sistemas de grande porte; *nl* é o número de ramos candidatos.)

O tamanho da cadeia de Markov inicial, N_0 , corresponde na prática ao número de tentativas de investimento que devem ser feitas no nível T_0 de temperatura. O mesmo vale para N_k e T_k e, enquanto a temperatura é decrementada para níveis cada vez mais baixos, a quantidade de tentativas é acrescida de cifras da ordem de 20%.

3. A cadeia N_k é dividida entre os p processadores; assim, as sub-cadeias terão o tamanho N_k/p . Isto significa que o esforço de trabalhar sobre as N_k tentativas é repartido entre os p processadores.
4. Assim que o *mestre* completa as iterações de sua sub-cadeia, ele armazena em variáveis temporárias a configuração aceita mais recente; em seguida, informa aos demais processadores que sua sub-cadeia foi concluída. Cada um dos $p-1$ processadores envia ao *mestre* a configuração aceita mais atual.
5. O *mestre* ao receber a configuração enviada por um dado *escravo*, compara a sua solução com a que recebe. Caso a configuração que chega seja melhor, ela será trocada pela do

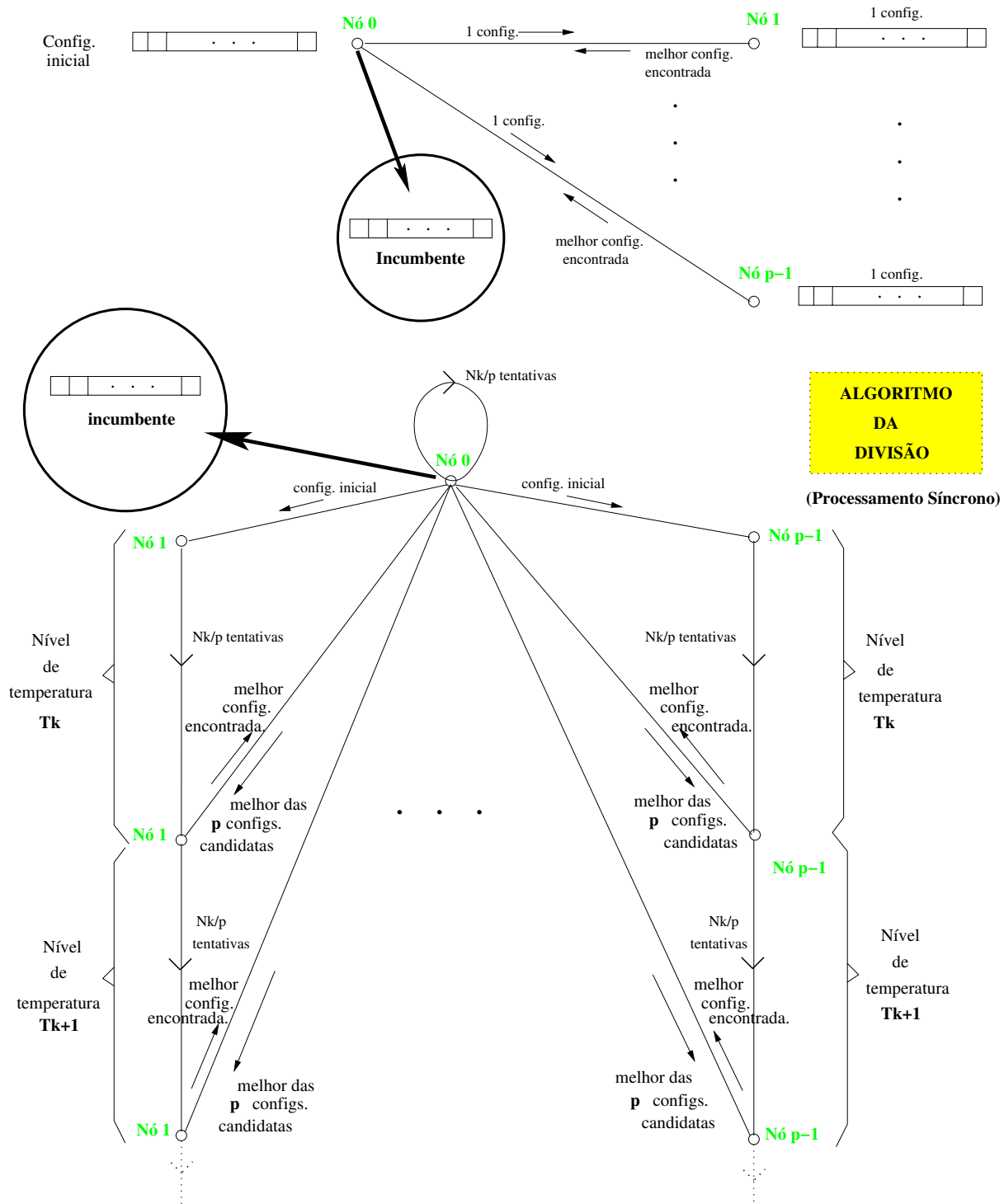
ALGORITMO "SIMULATED ANNEALING" PARALELO (SAPARNOR)

Figura 5.3: SAPARNOR - Princípio de funcionamento do algoritmo.

mestre. Se não, a configuração que chega será descartada. Este procedimento de seleção da configuração de melhor função objetivo prossegue até que todos os $p-1$ processadores tenham comunicado suas configurações ao *mestre*. Ao final ter-se-á a solução incumbente para esse nível de temperatura e para essa cadeia.

6. Para efeito de critério de parada, as configurações aceitas pelo *mestre* em cada nível de temperatura são contadas. O mesmo é feito em relação as configurações que chegam e que sejam idênticas a do *mestre*.
7. Em seguida, continuando o processamento, o *mestre* envia a solução incumbente a todos os $p-1$ processadores. Todos os *escravos* seguirão trabalhando sobre essa solução incumbente num novo nível de temperatura e numa nova cadeia.
8. O algoritmo ao selecionar a incumbente, só muda T_k e N_k após completar p ciclos de processamento. Neste algoritmo, maneja-se os parâmetros de controle: um fator de decremento da temperatura (*temperat*) e um incremento da cadeia (*cadeia*).

Um ciclo completo corresponde ao cálculo de uma sub-cadeia e a conseqüente escolha e envio da incumbente.

9. O processamento terminará se for satisfeito um dos seguintes *critérios de parada*:
 - (a) O número de iterações excedeu o limite máximo preestabelecido.
 - (b) O processo atingiu o “congelamento”.

O “congelamento” é alcançado se:

 - nenhuma configuração foi aceita pelo *mestre* num dado ciclo de processamento; ou
 - as configurações de todos os p processadores são idênticas num dado ciclo.

5.5 Algoritmos Genéticos Paralelos Propostos

A teoria de algoritmos genéticos mostra que existe um paralelismo intrínscio dentro do funcionamento de um algoritmo genético padrão (**GA**). Entretanto, quando da aplicação desta teoria na resolução do problema do planejamento da expansão da transmissão, pode-se implementar versões paralelas do algoritmo genético de acordo com duas propostas de caráter geral:

- Avaliação paralela das funções objetivo da população; e
- Alocação de subpopulação de configurações em cada processador.

Nesta seção, são apresentados diversos algoritmos baseados nas propostas acima, com o objetivo de investigar o desempenho dos mesmos em relação à versão seqüencial apresentada anteriormente, na resolução do problema do planejamento da expansão da transmissão. Uma síntese das proposições que se seguem pode ser vista no trabalho [De Oliveira, *et al.* - 2000].

Na seqüência, são detalhados os algoritmos propostos.

5.5.1 Avaliação paralela das funções objetivo da população

Considerando que o problema de planejamento da expansão de sistemas de transmissão está sendo resolvido com uma população de n configurações. Um algoritmo **GA** seqüencial deve resolver, de forma seqüencial, n problemas de programação linear (PPL) para determinar a função objetivo de cada uma das configurações da população. Deve-se lembrar que a resolução dos PPL's utiliza mais de 95% do tempo de processamento de um algoritmo genético seqüencial.

Nesta proposta, os processadores são usados para avaliar a função objetivo das configurações da população. E isto pode ser feito, dentre outras formas, de duas maneiras: atribuindo a cada processador o cálculo da função objetivo de cada configuração ou atribuindo a cada processador o cálculo de um conjunto de configurações.

Os algoritmos GAPAR1A e GAPAR1B que se seguem, resolvem o problema do planejamento com base nas duas proposições acima, respectivamente.

5.5.1.1 Algoritmo GAPAR1A

É proposto um algoritmo paralelo simples (ver Figura 5.4), que incorpora a paralelização da resolução dos PPL's, isto é, resolve os n PPL's usando n processadores (um PPL por processador) e o resto do processo é equivalente a um algoritmo **GA** seqüencial. Na prática, seria utilizar uma máquina paralela com 64 processadores (por exemplo, o nCube), para uma população de 64 configurações. A bibliografia geral de **GA**, [Goldberg - 1989] sugere este tipo de algoritmo mas menciona de que pode acontecer problemas de comunicação entre os processadores. Entretanto, este tipo de problema é mais típico naqueles problemas em que avaliar a função objetivo de cada configuração requer um esforço relativamente pequeno comparado com o esforço integral do **GA**, como pode acontecer com o problema do caixeiro viajante. Este fato, porém, não ocorre com o problema de planejamento da expansão do sistema de transmissão, como lembrado anteriormente.

Descrição do algoritmo GAPAR1A

- **Passo 0:** Definir o tamanho da população (n_{pop}) em função do número de processadores disponíveis, determinando previamente a população inicial com o auxílio do programa inicializador *INIC*;
definir o número máximo de gerações permitidas (k_{gera});
montar uma máquina paralela virtual com $p + 1$ processadores.

processo mestre

- **Passo 1:** Enviar a cada processador uma configuração da população.

processos escravos

- **Passo 1:** Determinar o valor da função objetivo de cada configuração correspondente, usando o algoritmo de PL (*MINOS*). PARE.

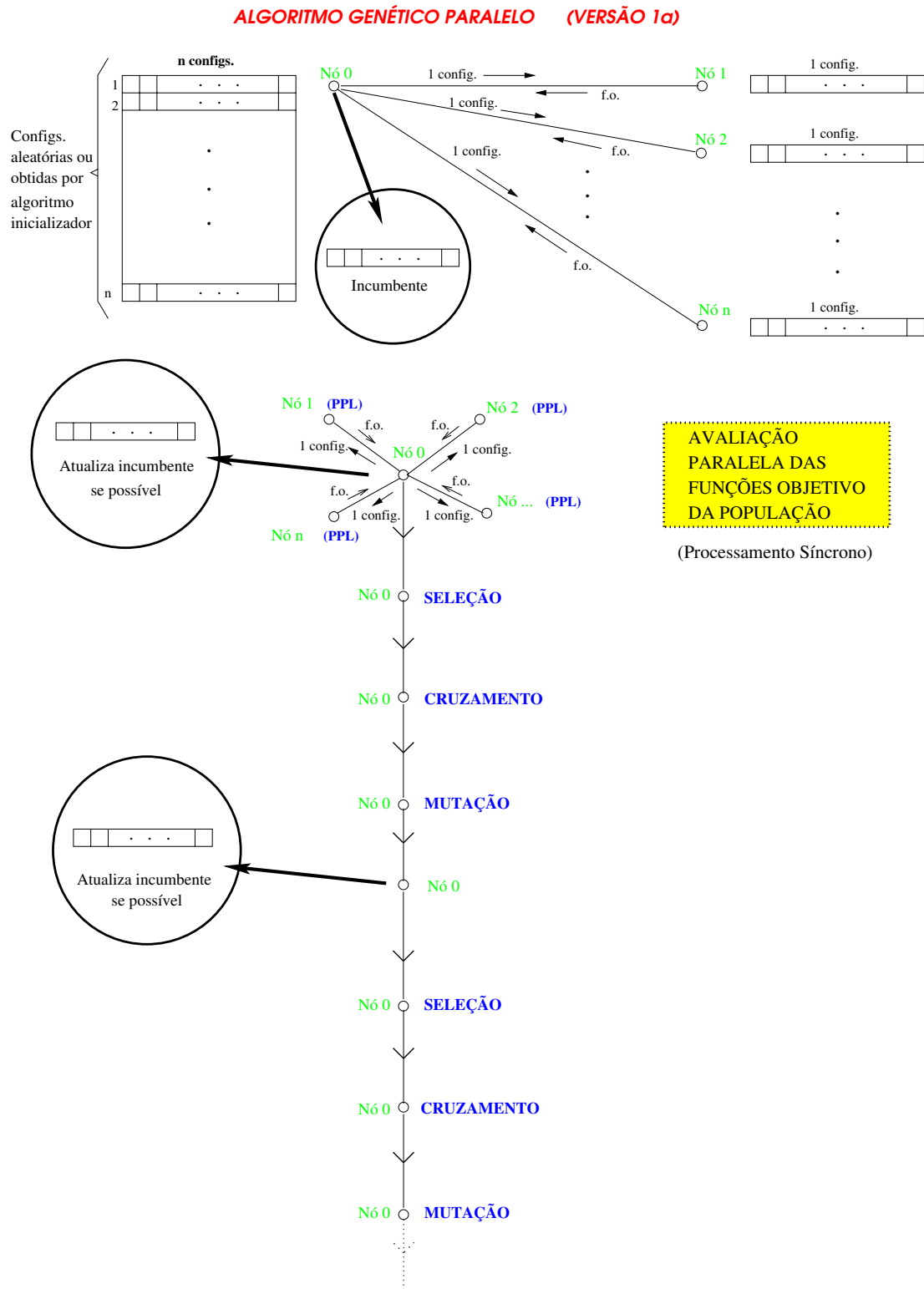


Figura 5.4: GAPAR1A - Avaliação paralela das funções objetivo da população, uma em cada processador.

processo mestre

- **Passo 2:** Receber de cada processador o valor da função objetivo de cada configuração.
- **Passo 3:** Atualizar a incumbente, se possível. Incrementar k_2 , o número de gerações.
- **Passo 4:** Se $k_2 > k_{gera}$ PARE, pois foi excedido o número máximo de gerações permitidas. Caso contrário, ir ao passo 5.
- **Passo 5:** Usando o mecanismo de seleção adequado, determinar o número de descendentes correspondente a cada configuração.
- **Passo 6:** Realizar recombinação entre cada par de configurações que participa 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 7:** Realizar a mutação. Este processo também é controlado estatisticamente pela taxa de mutação $nmutar$. Voltar ao passo 3.

5.5.1.2 Algoritmo GAPAR1B

Uma outra alternativa para a resolução do problema, é utilizar uma rede de estações de trabalho e uma biblioteca para processamento paralelo como o PVM ([Geist, *et al.* - 1994]), com o objetivo de dividir o cálculo dos n PPL's entre p processadores (estações), cabendo assim a cada processador: n/p PPL's. (Ver Fig. 5.5, a seguir.)

Descrição do algoritmo GAPAR1B

- **Passo 0:** Ler o tamanho da população (n_{pop}) e o número de configurações iniciais n_{para} determinado previamente com o auxílio do programa inicializador *INIC*; ler o número máximo de gerações permitidas (k_{gera}); montar uma máquina paralela virtual com $p + 1$ processadores.

processo mestre

- **Passo 1:** Enviar a cada processador escravo n/p configurações da população.

processos escravos

- **Passo 1:** Determinar os valores das funções objetivos correspondentes, usando o algoritmo de PL (*MINOS*).
- **Passo 2:** Enviar ao processo mestre os valores das respectivas n/p funções objetivos e PARE.

processo mestre

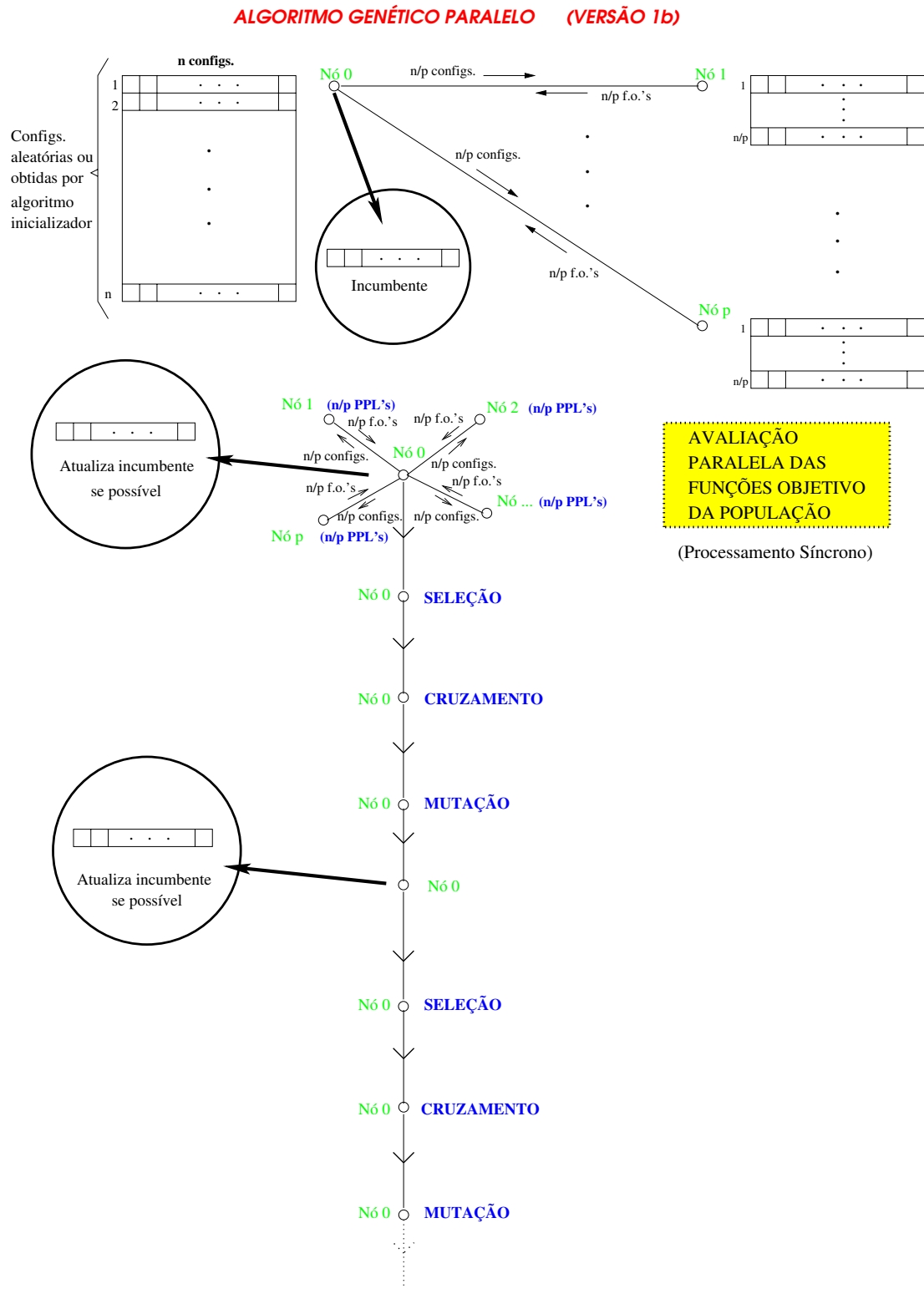


Figura 5.5: GAPAR1B - Avaliação paralela das funções objetivo da população, distribuídas para p processadores.

- **Passo 2:** Receber de cada processador os valores das n/p funções objetivos.
- **Passo 3:** Atualizar a incumbente, se possível. Incrementar k_2 , o número de gerações.
- **Passo 4:** Se $k_2 > k_{gera}$ PARE, pois foi excedido o número máximo de gerações permitidas. Caso contrário, ir ao passo 5.
- **Passo 5:** Usando o mecanismo de seleção adequado, determinar o número de descendentes correspondente a cada configuração.
- **Passo 6:** Realizar recombinação entre cada par de configurações que participa na formação da nova geração, até gerar um número de descendentes igual ao tamanho da população. A taxa de recombinação (γ_{cross}), controla estatisticamente o número de configurações que serão recombinadas.
- **Passo 7:** Realizar a mutação. Este processo também é controlado estatisticamente pela taxa de mutação γ_{mutar} . Voltar ao passo 3.

5.5.2 Alocação de subpopulação de configurações para cada processador

Neste caso, a população total é dividida em subpopulações e cada subpopulação trabalha de maneira independente em cada processador. É praticamente equivalente ao algoritmo da divisão e ao algoritmo híbrido implementado para “simulated annealing” (**SA**) [Gallego - 1997]. A vantagem adicional do **GA** em relação aos algoritmos de “simulated annealing” paralelos é que um algoritmo **GA** pode ser implementado de maneira assíncrona.

No **SA**, em cada nível de temperatura, cada processador realiza a busca de um determinado número de configurações candidatas e depois informa para o mestre a melhor configuração encontrada. Para que cada processador retome o trabalho de busca em **SA**, o mestre deve retornar a cada processador o novo valor da temperatura e a melhor configuração encontrada. No algoritmo genético paralelo, cada processador realizaria uma busca praticamente sem espera de nova informação pois simplesmente informaria ao mestre a melhor configuração encontrada e receberia do mesmo a informação da melhor configuração do processo (incumbente).

Para o exemplo mencionado anteriormente, a população de n configurações pode ser dividido em p subpopulações de n/p configurações cada uma e seria usado p processadores. Cada processador resolveria um ciclo geracional (seleção-recombinação-mutação-avaliação da função objetivo) e enviaria para o mestre a melhor configuração encontrada. Simultaneamente receberia do mestre a melhor configuração (ou as melhores configurações do processo integral) que o processador incorporaria na subpopulação para melhorar o processo de seleção. Logicamente, um processador não precisaria esperar que os outros processadores terminem seu correspondente ciclo geracional. Também a troca de informações não precisa ser a cada ciclo geracional, pode ser realizada após cada certo número de iterações. Para avaliar todas estas opções, é possível desenvolver versões como as apresentadas a seguir.

5.5.2.1 Algoritmo GAPAR2A

Uma primeira versão onde, após um número máximo de gerações (k), escolhido previamente, verifica-se a incumbente relativa a cada processador e escolhe-se a melhor dentre elas. (Ver Fig. 5.6.)

Descrição do algoritmo GAPAR2A

- **Passo 0:** Ler o tamanho da população (n_{pop}) e o número de configurações iniciais (n_{para}) determinado previamente com o auxílio do programa inicializador *INIC*; ler o número máximo de gerações permitidas (k_{gera}); montar uma máquina paralela virtual com $p + 1$ processadores.

processo mestre

- **Passo 1:** Enviar a cada processador escravo n/p configurações da população.

processos escravos

- **Passo 1:** Determinar os valores das funções objetivos das n/p configurações, usando o algoritmo de PL (*MINOS*).
- **Passo 2:** Atualizar a incumbente, se possível. Incrementar k_2 , o número de gerações.
- **Passo 3:** Se $k_2 > k_{gera}$, ir ao passo 7, pois foi excedido o número máximo de gerações permitidas. Caso contrário, ir ao passo 4.
- **Passo 4:** Usando o mecanismo de seleção adequado, determinar o número de descendentes correspondente a cada configuração.
- **Passo 5:** Realizar recombinação entre cada par de configurações que participa na formação da nova geração, até gerar um número de descendentes igual ao tamanho da população. A taxa de recombinação (y_{cross}), controla estatisticamente o número de configurações que serão recombinadas.
- **Passo 6:** Realizar a mutação. Este processo também é controlado estatisticamente pela taxa de mutação n_{mutar} . Voltar ao passo 2.
- **Passo 7:** Enviar a incumbente para o processo mestre e PARE.

processo mestre

- **Passo 2:** Receber a incumbente de cada processador escravo e escolher a melhor dentre elas. PARE

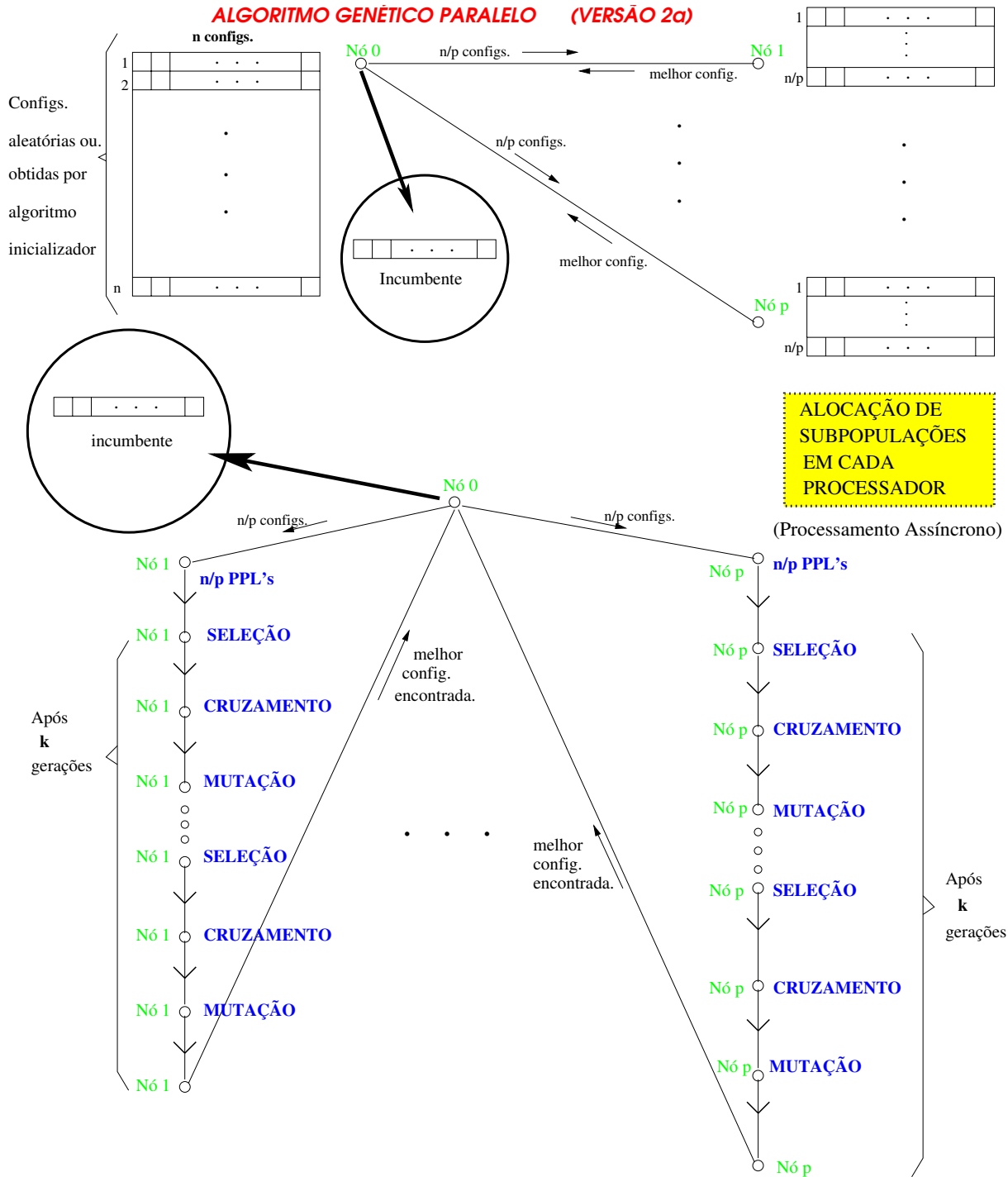


Figura 5.6: GAPAR2A - Alocação de subpopulações em cada processador, armazenando a incumbente no final de k gerações.

5.5.2.2 Algoritmo GAPAR2B

Uma segunda versão aonde, após um número pequeno de gerações (k_1), o mestre receberia as melhores configurações encontradas, mantendo o valor da melhor entre elas como uma referência e continua o processamento até um critério de parada ser satisfeito. (Ver Fig. 5.7.)

Descrição do algoritmo GAPAR2B

- **Passo 0:** Definir o tamanho da população ($npop$) determinando previamente a população inicial com o auxílio do programa inicializador *INIC*;
definir o número máximo de gerações permitidas (k_{gera});
definir o número de gerações (k_1) para troca de mensagens entre os processadores;
montar uma máquina paralela virtual com $p + 1$ processadores.

processo mestre

- **Passo 1:** Enviar a cada processador escravo n/p configurações da população.

processos escravos

- **Passo 1:** Determinar os valores das funções objetivos das n/p configurações, usando o algoritmo de PL (*MINOS*).
- **Passo 2:** Fazer $N = \text{int}(k_2/k_1)$, onde k_2 é o número de gerações;
atualizar a incumbente, se possível;
incrementar k_2 .
- **Passo 3:** Se $k_2 = k_{gera}$ ou $k_2 = N * k_1$, ir ao passo 7. Caso contrário, ir ao passo 4.
- **Passo 4:** Usando o mecanismo de seleção adequado, determinar o número de descendentes correspondente a cada configuração.
- **Passo 5:** Realizar recombinação entre cada par de configurações que participa 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:** Realizar a mutação. Este processo também é controlado estatisticamente pela taxa de mutação $nmutar$. Voltar ao passo 2.
- **Passo 7:** Enviar a incumbente para o processo mestre. Se $k_2 = k_{gera}$, enviar aviso para o mestre parar e PARE. Caso contrário, ir para o passo 8

processo mestre

- **Passo 2:** Receber de cada processador escravo a incumbente e escolher dentre elas a melhor. Se tem aviso para parar, PARE. Caso contrário, ir para o passo 3.

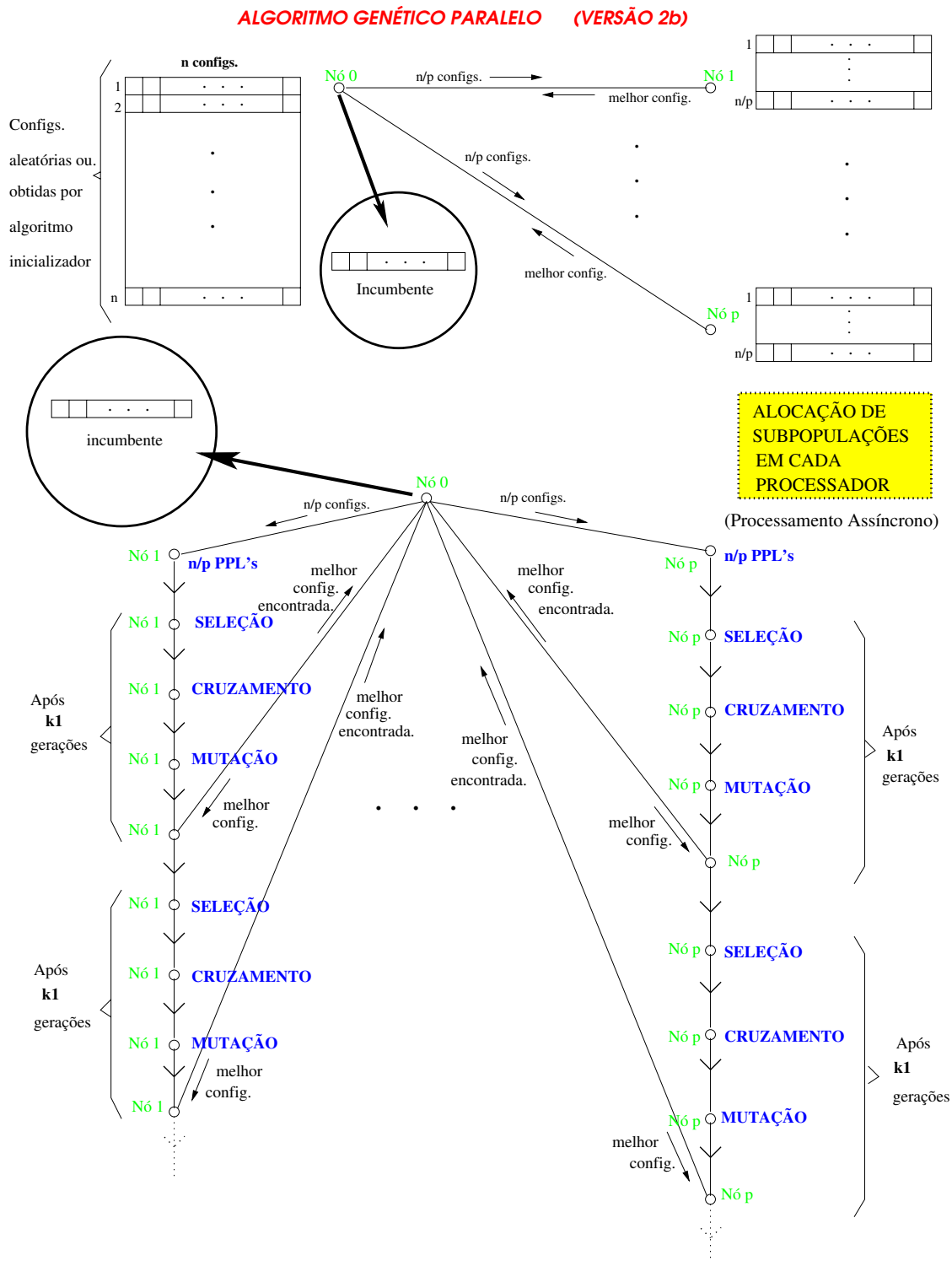


Figura 5.7: GAPAR2B - Alocação de subpopulações em cada processador, armazenando a incumbente após cada $k1$ gerações.

- **Passo 3:** Enviar a melhor configuração para os processadores escravos. Voltar para o passo 2.

processos escravos

- **Passo 8:** Receber a melhor configuração do mestre e voltar para o passo 2.

variante GAP2BDIM

Com a divisão da população em subpopulações e com o processamento destas últimas em cada processador, o número de configurações efetivamente manuseadas em cada subproblema torna-se pequeno em relação ao número de configurações da população normalmente utilizado para um determinado sistema em estudo. Assim, desenvolveu-se a variante GAP2BDIM, cuja única diferença em relação ao algoritmo GAPAR2B, anteriormente detalhado, está no aumento da dimensão das variáveis do programa relativas ao número de barras do sistema. Desta maneira, pode-se agora iniciar o algoritmo com dimensões de população inicial bem maiores do que as normalmente utilizadas para cada sistema, cabendo a cada processador uma subpopulação maior do que teríamos no algoritmo GAPAR2B.

Nesta variante, foram introduzidas também opções de escolha de diferentes métodos de seleção, via parâmetros de entrada para o programa.

variante GAP2BCIR

Uma outra variante desenvolvida, a GAP2BCIR, corresponde ao mesmo algoritmo GAPAR2B com as alterações efetuadas na variante anterior GAP2BDIM, mas agora com a possibilidade de se trabalhar com sistemas com dois circuitos por ramo. Em todas as outras versões, o sistema sempre deve ter, no caso de ter dois circuitos por ramo, seus circuitos equivalentes.

5.5.2.3 Algoritmo GAPAR3A

Uma outra versão onde, após um número pequeno de gerações (k_1), o mestre armazenaria m configurações elites, diversificadas. Repassando-as novamente a cada processador que as incorporariam à sua subpopulação, até um critério de parada ser satisfeito. (Ver Fig. 5.8.)

Descrição do algoritmo GAPAR3A

- **Passo 0:** Definir o tamanho da população ($npop$) determinando previamente a população inicial com o auxílio do programa inicializador *INIC*;
definir o número máximo de gerações permitidas (k_{gera});
definir o número de gerações (k_1) para troca de mensagens entre os processadores;
montar uma máquina paralela virtual com $p + 1$ processadores.

processo mestre

- **Passo 1:** Enviar a cada processador escravo n/p configurações da população.

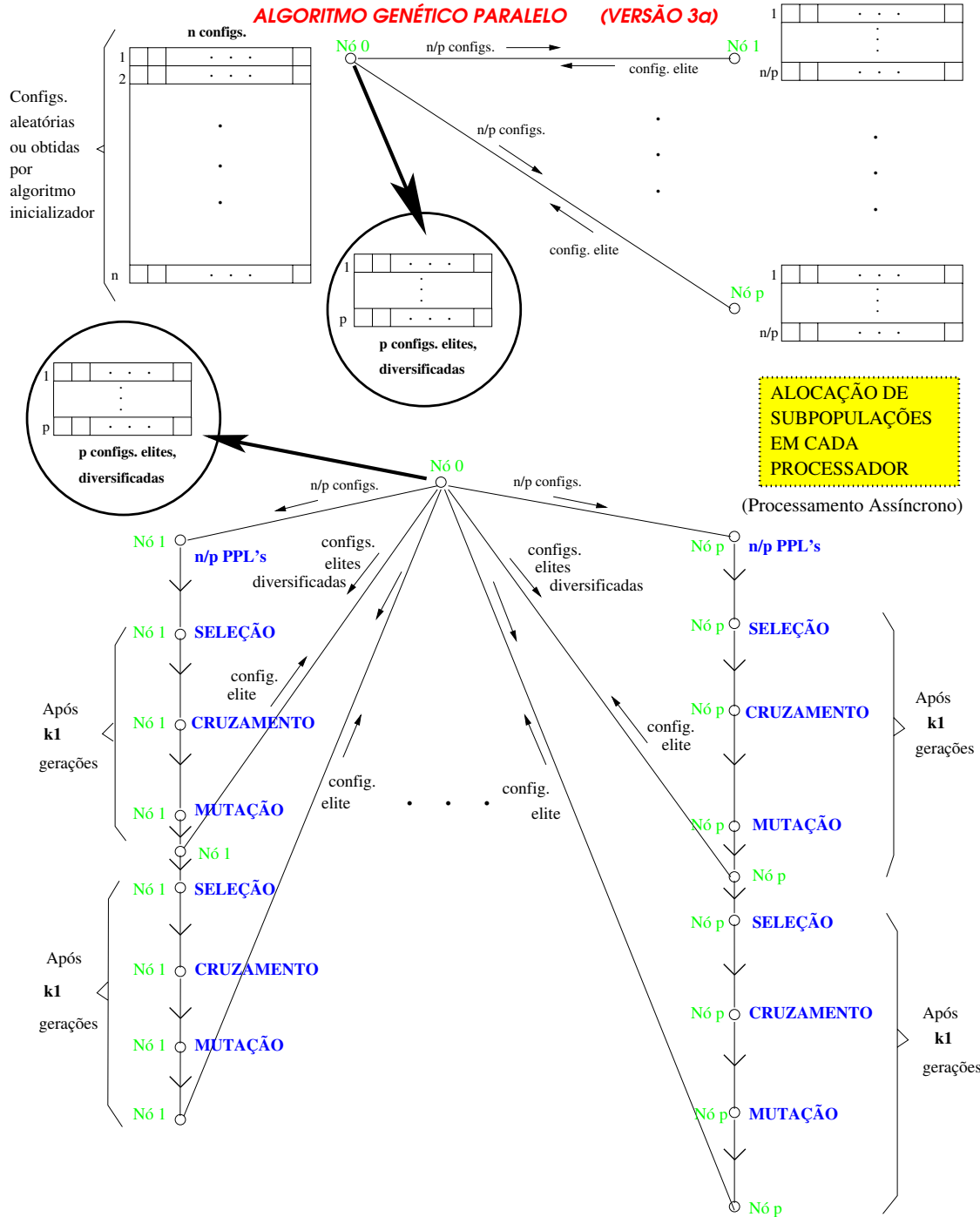


Figura 5.8: GAPAR3A - Alocação de subpopulações em cada processador, armazenando p configs. elites diversificadas no final de $k1$ gerações.

processos escravos

- **Passo 1:** Determinar os valores das funções objetivos das n/p configurações, usando o algoritmo de PL (*MINOS*).
- **Passo 2:** Fazer $N = \text{int}(k_2/k_1)$, onde k_2 é o número de gerações; determinar a configuração elite (incumbente da subpopulação); incrementar k_2 .
- **Passo 3:** Se $k_2 = k_{\text{gera}}$ ou $k_2 = N * k_1$, ir ao passo 7. Caso contrário, ir ao passo 4.
- **Passo 4:** Usando o mecanismo de seleção adequado, determinar o número de descendentes correspondente a cada configuração.
- **Passo 5:** Realizar recombinação entre cada par de configurações que participa 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:** Realizar a mutação. Este processo também é controlado estatisticamente pela taxa de mutação *nmutar*. Voltar ao passo 2.
- **Passo 7:** Enviar as configurações elites para o processo mestre. Se $k_2 = k_{\text{gera}}$, enviar aviso para o mestre parar e PARE. Caso contrário, ir para o passo 8

processo mestre

- **Passo 2:** Receber de cada processador escravo a incumbente da subpopulação (configuração elite) e atualizar a lista de **p** configurações elites diversificadas. Se tem aviso para parar, PARE. Caso contrário, ir para o passo 3.
- **Passo 3:** Enviar **p** configurações elites diversificadas para os processadores escravos. Voltar para o passo 2.

processos escravos

- **Passo 8:** Receber **p** configurações elites diversificadas do mestre e voltar para o passo 2.

5.5.2.4 Algoritmo GAPAR3B

E finalmente, uma versão em que, após um número pequeno de gerações ($k1$), o mestre receberia as configurações elites de cada processador, armazenando as elites das elites, e repassando-as novamente a cada processador que as incorporariam à sua subpopulação, até um critério de parada ser satisfeito. (Ver Fig. 5.9.)

Descrição do algoritmo GAPAR3B

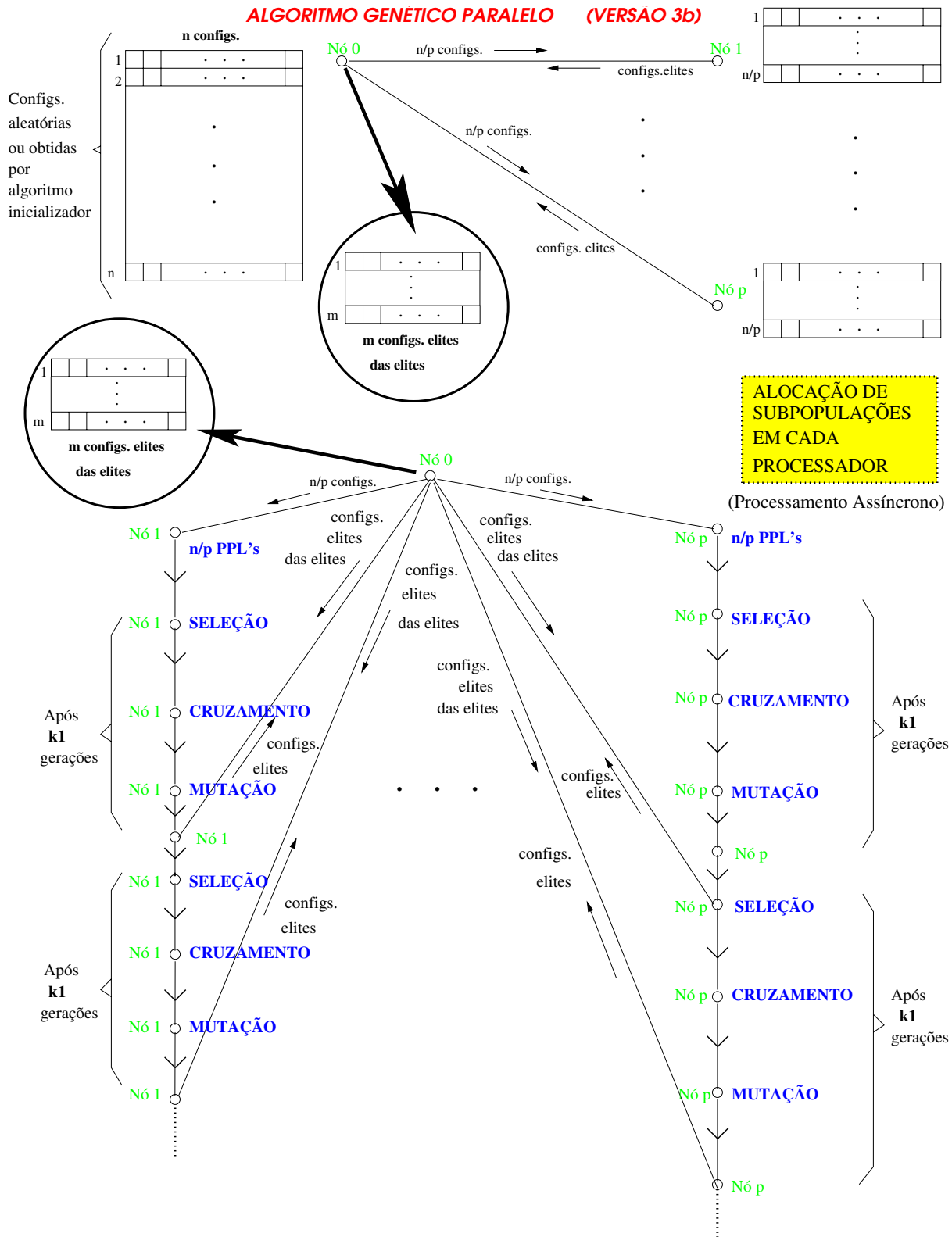


Figura 5.9: GAPAR3B - Alocação de subpopulações em cada processador, armazenando m configs. elites das elites após cada $k1$ gerações.

- **Passo 0:** Definir o tamanho da população (n_{pop}) determinando previamente a população inicial com o auxílio do programa inicializador *INIC*;
definir o número máximo de gerações permitidas (k_{gera});
definir o número de gerações (k_1) para troca de mensagens entre os processadores;
montar uma máquina paralela virtual com $p + 1$ processadores.

processo mestre

- **Passo 1:** Enviar a cada processador escravo n/p configurações da população.

processos escravos

- **Passo 1:** Determinar os valores das funções objetivos das n/p configurações, usando o algoritmo de PL (*MINOS*).
- **Passo 2:** Fazer $N = \text{int}(k_2/k_1)$, onde k_2 é o número de gerações;
atualizar a lista de configurações elites;
incrementar k_2 .
- **Passo 3:** Se $k_2 = k_{gera}$ ou $k_2 = N * k_1$, ir ao passo 7. Caso contrário, ir ao passo 4.
- **Passo 4:** Usando o mecanismo de seleção adequado, determinar o número de descendentes correspondente a cada configuração.
- **Passo 5:** Realizar recombinação entre cada par de configurações que participa 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:** Realizar a mutação. Este processo também é controlado estatisticamente pela taxa de mutação $nmutar$. Voltar ao passo 2.
- **Passo 7:** Enviar as configurações elites para o processo mestre. Se $k_2 = k_{jera}$, enviar aviso para o mestre parar e PARE. Caso contrário, ir para o passo 8

processo mestre

- **Passo 2:** Receber de cada processador escravo configurações elites e escolher dentre todas recebidas as m melhores configurações elites. Se tem aviso para parar, PARE. Caso contrário, ir para o passo 3.
- **Passo 3:** Enviar m configurações elites das elites para os processadores escravos. Voltar para o passo 2.

processos escravos

- **Passo 8:** Receber m configurações elites das elites do mestre e voltar para o passo 2.

Das versões citadas, foram desenvolvidas neste trabalho as versões de algoritmos genéticos: GAPAR1B, GAPAR2A, e GAPAR2B, GAP2BDIM, GAP2BCIR e GAPAR3A, além das alterações efetuadas no algoritmo serial mencionado inicialmente.

5.6 O Time Assíncrono de Metaheurísticas

Os agentes mencionados, bem como outros possíveis, podem ser incorporados ao **A-Team** de metaheurísticas (MHEUR) de diferentes maneiras; como, por exemplo, aquelas apresentadas a seguir. Na Figura 5.10, tem-se um exemplo de **A-Team** MHEUR que é composto por três agentes independentes que trabalham segundo suas respectivas heurísticas, compartilhando uma memória central com uma população de configurações. Neste exemplo, os agentes tomam uma configuração ou uma subpopulação e procuram melhorá-las. O agente destruidor é na prática uma função intrínseca do programa MHEUR que mantém fixo o tamanho da população na memória central, ou seja, as piores configurações são substituídas pelas melhores em cada ciclo, ou simplesmente uma nova configuração substitui uma outra existente, aleatoriamente.

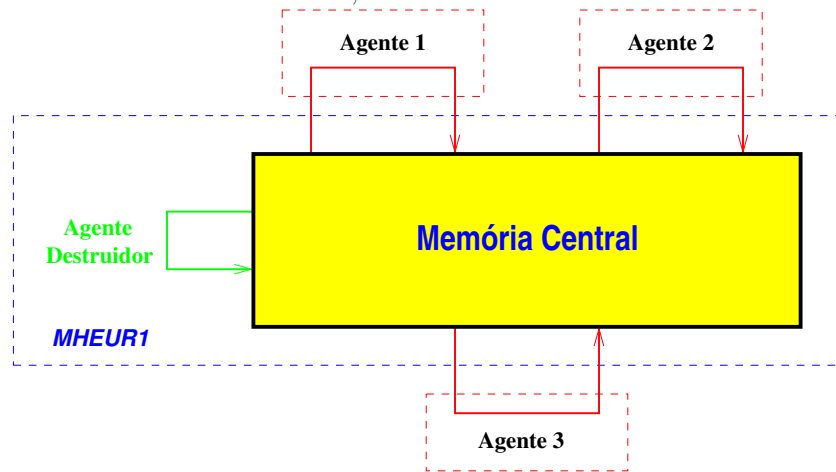


Figura 5.10: Exemplo 1 de time assíncrono de metaheurísticas.

Na Figura 5.11, têm-se outro exemplo de **A-Team** MHEUR com duas memórias compartilhadas, onde na central agem os agentes **1** e **2**, melhorando a população como um todo, e na secundária atua um agente **3** que toma aleatoriamente uma configuração para otimizá-la. Outras diferentes composições de agentes e memórias podem ser também exemplos de times assíncronos possíveis de serem implementados para a resolução do problema do planejamento da transmissão.

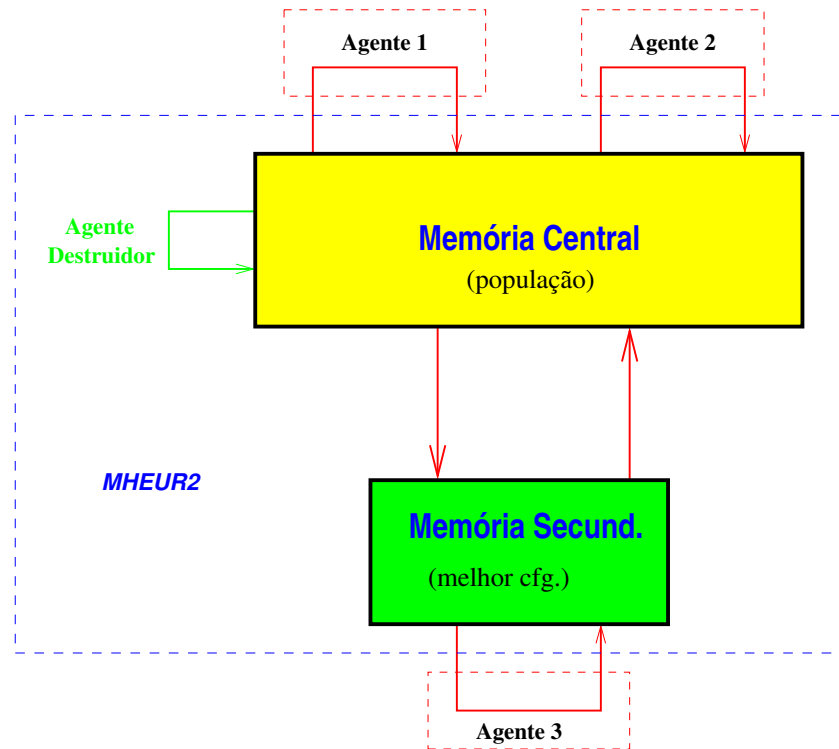


Figura 5.11: Exemplo 2 de time assíncrono de metaheurísticas.

Neste trabalho é proposto um **A-Team** da forma do Exemplo 1, com os agentes baseados nas heurísticas das versões sequenciais de “simulated annealing” e algoritmo genético; ou seja, os agentes são versões paralelas derivadas das mesmas. Nas seções que se seguem são apresentados mais detalhes do **A-Team** proposto.

5.6.1 Algoritmo MHEUR proposto

O *A-Team* apresentado neste trabalho é da forma da Figura 5.10, sendo a memória central representada pelo algoritmo MHEUR e seus agentes são versões paralelas derivadas dos algoritmos SANORD (“simulated annealing”) e GANORD (algoritmo genético); denominados SAM e GAM respectivamente. O agente SAM recebe do MHEUR a pior configuração da população atual e após cada ciclo de temperatura devolve à população a configuração incumbente que substitue aleatoriamente uma configuração da população. De maneira simultânea, o agente GAM recebe a população atual e trabalha durante *kk1* gerações previamente designadas, devolvendo à população a incumbente, colocando-a na posição da pior configuração da população atual. O agente destruidor, como mencionado anteriormente, representa uma função intrínseca do algoritmo MHEUR que mantém fixo o tamanho da população, com a substituição da pior configuração pela incumbente e/ou qualquer substituição aleatória de configuração.

A Figura 5.12, a seguir, mostra o esquema de funcionamento do **A-Team** MHEUR, quando

estão presentes, simultaneamente, os agentes SAM e GAM.

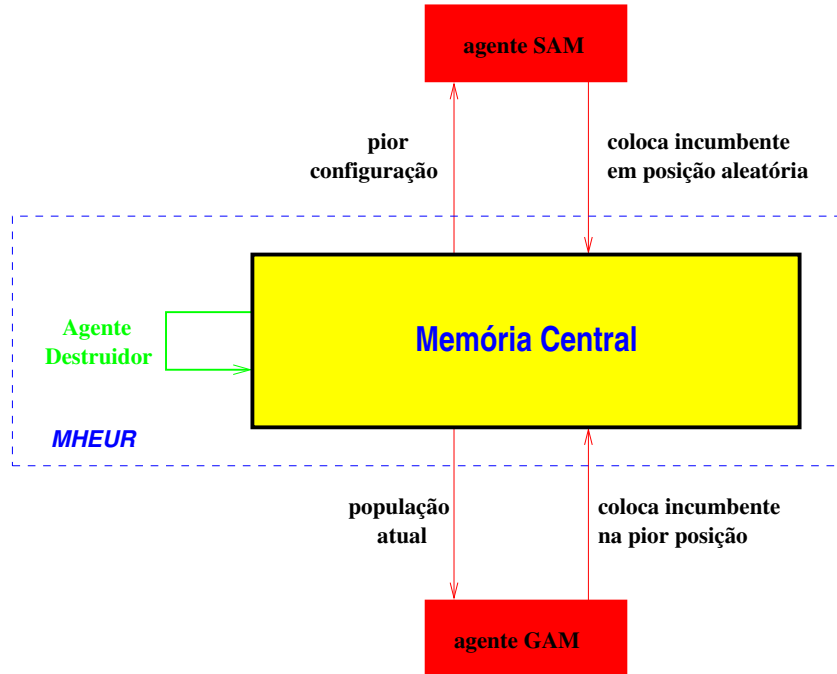


Figura 5.12: Esquema de funcionamento do time assíncrono de metaheurísticas.

5.6.1.1 Descrição do algoritmo MHEUR

O algoritmo MHEUR é descrito como segue:

- Passo 1.** Leitura dos parâmetros iniciais: tamanho da população (n_{pop}), número de configurações iniciais (n_{para}), nome da máquina (maq) e nome do agente $namea(agn)$;
- Passo 2.** Desova dos agentes e abertura do cronômetro;
- Passo 3.** Leitura dos dados do sistema a ser analisado;
- Passo 4.** Envio dos dados do sistema para todos os agentes;
- Passo 5.** Verifica presença de agentes na fila. Se for um novo agente, vá para o passo **10**, caso contrário continue.
- Passo 6.** Recebe configuração atualizada do agente;
- Passo 7.** Atualize a memória central;
- Passo 8.** Recebeu aviso para encerrar? Se sim, incrementa contador auxiliar $nsai$ e continue.

Passo 9. Verifique se $nsai$ é igual ao número de agentes. Se sim, fecha o cronômetro, imprime resultados e PARE; caso contrário, retorna para o passo 5.

Passo 10. Envia nova configuração para o agente, e retorna para o passo 5.

5.6.1.2 Descrição do agente SAM

O agente SAM é descrito como segue:

Passo 1. Recebimento dos dados do sistema a ser analisado;

Passo 2. Leitura dos parâmetros: $cadeia$, $temperat$, $nafa$, ix , $iterm$, $temp$, nut , $refcor$, $limcong$ e ncl . Ver significados de cada variável nas seções 5.3.1.1 e 5.3.1.2, das páginas 59 e 60, respectivamente;

Passo 3. Envia sinal solicitando primeira configuração;

Passo 4. Recebe configuração do algoritmo MHEUR;

Passo 5. Desenvolve processamento básico do algoritmo SANORD (passos 3 a 6.3, seção 5.3.1.2 da página 60) para um nível de temperatura;

Passo 6. Se o critério de parada for satisfeito, ir ao passo 8; caso contrário, ir ao passo 7.

Passo 7. Atualiza os parâmetros: N_{k+1} e T_{k+1} , envia a configuração atualizada para a memória principal e volta ao passo 4.

Passo 8. Envia aviso de encerramento para o MHEUR e PARE.

5.6.1.3 Descrição do agente GAM

Passo 1. Recebimento dos dados do sistema a ser analisado;

Passo 2. Leitura dos parâmetros: $nafa$, $kincumb$, $jgera$, $ycross$, $nmax$, $bconst$, $nmutar$, $kseed$, $ndes$, $temp$, $alfat$, $mcors$, ncl e $kk1$. Ver significados de cada variável nas seções 5.3.2.1 e 5.3.2.2, das páginas 61 e 63, respectivamente;

Passo 3. Envia sinal solicitando população atual;

Passo 4. Desenvolva processamento básico do algoritmo GANORD (passos 2 a 6, seção 5.3.2.2 da página 63), para cada geração e incremente contador auxiliar $kkcc$;

Passo 5. Verifica critério de parada. Se $kgera$ for maior ou igual a $jgera$ (número máximo de gerações), determina a incumbente, envia-a para MHEUR e vá ao passo 7; Caso contrário, continue.

Passo 6. Se $kkcc$ for igual a $kk1$ (número de gerações para troca de informações), determina a incumbente e a envia para MHEUR, retornando ao passo **3**. Caso contrário, retorna ao passo **4**;

Passo 7. Envia aviso de encerramento para o MHEUR e PARE.

No Capítulo que se segue são apresentados diversos resultados de simulações de todos os algoritmos implementados.

Capítulo 6

Resultados

6.1 Introdução

Os resultados e análise apresentados neste capítulo são referentes aos sistemas testes apresentados no *Apêndice A*, ou seja: como um sistema teste de uma rede de transmissão de pequeno porte utiliza-se o sistema Garver (6 barras / 15 ramos), para simular uma rede de transmissão de médio porte utiliza-se o sistema Sul brasileiro (46 barras / 79 ramos) e para simular uma rede de grande porte utiliza-se o sistema Norte-Nordeste brasileiro (87 barras / 179 ramos) e o sistema colombiano (93 barras / 155 ramos). No Apêndice A, todos os dados necessários são apresentados, bem como é apresentado o diagrama unifilar com as linhas existentes e candidatas para cada sistema.

Neste capítulo, para os sistemas Garver e Sul brasileiro, apresentam-se resultados referentes às simulações *sem* e *com* configurações iniciais factíveis determinadas previamente pelo algoritmo inicializador INIC. Dessa forma, pretende-se destacar a importância da inicialização dos algoritmos combinatórios na resolução do problema de planejamento da expansão da transmissão. Para os sistemas Norte-Nordeste brasileiro e colombiano, sistemas de grande porte, em todas as simulações são utilizadas configurações iniciais factíveis. Por questão de restrições de espaço, não são apresentadas as configurações iniciais utilizadas em cada simulação, sendo citado simplesmente o número de configurações iniciais utilizadas em cada caso. Quando não houver inicialização, todas as configurações estarão zeradas, isto é, o sistema opera inicialmente com a topologia base e sem adições. Para o sistema Garver, foram utilizadas 1 e 8 configurações iniciais, para o sistema Sul brasileiro, 1 e 20 configurações iniciais, para o sistema Norte-Nordeste, 1, 60, 90, 200 configurações iniciais e para o sistema colombiano, 20 e 80 configurações iniciais.

Na sequência, são apresentados alguns resultados para os sistemas Garver, com e sem redesenho; Sul Brasileiro, com e sem redesenho; Norte-Nordeste Brasileiro, planos 2002 e 2008 e Sistema Colombiano, plano 2012. Foram feitas simulações para todos os algoritmos implementados: SANORD, GANORD, SAPARNOR, GAPAR1B, GAPAR2A, GAPAR2B, GAPAR3A e MHEUR. Em cada tabela, são apresentados: um gráfico com a evolução da(s) incumbente(s), os parâmetros utilizados na simulação e a configuração ótima ou melhor configuração encontrada. No contexto do gráfico, são apresentados o valor da função objetivo correspondente à solução encontrada com seu

respectivo corte de carga, o tamanho da população e o número de configurações iniciais, quando for o caso.

No final do capítulo, para cada algoritmo implementado, são apresentadas tabelas com uma síntese dos resultados de todos os sistemas testes simulados, que entre outros dados, apresentam o tempo de obtenção da solução ótima e o tempo médio por PL, para uma análise e comparação de desempenho dos algoritmos propostos.

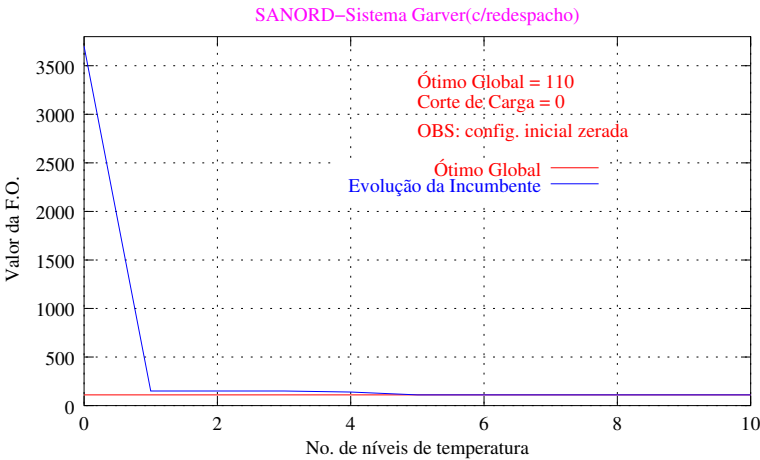
6.2 Sistema Garver

6.2.1 Sistema Garver (com redespacho)

6.2.1.1 Algoritmo SANORD

Observa-se pelos gráficos das Tabelas 6.1 e 6.2 que, sem o uso do inicializador a solução ótima do problema foi obtida em 5 ciclos de temperatura, com 401 PPLs e tempo de 23,6 segundos; enquanto que com o uso de uma configuração inicial (obtida via algoritmo de inicialização INIC) obteve-se a solução em 2 ciclos, 185 PPLs e tempo de 10,6 segundos.

Tabela 6.1: Sistema Garver - algoritmo SANORD (sem inicialização)



a) gráfico

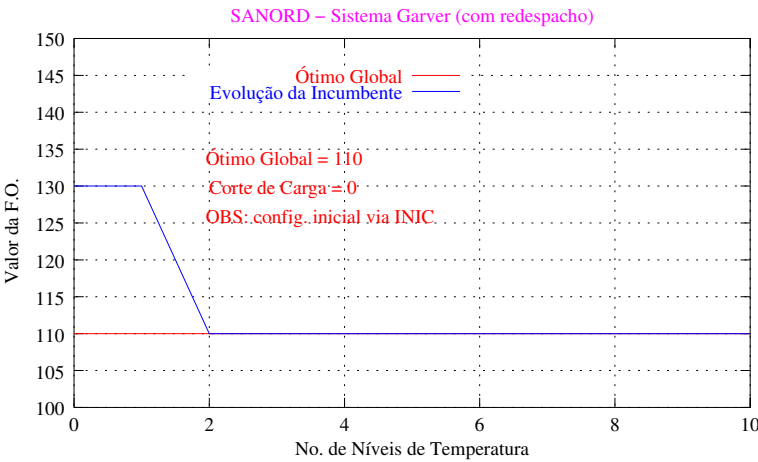
0	0	0	0	0	0	0	0	0	0	1	0	0	3	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

b) configuração ótima

parâmetro	valor
cadeia	1.2
temperat	0.7
nafa	10
ix	1114
item	2000
temp	300
nut	3
refcor	0.
limcong	3
ncl	0

c) Parâmetros

Tabela 6.2: Sistema Garver - algoritmo SANORD (com inicialização)



a) gráfico

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

b) configuração ótima

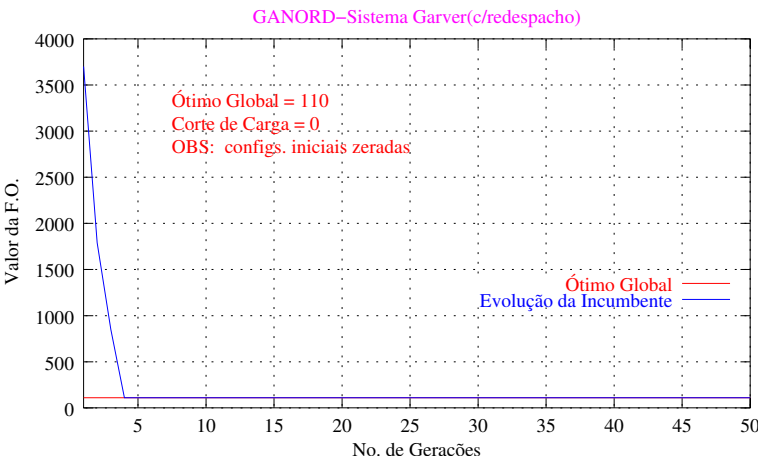
parâmetro	valor
cadeia	1.2
temperat	0.7
nafa	10
ix	7233
item	2000
temp	100
nut	3
refcor	0.
limcong	3
ncl	0

c) parâmetros

6.2.1.2 Algoritmo GANORD

Nos gráficos das Tabelas 6.3 e 6.4 também se verifica a importância da inicialização para o algoritmo GANORD, com a redução do número de gerações de quatro (409 PPLs, 23,6 s) para duas (130 PPLs, 7,5 s), com o uso de oito configurações iniciais em comparação à simulação sem nenhuma inicialização.

Tabela 6.3: Sistema Garver - algoritmo GANORD (sem inicialização)



a) gráfico

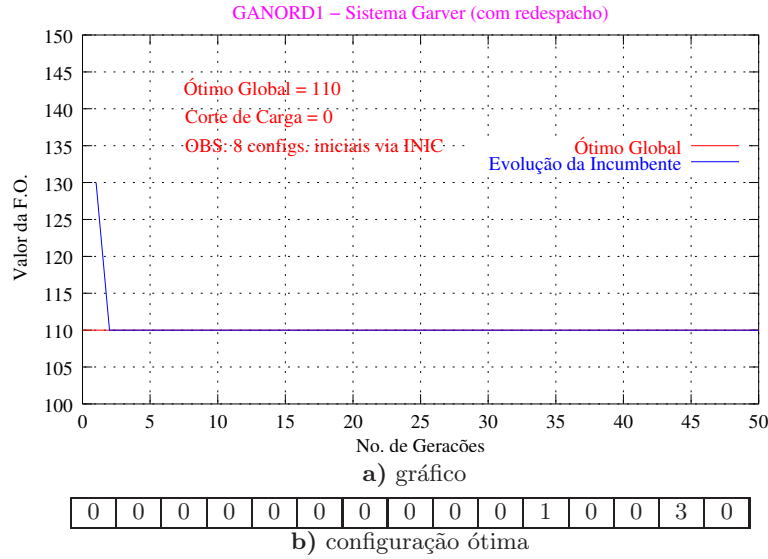
0	0	0	0	0	0	0	0	0	0	0	1	0	0	3	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

b) configuração ótima

parâmetro	valor
nafa	10
npop	40
kincumb	4000
jgera	50
npara	1
ycross	0.96
nmax	4
bconst	0.996
nmutar	4
kseed	1112
ndes	5%
temp	250
alfat	0.98
mcor	0
ncl	0

c) parâmetros

Tabela 6.4: Sistema Garver - algoritmo GANORD (com inicialização)



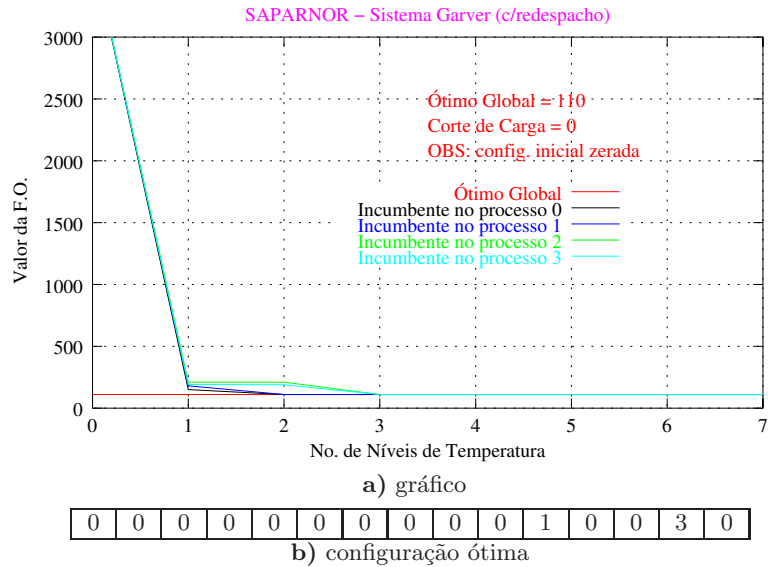
parâmetro	valor
nafa	10
npop	40
kincumb	4000
jgera	50
npara	8
ycross	0.96
nmax	4
bconst	0.996
nmutar	4
kseed	1112
ndes	5%
temp	250
alfat	0.98
mcors	0
ncl	0

c) parâmetros

6.2.1.3 Algoritmo SAPARNOR

Nas Tabelas 6.5 e 6.6, o uso do inicializador não diminuiu o número de gerações para o algoritmo SAPARNOR, mas a solução ótima foi obtida em 25 PPLs com um tempo de 0,6 segundos, contra 90 PPLs com um tempo de 1,9 segundos para o caso sem inicialização.

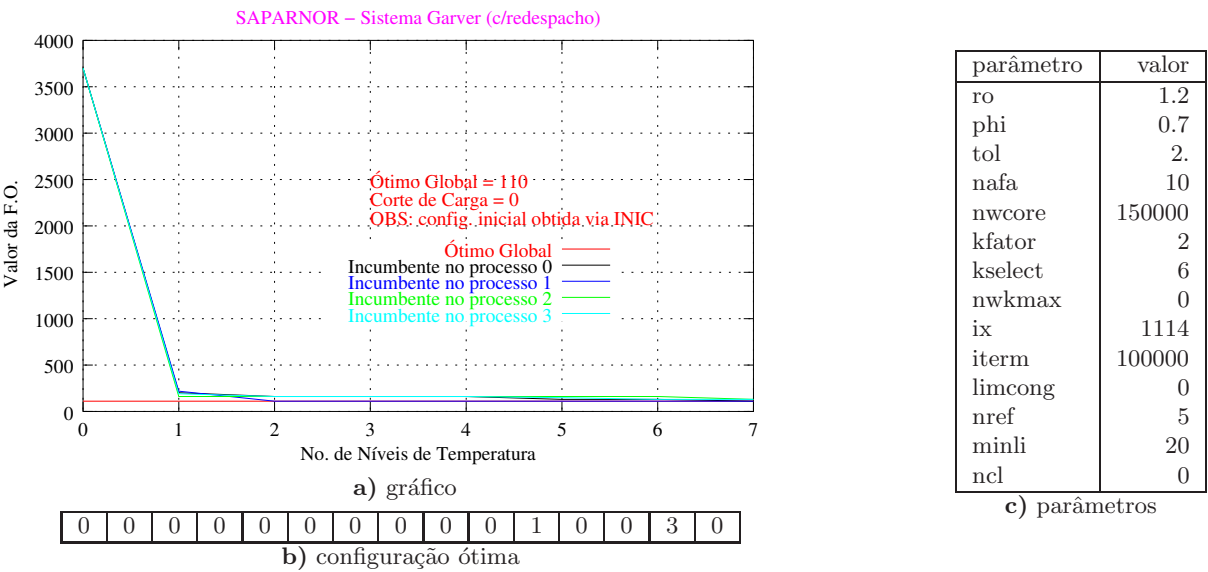
Tabela 6.5: Sistema Garver - algoritmo SAPARNOR (sem inicialização)



parâmetro	valor
ro	1.2
phi	0.6
tol	2.
nafa	10
nwcore	150000
kfator	3
kselect	6
nwkmax	0
ix	1114
iterm	100000
limcong	0
nref	5
minli	20
ncl	0

c) parâmetros

Tabela 6.6: Sistema Garver - algoritmo SAPARNOR (com inicialização)



6.2.1.4 Algoritmo GAPAR1B

Já para o algoritmo GAPAR1B, o uso de 8 configurações iniciais fez com que se obtivesse a solução ótima em apenas 2 gerações, com 100 PPLs em um tempo de 1,9 segundos (Tabela 6.8); enquanto que sem inicialização a solução ótima foi obtida em 11 gerações, com 1015 PPLs em um tempo de 20,5 segundos (Tabela 6.7).

Tabela 6.7: Sistema Garver - algoritmo GAPAR1B (sem inicialização)

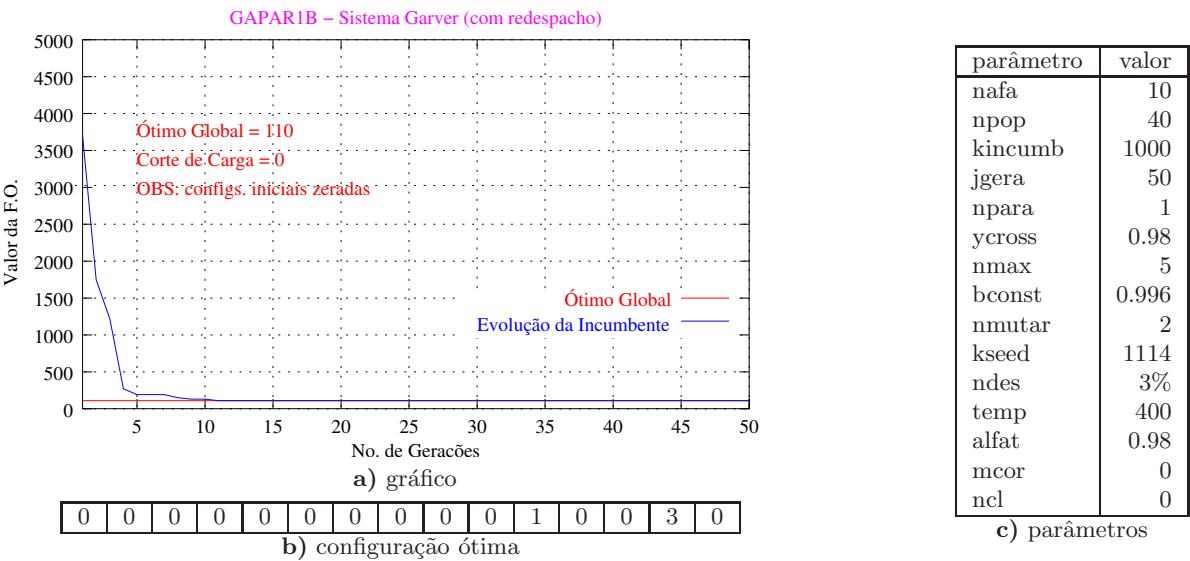
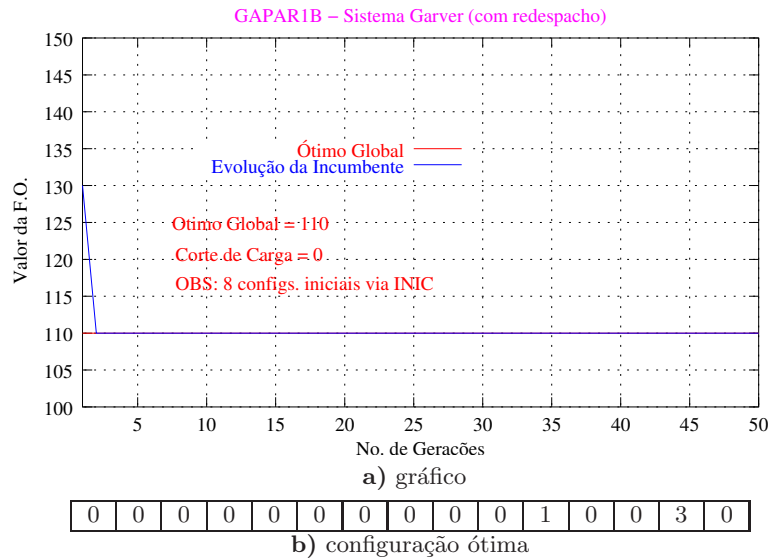


Tabela 6.8: Sistema Garver - algoritmo GAPAR1B (com inicialização)



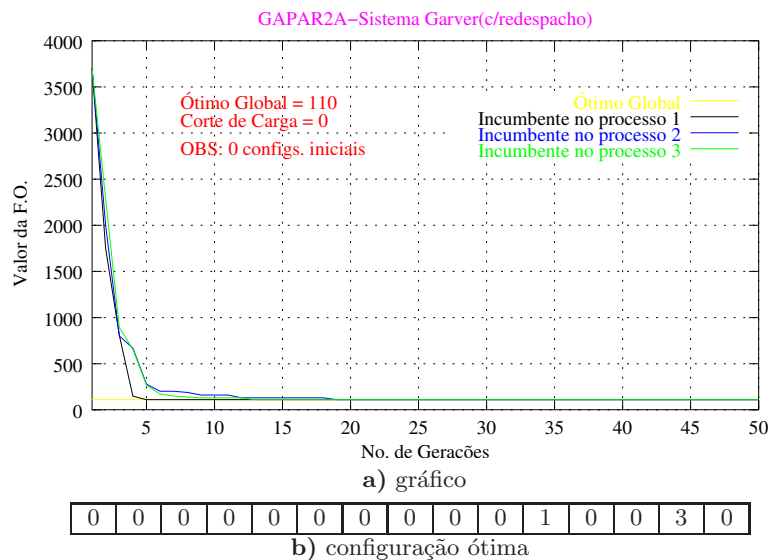
parâmetro	valor
nafa	10
npop	0
kincumb	1000
jgera	50
npara	8
ycross	0.98
nmax	5
bconst	0.996
nmutar	2
kseed	1114
ndes	3%
temp	400
alfat	0.98
mcor	0
ncl	0

c) parâmetros

6.2.1.5 Algoritmo GAPAR2A

Para o algoritmo GAPAR2A, o uso de 8 configurações iniciais, obteve-se a solução ótima em 2 gerações, com 99 PPLs em um tempo de 2,1 segundos (Tabela 6.10); enquanto que sem inicialização a solução ótima foi obtida em 5 gerações, com 394 PPLs em um tempo de 8,2 segundos (Tabela 6.9).

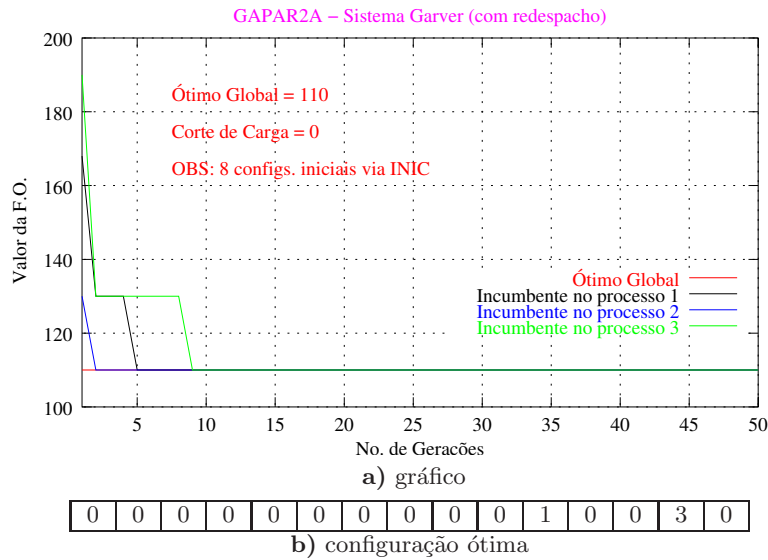
Tabela 6.9: Sistema Garver - algoritmo GAPAR2A (sem inicialização)



parâmetro	valor
nafa	10
npop	120
kincumb	4000
jgera	50
npara	1
ycross	0.96
nmax	4
bconst	0.996
nmutar	2
kseed	1112
ndes	8%
temp	250
alfat	0.98
mcor	0
ncl	0

c) parâmetros

Tabela 6.10: Sistema Garver - algoritmo GAPAR2A (com inicialização)



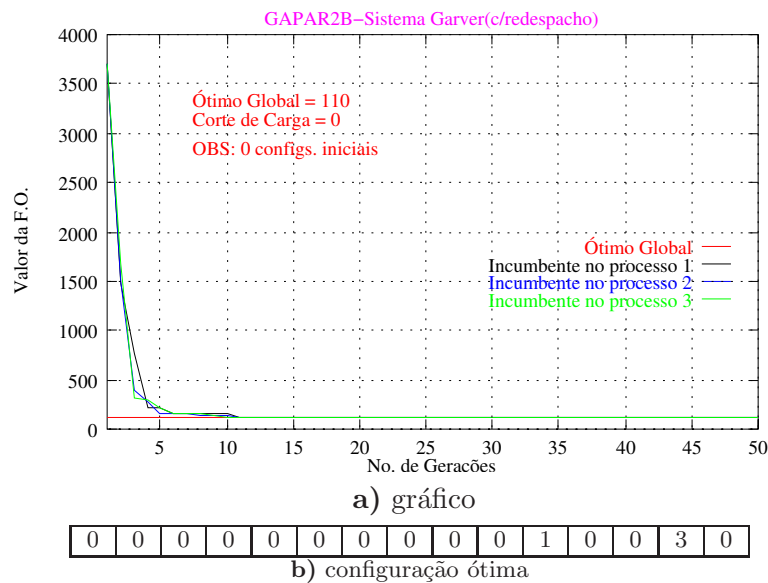
parâmetro	valor
nafa	10
npop	120
kincumb	1000
jgera	50
npara	8
ycross	0.96
nmax	4
bconst	0.996
nmutar	2
kseed	1112
ndes	5%
temp	250
alfat	0.98
mcor	0
ncl	0

c) parâmetros

6.2.1.6 Algoritmo GAPAR2B

Para o algoritmo GAPAR2B, sem inicialização, a solução ótima foi obtida em 7 gerações, com 679 PPLs em um tempo de 15,3 segundos (Tabela 6.11). Já com o uso de 8 configurações iniciais, obteve-se a solução ótima em 3 gerações, com 275 PPLs em um tempo de 5,7 segundos (Tabela 6.12).

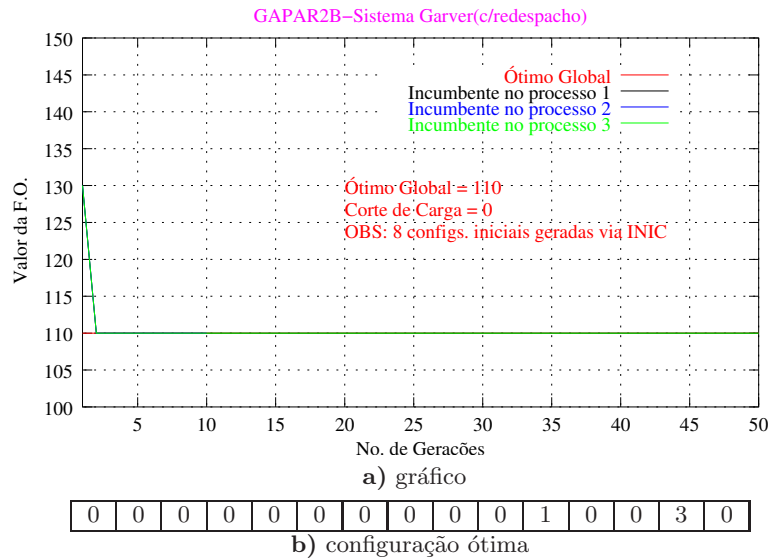
Tabela 6.11: Sistema Garver - algoritmo GAPAR2B (sem inicialização)



parâmetro	valor
nafa	10
npop	60
kincumb	4000
jgera	50
npara	1
ycross	0.96
nmax	4
bconst	0.996
nmutar	4
kseed	1112
ndes	5%
temp	250
alfat	0.98
mcor	0
kk1	5
ncl	0

c) parâmetros

Tabela 6.12: Sistema Garver - algoritmo GAPAR2B (com inicialização)



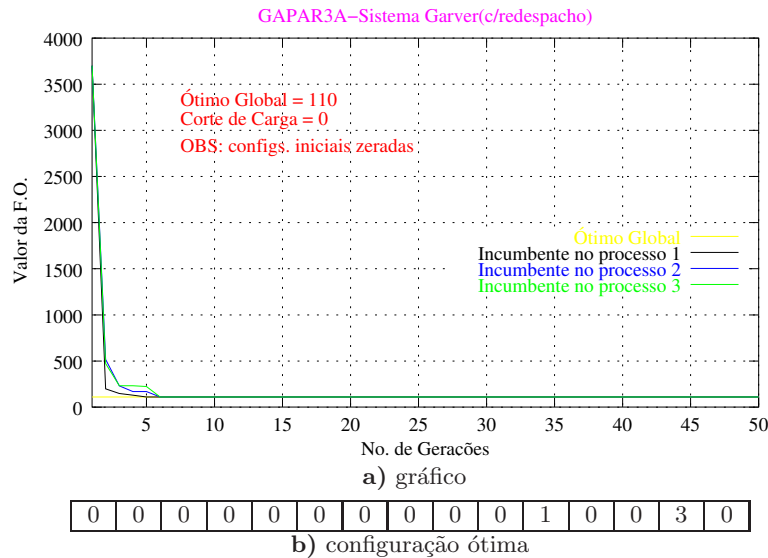
parâmetro	valor
nafa	10
npop	120
kincumb	4000
jgera	10
npara	8
ycross	0.96
nmax	4
bconst	0.996
nmutar	4
kseed	1112
ndes	5%
temp	250
alfat	0.98
mcor	0
kk1	5
ncl	0

c) parâmetros

6.2.1.7 Algoritmo GAPAR3A

Para o algoritmo GAPAR3A, sem inicialização, a solução ótima foi obtida em 5 gerações, com 903 PPLs em um tempo de 19,4 segundos (Tabela 6.13). Já com o uso de 8 configurações iniciais, obteve-se a solução ótima em 2 gerações, com 40 PPLs em um tempo de 0,9 segundos (Tabela 6.14).

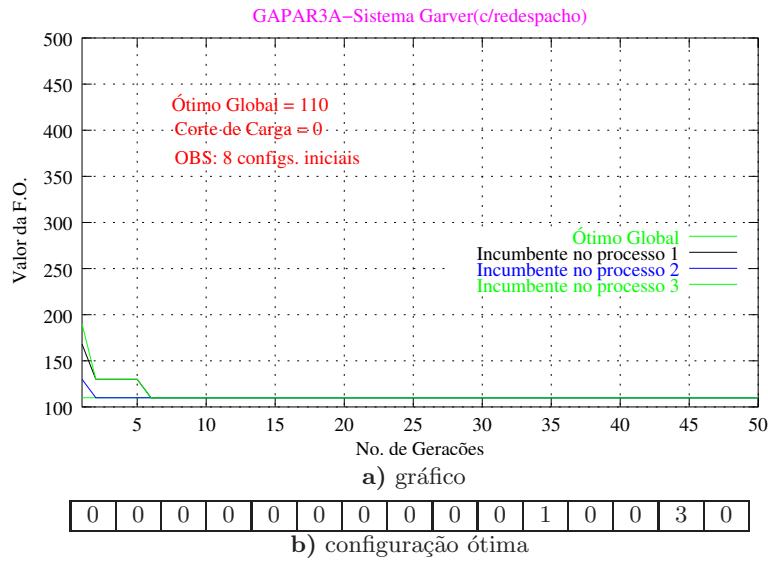
Tabela 6.13: Sistema Garver - algoritmo GAPAR3A (sem inicialização)



parâmetro	valor
nafa	10
npop	120
kincumb	6000
jgera	50
npara	1
ycross	0.98
nmax	4
bconst	0.996
nmutar	8
kseed	1114
ndes	7%
temp	270
alfat	0.98
mcor	0
kk1	5
ncl	0

c) parâmetros

Tabela 6.14: Sistema Garver - algoritmo GAPAR3A (com inicialização)



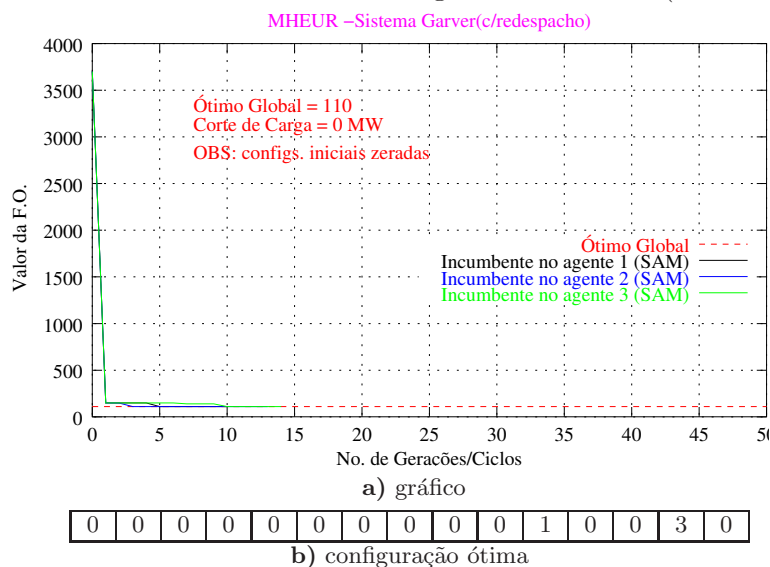
parâmetro	valor
nafa	10
npop	60
kincumb	6000
jgera	50
npara	8
ycross	0.98
nmax	4
bconst	0.996
nmutar	1
kseed	1114
ndes	7%
temp	270
alfat	0.98
mcor	0
kk1	5
ncl	0

c) parâmetros

6.2.1.8 Algoritmo MHEUR

Para o algoritmo MHEUR, sem inicialização, a solução ótima foi obtida em 3 gerações, com 170 PPLs em um tempo de 3,2 segundos (Tabela 6.15), com 3 agentes SAM. Já com o uso de 1 configuração inicial, obteve-se a solução ótima em 2 gerações, com 115 PLs, com um tempo de 2,3 segundos (Tabela 6.16), também com três agentes SAM.

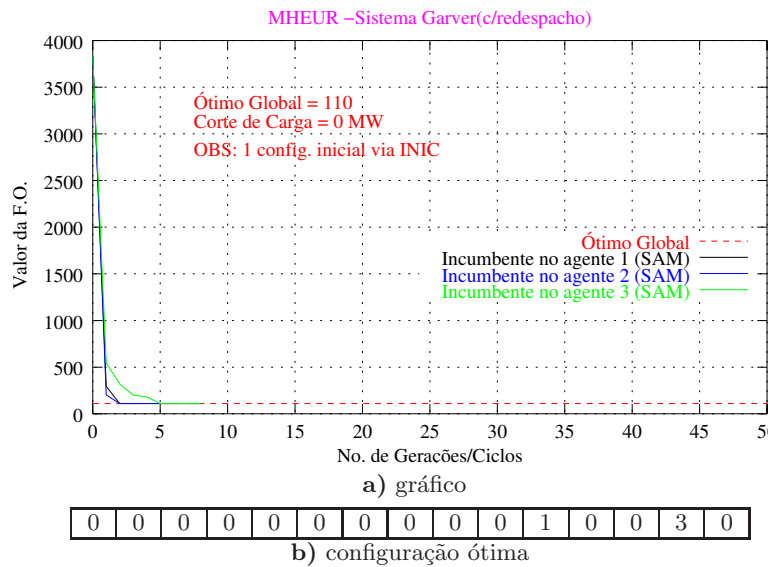
Tabela 6.15: Sistema Garver - algoritmo MHEUR (com 3 agentes SAM) - sem inicialização



parâmetro	valor
cadeia	1.2
temperat	0.7
nafa	10
ix	1114
iterm	10000
temp	300
nut	2
refcor	5.
limcong	2
ncl	0

c) parâmetros

Tabela 6.16: Sistema Garver - algoritmo MHEUR (com 3 agentes SAM) - com inicialização



parâmetro	valor
cadeia	1.2
temperat	0.7
nafa	10
ix	1114
item	1000
temp	100
nut	2
refcor	5.
limcong	2
ncl	0

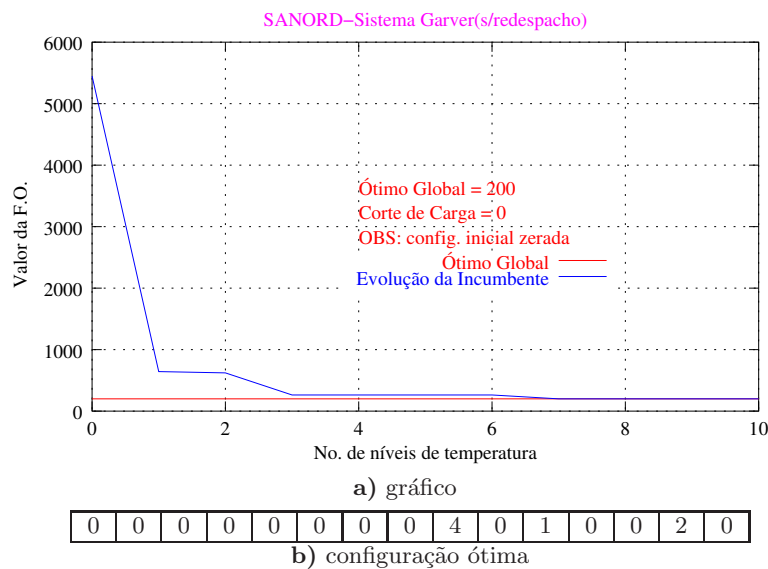
c) parâmetros

6.2.2 Sistema Garver (sem redespacho)

6.2.2.1 Algoritmo SANORD

Observa-se pelos gráficos das Tabelas 6.17 e 6.18 que, sem o uso do inicializador a solução ótima do problema foi obtida em 7 ciclos de temperatura, com 559 PPLs e tempo de 33,1 segundos; enquanto que com o uso de uma configuração inicial (obtida via algoritmo de inicialização INIC) obteve-se a solução em 4 ciclos, 208 PPLs e tempo de 11,8 segundos.

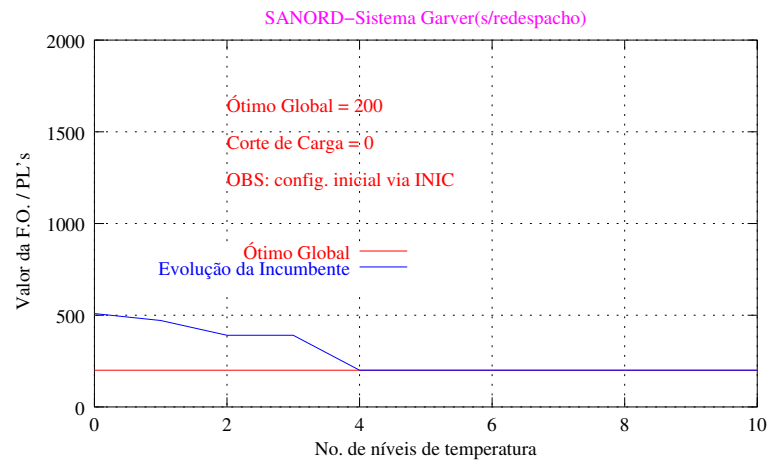
Tabela 6.17: Sistema Garver - algoritmo SANORD (sem inicialização)



parâmetro	valor
cadeia	1.2
temperat	0.7
nafa	10
ix	1114
item	2000
temp	300.
nut	2
refcor	0.
limcong	3
ncl	0

c) parâmetros

Tabela 6.18: Sistema Garver - algoritmo SANORD (com inicialização)



parâmetro	valor
cadeia	1.2
temperat	0.7
nafa	10
ix	1114
iterm	2000
temp	300.
nut	2
refcor	0.
limcong	3
ncl	0

c) parâmetros

a) gráfico

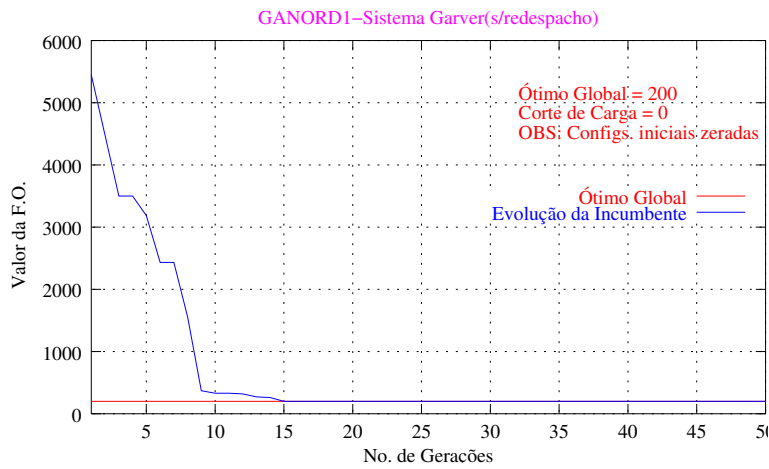
0	0	0	0	0	0	0	0	0	4	0	1	0	0	2	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

b) configuração ótima

6.2.2.2 Algoritmo GANORD

Nos gráficos das Tabelas 6.19 e 6.20 verifica-se a redução do número de gerações de quinze (588 PLs, 34,3 s) para seis (200 PLs, 12,5 s), com o uso de oito configurações iniciais em comparação à simulação sem nenhuma inicialização.

Tabela 6.19: Sistema Garver - algoritmo GANORD (sem inicialização)



parâmetro	valor
nafa	10
npop	20
kincumb	6000
jgera	50
npara	1
ycross	0.98
nmax	4
bconst	0.996
nmutar	1
kseed	1114
ndes	7%
temp	300
alfat	0.96
mcors	0
ncl	0

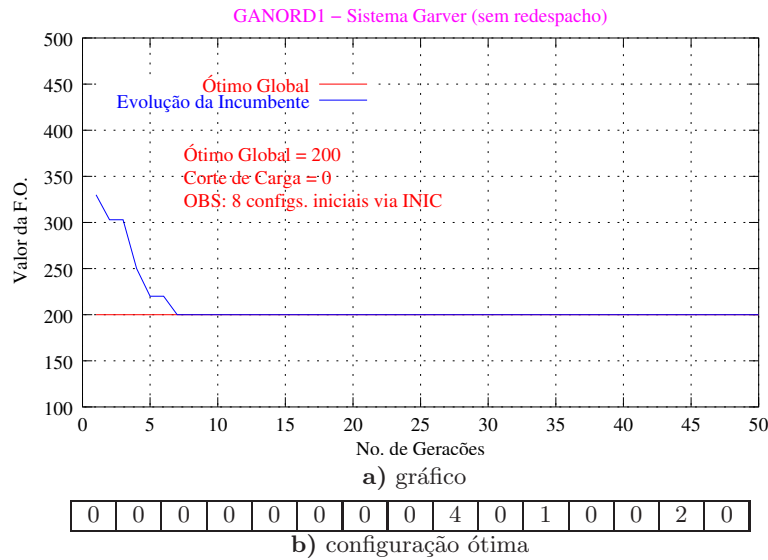
c) parâmetros

a) gráfico

0	0	0	0	0	0	0	0	0	4	0	1	0	0	2	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

b) configuração ótima

Tabela 6.20: Sistema Garver - algoritmo GANORD (com inicialização)



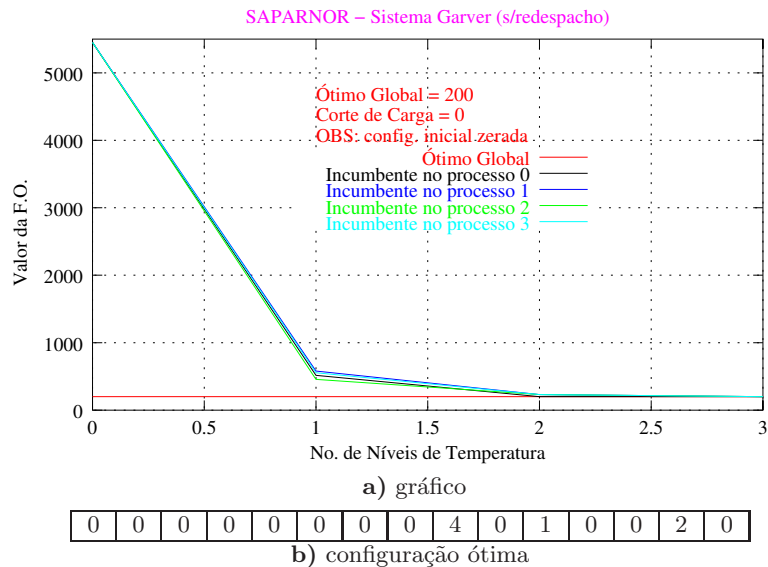
parâmetro	valor
nafa	10
npop	20
kincumb	6000
jgera	50
npara	8
ycross	0.98
nmax	4
bconst	0.996
nmutar	1
kseed	7323
ndes	7%
temp	270
alfat	0.98
mcors	0
ncl	0

c) parâmetros

6.2.2.3 Algoritmo SAPARNOR

Nas Tabelas 6.21 e 6.22, o uso do inicializador também não diminuiu o número de níveis de temperatura, mas obteve-se uma solução ótima em 43 PPLs e um tempo de 1,0 segundo, contra 67 PPLs e um tempo de 2,1 segundos para o caso sem inicialização.

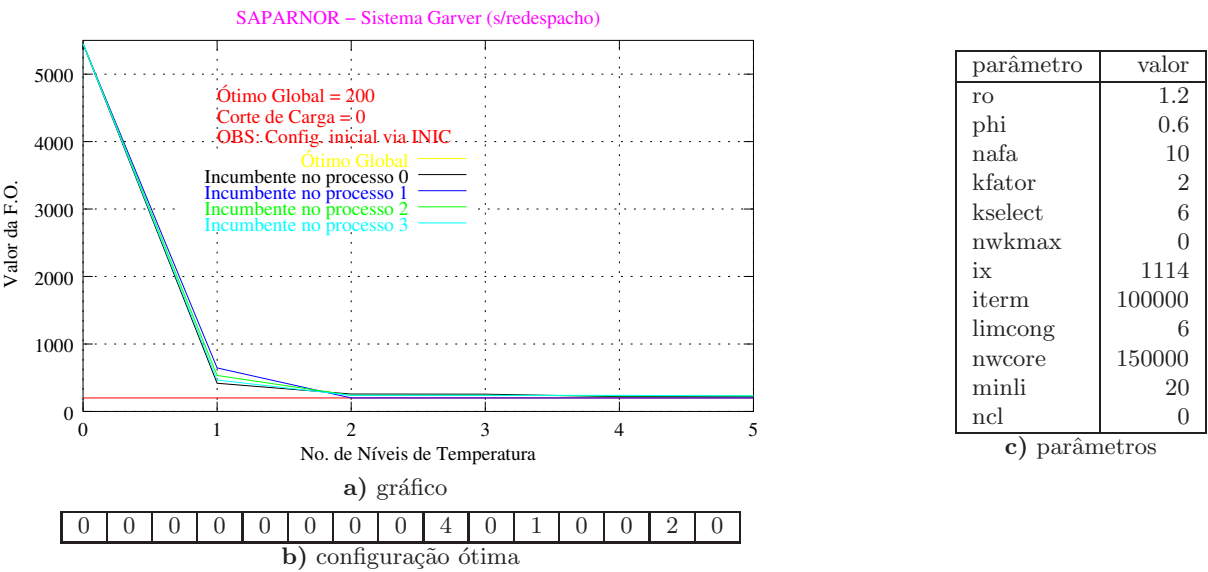
Tabela 6.21: Sistema Garver - algoritmo SAPARNOR (sem inicialização)



parâmetro	valor
ro	1.2
phi	0.7
nafa	10
kfator	2
kselect	6
nwkmax	0
ix	1114
iterm	100000
limcong	3
nwcore	150000
minli	20
ncl	0

c) parâmetros

Tabela 6.22: Sistema Garver - algoritmo SAPARNOR (com inicialização)



6.2.2.4 Algoritmo GAPAR1B

Já para o algoritmo GAPAR1B, o uso de 8 configurações iniciais, fez com que se obtivesse a solução ótima em 7 gerações, com 240 PPLs em um tempo de 4,1 segundos (Tabela 6.24); enquanto que sem inicialização a solução ótima foi obtida em 19 gerações, com 892 PPLs em um tempo de 17,7 segundos (Tabela 6.23).

Tabela 6.23: Sistema Garver - algoritmo GAPAR1B (sem inicialização)

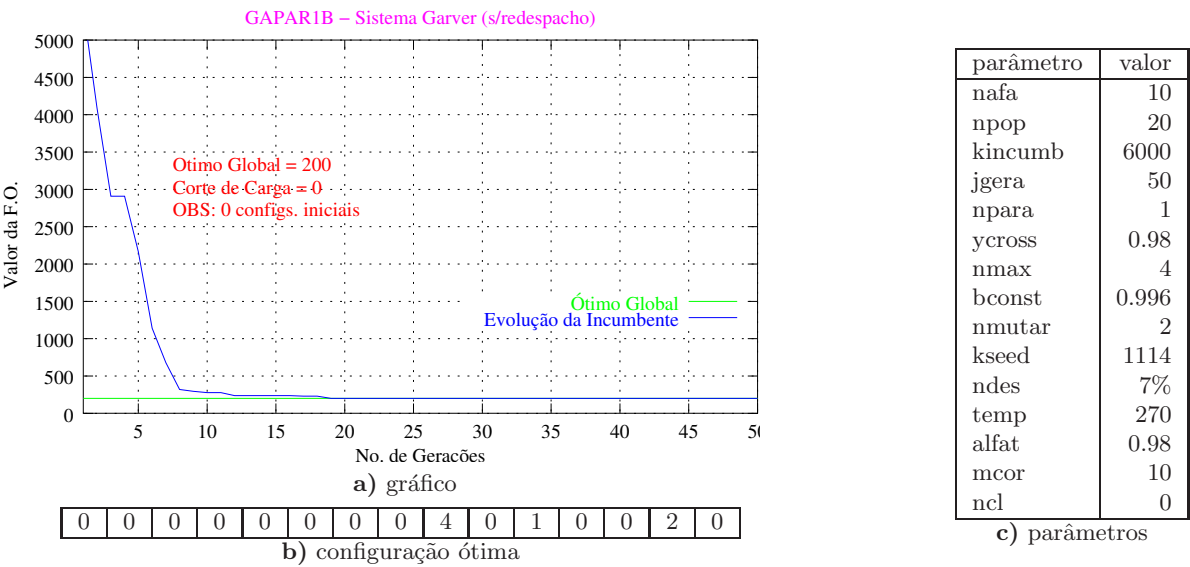
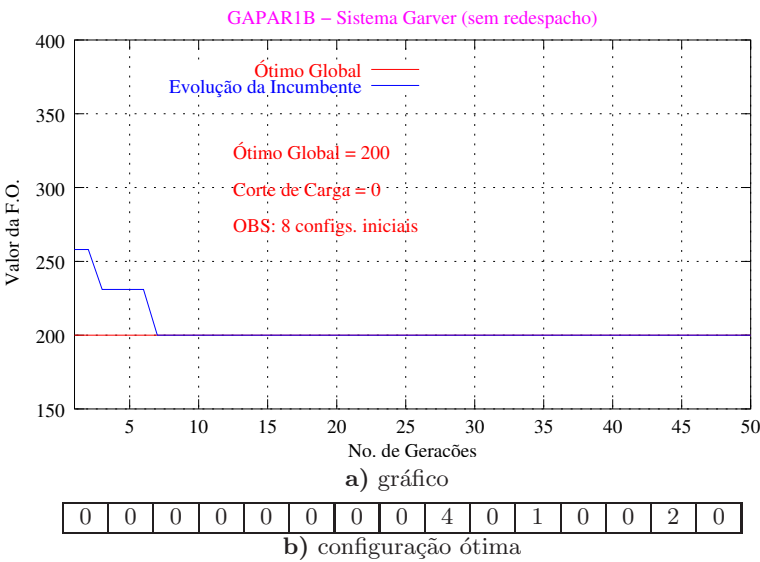


Tabela 6.24: Sistema Garver - algoritmo GAPAR1B (com inicialização)



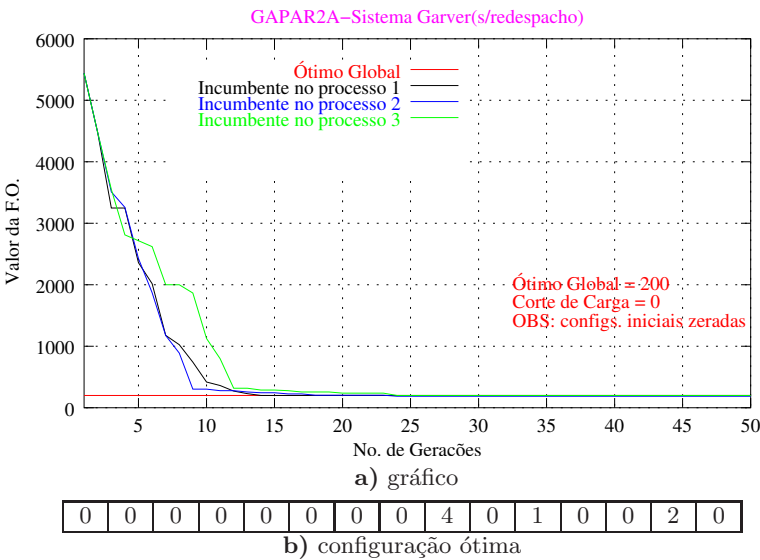
parâmetro	valor
nafa	10
npop	20
kincumb	6000
jgera	50
npara	8
ycross	0.98
nmax	4
bconst	0.996
nmutar	1
kseed	1114
ndes	7%
temp	270
alfat	0.98
mcors	0
ncl	0

c) parâmetros

6.2.2.5 Algoritmo GAPAR2A

Para o algoritmo GAPAR2A, o uso de 8 configurações iniciais, obteve-se a solução ótima em 5 gerações, com 320 PPLs em um tempo de 6,4 segundos (Tabela 6.26); enquanto que sem inicialização a solução ótima foi obtida em 14 gerações, com 1040 PPLs em um tempo de 21,2 segundos (Tabela 6.25).

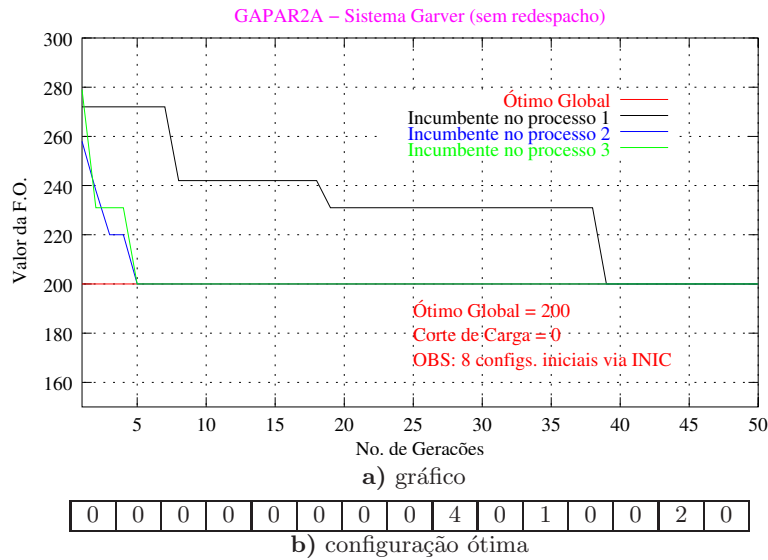
Tabela 6.25: Sistema Garver - algoritmo GAPAR2A (sem inicialização)



parâmetro	valor
nafa	10
npop	120
kincumb	6000
jgera	50
npara	1
ycross	0.98
nmax	4
bconst	0.996
nmutar	1
kseed	1114
ndes	7%
temp	270
alfat	0.98
mcors	0
ncl	0

c) parâmetros

Tabela 6.26: Sistema Garver - algoritmo GAPAR2A (com inicialização)



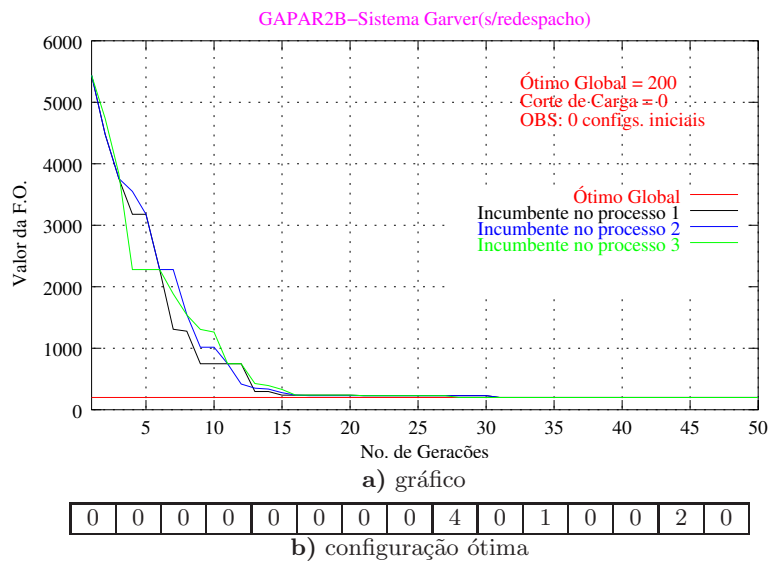
parâmetro	valor
nafa	10
npop	120
kincumb	6000
jgera	50
npara	8
ycross	0.98
nmax	4
bconst	0.996
nmutar	1
kseed	1114
ndes	7%
temp	270
alfat	0.98
mcor	0
ncl	0

c) parâmetros

6.2.2.6 Algoritmo GAPAR2B

Para o algoritmo GAPAR2B, sem inicialização, a solução ótima foi obtida em 28 gerações, com 876 PPLs em um tempo de 19,2 segundos (Tabela 6.27). Já com o uso de 8 configurações iniciais, obteve-se a solução ótima em 12 gerações, com 358 PPLs em um tempo de 8,1 segundos (Tabela 6.28).

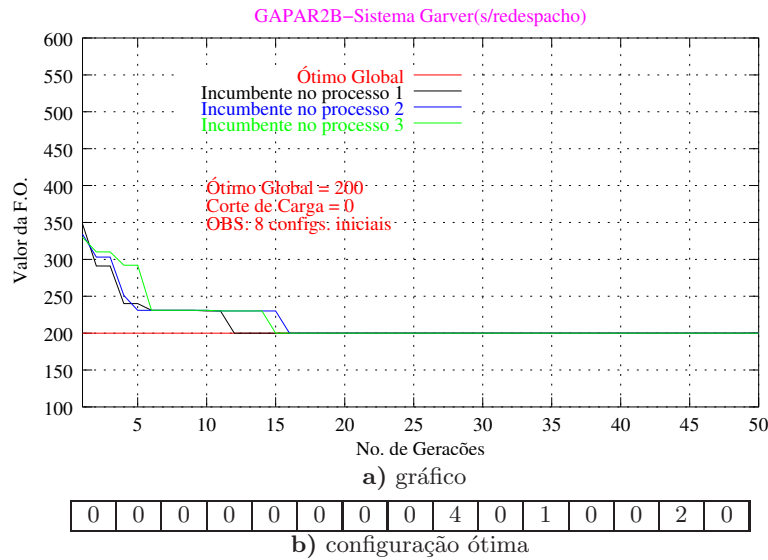
Tabela 6.27: Sistema Garver - algoritmo GAPAR2B (sem inicialização)



parâmetro	valor
nafa	10
npop	60
kincumb	6000
jgera	50
npara	1
ycross	0.98
nmax	4
bconst	0.996
nmutar	1
kseed	1114
ndes	7%
temp	270
alfat	0.98
mcor	0
kk1	5
ncl	0

c) parâmetros

Tabela 6.28: Sistema Garver - algoritmo GAPAR2B (com inicialização)



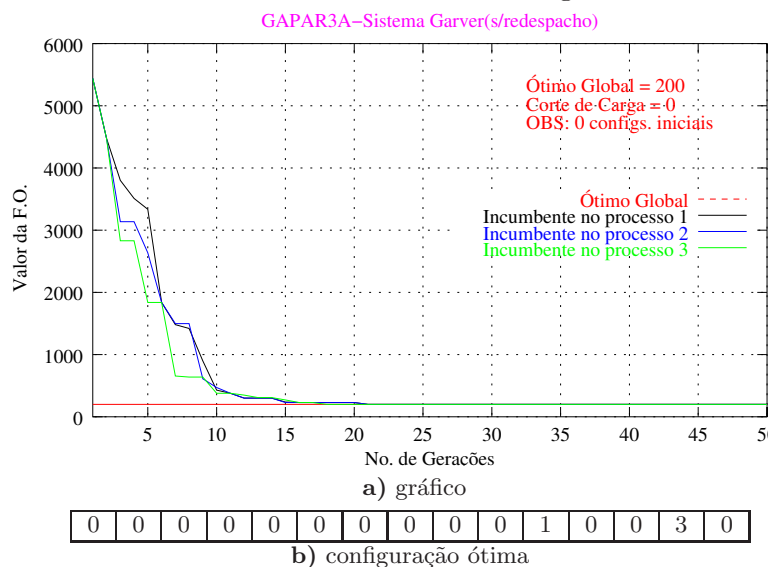
parâmetro	valor
nafa	10
npop	60
kincumb	6000
jgera	50
npara	8
ycross	0.98
nmax	4
bconst	0.996
nmutar	1
kseed	1114
ndes	7%
temp	270
alfat	0.98
mcor	0
kk1	5
ncl	0

c) parâmetros

6.2.2.7 Algoritmo GAPAR3A

Para o algoritmo GAPAR3A, sem inicialização, a solução ótima foi obtida em 18 gerações, com 840 PPLs em um tempo de 18,6 segundos (Tabela 6.29). Já com o uso de 8 configurações iniciais, obteve-se a solução ótima em 3 gerações, com 276 PPLs em um tempo de 6,7 segundos (Tabela 6.30).

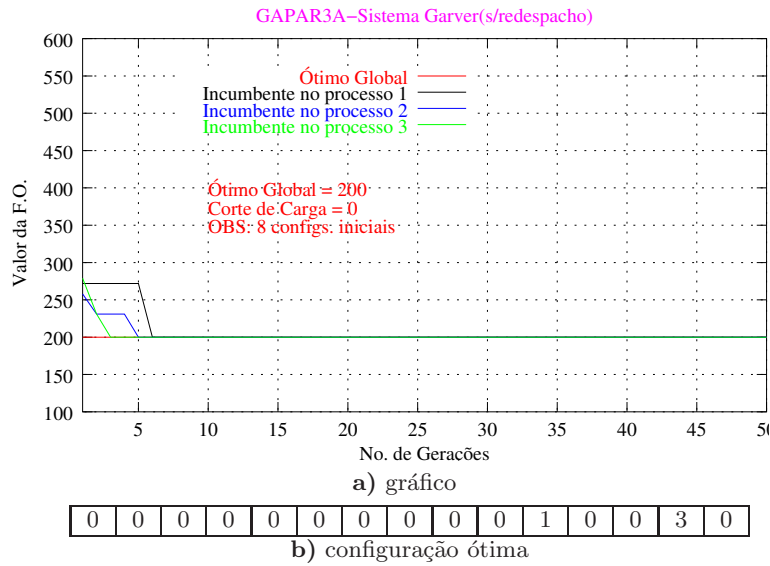
Tabela 6.29: Sistema Garver - algoritmo GAPAR3A (sem inicialização)



parâmetro	valor
nafa	10
npop	90
kincumb	6000
jgera	50
npara	1
ycross	0.98
nmax	4
bconst	0.996
nmutar	0
kseed	1114
ndes	7%
temp	270
alfat	0.98
mcor	0
kk1	5
ncl	0

c) parâmetros

Tabela 6.30: Sistema Garver - algoritmo GAPAR3A (com inicialização)



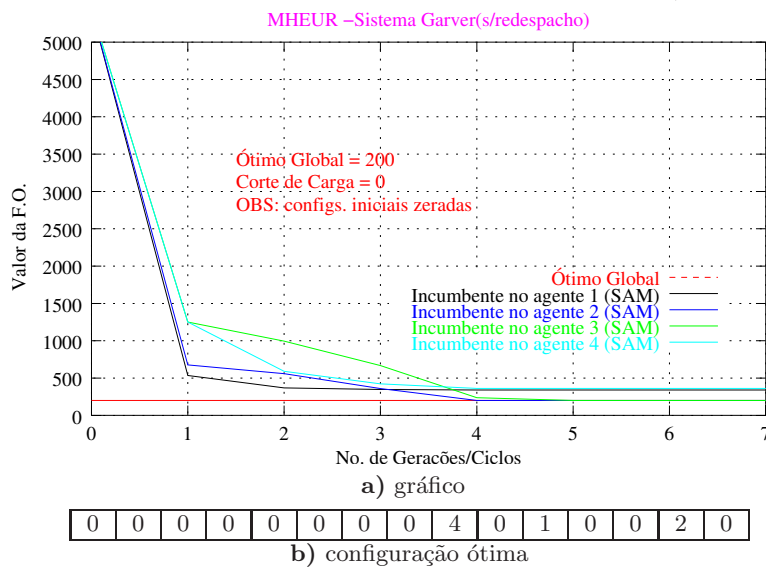
parâmetro	valor
nafa	10
npop	120
kincumb	4000
jgera	50
npara	8
ycross	0.96
nmax	4
bconst	0.996
nmutar	4
kseed	1112
ndes	5%
temp	250
alfat	0.98
mcor	0
kk1	5
ncl	0

c) parâmetros

6.2.2.8 Algoritmo MHEUR

Para o algoritmo MHEUR, sem inicialização, com quatro agentes SAM, a solução ótima foi obtida em 4 gerações, com 270 PPLs em um tempo de 6,3 segundos (Tabela 6.31). Já com o uso de 8 configurações iniciais, obteve-se a solução ótima em 4 gerações, com 156 PLs, e um tempo de 3,3 segundos (Tabela 6.32). Nestes casos, o time assíncrono foi composto com quatro agentes SAM.

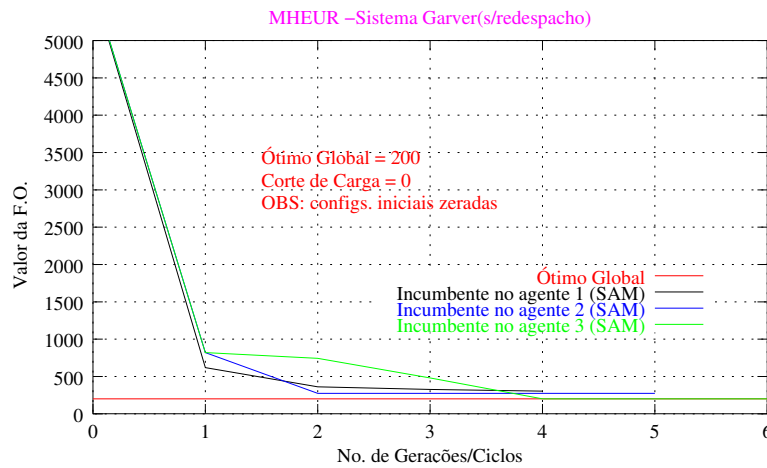
Tabela 6.31: Sistema Garver - algoritmo MHEUR (com 4 agentes SAM) - sem inicialização



parâmetro	valor
cadeia	1.2
temperat	0.6
nafa	10
ix	1114
iterm	1000
temp	100.
nut	2
refcor	0.
limcong	1
ncl	2

c) parâmetros

Tabela 6.32: Sistema Garver - algoritmo MHEUR (com 4 agentes SAM) - com inicialização



a) gráfico

0	0	0	0	0	0	0	0	0	4	0	1	0	0	2	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

b) configuração ótima

parâmetro	valor
cadeia	1.2
temperat	0.4
nafa	10
ix	1114
item	5000
temp	270.
nut	2
refcor	0.
limcong	6
ncl	0

c) parâmetros

6.3 Sistema Sul Brasileiro

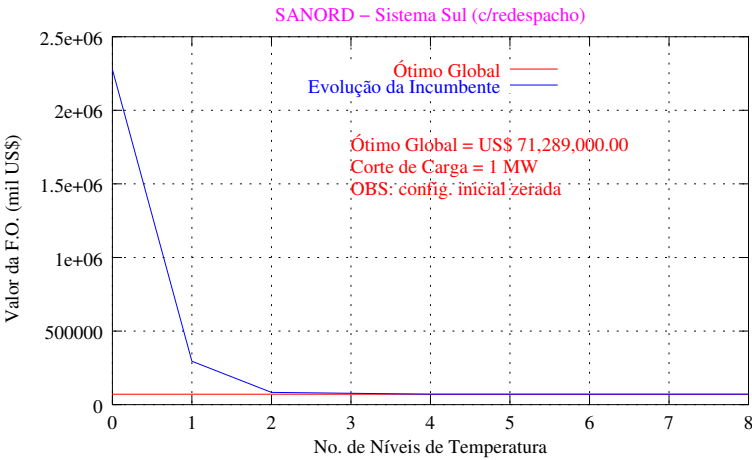
6.3.1 Sistema Sul Brasileiro (com redespacho)

Nas simulações efetuadas com este sistema e com os próximos, ou seja, sistema Norte-Nordeste brasileiro e sistema colombiano, os custos dos circuitos são considerados em milhares de dólares (US\$) e utiliza-se um $\alpha = 1000$; a unidade de α (nafa) está em milhares de US\$/MW. Assim, nos gráficos que se seguem, o valor apresentado representa o valor global da função objetivo (F.O.), isto é, o custo das linhas mais o custo do corte de carga.

6.3.1.1 Algoritmo SANORD

Observa-se pelos gráficos das Tabelas 6.33 e 6.34 que, sem o uso do inicializador, a solução ótima do problema foi obtida em 3 ciclos de temperatura, com 1617 PPLs e tempo de 5,1 minutos; enquanto que com o uso de uma configuração inicial (obtida via algoritmo de inicialização INIC) obteve-se a solução em 4 ciclos, mas com 1409 PPLs e tempo de 4,6 minutos.

Tabela 6.33: Sistema Sul Brasileiro - algoritmo SANORD (sem inicialização)



a) gráfico

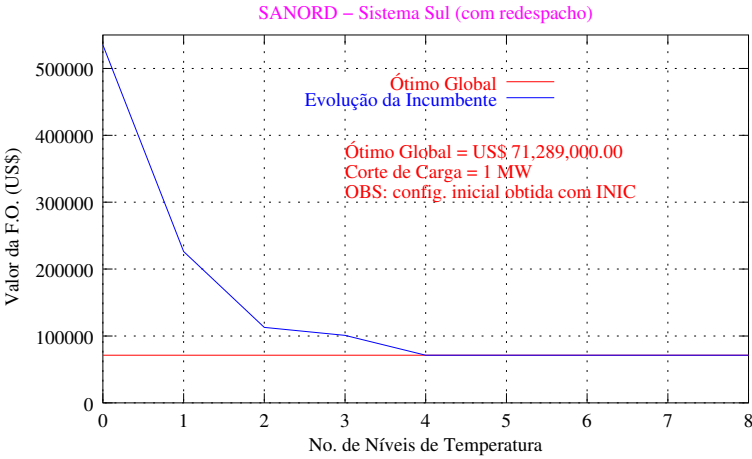
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	

b) configuração ótima

parâmetro	valor
cadeia	1.2
temperat	0.6
nafa	1000
ix	1114
item	10000
temp	50000.
nut	4
refcor	0.
limcong	6
ncl	10

c) parâmetros

Tabela 6.34: Sistema Sul Brasileiro - algoritmo SANORD (com inicialização)



a) gráfico

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

b) configuração ótima

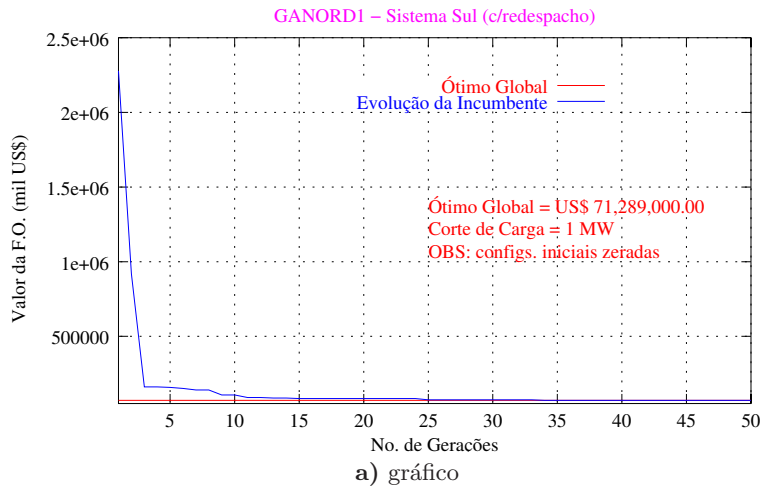
parâmetro	valor
cadeia	1.2
temperat	0.6
nafa	1000
ix	1114
item	5000
temp	50000.
nut	2
refcor	10.
limcong	6
ncl	20

c) parâmetros

6.3.1.2 Algoritmo GANORD

Nos gráficos das Tabelas 6.35 e 6.36 verifica-se a redução do número de gerações de 34 (6.410 PLs, 22,2 min) para 16 (2.556 PLs, 9,9 min), com o uso de vinte configurações iniciais em comparação à simulação sem nenhuma inicialização.

Tabela 6.35: Sistema Sul Brasileiro - algoritmo GANORD (sem inicialização)



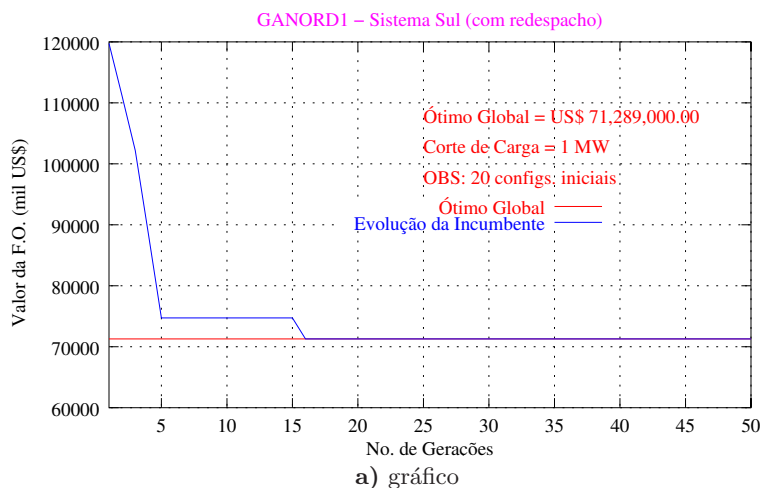
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	

b) configuração ótima

parâmetro	valor
nafa	1000
npop	40
kincumb	3000000
jgera	50
npara	1
ycross	0.90
nmax	4
bconst	0.996
nmutar	7
kseed	1114
ndes	3%
temp	21000
alfat	0.90
mcor	10
ncl	5

c) parâmetros

Tabela 6.36: Sistema Sul Brasileiro - algoritmo GANORD (com inicialização)



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

b) configuração ótima

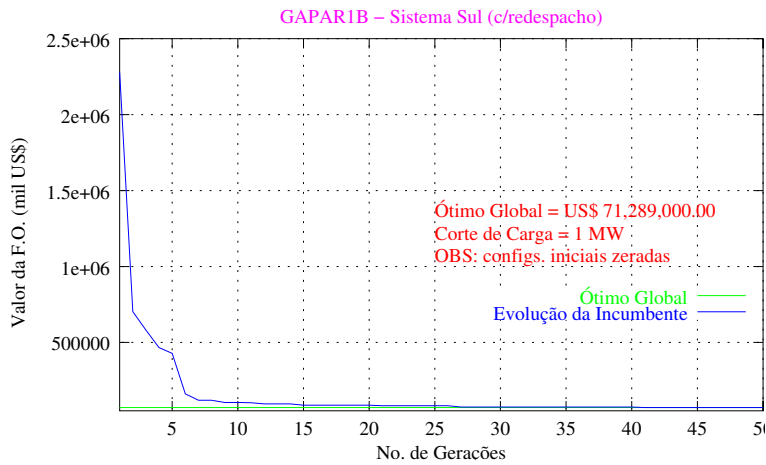
parâmetro	valor
nafa	1000
npop	90
kincumb	3000000
jgera	50
npara	20
ycross	0.90
nmax	4
bconst	0.996
nmutar	1
kseed	1114
ndes	3%
temp	21000
alfat	0.90
mcor	0
ncl	0

c) parâmetros

6.3.1.4 Algoritmo GAPAR1B

Já para o algoritmo GAPAR1B, o uso de 8 configurações iniciais, fez com que se obtivesse a solução ótima em 24 gerações, com 914 PPLs em um tempo de 1,2 minutos (Tabela 6.40); enquanto que sem inicialização a solução ótima foi obtida em 41 gerações, com 16.668 PPLs em um tempo de 20,5 minutos (Tabela 6.39).

Tabela 6.39: Sistema Sul Brasileiro - algoritmo GAPAR1B (sem inicialização)



a) gráfico

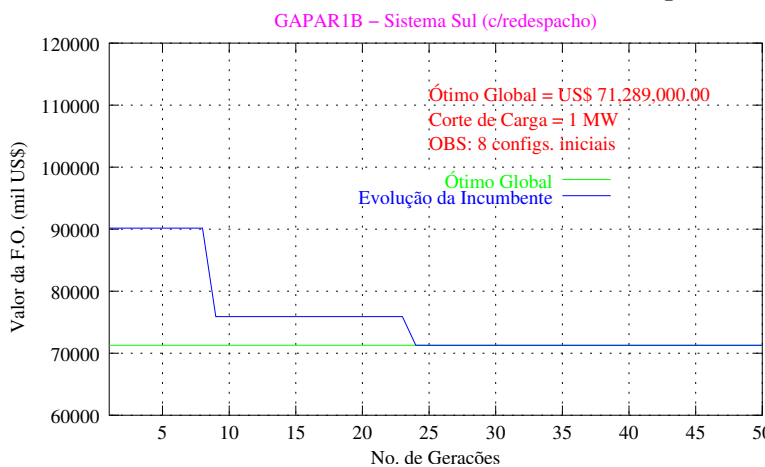
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	

b) configuração ótima

parâmetro	valor
nafa	1000
npop	120
kincumb	3000000
jgera	50
npara	1
ycross	0.98
nmax	4
bconst	0.996
nmutar	4
kseed	7326
ndes	5%
temp	21000
alfat	0.90
mcor	10
ncl	1

c) parâmetros

Tabela 6.40: Sistema Sul Brasileiro - algoritmo GAPAR1B (com inicialização)



a) gráfico

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

b) configuração ótima

parâmetro	valor
nafa	1000
npop	20
kincumb	300000
jgera	50
npara	8
ycross	0.99
nmax	8
bconst	0.996
nmutar	1
kseed	7326
ndes	8%
temp	40000
alfat	0.99
mcor	0
ncl	0

c) parâmetros

6.3.1.5 Algoritmo GAPAR2A

Para o algoritmo GAPAR2A, o uso de 20 configurações iniciais, obteve-se a solução ótima em 17 gerações, com 2.387 PPLs em um tempo de 4,6 minutos (Tabela 6.42); enquanto que sem inicialização a solução ótima foi obtida em 28 gerações, com 5.612 PPLs em um tempo de 11,2 minutos (Tabela 6.41).

Tabela 6.41: Sistema Sul Brasileiro - algoritmo GAPAR2A (sem inicialização)

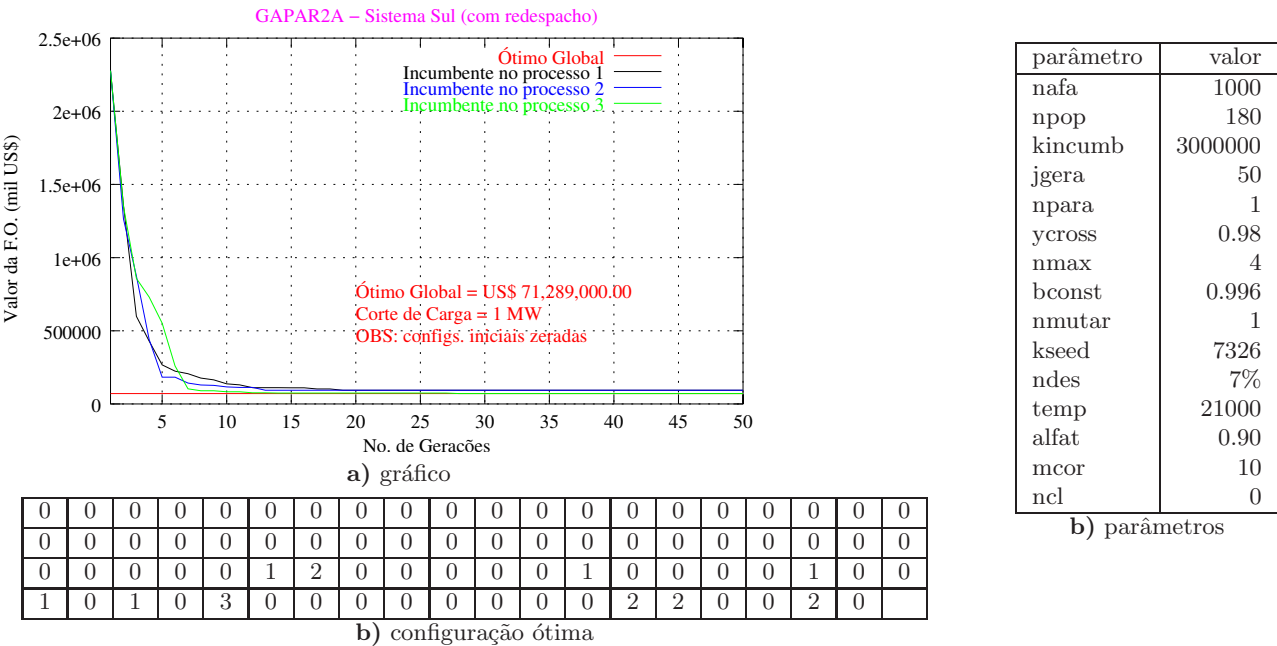
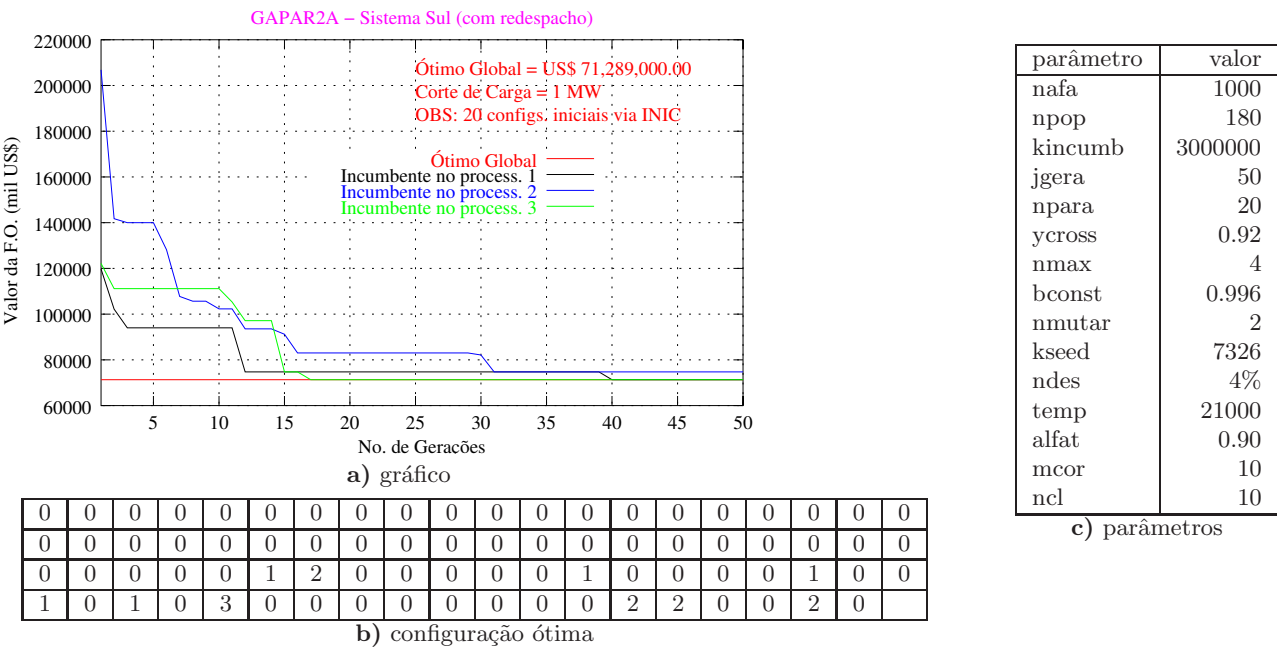


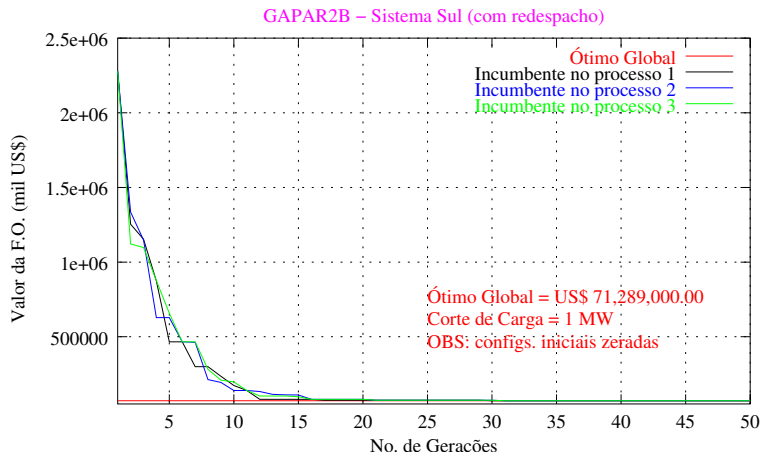
Tabela 6.42: Sistema Sul Brasileiro - algoritmo GAPAR2A (com inicialização)



6.3.1.6 Algoritmo GAPAR2B

Para o algoritmo GAPAR2B, sem inicialização, a solução ótima foi obtida em 30 gerações, com 2362 PPLs em um tempo de 7,7 minutos (Tabela 6.43). Já com o uso de 10 configurações iniciais, obteve-se a solução ótima em 19 gerações, com 1193 PPLs em um tempo de 2,3 minutos (Tabela 6.44).

Tabela 6.43: Sistema Sul Brasileiro - algoritmo GAPAR2B (sem inicialização)



a) gráfico

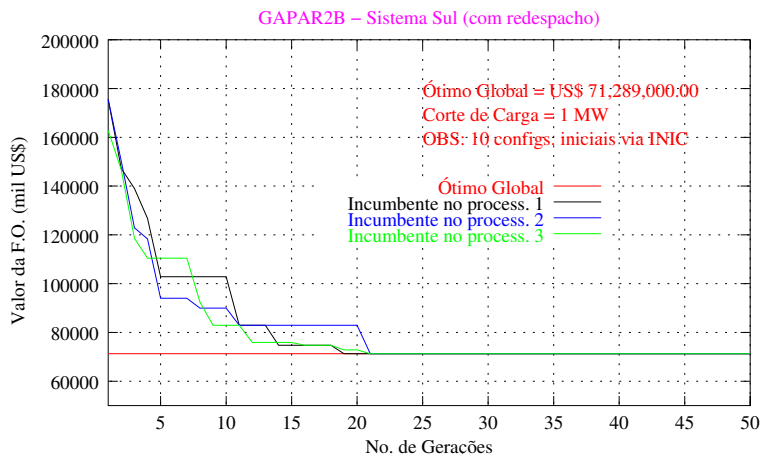
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	2	0	0	0	0	0	0	1	0	0	0	0	1
1	0	1	0	3	0	0	0	0	0	0	0	0	0	2	2	0	0	2	0

b) configuração ótima

parâmetro	valor
nafa	1000
npop	120
kincumb	2500000
jgera	50
npara	1
ycross	0.98
nmax	4
bconst	0.996
nmutar	4
kseed	1114
ndes	5%
temp	21000
alfat	0.98
mcor	0
kk1	5
ncl	10

c) parâmetros

Tabela 6.44: Sistema Sul Brasileiro - algoritmo GAPAR2B (com inicialização)



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

b) configuração ótima

parâmetro	valor
nafa	1000
npop	60
kincumb	3000000
jgera	50
npara	10
ycross	0.98
nmax	4
bconst	0.996
nmutar	4
kseed	7326
ndes	5%
temp	21000
alfat	0.90
mcor	10
kk1	10
ncl	1

c) parâmetros

6.3.1.7 Algoritmo GAPAR3A

Para o algoritmo GAPAR3A, sem inicialização, a solução ótima foi obtida em 26 gerações, com 4.289 PPLs em um tempo de 19,4 minutos (Tabela 6.45). Já com o uso de 20 configurações iniciais, obteve-se a solução ótima em 15 gerações, com 2.502 PPLs em um tempo de 5,4 minutos (Tabela 6.46).

Tabela 6.45: Sistema Sul Brasileiro - algoritmo GAPAR3A (sem inicialização)

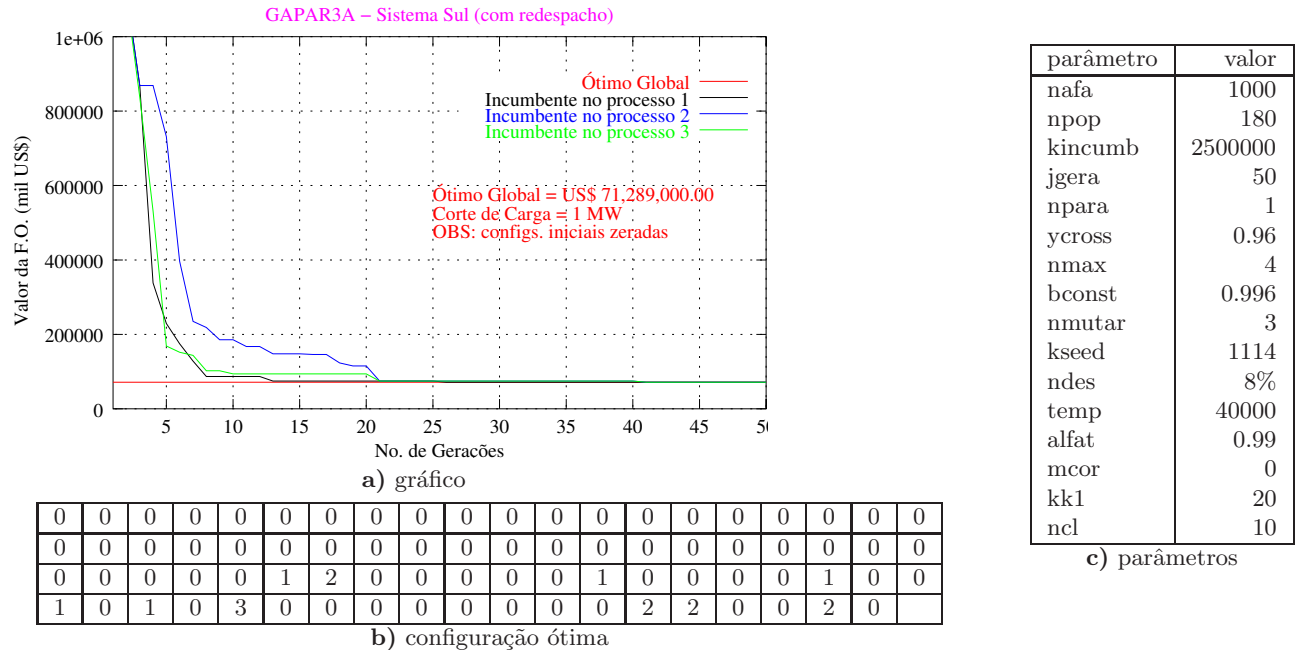
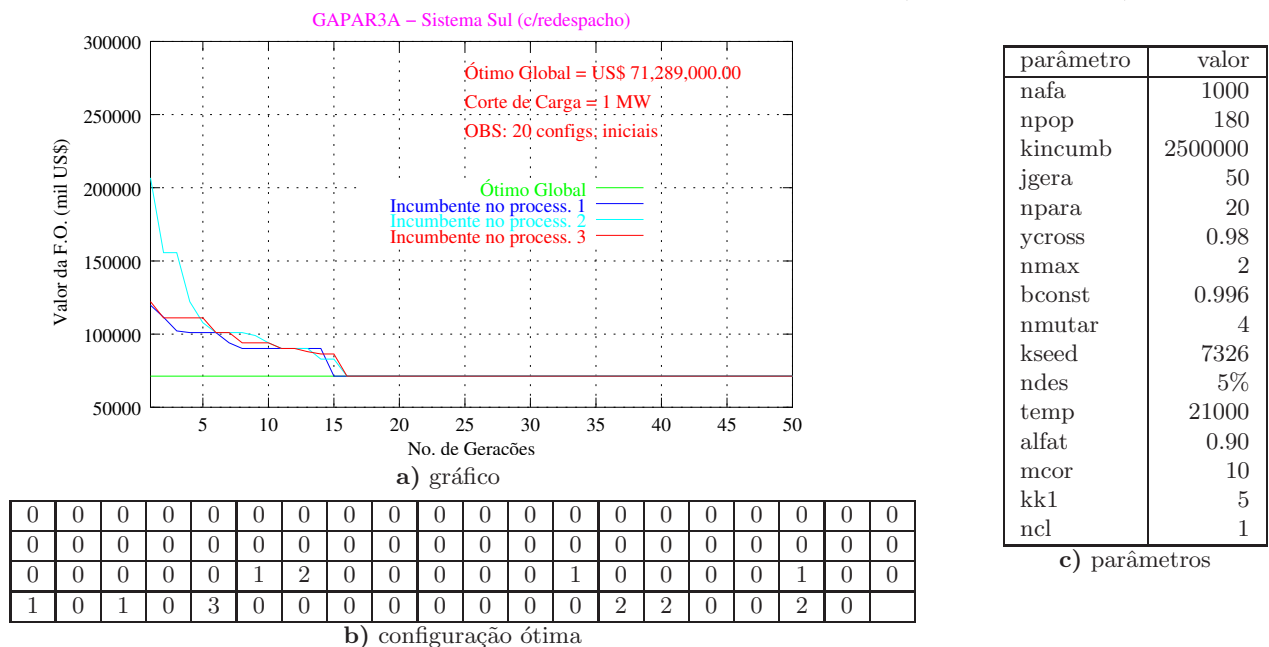


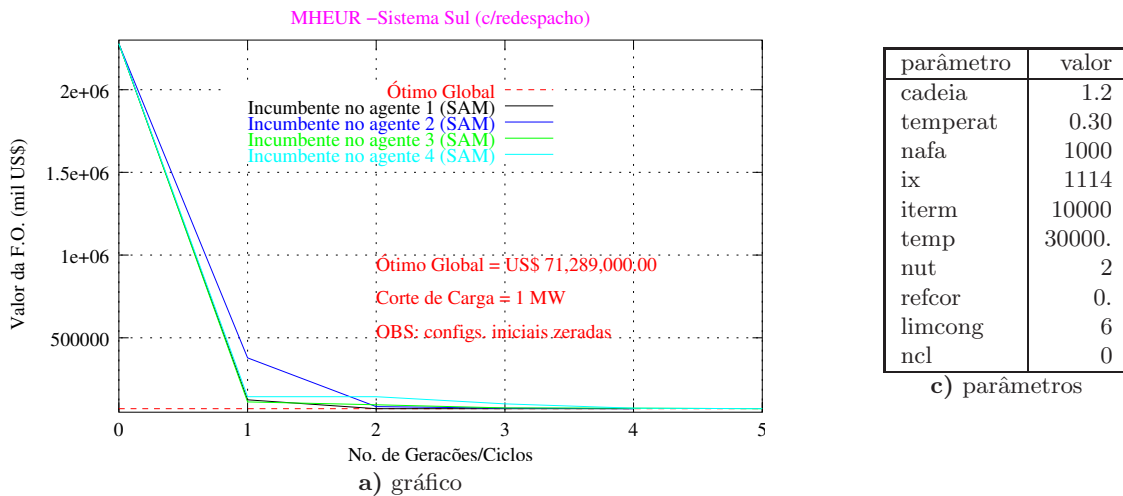
Tabela 6.46: Sistema Sul Brasileiro - algoritmo GAPAR3A (com inicialização)



6.3.1.8 Algoritmo MHEUR

Para o algoritmo MHEUR, sem inicialização a solução ótima foi obtida em 2 gerações, com 623 PPLs em um tempo de 1,2 minutos (Tabela 6.47). Já com o uso de 20 configurações iniciais, obteve-se a solução ótima em 6 ciclos, com 5626 PPLs, e um tempo de 11,4 minutos (Tabela 6.48). Nestes casos, o time assíncrono foi composto com quatro agentes SAM.

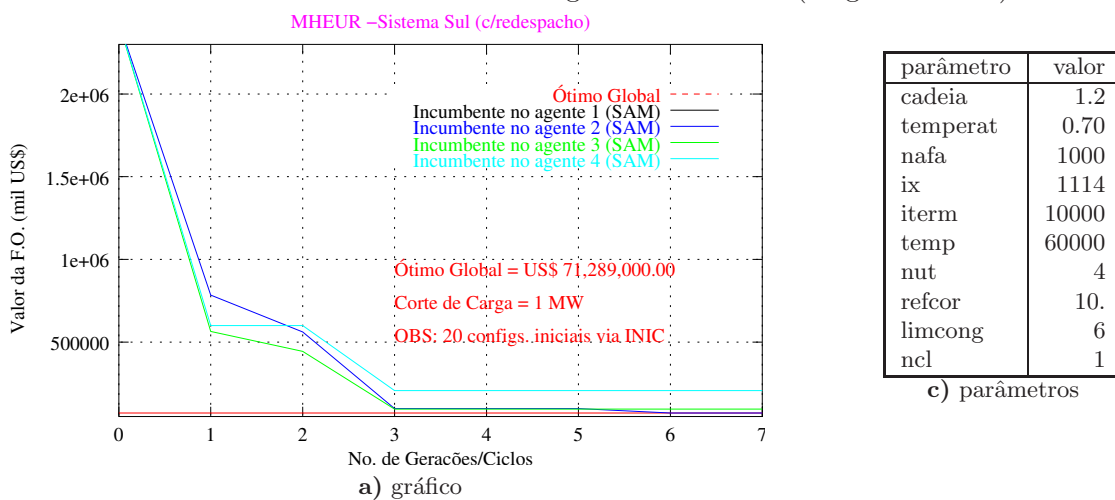
Tabela 6.47: Sistema Sul Brasileiro - algoritmo MHEUR (4 agentes SAM) - sem inicialização



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

b) configuração ótima

Tabela 6.48: Sistema Sul Brasileiro - algoritmo MHEUR (4 agentes SAM) - com inicialização



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

b) configuração ótima

6.3.2 Sistema Sul Brasileiro (sem redespacho)

6.3.2.1 Algoritmo SANORD

Observa-se pelos gráficos das Tabelas 6.49 e 6.50 que, sem o uso do inicializador, a solução ótima do problema foi obtida em 7 ciclos de temperatura, com 6071 PPLs e tempo de 27,8 minutos; enquanto que com o uso de uma configuração inicial (obtida via algoritmo de inicialização INIC) obteve-se a solução em 6 ciclos, mas com 4426 PPLs e tempo de 17,2 minutos.

Tabela 6.49: Sistema Sul Brasileiro - algoritmo SANORD (sem inicialização)

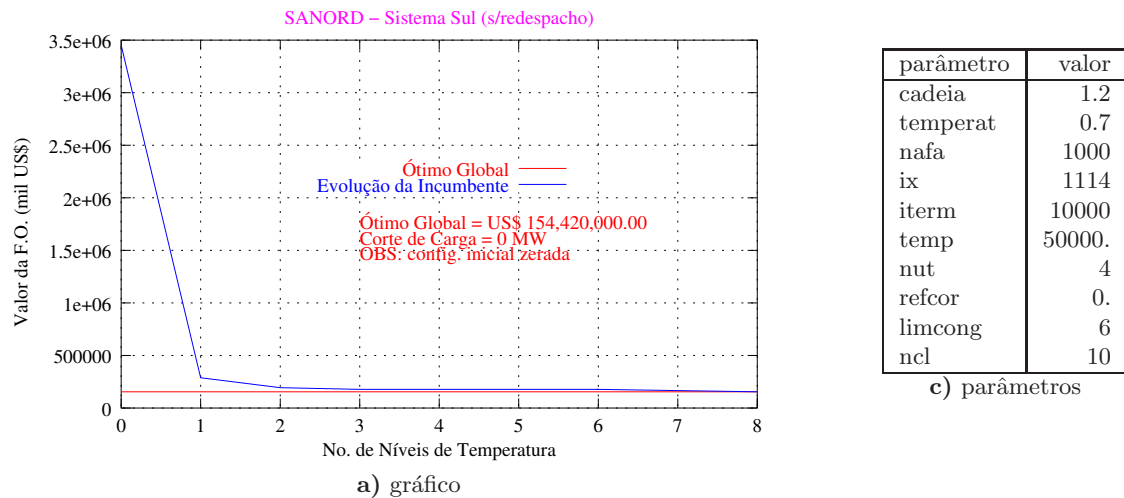
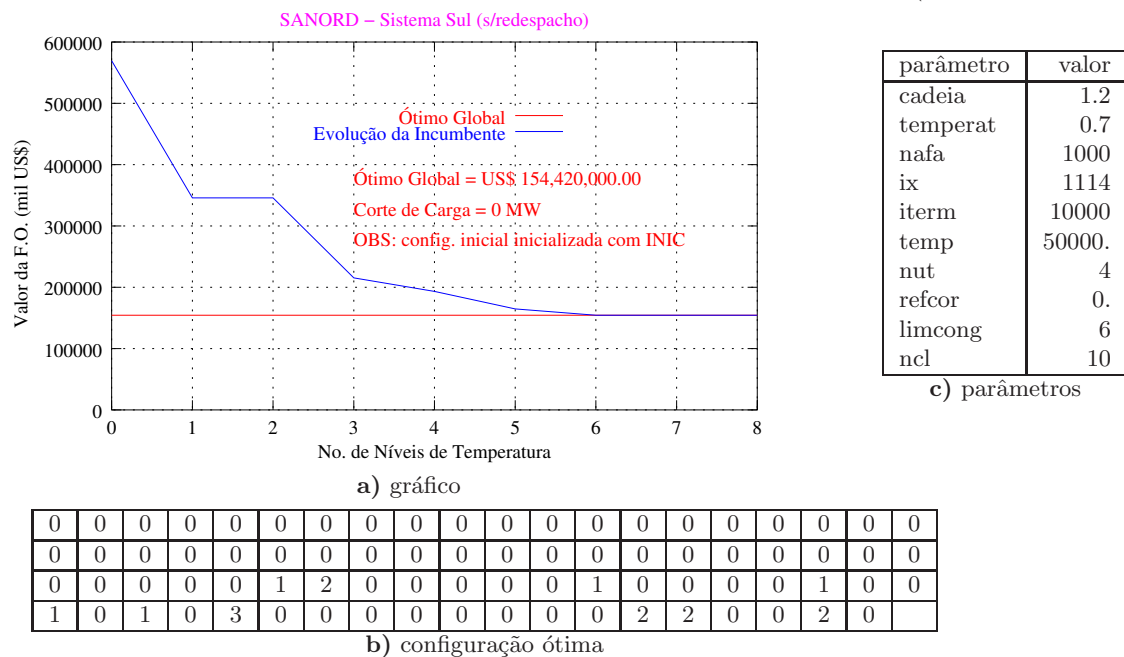


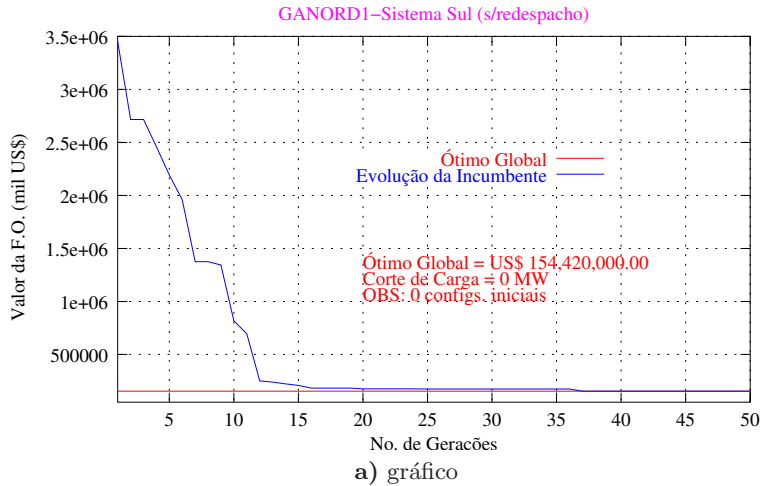
Tabela 6.50: Sistema Sul Brasileiro - algoritmo SANORD (com inicialização)



6.3.2.2 Algoritmo GANORD

Nos gráficos das Tabelas 6.51 e 6.52 verifica-se a redução do número de gerações de 37 (2.865 PPLs, 10,5 min) para 22 (3.295 PPLs, 12,2 min), com o uso de vinte configurações iniciais em comparação à simulação sem nenhuma inicialização.

Tabela 6.51: Sistema Sul Brasileiro - algoritmo GANORD (sem inicialização)



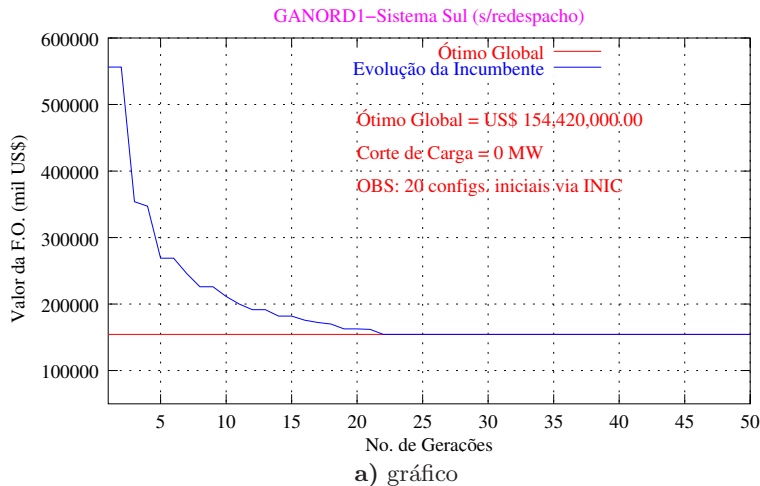
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0	0	0	0	0	1	0	0	0	0	1	0
1	0	1	0	3	0	0	0	0	0	0	0	0	2	2	0	0	2	0

b) configuração ótima

parâmetro	valor
nafa	1000
npop	40
kincomb	2500000
jgera	50
npara	1
ycross	0.90
nmax	4
bconst	0.996
nmutar	1
kseed	1211
ndes	5%
temp	20000
alfat	0.96
mcor	0
ncl	0

c) parâmetros

Tabela 6.52: Sistema Sul Brasileiro - algoritmo GANORD (com inicialização)



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

b) configuração ótima

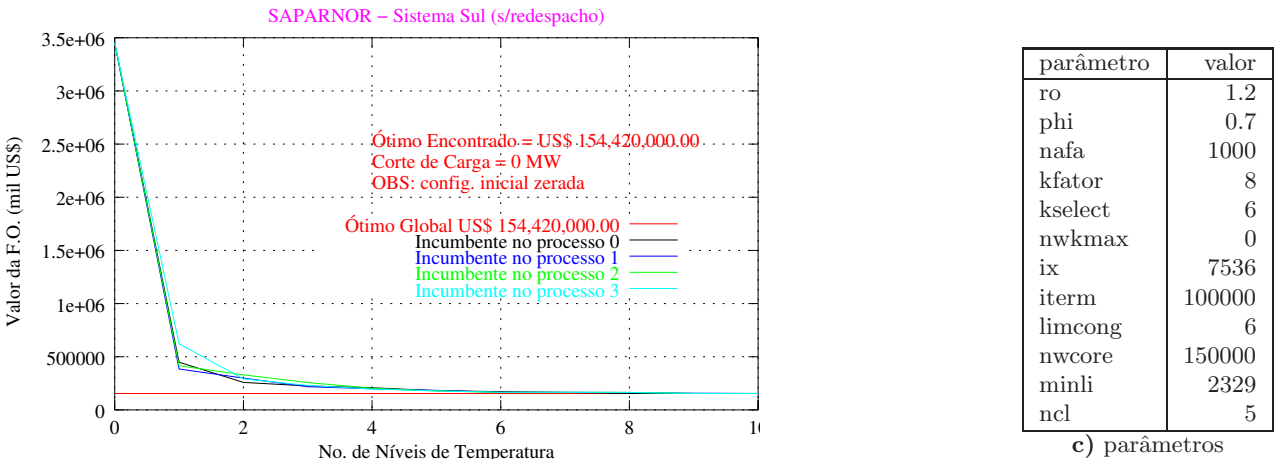
parâmetro	valor
nafa	1000
npop	40
kincomb	2500000
jgera	50
npara	20
ycross	0.90
nmax	4
bconst	0.996
nmutar	5
kseed	1211
ndes	5%
temp	20000
alfat	0.90
mcor	0
ncl	0

c) parâmetros

6.3.2.3 Algoritmo SAPARNOR

Nas Tabelas 6.53 e 6.54, vê-se que o uso do inicializador diminuiu o número de gerações para o algoritmo SAPARNOR, a solução ótima foi obtida em 6 ciclos com um tempo de 7,9 minutos e 3.488 PPLs, contra 8 ciclos com um tempo de 22,2 minutos e 10.428 PPLs para o caso sem inicialização.

Tabela 6.53: Sistema Sul Brasileiro - algoritmo SAPARNOR (sem inicialização)

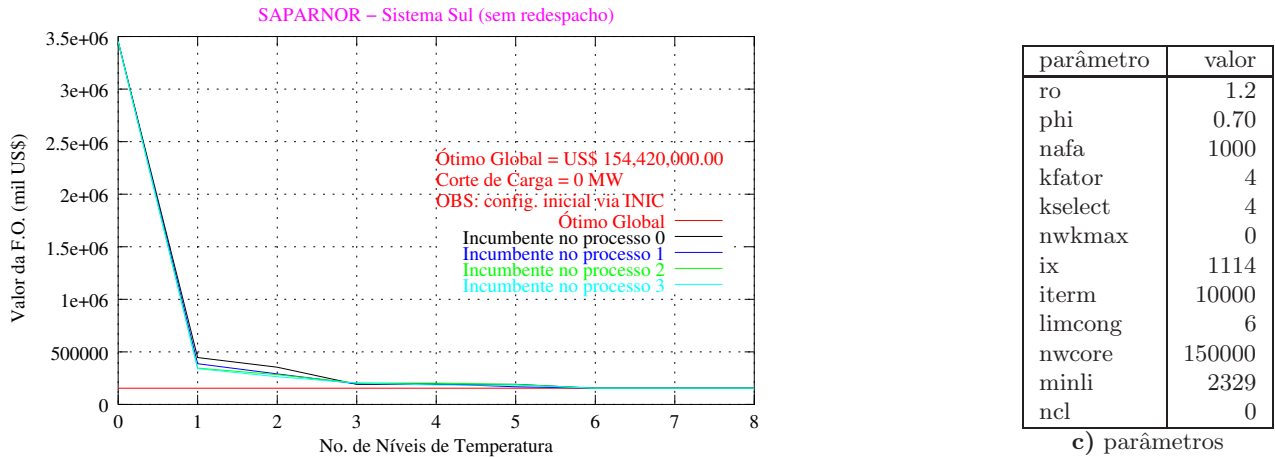


a) gráfico

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

Melhor Configuração Obtida

Tabela 6.54: Sistema Sul Brasileiro - algoritmo SAPARNOR (com inicialização)



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

b) configuração ótima

6.3.2.4 Algoritmo GAPAR1B

Já para o algoritmo GAPAR1B, o uso de 20 configurações iniciais, fez com que se obtivesse a solução ótima em 22 gerações, com 2.939 PPLs em um tempo de 6,5 minutos (Tabela 6.56); enquanto que sem inicialização a solução ótima foi obtida em 27 gerações, com 3687 PPLs em um tempo de 15,4 minutos (Tabela 6.55).

Tabela 6.55: Sistema Sul Brasileiro - algoritmo GAPAR1B (sem inicialização)

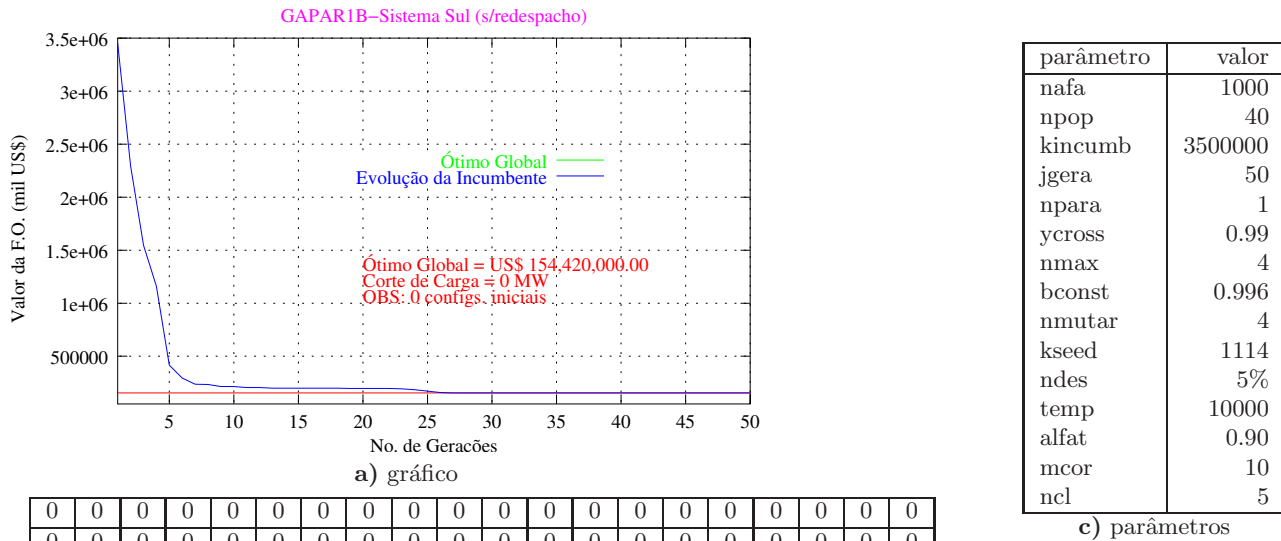
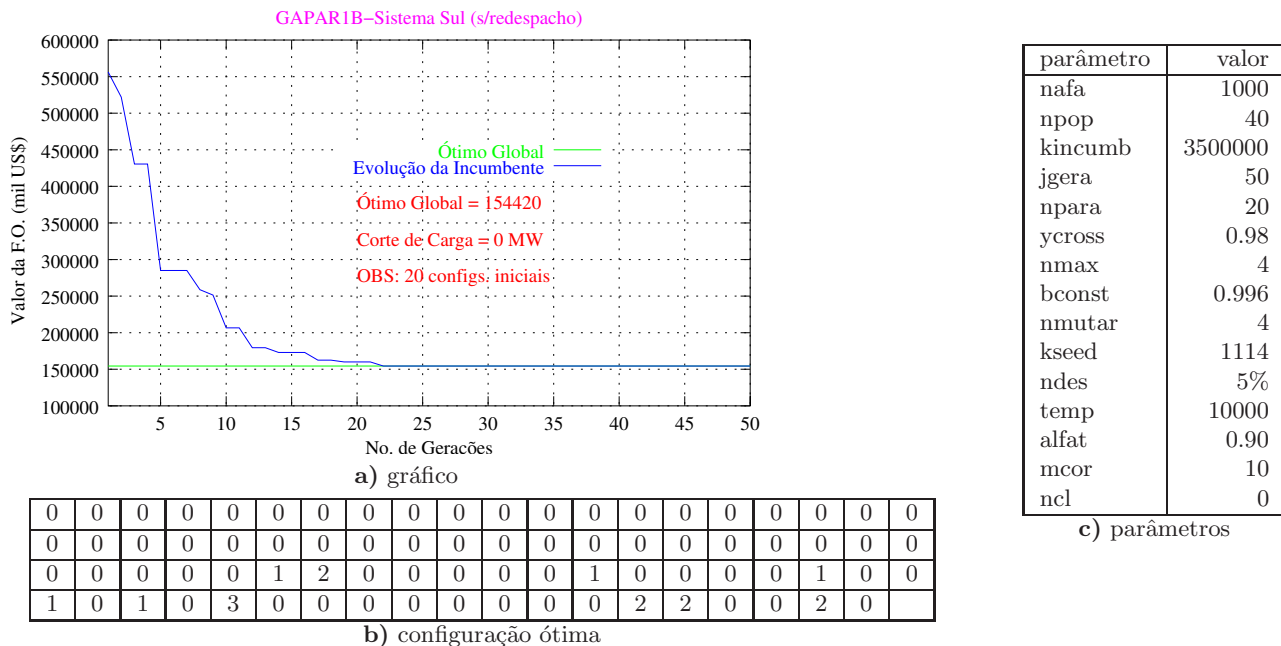


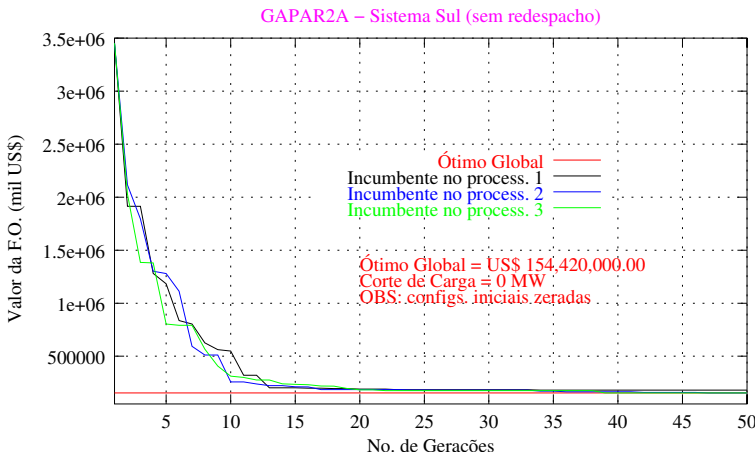
Tabela 6.56: Sistema Sul Brasileiro - algoritmo GAPAR1B (com inicialização)



6.3.2.5 Algoritmo GAPAR2A

Para o algoritmo GAPAR2A, o uso de 20 configurações iniciais, obteve-se a solução ótima em 20 gerações, com 1.520 PPLs em um tempo de 3,5 minutos (Tabela 6.58); enquanto que sem inicialização a solução ótima foi obtida em 39 gerações, com 5.394 PPLs em um tempo de 12,2 minutos (Tabela 6.57).

Tabela 6.57: Sistema Sul Brasileiro - algoritmo GAPAR2A (sem inicialização)



a) gráfico

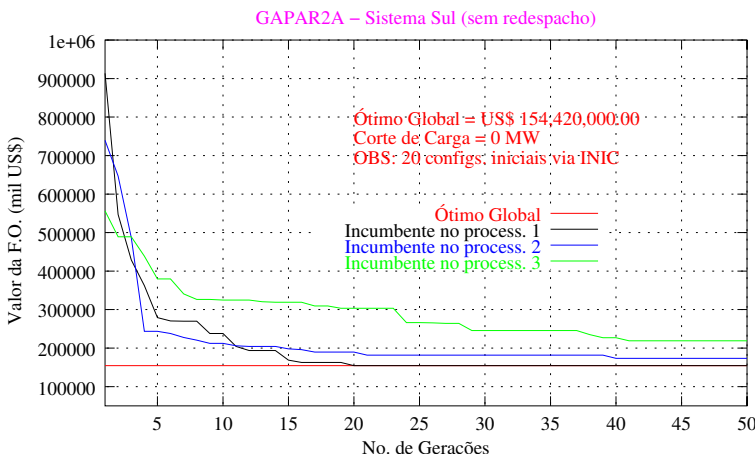
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0	0	0	0	0	0	1	0	0	0	0	1	0
1	0	1	0	3	0	0	0	0	0	0	0	0	0	2	2	0	0	2	0

b) configuração ótima

parâmetro	valor
nafa	1000
npop	120
kincumb	3500000
jgera	50
npara	1
ycross	0.98
nmax	4
bconst	0.996
nmutar	4
kseed	7326
ndes	5%
temp	21000.
alfat	0.90
mcor	10
ncl	0

c) parâmetros

Tabela 6.58: Sistema Sul Brasileiro - algoritmo GAPAR2A (com inicialização)



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

b) configuração ótima

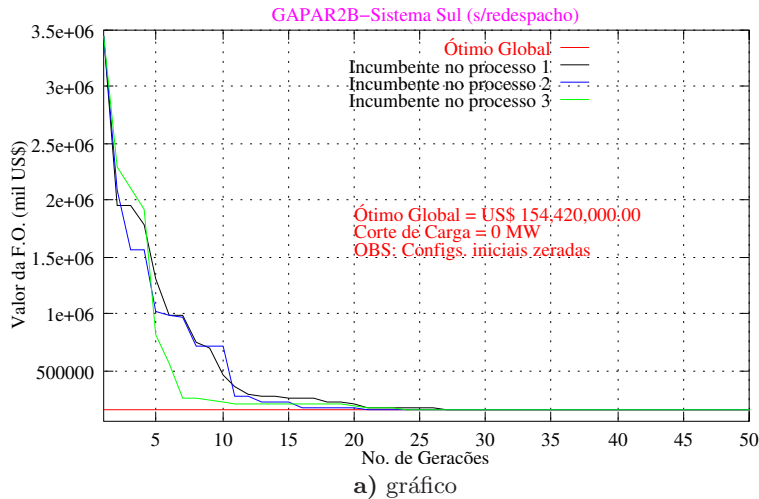
parâmetro	valor
nafa	1000
npop	120
kincumb	1000000
jgera	50
npara	20
ycross	0.98
nmax	4
bconst	0.996
nmutar	1
kseed	1211
ndes	5%
temp	20000.
alfat	0.96
mcor	10
ncl	10

c) parâmetros

6.3.2.6 Algoritmo GAPAR2B

Para o algoritmo GAPAR2B, sem inicialização, a solução ótima foi obtida em 27 gerações, com 7.395 PPLs em um tempo de 34,5 minutos (Tabela 6.59). Já com o uso de 20 configurações iniciais, obteve-se a solução ótima em 15 gerações, com 2141 PPLs em um tempo de 4,5 minutos (Tabela 6.60).

Tabela 6.59: Sistema Sul Brasileiro - algoritmo GAPAR2B (sem inicialização)



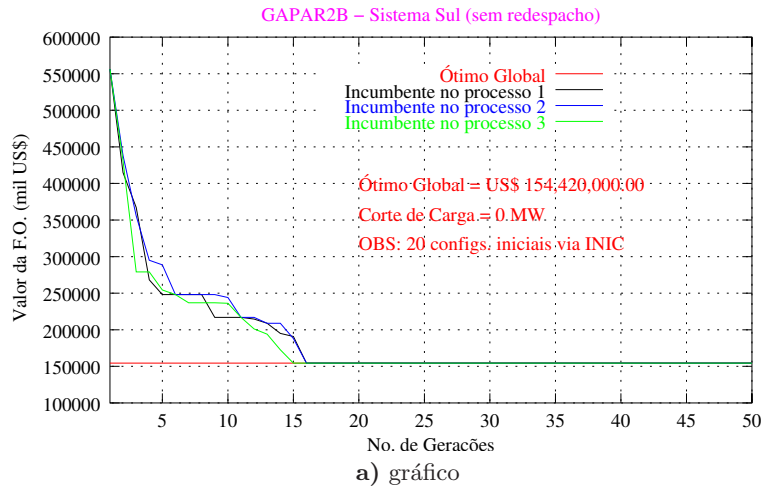
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0	0	0	0	0	0	1	0	0	0	0	1	0
1	0	1	0	3	0	0	0	0	0	0	0	0	0	2	2	0	0	2	0

b) configuração ótima

parâmetro	valor
nafa	1000
npop	180
kincumb	2500000
jgera	50
npara	1
ycross	0.90
nmax	4
bconst	0.996
nmutar	7
kseed	7212
ndes	3%
temp	40000
alfat	0.97
mcor	0
kk1	20
ncl	10

c) parâmetros

Tabela 6.60: Sistema Sul Brasileiro - algoritmo GAPAR2B (com inicialização)



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	0	0	2	2	0	0	2	0

b) configuração ótima

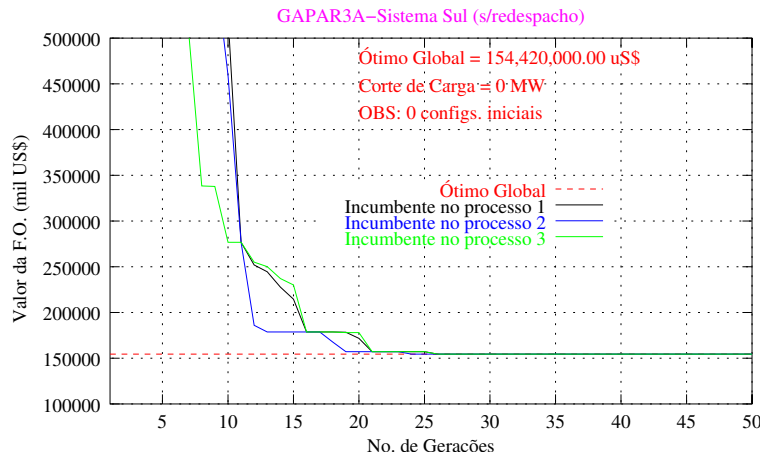
parâmetro	valor
nafa	1000
npop	180
kincumb	2500000
jgera	50
npara	20
ycross	0.98
nmax	4
bconst	0.996
nmutar	3
kseed	1114
ndes	5%
temp	10000
alfat	0.97
mcor	0
kk1	5
ncl	10

c) parâmetros

6.3.2.7 Algoritmo GAPAR3A

Para o algoritmo GAPAR3A, sem inicialização, a solução ótima foi obtida em 24 gerações, com 3.410 PPLs em um tempo de 8 minutos (Tabela 6.61). Já com o uso de 20 configurações iniciais, obteve-se a solução ótima em 15 gerações, com 2.161 PPLs em um tempo de 5,2 minutos (Tabela 6.62).

Tabela 6.61: Sistema Sul Brasileiro - algoritmo GAPAR3A (sem inicialização)



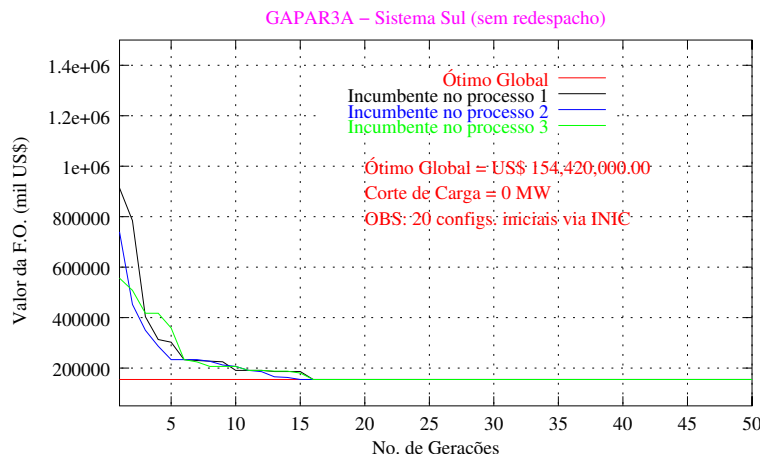
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0	0	0	0	0	0	1	0	0	0	0	1	0
1	0	1	0	3	0	0	0	0	0	0	0	0	0	2	2	0	0	2	0

b) configuração ótima

parâmetro	valor
nafa	1000
npop	180
kincumb	2500000
jgera	50
npara	1
ycross	0.98
nmax	4
bconst	0.996
nmutar	3
kseed	1114
ndes	5%
temp	10000
alfat	0.98
mcor	0
kk1	5
ncl	10

c) parâmetros

Tabela 6.62: Sistema Sul Brasileiro - algoritmo GAPAR3A (com inicialização)



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

b) configuração ótima

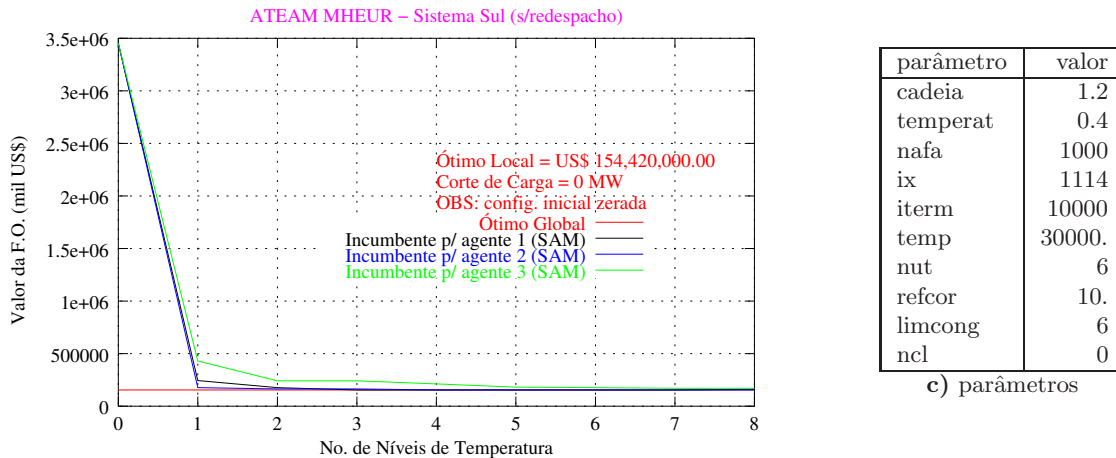
parâmetro	valor
nafa	1000
npop	180
kincumb	2500000
jgera	50
npara	20
ycross	0.98
nmax	4
bconst	0.996
nmutar	3
kseed	1114
ndes	5%
temp	10000
alfat	0.97
mcor	0
kk1	5
ncl	10

c) parâmetros

6.3.2.8 Algoritmo MHEUR

Para o algoritmo MHEUR, sem inicialização, obteve-se a solução ótima em 3 gerações, com 2.281 PLs, e um tempo de 4,6 minutos (Tabela 6.63). O time assíncrono foi composto com 3 agentes SAM. Com 20 configurações iniciais, a solução ótima em 5 gerações, com 7.382 PPLs, e um tempo de 15,2 minutos (Tabela 6.64).

Tabela 6.63: Sistema Sul Brasileiro - algoritmo MHEUR (3 agentes SAM) - sem inicialização

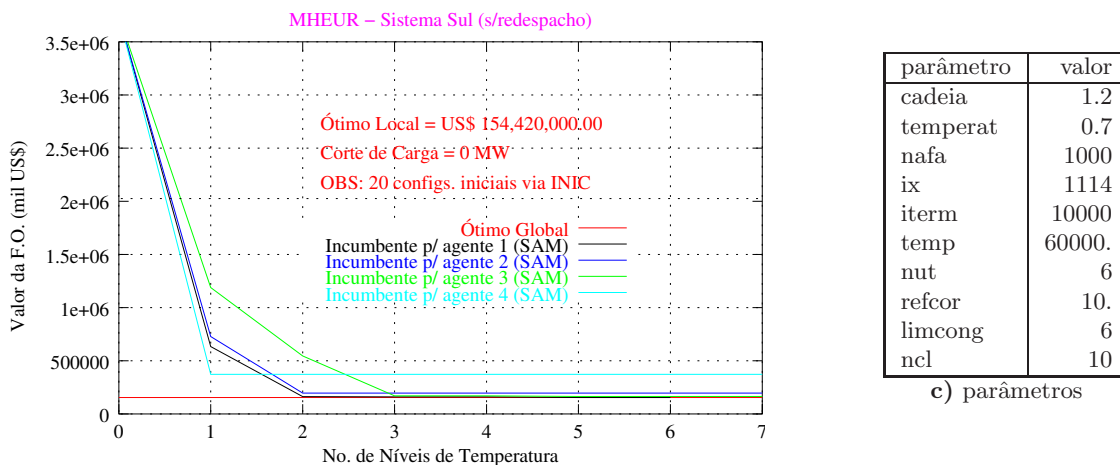


a) gráfico

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

b) configuração ótima

Tabela 6.64: Sistema Sul Brasileiro - algoritmo MHEUR (4 agentes SAM) - com inicialização



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

b) configuração ótima

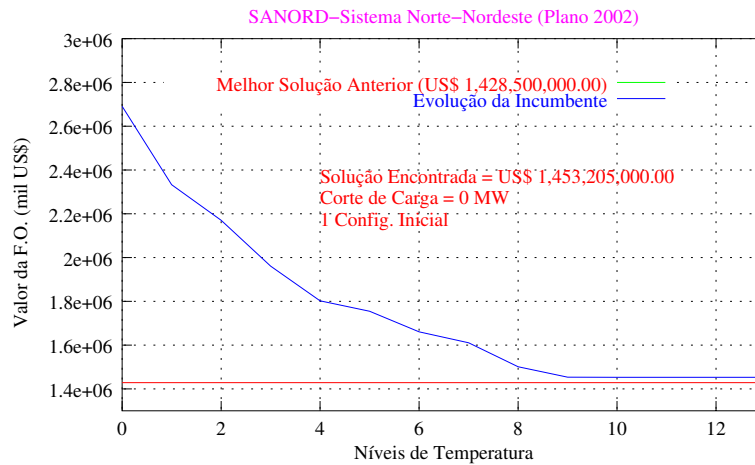
6.4 Sistema Norte-Nordeste Brasileiro

6.4.1 Sistema Norte-Nordeste Brasileiro - Plano 2002

6.4.1.1 Algoritmo SANORD

Observa-se pelo gráfico da Tabela 6.65 que a melhor solução do problema com o algoritmo SANORD, com uma configuração inicial (obtida via algoritmo de inicialização INIC), foi obtida em 10 ciclos de temperatura, com 55.134 PPLs e tempo de 8,2 horas.

Tabela 6.65: Sistema Norte Nordeste (Plano 2002) - algoritmo SANORD



parâmetro	valor
cadeia	1.2
temperat	0.70
nafa	1000
ix	1114
item	200000
temp	50000
nut	5
refcor	300.
limcong	8
ncl	30

c) parâmetros

a) gráfico

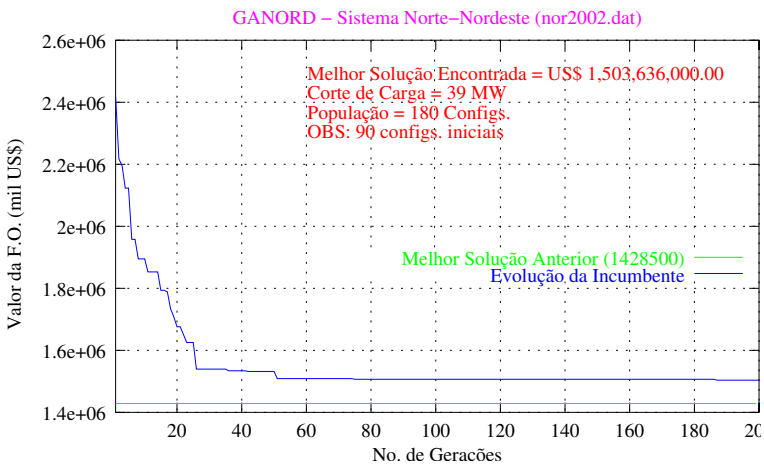
0	0	0	2	0	0	1	3	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0
0	0	3	1	0	0	0	0	0	6	0	3	0	0	1	1	0	0	1
0	1	0	0	0	2	0	1	0	0	0	1	0	1	0	0	0	0	0
0	0	0	0	0	1	0	1	2	0	2	1	2	0	0	0	0	2	0
1	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1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

b) melhor configuração encontrada

6.4.1.2 Algoritmo GANORD

No gráfico da Tabela 6.66 é apresentada a evolução da incumbente com a obtenção da melhor solução na geração 187, com 66.342 PPLs, e um tempo de processamento de 11,6 horas. Nesta simulação foram utilizadas 90 configurações iniciais obtidas previamente via algoritmo inicializador INIC.

Tabela 6.66: Sistema Norte Nordeste (Plano 2002) - algoritmo GANORD



a) gráfico

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

b) melhor configuração encontrada

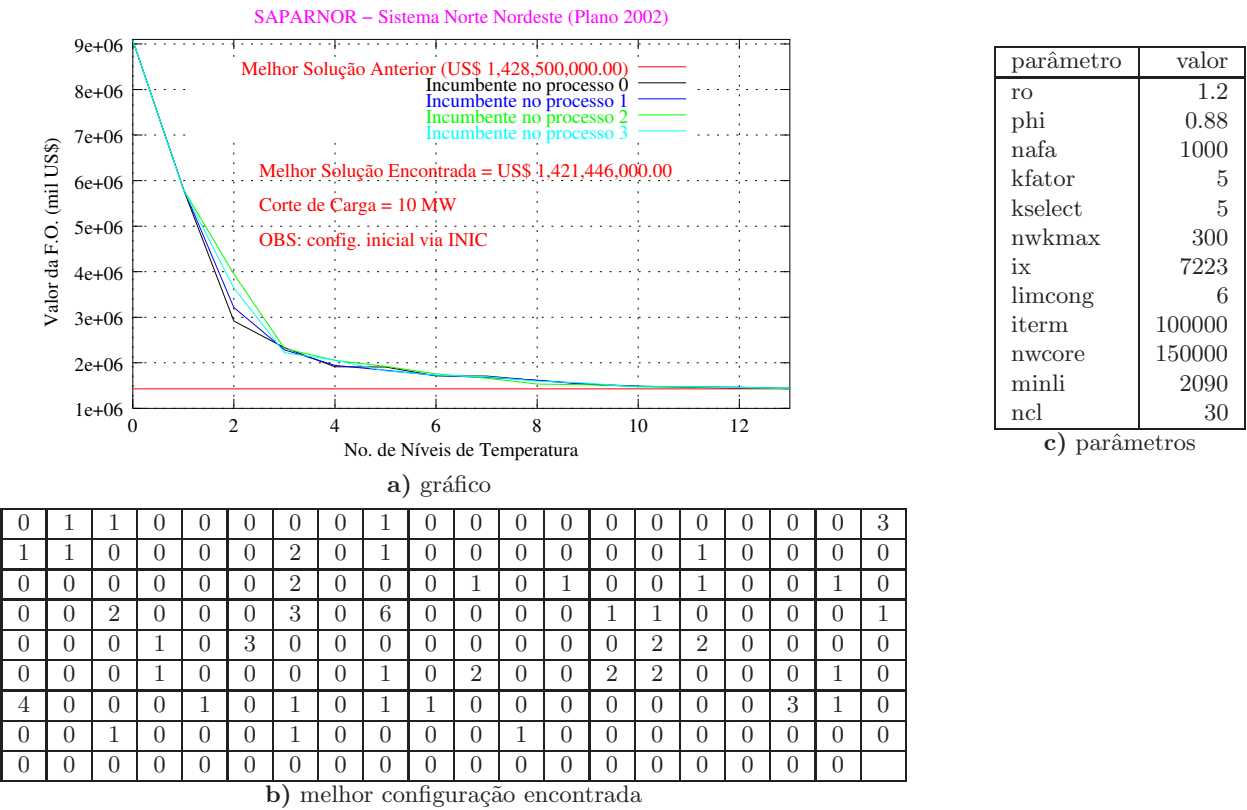
parâmetro	valor
nafa	1000
npop	180
kincomb	4500000
jgera	200
npara	90
ycross	0.98
nmax	11
bconst	0.996
nmutar	1
kseed	1114
ndes	4%
temp	100000.
alfat	0.996
mcors	300
ncl	30

c) parâmetros

6.4.1.3 Algoritmo SAPARNOR

Na Tabela 6.67, com o uso de uma configuração inicial para o algoritmo SAPARNOR, a solução ótima foi obtida em 16 ciclos de temperatura, com 77.030 PPLs e um tempo de 7,9 horas.

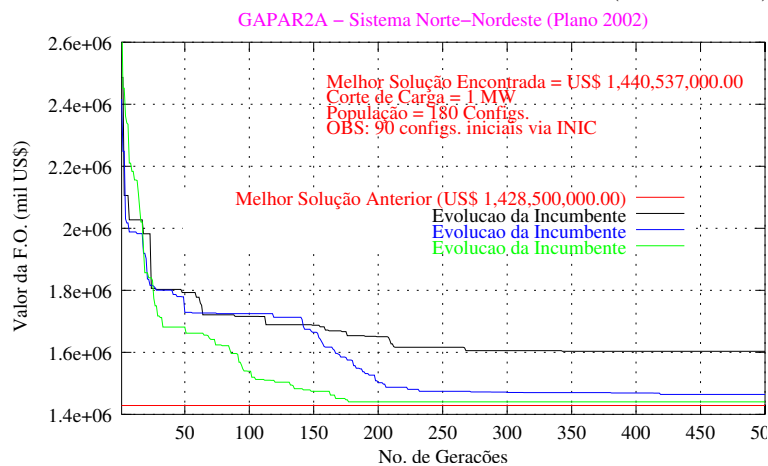
Tabela 6.67: Sistema Norte_Nordeste (Plano 2002) - algoritmo SAPARNOR



6.4.1.4 Algoritmo GAPAR2A

Para o algoritmo GAPAR2A, com o uso de 90 configurações iniciais obteve-se a melhor solução em 177 gerações, com 27.742 PPLs em um tempo de 2,7 horas (Tabela 6.68).

Tabela 6.68: Sistema Norte-Nordeste (Plano 2002) - algoritmo GAPAR2A



a) gráfico

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

b) melhor configuração encontrada

parâmetro	valor
nafa	1000
npop	180
kincomb	3000000
jgera	500
npara	90
ycross	0.98
nmax	11
bconst	0.996
nmutar	3
kseed	1114
ndes	4%
temp	50000
alfat	0.99
mcors	300
ncl	30

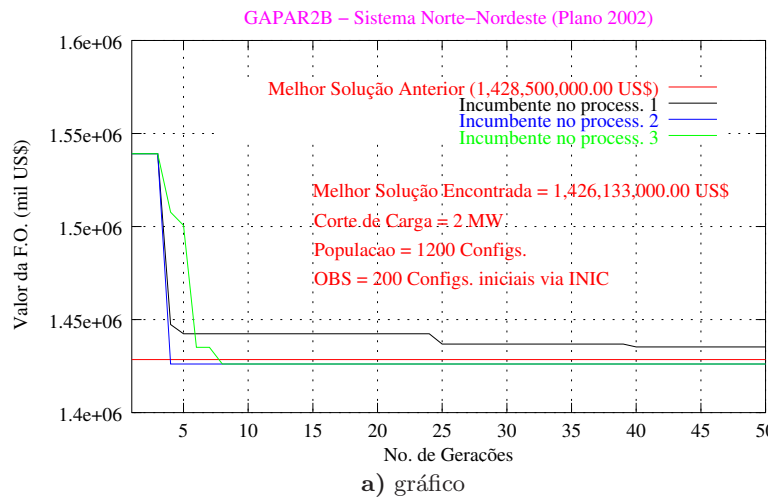
c) parâmetros

6.4.1.5 Algoritmo GAPAR2B

Para uma análise do comportamento do algoritmo GAPAR2B apresenta-se a seguir algumas simulações com diferentes parâmetros. Na Tabela 6.69, para 200 configurações iniciais, obteve-se a melhor solução em 4 gerações, com 5.570 PPLs e um tempo de processamento de 28,9 minutos. Na Tabela 6.70, para 200 configurações iniciais, obteve-se uma solução em 167 gerações, com 135.088 PPLs e um tempo de processamento de 11,3 horas. Na Tabela 6.71, também para 200 configurações iniciais, obteve-se a melhor solução em 122 gerações, com 112.072 PPLs e um tempo de processamento de 9,8 horas. Ainda para 200 configurações iniciais, conforme mostrado na Tabela 6.72, obteve-se a melhor solução em 183 gerações, com 301.351 PPLs e um tempo de processamento de 24,8 horas. E finalmente, com o uso de 400 configurações iniciais, a solução ótima foi obtida em 4 gerações, com 3.708 PPLs em um tempo de 18,7 minutos (Tabela 6.73).

Esta sequência de simulações, mostra além da influência da inicialização do algoritmo, a variações no desempenho do mesmo para mudanças nos parâmetros utilizados.

Tabela 6.69: Sistema Norte Nordeste (Plano 2002) - algoritmo GAPAR2B (resultado 1)



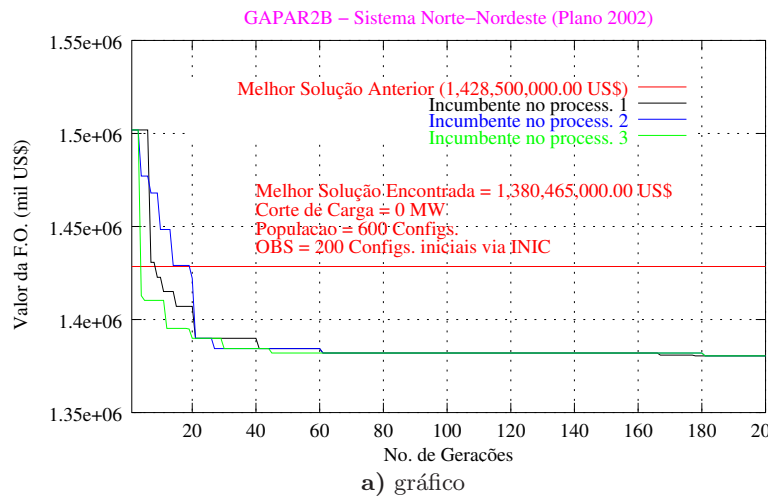
0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	1	2	0
0	0	3	0	0	0	0	0	6	0	3	0	0	1	1	0	0	0	1
0	0	0	0	0	3	0	1	0	0	0	0	0	1	0	0	0	0	1
2	0	0	0	0	1	2	3	1	0	0	1	1	2	2	0	0	1	0
2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	2	0	1	0	0	0	1	0	1	1	0	1
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	1200
kincumb	1700000
jgera	50
npara	200
ycross	0.98
nmax	7
bconst	0.997
nmutar	7
kseed	7326
ndes	3%
temp	120000
alfat	0.996
mcor	300
kk1	50
ncl	30

c) parâmetros

Tabela 6.70: Sistema Norte Nordeste (Plano 2002) - algoritmo GAPAR2B (resultado 2)



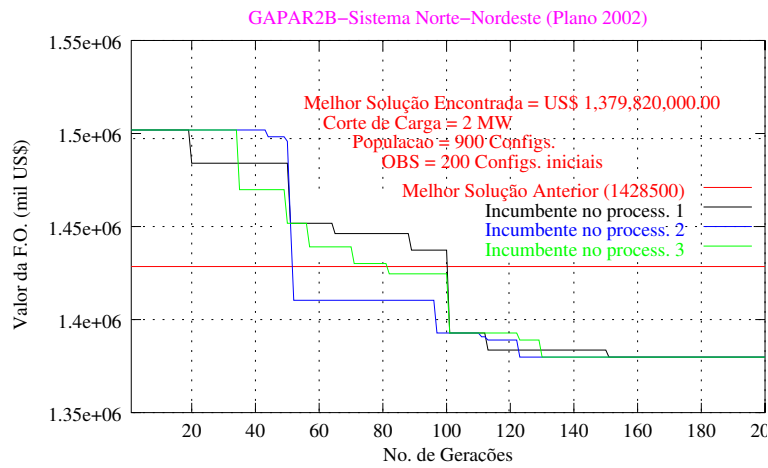
0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	1	2	0
0	0	3	0	0	0	0	0	6	0	3	0	0	1	1	0	0	0	1
0	0	0	0	0	3	0	1	0	0	0	0	0	1	0	0	0	0	1
2	0	0	0	0	1	2	3	1	0	0	1	1	2	2	0	0	1	0
2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	2	0	1	0	0	0	1	0	1	1	0	1
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	1200
kincumb	1700000
jgera	50
npara	200
ycross	0.98
nmax	7
bconst	0.997
nmutar	7
kseed	7326
ndes	3%
temp	120000
alfat	0.996
mcor	300
kk1	50
ncl	30

c) parâmetros

Tabela 6.71: Sistema Norte Nordeste (Plano 2002) - algoritmo GAPAR2B (resultado 3)



a) gráfico

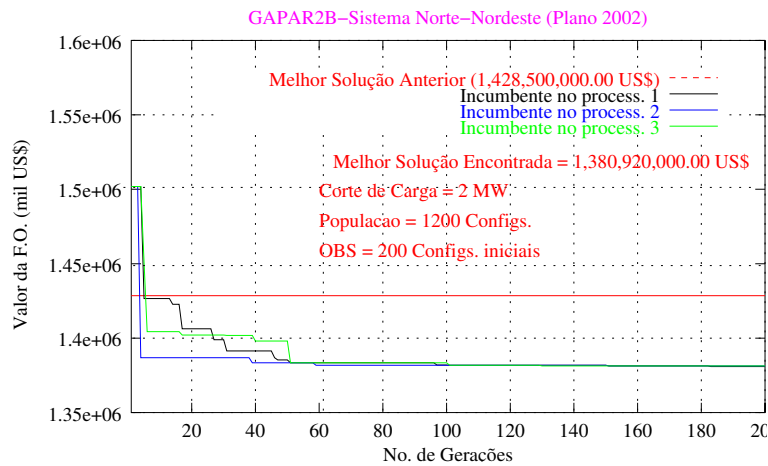
0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	1	2	0
0	0	3	0	0	0	0	0	6	0	3	0	0	1	1	0	0	0	1
0	0	0	0	0	3	0	1	0	0	0	0	0	1	0	0	0	0	1
2	0	0	0	0	1	1	3	1	0	0	2	1	2	2	0	0	1	0
2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	3	0	0	0	0	0	1	0	0	1	0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

b) melhor configuração encontrada

parâmetro	valor
nafa	1000
npop	600
kincumb	1600000
jgera	200
npara	200
ycross	0.98
nmax	7
bconst	0.997
nmutar	7
kseed	7326
ndes	3%
temp	120000
alfat	0.996
mcors	300
kk1	50
ncl	30

c) parâmetros

Tabela 6.72: Sistema Norte Nordeste (Plano 2002) - algoritmo GAPAR2B (resultado 4)



a) gráfico

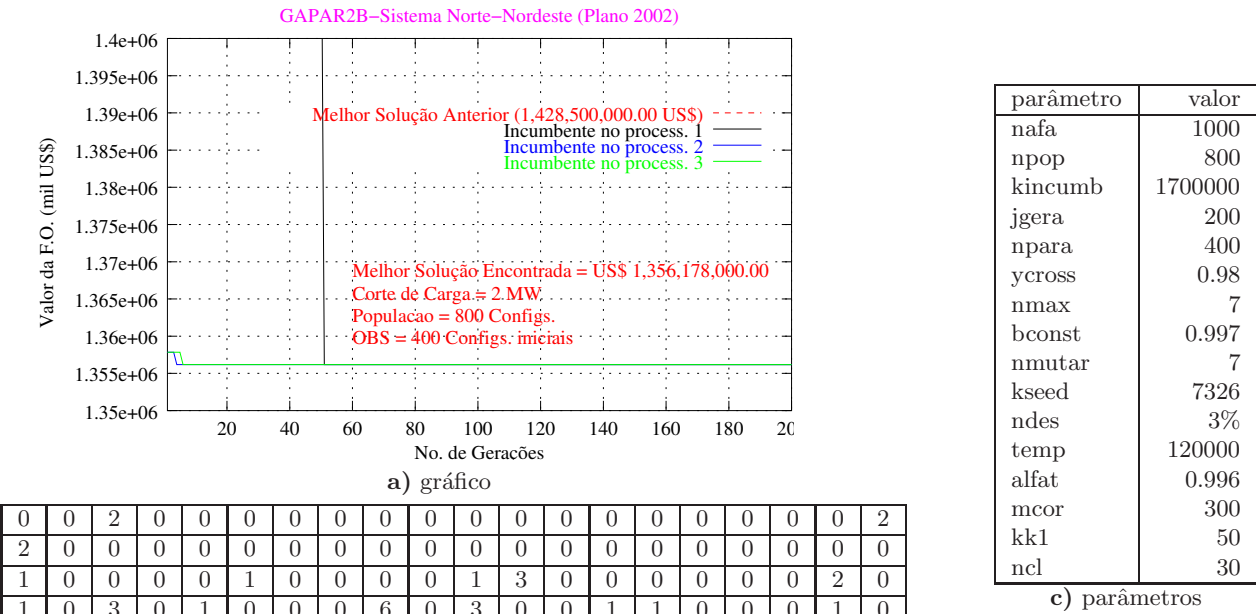
0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	1	2	0
0	0	3	0	0	0	0	0	6	0	3	0	0	1	1	0	0	0	1
0	0	0	0	0	3	0	1	0	0	0	0	0	1	0	0	0	0	1
2	0	0	0	0	1	2	3	1	0	0	1	1	2	2	0	0	2	0
1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	3	0	0	0	0	0	1	0	0	1	0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

b) melhor configuração encontrada

parâmetro	valor
nafa	1000
npop	1200
kincumb	1700000
jgera	200
npara	200
ycross	0.98
nmax	7
bconst	0.997
nmutar	7
kseed	7326
ndes	3%
temp	120000
alfat	0.996
mcors	300
kk1	50
ncl	30

c) parâmetros

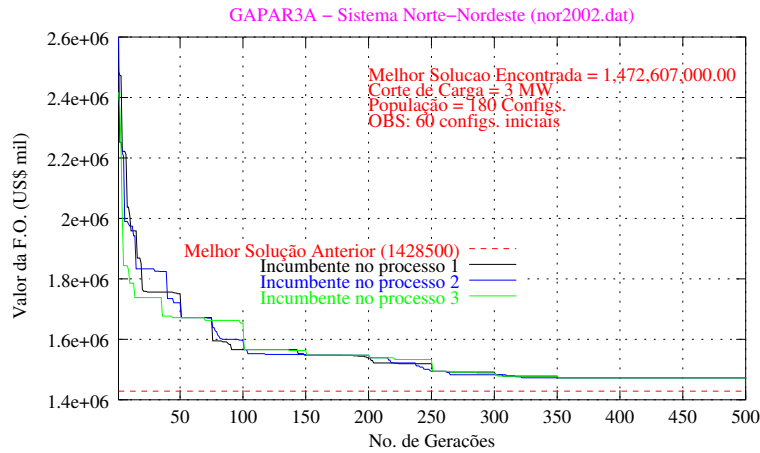
Tabela 6.73: Sistema Norte Nordeste (Plano 2002) - algoritmo GAPAR2B (resultado 5)



6.4.1.6 Algoritmo GAPAR3A

Com uma população de 360 cromossomos e 60 configurações iniciais, o algoritmo GAPAR3A obteve para o sistema Norte-Nordeste 2002, para os parâmetros apresentados na Tabela 6.74, a melhor solução em 468 gerações, 115.517 PPLs e um tempo de processamento de 13 horas. Ainda com uma população de 360 cromossomos e 60 configurações iniciais e os parâmetros apresentados na Tabela 6.75, obteve-se a melhor solução para 480 gerações, 119.728 PPLs e um tempo de processamento de 11,3 horas.

Tabela 6.74: Sistema Norte Nordeste (Plano 2002) - algoritmo GAPAR3A (Resultado 1)



a) gráfico

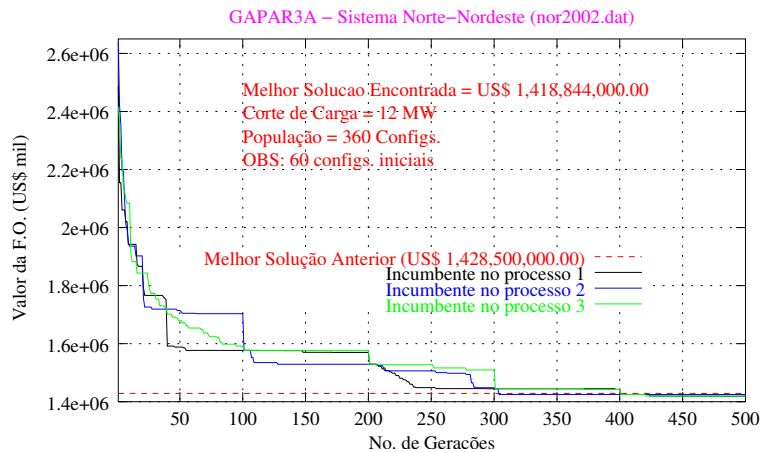
0	0	0	2	0	0	1	2	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0	0	2	0
0	0	3	0	1	0	0	0	6	0	3	0	0	1	1	0	0	0	1
0	1	0	0	0	2	0	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	2	1	0	1	0	0	1	1	0	0	1	0
2	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	2	0
0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	1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

b) melhor configuração encontrada

parâmetro	valor
nafa	1000
npop	360
kincumb	4500000
jgera	800
npara	60
ycross	0.90
nmax	12
bconst	0.996
nmutar	2
kseed	1114
ndes	4%
temp	80000
alfat	0.996
mcor	300
kk1	50
ncl	30

c) parâmetros

Tabela 6.75: Sistema Norte Nordeste (Plano 2002) - algoritmo GAPAR3A (Resultado 2)



a) gráfico

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

b) melhor configuração encontrada

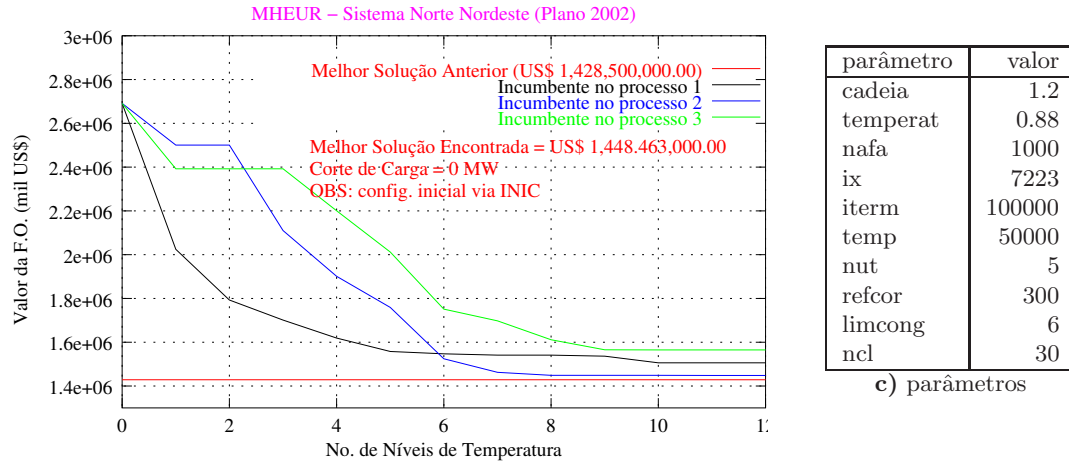
parâmetro	valor
nafa	1000
npop	360
kincumb	4500000
jgera	500
npara	60
ycross	0.90
nmax	12
bconst	0.996
nmutar	2
kseed	1114
ndes	4%
temp	70000
alfat	0.996
mcor	300
kk1	100
ncl	30

c) parâmetros

6.4.1.7 Algoritmo MHEUR

Para o algoritmo MHEUR, com o uso de uma configuração inicial, obteve-se a melhor solução em 11 ciclos, com 90.707 PPLs, e um tempo de 9,4 horas (Tabela 6.76). O time assíncrono foi composto com 3 agentes SAM.

Tabela 6.76: Sistema Norte Nordeste (Plano 2002) - algoritmo MHEUR (3 agentes SAM)



a) gráfico

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

b) melhor configuração encontrada

6.4.2 Sistema Norte-Nordeste Brasileiro - Plano 2008

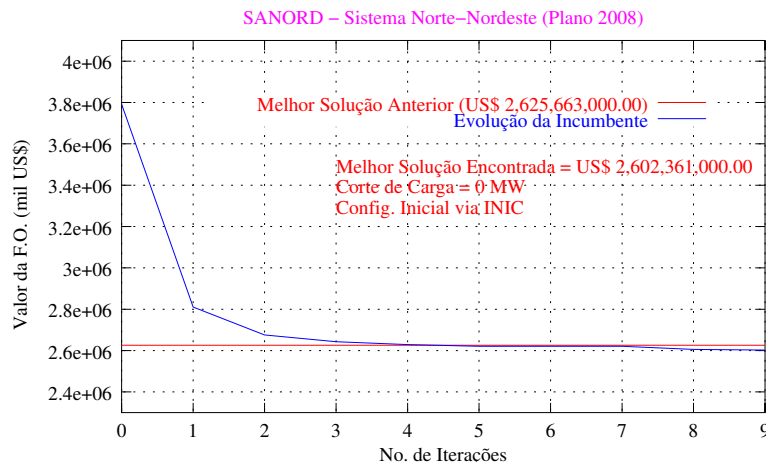
6.4.2.1 Algoritmo SANORD

Observa-se pelo gráfico da Tabela 6.77 que a melhor solução encontrada com o algoritmo SANORD, com uma configuração inicial (obtida via algoritmo inicializador INIC), foi obtida em 9 ciclos de temperatura, com 99.039 PPLs e tempo de 19,2 horas.

6.4.2.2 Algoritmo GANORD

No gráfico da Tabela 6.78 é apresentado a evolução da incumbente com a obtenção da melhor solução na geração 147, com 89.807 PPLs, e um tempo de processamento de 23,0 horas. Nesta simulação foram utilizadas 60 configurações iniciais obtidas previamente via algoritmo inicializador.

Tabela 6.77: Sistema Norte Nordeste (Plano 2008) - algoritmo SANORD

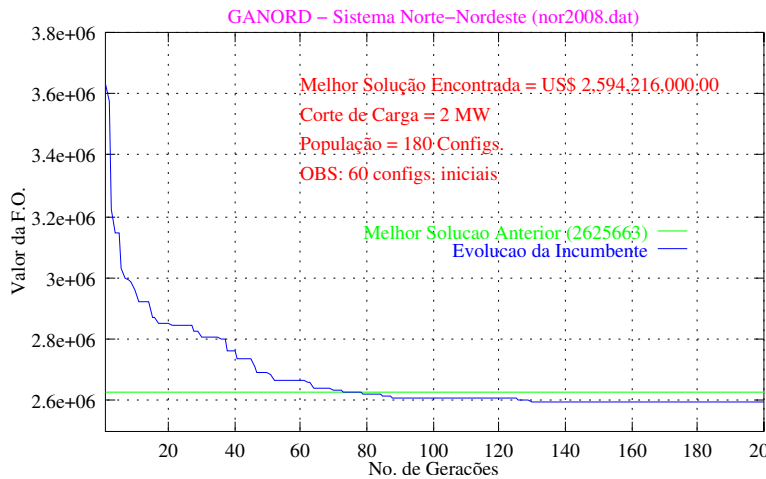


a) gráfico

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

Tabela 6.78: Sistema Norte Nordeste (Plano 2008) - algoritmo GANORD



a) gráfico

1	1	0	0	0	0	0	0	5	0	0	0	0	0	3	0	0	0	1	4
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	4	0	0	0	0	0	1	3	0
1	0	6	0	0	0	0	0	11	0	5	0	0	0	0	0	0	2	0	0
2	0	0	1	0	4	0	0	0	0	0	0	0	0	1	2	0	0	0	0
0	1	0	2	0	2	0	1	1	2	2	1	0	2	3	0	0	1	0	0
4	0	1	0	0	0	1	0	1	1	0	0	1	0	0	0	0	3	1	0
0	0	1	0	0	1	1	3	0	0	0	0	0	1	0	0	1	0	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
cadeia	1.4
temperat	0.9
nafa	1000
ix	1114
item	100000
temp	50000
nut	5
refcor	300.
limcong	8
ncl	20

c) parâmetros

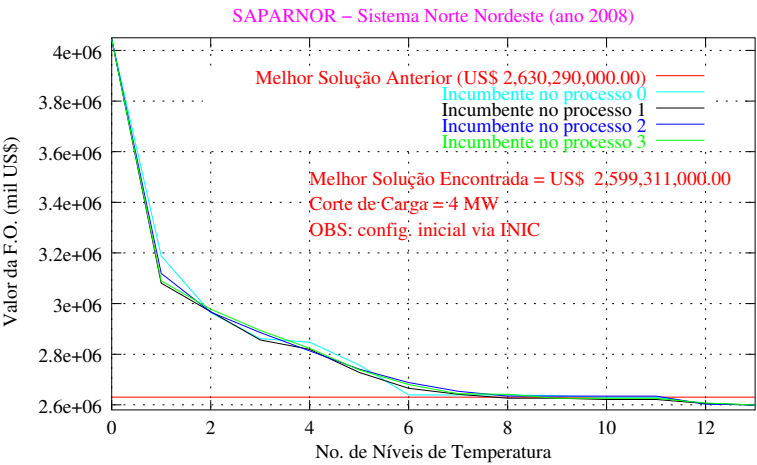
parâmetro	valor
nafa	1000
npop	180
kincumb	4500000
jgera	200
npara	60
ycross	0.90
nmax	12
bconst	0.996
nmutar	4
kseed	1211
ndes	2%
temp	150000
alfat	0.996
mcor	300
ncl	30

c) parâmetros

6.4.2.3 Algoritmo SAPARNOR

Na Tabela 6.79, com o uso de uma configuração inicial para o algoritmo SAPARNOR, a solução ótima foi obtida em 13 ciclos de temperatura, com 52.099 PPLs e um tempo de 4,0 horas.

Tabela 6.79: Sistema Norte_Nordeste (Plano 2008) - algoritmo SAPARNOR



parâmetro	valor
ro	1.2
phi	0.90
nafa	1000
kfator	6
kselect	2
nwkmax	300
ix	5775
item	50000
nwcore	150000
minli	2090
ncl	30

c) parâmetros

a) gráfico

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

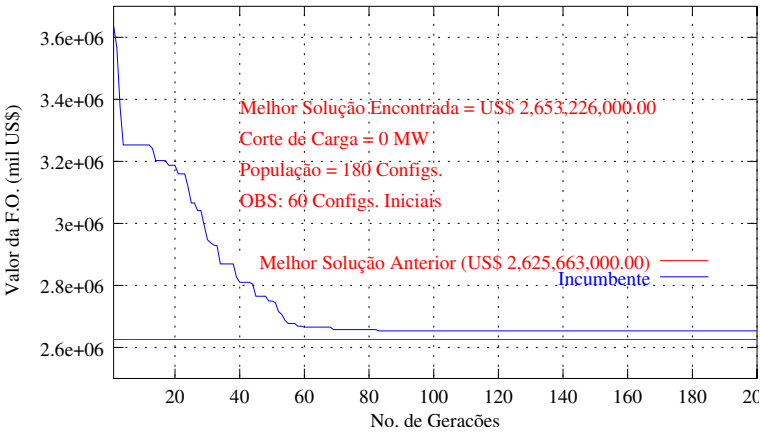
Melhor Configuração Encontrada

6.4.2.4 Algoritmo GAPAR1B

Já para o algoritmo GAPAR1B, com o uso de 60 configurações iniciais, a melhor solução foi obtida em 83 gerações, com 29.398 PPLs em um tempo de processamento de 2,3 horas (Tabela 6.80).

Tabela 6.80: Sistema Norte Nordeste (Plano 2008) - algoritmo GAPAR1B

GAPAR1B-Sistema Norte-Nordeste (Plano 2008)



a) gráfico

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

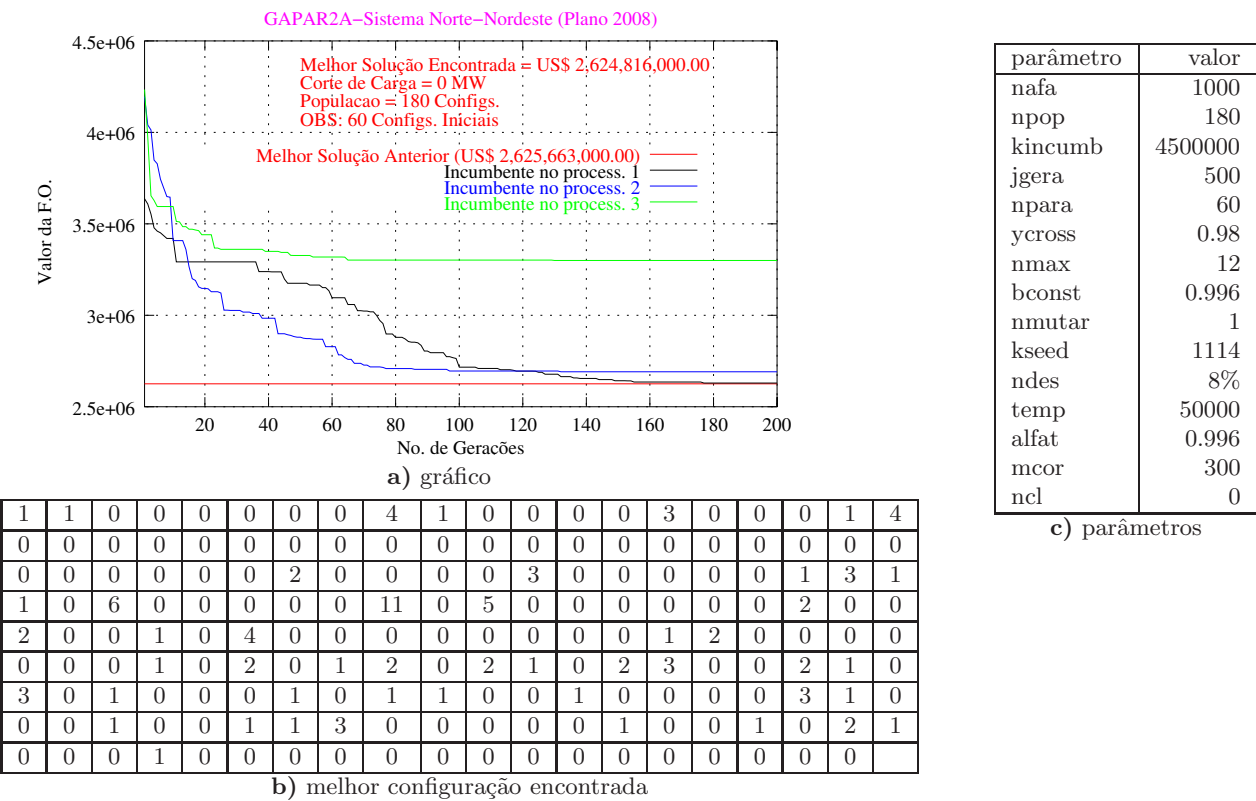
parâmetro	valor
nafa	1000
npop	180
kincumb	4000000
jgera	200
npara	60
ycross	0.99
nmax	11
bconst	0.996
nmutar	1
kseed	1114
ndes	8%
temp	70000
alfat	0.99
mcor	1000
ncl	30

c) parâmetros

6.4.2.5 Algoritmo GAPAR2A

Para o algoritmo GAPAR2A, com o uso de 60 configurações iniciais obteve-se a melhor solução em 201 gerações, com 24.002 PPLs em um tempo de 2,7 horas (Tabela 6.81).

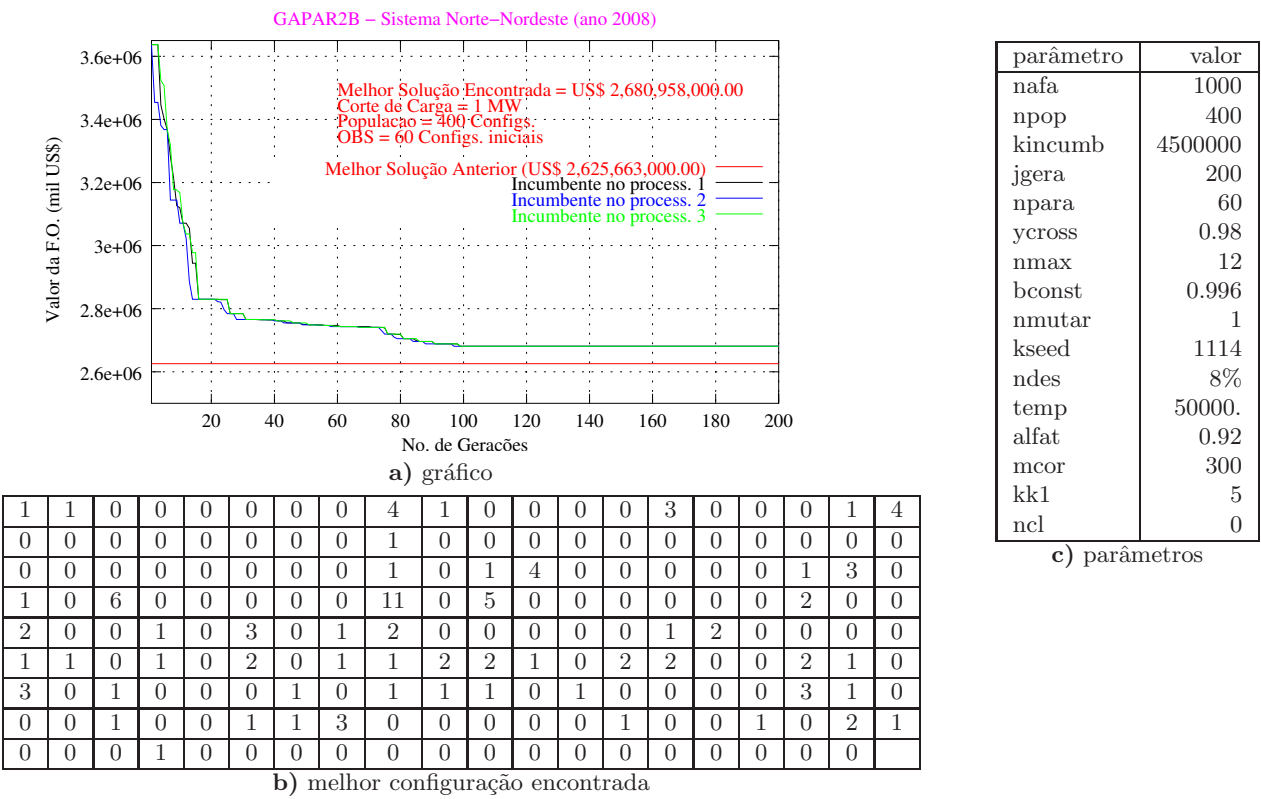
Tabela 6.81: Sistema Norte-Nordeste (Plano 2008) - algoritmo GAPAR2A



6.4.2.6 Algoritmo GAPAR2B

Na Tabela 6.82, para 60 configurações iniciais, obteve-se a melhor solução em 99 gerações, com 21.127 PPLs e um tempo de processamento de 2,3 horas.

Tabela 6.82: Sistema Norte Nordeste (Plano 2008) - algoritmo GAPAR2B

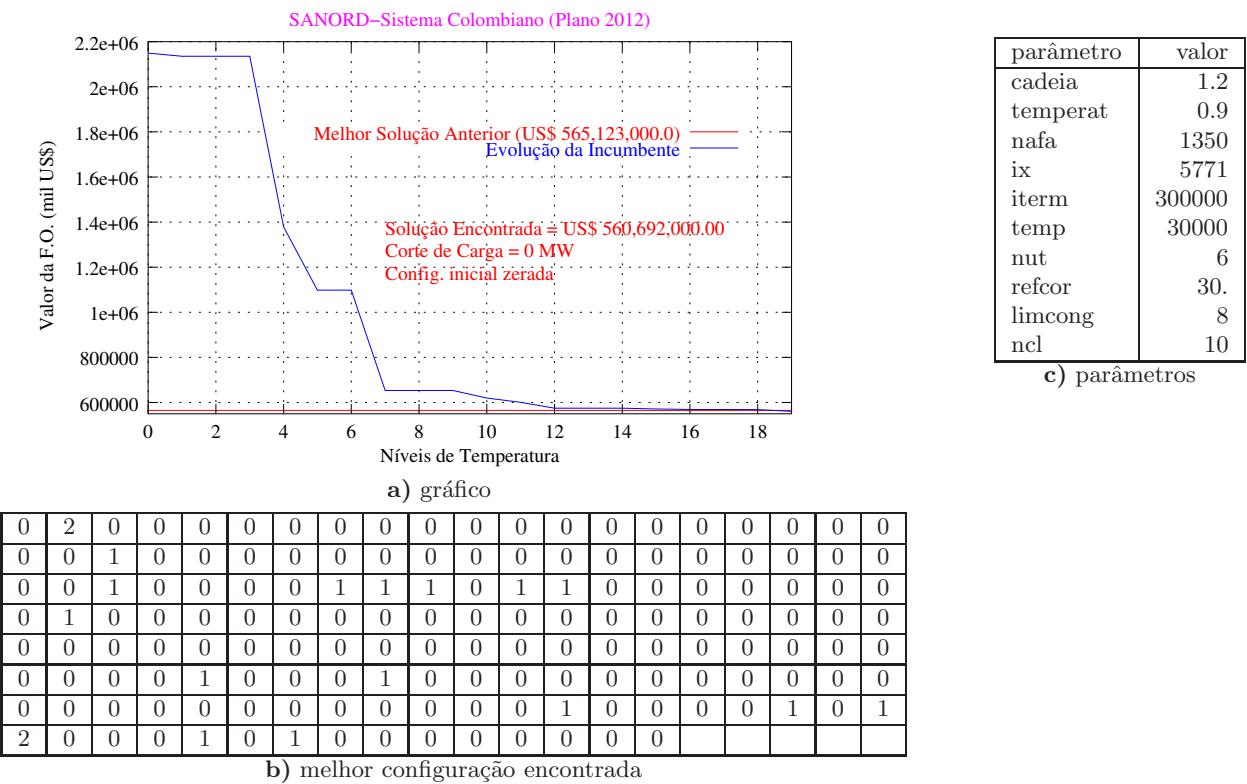


6.5 Sistema Colombiano - Plano 2012

6.5.1 Algoritmo SANORD

Observa-se pelo gráfico da Tabela 6.83 que a melhor solução encontrada com o algoritmo SANORD, sem inicialização, foi obtida em 19 ciclos de temperatura, com 301.086 PPLs e tempo de 74,9 horas.

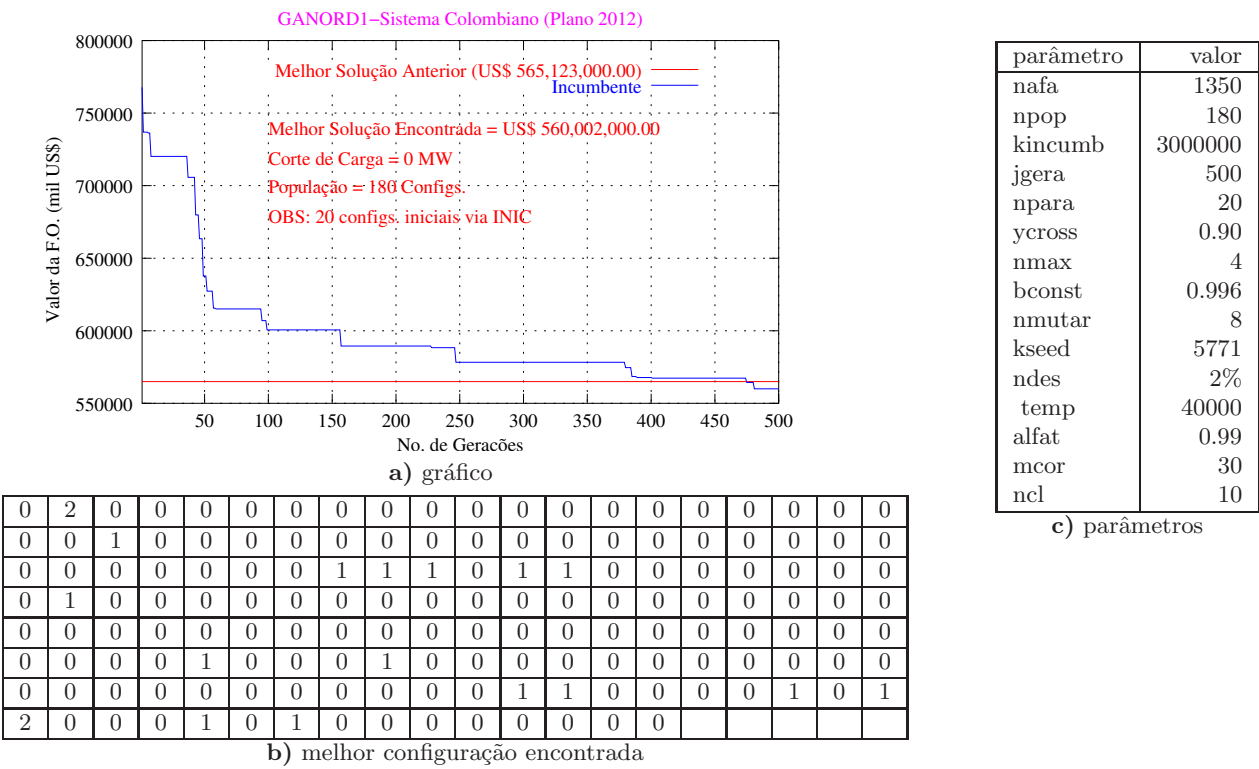
Tabela 6.83: Sistema Colombiano (Plano 2012) - algoritmo SANORD (configuração inicial zerada)



6.5.2 Algoritmo GANORD

No gráfico da Tabela 6.84 é apresentada a evolução da incumbente com a obtenção da melhor solução na geração 481, com 278.196 PPLs, e um tempo de processamento de 66,6 horas. Nesta simulação foram utilizadas 20 configurações iniciais obtidas previamente via algoritmo inicializador INIC.

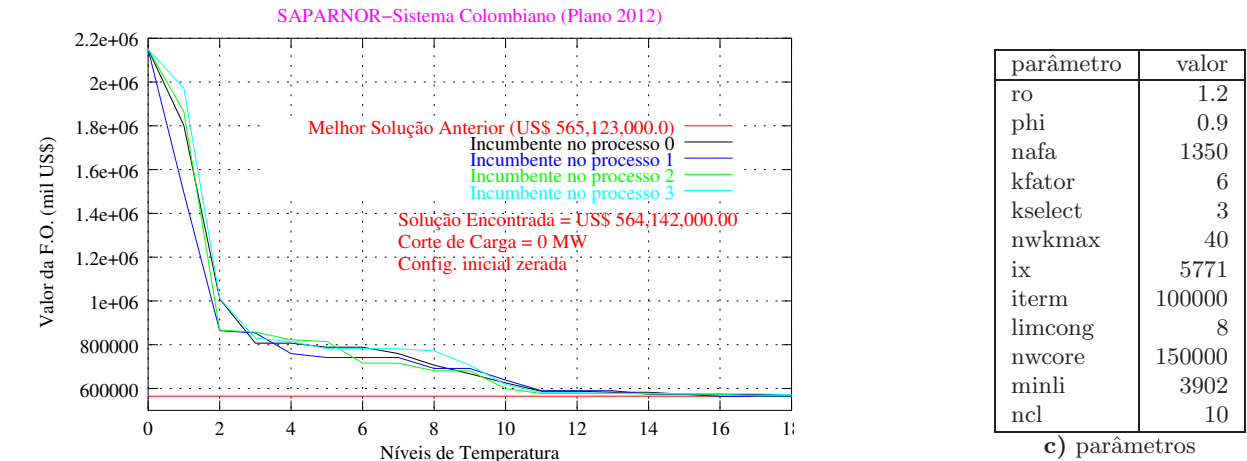
Tabela 6.84: Sistema Colombiano (Plano 2012) - algoritmo GANORD



6.5.3 Algoritmo SAPARNOR

Na Tabela 6.85, sem inicialização para o algoritmo SAPARNOR, a melhor solução foi obtida em 16 ciclos de temperatura, com 76.046 PPLs e um tempo de 12,3 horas.

Tabela 6.85: Sistema Colombiano (Plano 2012) - algoritmo SAPARNOR (sem inicialização)



a) gráfico

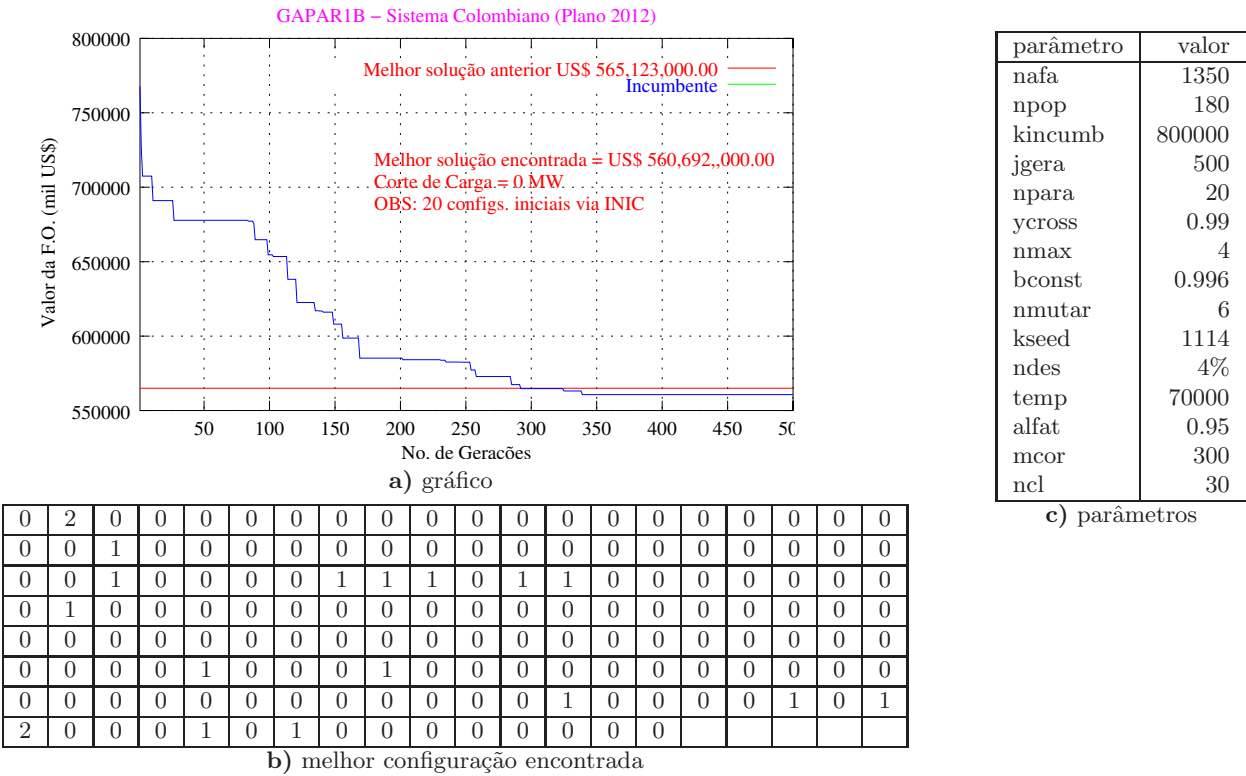
0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	1
2	0	0	0	1	0	1	0	0	0	0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

b) melhor configuração encontrada

6.5.4 Algoritmo GAPAR1B

Já para o algoritmo GAPAR1B, com o uso de 20 configurações iniciais, a melhor solução foi obtida em 339 gerações, com 192.642 PPLs em um tempo de processamento de 34,1 horas (Tabela 6.86).

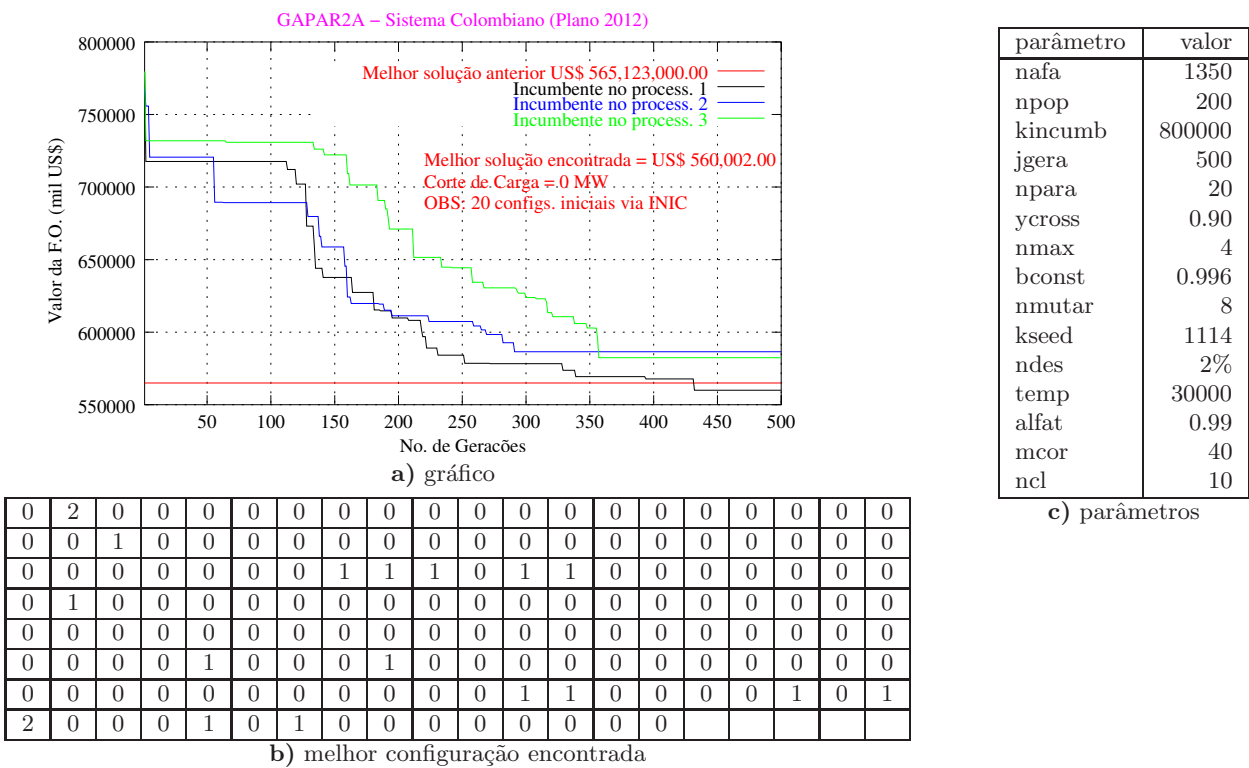
Tabela 6.86: Sistema Colombiano (Plano 2012) - algoritmo GAPAR21B



6.5.5 Algoritmo GAPAR2A

Para o algoritmo GAPAR2A, com o uso de 20 configurações iniciais obteve-se a solução ótima para o problema em 432 gerações, com 100.853 PPLs em um tempo de 16,8 horas (Tabela 6.87).

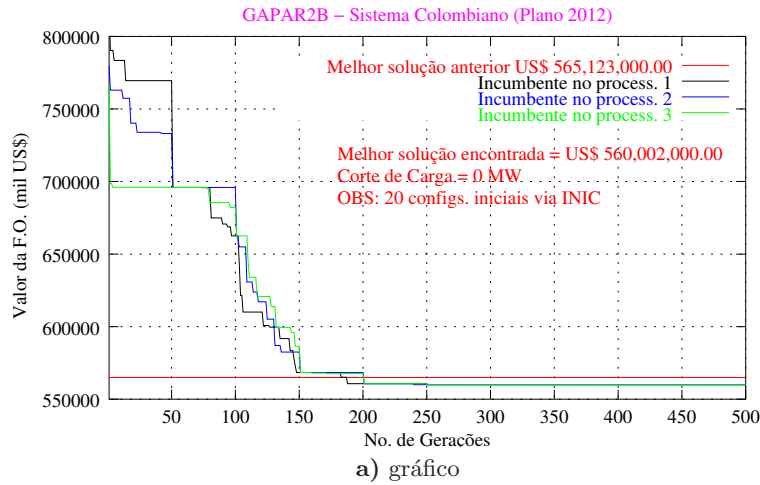
Tabela 6.87: Sistema Colombiano (Plano 2012) - algoritmo GAPAR2A



6.5.6 Algoritmo GAPAR2B

Na sequência, são apresentadas alguns resultados de simulações com o algoritmo GAPAR2B. Na Tabela 6.88, para 10 configurações iniciais, obteve-se a solução ótima para o problema (US\$ 560,002,000.00 - 0 MW) em 240 gerações, com 21.929 PPLs e um tempo de processamento de 3,4 horas. Na Tabela 6.89, para 20 configurações iniciais, obteve-se a solução ótima para o problema em 432 gerações, com 95.974 PPLs e um tempo de processamento de 14,7 horas. Já com 80 configurações iniciais, obteve-se a solução de US\$ 560,692,000.00 com corte de carga 0 MW, em 29 gerações, com 22.383 PPLs e um tempo de processamento de 3,5 horas (Tabela 6.90).

Tabela 6.88: Sistema Colombiano (Plano 2012) - algoritmo GAPAR2B (resultado 1)



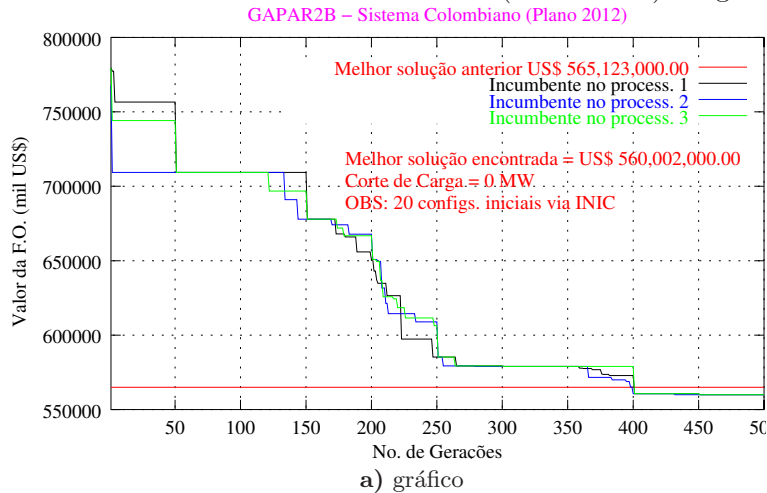
0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0
2	0	0	0	1	0	1	0	0	0	0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

b) melhor configuração encontrada

parâmetro	valor
nafa	1350
npop	100
kincumb	800000
jgera	500
npara	10
ycross	0.98
nmax	4
bconst	0.996
nmutar	4
kseed	1114
ndes	6%
temp	10000
alfat	0.99
mcor	100
kk1	50
ncl	10

c) parâmetros

Tabela 6.89: Sistema Colombiano (Plano 2012) - algoritmo GAPAR2B (resultado 2)



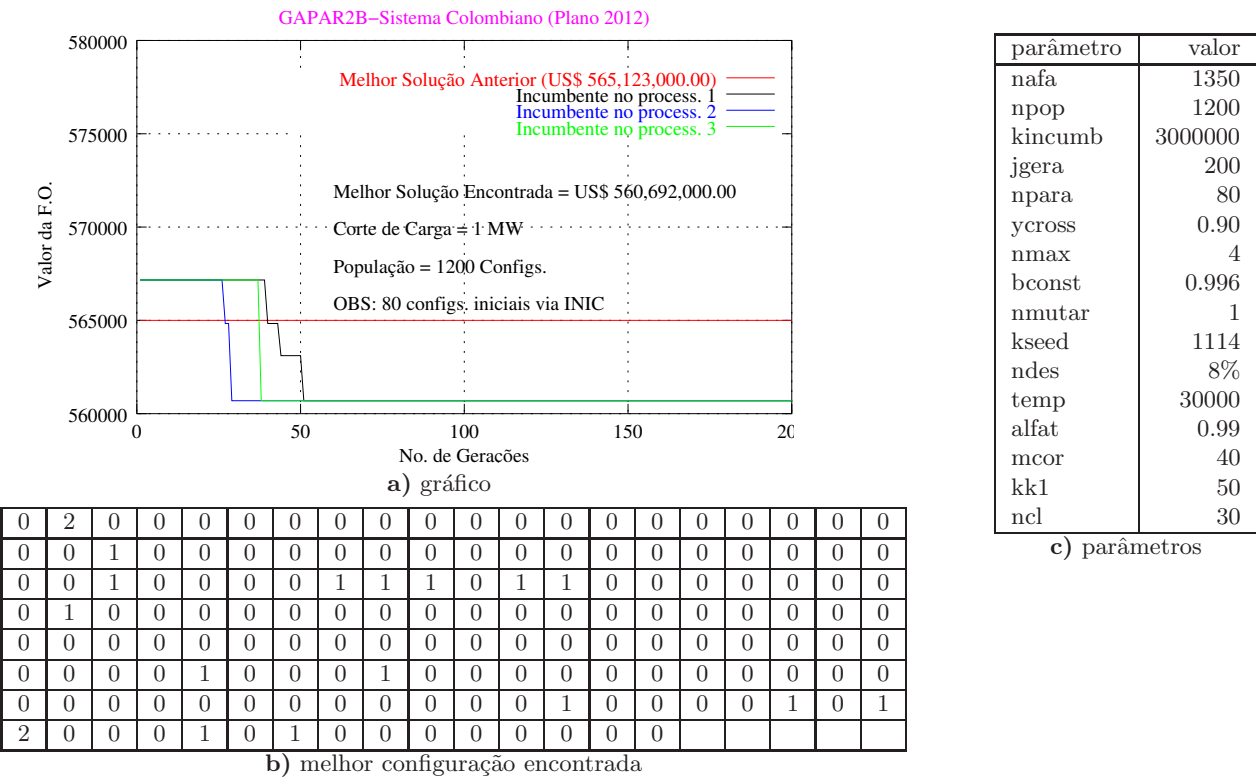
0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0
2	0	0	0	1	0	1	0	0	0	0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

b) melhor configuração encontrada

parâmetro	valor
nafa	1350
npop	200
kincumb	800000
jgera	500
npara	20
ycross	0.90
nmax	4
bconst	0.996
nmutar	8
kseed	5771
ndes	2%
temp	30000
alfat	0.99
mcor	40
kk1	50
ncl	10

c) parâmetros

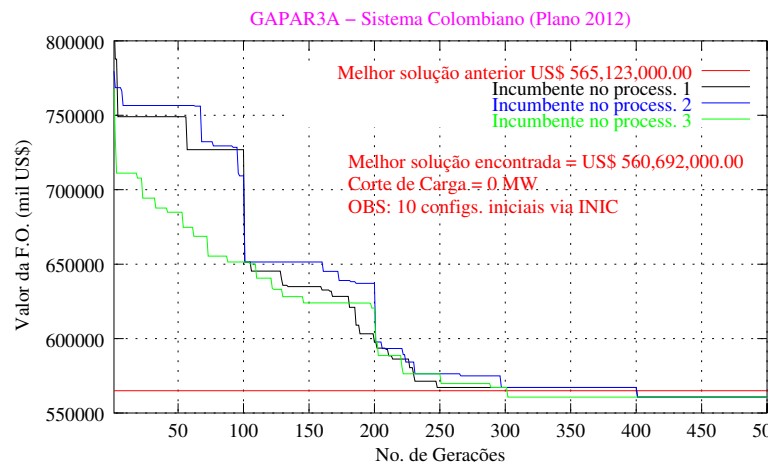
Tabela 6.90: Sistema Colombiano (Plano 2012) - algoritmo GAPAR2B (resultado 3)



6.5.7 Algoritmo GAPAR3A

Com uma população de 100 cromossomos e 10 configurações iniciais, o algoritmo GAPAR3A obteve para o sistema Colombiano 2012, para os parâmetros apresentados na Tabela 6.91, a solução ótima para o problema em 302 gerações, 26.769 PPLs e um tempo de processamento de 4,1 horas.

Tabela 6.91: Sistema Colombiano (Plano 2012) - algoritmo GAPAR3A (com inicialização)



a) gráfico

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

b) melhor configuração encontrada

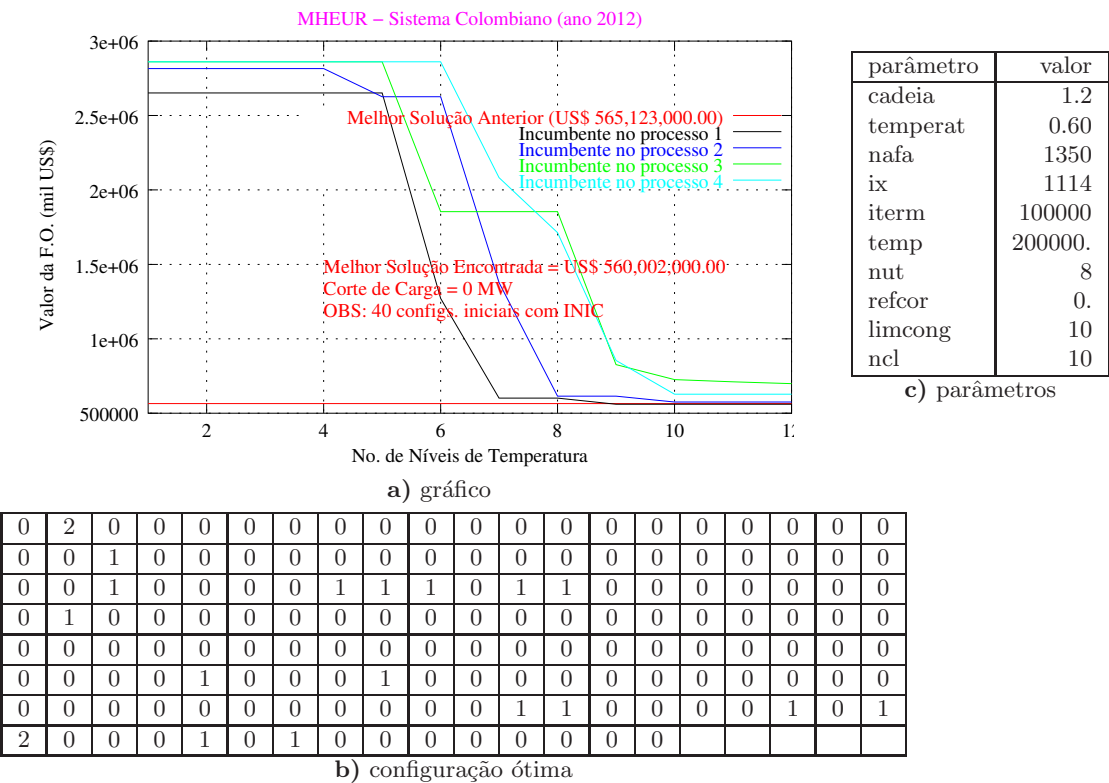
parâmetro	valor
nafa	1350
npop	100
kincumb	800000
jgera	500
npara	10
ycross	0.98
nmax	4
bconst	0.996
nmutar	4
kseed	1114
ndes	6%
temp	10000
alfat	0.99
mcor	100
kk1	100
ncl	10

c) parâmetros

6.5.8 Algoritmo MHEUR

Para o algoritmo MHEUR, com uma população de 160 configurações e o uso de 40 configurações iniciais, obteve-se a melhor solução em 10 ciclos, com 50.395 PPLs, e um tempo de 7,9 horas (Tabela 6.92). O time assíncrono foi composto com 4 agentes SAM.

Tabela 6.92: Sistema Colombiano (Plano 2012) - algoritmo MHEUR (4 agentes SAM)



Mais detalhes sobre os resultados podem ser vistos no *Apêndice C* onde são apresentados alguns trabalhos publicados em congressos nacionais e internacionais. A seguir é apresentado um resumo em forma de tabelas dos resultados para uma análise comparativa dos algoritmos propostos.

6.6 Resumo dos Resultados

Nesta seção, são apresentados os resultados de diversas simulações realizadas com as diferentes versões implementadas neste trabalho para a resolução do problema do planejamento da expansão da transmissão, para os exemplos de redes de pequeno, médio e grande porte citados anteriormente.

Nas Tabelas 6.93 e 6.94 a seguir, são apresentados resultados de simulações, para os diferentes sistemas testes, com os algoritmos “simulated annealing” serial e algoritmo genético serial, respectivamente.

Tabela 6.93: Resultados gerais com algoritmo “Simulated Annealing” serial - SANORD

algoritmo		SANORD			
itens		Número de Níveis de Temperatura	Número de PPLs	Tempo da Solução ótima	Tempo Médio por PL
sistemas					
Garver c/ Redespacho	s/ inic	5	401	23,6	0,0588
	c/ inic	2	185	10,6	0,0575
Garver s/ Redespacho	s/ inic	7	559	33,1	0,0593
	c/ inic	4	208	11,8	0,0565
Sul Brasileiro c/ Redespacho	s/ inic	3	1.617	5,1 min	0,1904
	c/ inic	4	1.409	4,6 min	0,1949
Sul Brasileiro s/ Redespacho	s/ inic	7	6.071	27,8 min	0,2750
	c/ inic	6	4.426	17,2 min	0,2328
Nor-Nord.-2002	c/ inic	10	55.134	8,2 h ^a	0,5334
Nor-Nord.-2008	c/ inic	9	99.039	19,2 h ^b	0,6980
Colomb.-2012	s/ inic	19	301.086	74,9 h ^c	0,8951

^aótimo local US\$ 1,453,205,000.00 - corte de carga 0 MW

^bótimo local US\$ 2,602,361,000.00 - corte de carga 0 MW

^cótimo local US\$ 560,692,000.00 - corte de carga 0 MW

Tabela 6.94: Resultados gerais com Algoritmo Genético serial - GANORD

algoritmo		GANORD					
itens		Pop.	N. de Gers.	N. de PPLs	Confs. Inics.	Tempo da Solução ótima	Tempo Médio por PL
sistemas							
Garver c/ Redespacho	s/ inic	40	4	409	-	23,6	0,0577
	c/ inic	40	2	130	8	7,5	0,0576
Garver s/ Redespacho	s/ inic	20	15	588	-	34,3	0,0583
	c/ inic	20	6	200	8	12,5	0,0627
Sul Brasileiro c/ Redespacho	s/ inic	40	34	6.410	-	22,2 min	0,2077
	c/ inic	40	16	2.556	20	9,9 min	0,2322
Sul Brasileiro s/ Redespacho	s/ inic	40	37	2.856	-	10,5 min	0,2208
	c/ inic	40	22	3.295	20	12,2 min	0,2213
Nor-Nord.-2002	c/ inic	180	187	66.342	90	11,6 h ^a	0,6297
Nor-Nord.-2008	c/ inic	180	147	89.807	60	23,0 h ^b	0,9230
Colomb.-2012	c/ inic	180	481	278.196	20	66,6 h ^c	0,8619

^aótimo local US\$ 1,503,636,000.00 - corte de carga 39 MW

^bótimo local US\$ 2,594,216,000.00 - corte de carga 2 MW

^cótimo global US\$ 560,002,000.00 - corte de carga 0 MW

A Tabela 6.95 apresenta os resultados da simulação para o algoritmo “simulated annealing” paralelo, para os citados sistemas testes.

Tabela 6.95: Resultados Gerais com Algoritmo “Simulated Annealing” Paralelo - SAPARNOR

algoritmo itens		SAPARNOR					
sistemas		N. de Níveis de Temper.	N. de PPLs	N. de máqs.	Config. Inicial	Tempo da Sol. ótima	Tempo Méd. por PL
Garver c/ Redespacho	s/ inic	2	90	4	-	1,9	0,0215
	c/ inic	2	25	4	1	0,6	0,0224
Garver s/ Redespacho	s/ inic	2	67	4	-	2,1	0,0313
	c/ inic	2	43	4	1	1,0	0,0243
Sul Brasileiro c/ Redespacho	s/ inic	5	2.413	4	-	4,9 min	0,1214
	c/ inic	6	3.879	4	1	7,9 min	0,1223
Sul Brasileiro s/ Redespacho	s/ inic	8	10.428	4	-	22,2 min	0,1275
	c/ inic	6	3.488	4	1	7,9 min	0,1358
Nor-Nord.-2002	c/ inic	16	77.030	4	1	7,9 h ^a	0,3707
Nor-Nord.-2008	c/ inic	13	52.099	4	1	4h ^b	0,2786
Colomb.-2012	s/ inic	16	76.046	4	-	12,3h ^c	0,5837

^aótimo local US\$ 1,421,446,000.00 - corte de carga 10 MW

^bótimo local US\$ 2,599,311,000.00 - corte de carga 4 MW

^cótimo local US\$ 564,142,000.00 - corte de carga 0 MW

Se compararmos os resultados obtidos para os diferentes sistemas simulados, entre as versões serial (Tabela 6.93) e paralela (Tabela 6.95) do “simulated annealing” verifica-se que, para a maioria das simulações, os tempos encontrados com a versão paralela foram bem menores que os da versão serial, sendo o tempo médio por PL um indicador disto.

Nas Tabelas 6.96 a 6.99, são apresentados, respectivamente, os resultados para as versões paralelas de algoritmo genético GAPAR1B, GAPAR2A, GAPAR2B e GAPAR3A, simulados para os diferentes sistemas testes. Estes resultados devem ser comparados com os resultados da Tabela 6.94, ou seja, com a melhor versão serial do algoritmo genético.

E finalmente, na Tabela 6.100, são apresentados os resultados para os diferentes sistemas para o time assíncrono MHEUR.

Tabela 6.96: Resultados Gerais com Algoritmo Genético Paralelo - GAPAR1B

algoritmo		GAPAR1B						
sistemas	itens	N. de	N. de	N. de	Pop.	Confs.	Tempo da	Tempo Méd.
	Gers.	PPLs	máqs.		Inics.	Sol. ótima	por PL	
Garver c/ Redespacho	s/ inic	11	1.015	3	40	-	20,5	0,0202
	c/ inic	2	100	3	40	8	1,9	0,0194
Garver s/ Redespacho	s/ inic	20	754	3	20	-	15,9	0,0211
	c/ inic	7	240	3	20	8	4,1	0,0207
Sul Brasileiro c/ Redespacho	s/ inic	41	16.668	3	120	-	20,5 min	0,0237
	c/ inic	24	914	3	20	8	1,2 min	0,0760
Sul Brasileiro s/ Redespacho	s/ inic	27	3.687	3	40	-	15,4 min	0,1334
	c/ inic	22	2.939	3	40	20	6,5 min	0,1332
Nor-Nord.-2008	c/ inic	83	29.398	3	180	60	2,3 h ^a	1,0454
Colomb.-2012	c/ inic	339	192.642	3	180	20	34,1 h ^b	1,0454

^aótimo local US\$ 2,653,226,000.00 - corte de carga 0 MW^bótimo local US\$ 560,692,000.00 - corte de carga 0 MW

Tabela 6.97: Resultados Gerais com Algoritmo Genético Paralelo - GAPAR2A

algoritmo		GAPAR2A						
sistemas	itens	N. de	N. de	N. de	Pop.	Confs.	Tempo da	Tempo Méd.
		Gerações	PPLs	máqs.		Inics.	Sol. ótima	por PL
Garver c/ Redespacho	s/ inic	5	394	3	120	-	8,2	0,0208
	c/ inic	2	99	3	120	8	2,1	0,0209
Garver s/ Redespacho	s/ inic	14	1.040	3	120	-	21,2	0,0204
	c/ inic	5	320	3	120	8	6,4	0,0202
Sul Brasileiro c/ Redespacho	s/ inic	28	5.612	3	180	-	11,2 min	0,1194
	c/ inic	17	2.387	3	180	20	4,6 min	0,1150
Sul Brasileiro s/ Redespacho	s/ inic	39	5.364	3	120	-	12,2 min	0,1365
	c/ inic	20	1.520	3	120	20	3,5 min	0,1385
Nor-Nord.-2002	c/ inic	177	27.742	3	180	90	2,7 h ^a	0,3529
Nor-Nord.-2008	c/ inic	201	24.002	3	180	60	2,7 h ^b	0,3988
Colomb.-2012	c/ inic	432	100.853	3	200	20	16,8 h ^c	0,6013

^aótimo local US\$ 1,440,537,000.00 - corte de carga 1 MW^bótimo local US\$ 2,624,816,000.00 - corte de carga 0 MW^cótimo global US\$ 560,002,000.00 - corte de carga 0 MW

Tabela 6.98: Resultados Gerais com Algoritmo Genético Paralelo - GAPAR2B

algoritmo		GAPAR2B							
sistemas	itens	Núm. de Gers.	Núm. de PPLs	Núm. de máqs.	Pop.	Confs. Iniciais	troca informs . (gers.)	Tempo Solução ótima	Tempo Médio por PL
Garver c/ Redespacho	s/ inic	7	679	3	120	-	5	15,3	0,0226
	c/ inic	3	275	3	120	8	5	5,7	0,0208
Garver s/ Redespacho	s/ inic	28	876	3	60	-	5	19,2	0,0219
	c/ inic	12	358	3	60	8	5	8,1	0,0225
Sul Brasileiro c/ Redespacho	s/ inic	37	2.071	3	60	-	5	4,1 min	0,1198
	c/ inic	19	1.193	3	60	10	10	2,3 min	0,1142
Sul Brasileiro s/ Redespacho	s/ inic	27	7.395	3	180	-	10	34,5 min	0,2797
	c/ inic	15	2.141	3	180	20	5	4,5 min	0,1275
Nor-Nord.-2002	c/ inic	4	3.708	3	800	400	50	18,7 min ^a	0,3029
Nor-Nord.-2008	c/ inic	99	21.127	3	400	60	5	2,3 h ^b	0,3841
Colomb.-2012	c/ inic	240	21.929	3	100	10	50	3,4 h ^c	0,5518

^aótimo local US\$ 1,356,178,000.00 - corte de carga 2 MW^bótimo local US\$ 2,680,958,000.00 - corte de carga 0 MW^cótimo global US\$ 560,002,000.00 - corte de carga 0 MW

Tabela 6.99: Resultados Gerais com Algoritmo Genético Paralelo - GAPAR3A

algoritmo		GAPAR3A							
sistemas	itens	Núm. de Gers.	Núm. de PPLs	Núm. de máqs.	Pop.	Confs. Inics.	troca informs . (gers.)	Tempo Solução ótima	Tempo Médio por PL
Garver c/ Redespacho	s/ inic	5	903	3	120	-	5	19,4	0,0215
	c/ inic	2	40	3	60	8	5	0,9	0,0219
Garver s/ Redespacho	s/ inic	18	840	3	90	-	5	18,6	0,0221
	c/ inic	3	276	3	120	8	5	6,7	0,0244
Sul Brasileiro c/ Redespacho	s/ inic	26	4.289	3	180	-	20	19,4 min	0,2711
	c/ inic	15	2.502	3	180	20	5	5,4 min	0,1299
Sul Brasileiro s/ Redespacho	s/ inic	24	3.410	3	180	-	5	8,0 min	0,1413
	c/ inic	15	2.161	3	180	20	5	5,2 min	0,1449
Nor-Nord.-2002	c/ inic	480	119.728	3	360	60	100	11,3 h ^a	0,3383
Colomb.-2012	c/ inic	302	26.769	3	100	10	100	4,1 h ^b	0,5523

^aótimo local US\$ 1,418,844,000.00 - corte de carga 12 MW^bótimo local US\$ 560,692,000.00 - corte de carga 0 MW

Tabela 6.100: Resultados Gerais com Times Assíncronos - MHEUR

Algoritmo Itens Sistemas		MHEUR							
		Núm. de Gers.	Núm. de PPLs	Núm. de máqs.	Pop.	Confs. Inics.	Tempo informs . ótima	Tempo Solução PPL	Agentes
Garver c/ Redespacho	s/ inic.	3	170	3	16	-	3,2	0,0191	3 SAM
	c/ inic.	2	115	3	16	1	2,3	0,0203	3 SAM
Garver s/ Redespacho	s/ inic.	4	270	4	16	-	6,3	0,0233	4 SAM
	c/ inic.	4	156	4	16	8	3,3	0,0209	4 SAM
Sul Brasileiro c/ Redespacho	s/ inic.	2	623	4	40	-	1,2 min	0,1156	4 SAM
	c/ inic.	6	5.626	4	40	20	11,4 min	0,1213	4 SAM
Sul Brasileiro s/ Redespacho	s/ inic.	3	2.281	4	40	-	4,6 min	0,1214	4 SAM
	c/ inic.	5	7.382	4	180	20	15,2 min	0,1239	4 SAM
Nor-Nord.-2002	c/ inic.	11	90.707	3	180	1	9,4 h ^a	0,3742	3 SAM
Colomb.-2012	c/ inic	10	50.395	4	160	40	7,9 h ^b	0,5654	4 SAM

^aótimo local US\$ 1,448,463,000.00 - corte de carga 0 MW

^bótimo global US\$ 560,002,000.00 - corte de carga 0 MW

Uma análise comparativa entre a Tabela 6.94 e as Tabelas 6.96 a 6.99, mostra que também as versões paralelas de algoritmos genéticos desenvolvidas mostraram eficiência em obter os valores ótimos globais para os sistemas de pequeno e médio porte, bem como ótimos locais com custo menores para sistemas de grande porte. Os resultados destacam, de maneira geral, uma redução significativa do tempo médio por PL para os sistemas analisados. Houve, com o uso das versões paralelas, uma redução no número de gerações em que se obteve os ótimos globais ou locais, para os diferentes sistemas e como consequência uma diminuição no número de PL necessário para a obtenção destes ótimos.

De uma maneira geral os agentes mostraram-se aptos em obter melhores resultados e em um tempo menor, refletido pela diminuição do tempo médio por PL ou pelo tempo total na obtenção dos ótimos. Dessa forma, os mesmos podem ser agrupados em times assíncronos e processados em uma máquina paralela virtual composta por estações de trabalho funcionando em rede, conforme a proposta apresentada no Capítulo 5, página 57.

Capítulo 7

Conclusões e Sugestões para Futuros Trabalhos

Neste trabalho foram apresentadas diferentes algoritmos para a resolução do problema do planejamento da expansão da transmissão de energia elétrica, tratado do ponto de vista estático, envolvendo as seguintes metaheurísticas: “simulated annealing”, algoritmos genéticos e times assíncronos.

A metaheurística “simulated annealing” desempenha uma busca estocástica do espaço de vizinhança onde são permitidas modificações para a solução atual que aumente o valor da função objetivo (no caso de minimização); ou seja, modificações que reduz a função objetivo são sempre aceitas, enquanto que modificações que a aumentem de um valor são aceitas com uma probabilidade deste valor. Esta probabilidade vai diminuindo com a diminuição da temperatura que segue um determinado plano de resfriamento determinado pelos parâmetros: valor inicial de temperatura, número de iterações para cada nível de temperatura e taxa de resfriamento.

Os algoritmos genéticos apresentam a idéia básica de manter uma população de soluções candidatas que se desenvolve sob um processo seletivo que favorece a sobrevivência do mais preparado, podendo ser vista como uma classe de métodos de busca local que empregam um mecanismo de geração de solução que opera sobre atributos de um conjunto de soluções ao invés de atributos de uma única solução, usando um mecanismo de geração de movimento determinado principalmente pela taxa de recombinação e taxa de mutação.

Os times assíncronos consistem em uma organização de agentes autônomos (novos agentes podem ser inseridos sem necessidade de gerenciamento) que se comunicam assíncronamente (todos os agentes trabalham em paralelo todo o tempo) através de memórias compartilhadas e contém um fluxo de dados cíclico que possibilita iteração e “feedback”. Assim, os algoritmos podem fornecer soluções que são compartilhadas por todos, levando às soluções ótimas ou próximas do ótimo.

Numa análise comparativa entre as metaheurísticas pode-se afirmar que existem algumas similaridades entre as mesmas, ou seja, comparando os algoritmos genéticos com os times assíncronos observa-se que ambos têm um fluxo de dados cíclico, têm uma população de soluções, usam processos randômicos durante a execução (seleção, geração e eliminação de soluções). Comparando

“simulated annealing” com times assíncronos, observa-se que ambos são iterativos (fluxo de dados cíclico), fazem uso de parâmetros aleatórios e pode-se afirmar que existe uma sutil similaridade entre os mesmos quanto ao efeito restritivo que o decréscimo de temperatura e o agente destruidor causam, respectivamente, nas soluções que estão sendo criadas. O decréscimo de temperatura reduz a probabilidade de aceitação de novas soluções enquanto que o agente destruidor (que mantém fixo o tamanho da população) elimina as piores soluções da população.

Na prática, os algoritmos desenvolvidos neste trabalho para um ambiente de processamento distribuído, apresentaram na maioria dos casos, um tempo de processamento inferior aos dos algoritmos seriais usados como referência. A Tabela 7.1 a seguir resume o desempenho dos mesmos, levando em conta os melhores resultados obtidos nas simulações apresentadas nas tabelas 6.93 a 6.100. Nesta tabela, o “*speedup*” é calculado como a relação entre o tempo de processamento do melhor algoritmo serial e o tempo de processamento do algoritmo paralelo em questão, e a *eficiência* (dada em porcentagem) como a relação entre o “speedup” e o número de processadores utilizados. Para análise do desempenho do SAPARNOR, utiliza-se o SANORD como o melhor algoritmo serial e na análise de desempenho dos algoritmos genéticos paralelos GAPAR1B, GAPAR2A, GAPAR2B, GAPAR3A, utiliza-se o GANORD como o melhor algoritmo genético serial. No desempenho do algoritmo MHEUR, leva-se em conta o tipo de agente utilizado.

Tabela 7.1: Desempenho dos algoritmos propostos

Sistemas		Garver		Sul		Norte-Nord.		Colomb.
Algoritmos		c/redesp.	s/redesp.	c/redesp.	s/redesp.	2002	2008	2012
		sp./ef.%	sp./ef.%	sp./ef.%	sp./ef.%	sp./ef.%	sp./ef.%	sp./ef.%
SAPARNOR	s/inic.	12,4/310	15,8/394	1,0/26	1,3/31	-	-	6,0/152
	c/inic.	17,7/442	11,8/295	0,6/15	2,2/54	1,0/26	4,8/120	-
GAPAR1B	s/inic.	1,2/38	2,2/72	1,1/36	0,7/23	-	-	-
	c/inic.	3,9/132	3,0/100	8,3/275	1,9/63	-	10/250	2,0/65
GAPAR2A	s/inic.	2,9/96	1,6/54	2,0/66	0,9/29	-	-	-
	c/inic.	3,6/119	1,9/64	2,2/72	3,5/116	4,3/143	8,5/284	4,0/132
GAPAR2B	s/inic.	1,5/51	1,8/60	5,4/180	0,3/10	-	-	-
	c/inic.	1,3/44	1,5/51	4,3/143	2,7/90	37,2/1241	10,0/333	19,6/653
GAPAR3A	s/inic.	1,2/41	1,8/60	1,1/38	1,3/44	-	-	-
	c/inic.	8,3/278	1,9/62	1,8/60	2,3/78	1,0/34	-	16,2/541
MHEUR	s/inic.	7,4/246	5,3/131	4,3/106	6,0/151	-	-	-
	c/inic.	4,6/154	3,6/89	0,4/10	1,1/28	0,9/29	-	9,5/237

Observa-se na Tabela 7.1 que, por exemplo, o algoritmo GAPAR2B aplicado ao sistema Norte-Nordeste brasileiro, plano 2002, apresenta um “speedup” de 37,2 para 3 processadores, ou seja, 12,4 vezes mais eficiente se comparado ao algoritmo serial GANORD. Por outro lado, para o mesmo sistema e a mesma quantidade de processadores, o algoritmo MHEUR apresenta um “speedup” de 0,9, ou seja, 3,4 vezes menos eficiente em relação ao GANORD. Estas variações no desempenho, deve-se principalmente a divisão da população em subpopulações, que requerem uma nova calibração nos parâmetros do algoritmo paralelo utilizado, e ao momento apropriado para a troca de mensagem (incumbente) entre os processadores. Ou seja, a “correta” calibração dos

parâmetros destes algoritmos é a questão crucial nos seus desempenhos.

Outro fato a ser observado, que explica de um modo geral o bom desempenho dos algoritmos paralelos, é a utilização do time assíncrono inicializador baseado em algoritmos heurísticos contrutivos para gerar as configurações iniciais, aliado ao uso de rotinas para o tratamento dos blocos construtivos associados aos ramos das configurações candidatas. Ou seja, a inicialização dos algoritmos com soluções iniciais factíveis determinadas segundo heurísticas que levam em conta o comportamento dos sistemas em relação ao problema do planejamento da expansão da transmissão propriamente dito, com mais razão do que inicializados com configurações aleatórias; bem como, a manutenção/retirada de todo o bloco construtivo relacionado a um ramo candidato de modo que a configuração não apresente regiões desconexas (ilhamento) e/ou linhas isoladas eletricamente, têm mostrado mais eficiência na obtenção das soluções ótimas globais e/ou das melhores soluções e na melhoria de desempenho em relação ao esforço computacional.

Na construção de soluções iniciais para sistemas de grande porte, deve-se fazer mais de duas cópias por configuração, pois isto possibilita que as melhores características da mesma sobrevivam. Tendo somente uma cópia da configuração, a probabilidade que as características se percam é alta.

Quanto ao comportamento dos algoritmos num ambiente de processamento distribuído, pode-se observar que devido ao uso compartilhado dos recursos computacionais com outros usuários, em algumas simulações, principalmente para sistemas de grande porte, ocorreram travamento dos processos nos processadores, sendo necessário, portanto, dotar os algoritmos de rotinas de tolerância à falhas.

Do ponto de vista matemático, a resolução do problema do planejamento da expansão da transmissão ainda é um problema em aberto para sistemas reais de grande porte. Não se tem uma metodologia ou algoritmo que resolva todos os sistemas. O problema do planejamento estático é um subproblema do planejamento dinâmico e do planejamento multi-estágio coordenado e, portanto, é importante ter ferramentas cada vez mais eficientes para a resolução destes subproblemas. Acredita-se que, com a aplicação das metaheurísticas, representadas pelos diferentes algoritmos propostos, caminhou-se nesta direção.

O uso de outras metaheurísticas, tais como Busca Tabu, GRASP, bem como algoritmos híbridos destas e das metaheurísticas utilizadas neste trabalho, podem ser analisadas para um ambiente de processamento distribuído na resolução do problema.

Conclui-se este trabalho observando-se que com a tendência cada vez maior do uso de redes de computadores nas concessionárias de energia elétrica, aliado à disponibilidade de bibliotecas para processamento paralelo, tem-se o ambiente propício para a utilização de modelos de programação distribuída nos problemas complexos e de natureza combinatória como o problema de planejamento da expansão da transmissão de energia elétrica.

Capítulo 8

Referências Bibliográficas

AVILA-ABASCAL, P.; TALUKDAR, S. N. Cooperative Algorithms and abductive causal networks for the automatic generation of intelligent substation alarm processors. In: ISCAS, 1996. *Proceedings...* 1996.

BAJAY, S. V. Uma revisão crítica do atual planejamento da expansão do setor elétrico brasileiro. In: CONGRESSO BRASILEIRO DE ENERGIA, 9., 2002, Rio de Janeiro. *Anais...* Rio de Janeiro: SBPE, COPPE/UFRJ, Clube de Engenharia, v. 2, 2002. p. 899-905.

BARÁN, B.; KASZKUREWICZ, E.; BHAYA, A. Distributed Asynchronous Team Algorithms: Application to the Load Flow Problem. In: CONFERÊNCIA LATINOAMERICANA DE INFORMÁTICA, 19., 1993, Buenos Aires. *Anais...* Buenos Aires: 1993.

BENDERS, J. F. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, v. 4, 1962. p. 238-252.

BRITO, R. B. S.; BADANHAN, L. F. Expansão da oferta de energia: ciclo de planejamento 2001-2010. In: CONGRESSO BRASILEIRO DE ENERGIA, 9., 2002, Rio de Janeiro. *Anais...* Rio de Janeiro: SBPE, COPPE/UFRJ, Clube de Engenharia, v. 2, 2002. p. 938-948.

CAVALCANTE, V. F.; DE SOUZA, P. S. *Solving the job shop scheduling problem by asynchronous teams*. Campinas: DCC-IMECC-UNICAMP, 1995, 10 p. (Relatório Técnico)

ČERNÝ, V. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications*, n. 45, 1985, p. 41-51.

CHEN, S. Y.; TALUKDAR, S. N.; SADEH, N. M. Job-shop-scheduling by a team of asynchronous agents. In: IJCAI - WORKSHOP ON KNOWLEDGE-BASED PRODUCTION, SCHEDULING AND CONTROL, 1993, Chambery. *Conference Paper*. Chambery: 1993.

DAVIS, L.(editor) *Handbook of genetic algorithms*. New York: Van Nostrand Reinhold, 1991.

DECHAMPS, C.; JAMOULLE, A. Interactive computer program for planning the expansion of meshed transmission networks. *International Journal of Electrical Power Energy Systems*, v. 2, n. 2, p. 103-108, Apr. 1980.

DE OLIVEIRA, S. A.; GALLEGGO, R. A.; ESCOBAR, A.; RESTREPO, R. Parallel combinatorial algorithm for statical planning of a transmission system. In: INTERNATIONAL CONFERENCE ON CAD/CAM, ROBOTICS AND FACTORIES AT THE FUTURE, 17., 2001, Durban. *Proceedings...* Durban: 2001. p. 1183-1191.

DE OLIVEIRA, S. A.; ROMERO, R. A.; GALLEGGO, R. A.; MONTICELLI, A. Algoritmo genético paralelo aplicado ao planejamento da expansão da transmissão. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA - CBA200, 13., 2000, Florianópolis. *Anais...* Florianópolis: 2000. p. 1790-1795. (CD-ROM)

DE OLIVEIRA, S. A.; DE ALMEIDA, C. R. T.; MONTICELLI, A. Times assíncronos inicializador de métodos combinatoriais para planejamento da expansão da transmissão. In: SEMINÁRIO NACIONAL DE PRODUÇÃO E TRANSMISSÃO DE ENERGIA ELÉTRICA - SNPTEE, 15., 1999, Foz do Iguaçu. *Anais...* Foz do Iguaçu: Cigré Brasil/Itaipu Binacional, 1999. Grupo VII-GPL/IT 04.

DE OLIVEIRA, S. A. *Parâmetros ótimos para algoritmo genético aplicado ao planejamento da expansão da transmissão*. Campinas: DSEE/ FEEC/UNICAMP, dezembro/1998, 111 p. (Relatório Técnico)

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, 12., 1998, Uberlândia. *Proceedings...* Uberlândia: SBA-UFU, v. 3, 1998. p. 1029-1034.

DE OLIVEIRA, S. A.; DE ALMEIDA, C. R. T. *Times Assíncronos: protótipos e testes*. Campinas: DSEE/FEEC/UNICAMP, junho/1997, 33 p. (Relatório Técnico)

DE OLIVEIRA, S. A. *Metaheurísticas - um estudo introdutório*. Campinas: DSEE/FEEC/ UNICAMP, dezembro/1996, 58 p. (Relatório Técnico)

DE SOUZA, P. S. *Asynchronous organizations for multi-algorithm problems*. Pittsburgh: Department of Electrical and Computer Engineering, Carnegie Mellon University, 1993. (Thesis, Ph.D.)

GALLEGGO, R. A.; ESCOBAR, A.; ROMERO, R. A.; DE OLIVEIRA, S. A. Hybrid model analysis and constructive algorithms in the performance of a genetic algorithm for the planning of an electric power transmission system. In: INTERNATIONAL CONFERENCE ON CAD/CAM, ROBOTICS AND FACTORIES AT THE FUTURE, 17., 2001, Durban. *Proceedings...* Durban: 2001. p. 850-859.

GALLEGGO, R. A. *Planejamento a longo prazo de sistemas de transmissão usando técnicas de otimização combinatorial*. Campinas, 1997. 122 f. Tese (Doutorado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.

GALLEGO, R. A.; DE OLIVEIRA, S. A. *SANORD - modelo de "simulated annealing" para o planejamento da transmissão de sistemas de grande porte: manual do usuário*. Campinas, FEEC/Unicamp, mar/1997. 6 p.

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, FEEC/Unicamp, mar/1997. 7 p.

GALLEGO, R. A.; DE OLIVEIRA, S. A. *GANORD3 - algoritmo genético para o planejamento da transmissão de sistemas de grande porte com circuitos duplos: manual do usuário*. Ilha Solteira, DEE/FEIS/UNESP, jul/2000. 7 p.

GALLEGO, R. A.; ALVES, A. B.; MONTICELLI, A., ROMERO, R. Parallel simulated annealing applied to long term transmission network expansion planning. *IEEE Transactions on Power Systems*, v. 12, n. 1, p. 181-188, Feb. 1997.

GARVER, L. L. Transmission network estimation using linear programming. *IEEE Transactions on Power Apparatus and Systems*, v. PAS-89, n. 7, p. 1688-1697, Sept./Oct.1970.

GEIST, A. *et al.* *PVM3 users's guide and reference manual*, Oak Ridge National Laboratory, Oak Ridge, 1994.

GOLDBERG, D.E. *Genetic algorithms in search, optimization e machine learning*. Reading: Addison-Wesley, 1989.

GREFENSTETTE, J. *et al.* Genetic algorithms for the traveling salesman problem. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, July 1985, Pittsburgh. *Proceedings* Pittsburgh: 1985.

KALTENBACH, J. C.; PESCHON, J.; GEHRIG, E. H. A mathematical optimization technique for the expansion of electric transmission systems. *IEEE Transactions on Power Apparatus and Systems*, v. PAS-90, n. 1, p. 113-119, Jan. 1970.

KAO, J. H.; HEMMERLE, J. S., PRINZ, F. P. Asynchronous-teams based collision avoidance in PAWS. *EDRC Report*, Carnegie Mellon University, June 1995.

KIRKPATRICK, S.; GELATTI, C. D.; CECCHI, P. M. Optimization by Simulated Annealing. *Science*, n. 22, 1983, p. 671.

LAWLER, E.L. *Combinatorial optimization: Networks and Matroids*. New York: Holt, Rinehart and Winston, 1976.

METROPOLIS, N. *et al.* Equation of state calculations by fast computing machines, *Journal of Chemical Physics*, n. 21, p. 1087-1092, 1953.

MONTICELLI, A.; SANTOS JR., A.; PEREIRA, M. V. F.; CUNHA, S. H.; PRAÇA, J. C. G.;

- PARK, B. J. Interactive transmission network planning using a least-effort criterion. *IEEE Transactions on Power Apparatus and Systems*, v. PAS-101, n. 10, p. 3919-3925, Oct. 1982.
- MURTAGH, B. A.; SAUNDERS, M. A. *MINOS 5.1 user's guide*. Stanford: Stanford University, SOL 83-20R, 1987, 125 p.
- NII, H. P. Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures, Parts I and II, *AI Magazine*, 7:2 and 7:3, 1986.
- OSMAN, I. H.; LAPORTE, G. Metaheuristic: a bibliography, *Annals of Operational Research*, n. 63, p. 513-528, 1996.
- OSMAN, I. H. *An introduction to meta-heuristics*. Canterbury: Institute of Mathematics and Statistics, University of Kent, 1996, 33p. (Technical Report)
- OSMAN, I. H.; CHRISTOFIDES, N. Capacitated clustering problems by hybrid simulated annealing and tabu search, *International Transactions in Operational Research*, n. 1, p. 317-336, 1994.
- 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 Transactions on Power Apparatus and Systems*, v. PAS-104, n. 11, p. 3074-3083, Nov. 1985.
- PEREIRA, M. V. F.; PINTO, L. M. V. G. Application of sensibility analysis of load supplying capability to interactive transmission expansion planning, *IEEE Transactions on Power Apparatus and Systems*, v. PAS-104, n. 2, p. 381-389, Feb. 1985.
- RAMESH, V. C. *Inertial search and asynchronous decompositions*. Pittsburgh: Department of Electrical and Computer Engineering, Carnegie Mellon University, 1994. (Thesis, Ph.D.)
- RODRIGUES, R. F.; DE SOUZA, P. S. *Manipulando conjuntos de soluções multidimensionais através de times assíncronos*. Campinas: DCC-IMECC-UNICAMP, 1996, 6 p. (Relatório Técnico)
- RODRIGUES, R. F.; DE SOUZA, P. S. *Asynchronous teams: a multi-algorithm approach for solving combinatorial optimizations problems*. Campinas: DCC-IMECC-UNICAMP, Nov. 1995, 11p. (Relatório Técnico DCC-95-18)
- ROMERO, R.; MONTICELLI, A.; GARCIA, A.; HAFFNER, S. Test systems and mathematical models for transmission network expansion planning. *IEE Proceedings - Generation, Transmission and Distribution*, v. 149, n. 1, p. 27-36, Jan. 2002.
- ROMERO, R.; GALLEGU, R.A.; MONTICELLI, A. Transmission system expansion planning by simulated annealing. *IEEE Transactions on Power Systems*, v. 11, n. 1, p. 364-369, Feb. 1996.
- ROMERO, R.; MONTICELLI, A. A hierarchical decomposition approach for transmission network expansion planning. *IEEE Transactions on Power Systems*, v.9, n. 1, p. 373-380, Feb. 1994.

- ROMERO, R.; MONTICELLI, A. A zero-one implicit enumeration method for optimizing investments in transmission expansion planning. *IEEE Transactions on Power Systems*, v. 9, n. 3, p. 1385-1391, Aug. 1994.
- SANTOS, A.; FRANÇA, P. M.; SAID, A. As optimization model for long-range transmission expansion planning. *IEEE Transactions on Power Systems*, v. 4, n. 1, p. 94-101, Fev. 1989.
- STEWART, B. S.; LIAW, C. F.; WHITE, C. C. A bibliography of heuristic search throught 1992. *IEEE Transactions on Systems, Man, and Cybernetics*, n. 24, p. 268-293, 1994.
- TALUKDAR, S. N. *et al.* Cooperation schemes for autonomous agents. In: SEMINÁRIO A-TEAMS - TIMES ASSÍNCRONOS PARA PROBLEMAS INDUSTRIAIS, 1., 20 de agosto de 1996, São Paulo. *Anais...* São Paulo: SOFTEX2000, CTI, IPT, 1996. p. 2-36.
- TALUKDAR, S. N.; RAMESH, V. C. A multi-agent technique for contingency constrained optimal power flows. *IEEE Transactions on Power Systems*, v. 9, n. 2, p.855-861, May 1994.
- TALUKDAR, S. N.; DE SOUZA, P. S. Scale efficient organizations. In: INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS, Oct. 18-21, 1992, Chicago. *Proceedings...* New York: IEEE, 1993. p. 1458-1463.
- TODA, M.; KUBO, R.; SAITÔ, N. *Statistical physics*. Berlin: Springer-Verlag, 1983.
- VILLASANA, R.; GARVER, L. L.; SALON, S. J. Transmission network planning using linear programming. *IEEE Transactions on Power, Apparatus and Systems*, v. PAS-104, n. 2, p. 349-356, Feb. 1985.
- WHITLEY, D. The GENITOR algorithm and selection pressure: why ranking based allocation of reproductive trials is best. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 3., 1989, San Mateo. *Proceedings* San Mateo: Morgan Kauffmann Publishers, 1989. p. 391-397.

Capítulo 9

Bibliografia

Nas páginas que se seguem são apresentadas a título de complementação do trabalho, uma bibliografia separada por assunto para eventuais consultas e/ou prosseguimento deste trabalho de pesquisa.

9.1 Gerais

AARTS, E. H. L.; KORST, J. *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing*. Chichester: John Wiley & Sons - Interscience Series in Discrete Mathematics and Optimization, 1989, 272 p.

AHO, A. V.; HOPCROFT, J. E., ULLMAN, J. D. *The design and analysis of computer algorithms*. Reading: Addison-Wesley, 1974.

COOK, S.A. The complexity of theorem-proving procedures. In: ANNUAL ACM SYMPOSIUM ON THE THEORY OF COMPUTING, 3., 1971. *Proceedings...* 1971. p. 151-158.

DA SILVA, E. F. *Um Estudo de Metaheurísticas aplicadas ao Problema "Job Shop Scheduling"*. Rio de Janeiro, 2000. 86 p. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal Fluminense.

DANTZIG, G. B. *Linear programming and extensions*. Princeton: Princeton University Press, 1963.

DIAZ, A.; GLOVER, F.; GHAZIRI, H. M.; GONZÁLEZ, J. L.; LAGUNA, M.; MOSCATO, P.; TSENG, F. T. *Optimización heurística y redes neuronales en dirección de operaciones e ingeniería*. Madri: Editorial Paraninfo, 1996. 235 p., 24 cm. (ISBN 84-283-2269-4)

DOWSLAND, K. Variants of simulated annealing for practical problem solving. In: RAYWARD-SMITH, V. *Applications of modern heuristic methods*. Henley-on-Thames: Alfred Waller Ltd./

UNICOM, 1995.

EGLESE, R. W. Heuristics in operational research. In: BELTON, V.; O'Keefe, B. *Recent developments in operational research* Oxford: Pergamon Press, 1986.

FORD, L. R.; FULKERSON, D. R. *Flows in networks*. Princenton: Princenton University Press, 1962.

GAREY, M. R.; JOHNSON, D. S. *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco: W. H. Freeman and Co., 1979.

GLASS, C. A.; POTTS, C. N.; SHADE, P. Unrelated parallel machine scheduling using local search, *Mathematical and Computer Modelling*, n.20, p.41, 1994.

HASAN, M.; OSMAN, I. H. Local search algorithms for the maximal planar layout problem, *International Transactions in Operations Research*, n. 2, p. 89-106, 1995.

HELD, M.; KARP, R. M. The travelling-salesman problem and minimum spanning trees, *Operations Research*, v. 18, p. 1138-1162, 1970.

HU, T.C. *Integer programming and network flows*. Reading: Addison-Wesley, 1970.

KERR JR., A. *Subacute bacterial endocardites*. Springfield: Charles C. Thomas, 1955.

LASDON, L. S. *Optimisation theory for Large Systems*. New York: MacMillan, 1970.

LAWLER, E. L. et al. *The travelling-salesman problem*. Chichester: John Wiley & Sons, 1985.

LAWLER, E. L. et al. *The traveling salesman problem: a guided tour of combinatorial optimization*. Chichester: Wiley, 1985.

LEE, H.; MURTHY, S.; MORSE, D. Primary production scheduling at steel making industries. *IBM Journal of Research and Development*, v. 40, n. 2, p. 231-252, 1996.

LIN, S.; KERNINGHAN, B. W. An effective heuristic algorithm for the traveling-salesman problem, *Operations Research*, v. 21, p. 498-516, 1973.

LOKKETANGEN, A.; GLOVER, F. Probabilistic move selection in tabu search for 0-1 mixed integer programming. In: OSMAN, I. H., KELLY, J. P., *Metaheuristics: theory and applications*. Boston: Kluwers Academic Publishers, 1995.

MANTAWY, A. H.; ABDEL-MAGID, Y. L.; SELIM, S. Z. Integrating genetic algorithms, tabu search and simulated annealing for the unit commitment problem. *IEEE Transactions on Power Systems*, v. 14, n. 3, p. 829-836, Aug. 1999.

MILLER, C. E.; THATCHER, J. W. *Complexity of computer computations*. New York: Plenum Press, 1972.

- NARA, N. State of the arts of the modern heuristics application to power systems. In: IEEE POWER ENGINEERING SOCIETY WINTER MEETING, Jan. 2000. *Proceedings...* v. 2, 2000. p. 1279-1283.
- NICHOLSON, T. Optimization techniques. In: OPTIMIZATION IN INDUSTRY, 1971, London. *Proceedings...* London: Longman Press, 1971.
- OSMAN, I. H. Heuristics for the generalized assignment problem: simulated annealing and tabu search approaches, *OR Spectrum*, v. 17, n. 4, p. 211-225, Oct. 1995.
- OSMAN, I. H. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problems, *Annals of Operations Research*, n. 41, p. 421-451, 1993.
- PAPADIMITRIOU, C. H.; STEIGLITZ, K. *Combinatorial optimization: algorithms and complexity*. New York: Prentice-Hall, 1982.
- REEVES, C. R. *Modern heuristic techniques for combinatorial problems*, New York: McGraw-Hill Book Company, 1995.
- REEVES, C. R.; BEASLEY, J. Introduction. In: MODERN HEURISTIC TECHNIQUES FOR COMBINATORIAL PROBLEMS, 1993, Oxford. *Proceedings...* Oxford: Blackwell Scientific Publications, 1993.
- RODRIGUES, F. L. *Metaheurística e sistema de suporte à decisão no gerenciamento de recursos florestais*. Viçosa, 2001. 225 p. Tese (Doutorado em Ciência Florestal) - Universidade Federal de Viçosa.
- SCHRIJVER, A. *Theory of linear and integer programming*. Chichester: Wiley, 1986.
- SILVA, M. B. *Metaheurísticas seqüenciais e paralelas para uma generalização do problema do caixeiro viajante*. Rio de Janeiro, 2001. 75 p. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal Fluminense.
- SILVER, A.; VIDAL, R. V. V.; DE WERRA, D. A tutorial on heuristic methods. *European Journal of Operational Research*, n. 5, 1980. p. 153.
- SIMONNARD, M. *Linear programming*. Englewood Cliffs: Prentice Hall, 1966.
- VAN LAARHOVEN, P. J. M. et al. New upper bounds for the football pool problem for 6, 7 and 8 matches. *Journal of Combinatorial Theory A*, 1988.
- VIEIRA, C. E. C. *Metaheurísticas aplicadas ao problema de alocação de canal em sistemas de telecomunicações móveis*. Rio de Janeiro, 2001. 186 p. Dissertação (Mestrado em Sistemas de Comunicação) - Instituto Militar de Engenharia.
- ZANAKIS, S.; EVANS, J.; VAZACOPOULOS, A. Heuristic methods and applications: a category survey. *European Journal of Operational Research*, n. 43, p. 88, 1989.

WHITLEY, D.; STARKWEATHER, T.; FUQUAY, D. Scheduling problems and traveling salesmen: The genetic edge recombination operator. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 3., 1989, San Mateo. *Proceedings...* San Mateo: Morgan Kauffmann Publishers, 1989.

WOODRUFF, D. L.; ZEMEL, Z. Hashing vectors for tabu search, *Annals of Operations Research*, n. 41, p. 123-138, 1993.

9.2 Planejamento da Expansão da Transmissão

ADAMS, R. N.; LAUGHTON, M. A. Optimal planning of power networks using mixed-integer programming: part 1: static and time-phased network synthesis. *Proceedings of Inst. Electr. Eng.*, v. 121, n. 2, p. 139-147, 1974.

BAHIENSE, L.; OLIVEIRA, G. C.; PEREIRA, M.; GRANVILLE, S. A mixed integer disjunctive model for transmission network expansion. *IEEE Transactions on Power Systems*, v. 16, n. 3, p. 560-565, Aug. 2001.

BAJAY, S. V. Uma revisão crítica do atual planejamento da expansão do setor elétrico brasileiro. In: CONGRESSO BRASILEIRO DE ENERGIA, 9., 2002, Rio de Janeiro. *Anais...* Rio de Janeiro: SBPE, COPPE/UFRJ, Clube de Engenharia, v. 2, 2002. p. 899-905.

BALDICK, R.; KAHN, E. Transmission planning issues in a competitive economic environment. *IEEE Transactions on Power Systems*, v. 8, n. 4, p. 1497-1503, Nov. 1993.

BALDWIN, C. J.; DE SALVO, C. A.; HOFFMAN, C. H.; KU, W. S. A model for transmission planning by logic. *IEEE Transactions on Am. Inst. Electr. Eng.*, v. 78, p. 1638-1645, Feb. 1960.

BARROS, J. R. P. *Uma metodologia de planejamento da expansão da transmissão baseada em "trade-off" e indicador de mérito econômico*. Recife, 2000. 150 p. Dissertação (Mestrado em Engenharia Elétrica) Universidade Federal de Pernambuco.

BHAVARAJU, M. P.; BILLINGTON, R. Transission planning using a quantitative reliability criterion. In: POWER IND. COMPUT. APPLICATION CONFERENCE, 6., 1969. *Proceedings...* 1969. p. 115-124.

BINATO, S.; DE OLIVEIRA, G. C.; DE ARAUJO, J. L. A greedy randomized adaptative search procedure for transmission expansion planning under an improved. *IEEE Transactions on Power Systems*, v. 16, n. 2, p. 247-253, Aug. 2000.

BRITO, R. B. S.; BADANHAN, L. F. Expansão da oferta de energia: ciclo de planejamento 2001-2010. In: CONGRESSO BRASILEIRO DE ENERGIA, 9., 2002, Rio de Janeiro. *Anais...* Rio de Janeiro: SBPE, COPPE/UFRJ, Clube de Engenharia, v. 2, 2002. p. 938-948.

CHANDA, R. S.; BHATTACHARJEE, P.K. Application of computer software in transmission expansion planning using variable load structure. *IEEE Electric Power Systems Research*, v. 31, p. 13-20, 1994.

CCPE. *Plano Decenal de Expansão: 2001-2010*. Brasília: Ministério de Minas e Energia, 2001. 270 p.

CONTRERAS, J. *A co-operative game theory approach to transmission planning in power system*. Boulder: University of Colorado at Boulder, 1997. (PhD thesis)

CONTRERAS, J.; WU, F.F. A kernel-oriented algorithm for transmission expansion planning. *IEEE Transactions on Power Systems*, v. 15, n. 4, p. 1434-1440, Nov. 2000.

DA SILVA, E. L.; GIL, H. A.; AREIZA ORTIZ, J. M.; DE OLIVEIRA, G. C.; BINATO, S. Transmission network expansion planning under a tabu search approach. *IEEE Transactions on Power Systems*, v. 16, n. 1, p. 62-68, Feb. 2001.

DA SILVA, E. L.; GIL, H. A.; AREIZA, J. M. Transmission network expansion planning under an improved genetic algorithm. *IEEE Transactions on Power Systems*, v. 15, n. 3, p. 1168-1174, Aug. 2000.

DA SILVA, E. L.; FONTOURA, R. N.; GIL, H. A. Planejamento da expansão da transmissão em ambiente competitivo. In: SEMINÁRIO NACIONAL DE PRODUÇÃO E TRANSMISSÃO DE ENERGIA ELÉTRICA - SNPTEE, 15., 1999, Foz do Iguaçu. *Anais...* Foz do Iguaçu: Cigré Brasil/Itaipu Binacional, 1999. Grupo VII-GPL.

DECHAMPS, C.; JAMOULLE, A. Interactive computer program for planning the expansion of meshed transmission networks. *International Journal of Electrical Power Energy Systems*, v. 2, n. 2, p. 103-108, Apr. 1980.

DE OLIVEIRA, S. A.; GALLEGOS, R. A.; ESCOBAR, A.; RESTREPO, R. Parallel combinatorial algorithm for statical planning of a transmission system. In: INTERNATIONAL CONFERENCE ON CAD/CAM, ROBOTICS AND FACTORIES AT THE FUTURE, 17., 2001, Durban. *Proceedings...* Durban: 2001. p. 1183-1191.

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, 12., 1998, Uberlândia. *Proceedings...* Uberlândia: SBA-UFU, v. 3, 1998. p. 1029-1034.

DE OLIVEIRA, S. A. *Parâmetros ótimos para algoritmo genético aplicado ao planejamento da expansão da transmissão*. Campinas: DSEE/ FEEC/UNICAMP, dezembro/1998, 111 p. (Relatório Técnico)

DE SALVO, C. A.; SMITH, H. L. Automated transmission planning with AC load flow and incremental transmission loss evaluation. *IEEE Transactions on Power Apparatus and Systems*, v. PAS-84, p. 156-163, 1965.

DODU, J.; HERTZ, A.; QUADRAT, J. A new probabilistic model for long-term planning studies of EHV power transmission systems using a stochastic subgradient method. In: POWER SYSTEM COMPUTATION CONFERENCE, 7., 12-17 July 1981, Lausanne. *Proceedings...* Lausanne: Butterworths (Ed.), 1981. p. 125-133.

DODU, J.; MERLIN, A.; DAVID, J. On the search of optimal switching configurations in power transmission systems studies. In: POWER SYSTEM COMPUTATION CONFERENCE, 7., 12-17 July 1981, Lausanne. *Proceedings...* Lausanne: Butterworths (Ed.), 1981. p. 281-289.

DUSONCHET, Y. P.; EL-ABIAD, A. H. Transmission planning using discrete dynamic optimizing. *IEEE Transactions on Power Apparatus and Systems*, v. PAS-92, n. 4, p. 1358-1371, 1973.

ENAMORADO, J. C.; GÓMEZ, T.; RAMOS, A. Multi-area regional interconnection planning under uncertainty. In: POWER SYSTEM COMPUTATION CONFERENCE, 13., June 28 - July 2nd, 1999, Trondheim. *Proceedings...* Trondheim: 1999. p. 599-606.

FISCHL, R. Optimal system expansion planning. A critical overview. In: ERDA & EPRI CONFERENCE, Henniker, 1975. *Proceedings...* Henniker: 1975.

FISCHL, R.; PUNTEL, W. R. Efficient method for computing electric power transmission network sensitivities. In: IEEE POWER ENGINEERING SOCIETY WINTER POWER MEETING, 1972. *Proceedings...* 1972. (paper n. C72 167-0)

FORTUNATO, L. A. M.; ARARIPE NETO, T. A. A.; ALBUQUERQUE, J. C. R.; PEREIRA, M. V. F. *Introdução a Expansão e Operação dos Sistemas de Potência*. Niterói: Universidade Federal Fluminense, 232 p., 1990. (Relatório Técnico)

GALLEGO, R. A.; ESCOBAR, A.; ROMERO, R. A.; DE OLIVEIRA, S. A. Hybrid model analysis and constructive algorithms in the performance of a genetic algorithm for the planning of an electric power transmission system. In: INTERNATIONAL CONFERENCE ON CAD/CAM, ROBOTICS AND FACTORIES AT THE FUTURE, 17., 2001, Durban. *Proceedings...* Durban: 2001. p. 850-859.

GALLEGO, R. A.; ROMERO, R.; MONTICELLI, A. Tabu search algorithm for network synthesis. *IEEE Transactions on Power Systems*, v. 15, n. 2, p. 490-495, 2000.

GALLEGO, R. A.; ROMERO, R.; ESCOBAR, A. Planeación del sistema eléctrico de Pereira usando un algoritmo genético eficiente. In: CONGRESO INTERNACIONAL DE GENERACIÓN, TRANSMISIÓN Y DISTRIBUCIÓN, nov. 1999, Bogotá. *Anales...* Bogotá: CIER, 1999.

GALLEGO, R. A.; MONTICELLI, A.; ROMERO, R. Comparative studies of non-convex optimization methods for transmission network expansion planning. *IEEE Transactions on Power Systems*, v. 13, n. 2, p. 822-828, May 1998.

GALLEGO, R. A.; MONTICELLI, A.; ROMERO, R. Transmission system expansion planning by extended genetic algorithm, *IEEE Proceedings Part C - Generation, Transmission and Distribution*, v. 145, n. 3, p. 329-335, May 1998.

GALLEGO, R. A.; ALVES, A. B.; MONTICELLI, A.; ROMERO, R. Parallel simulated annealing applied to long term transmission network expansion planning. *IEEE Transactions on Power Systems*, v. 12, n. 1, p. 181-188, Feb. 1997.

GALLEGO, R. A. *Planejamento a longo prazo de sistemas de transmissão usando técnicas de otimização combinatorial*. Campinas, 1997. 249 p. Tese (Doutorado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.

GANDINI, A. C. *Método de programação linear dual para o cálculo do corte de cargas em planejamento da expansão da transmissão de energia elétrica*. Campinas, 1998. 100 p. Dissertação (Mestrado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.

GARVER, L. L. Transmission network estimation using linear programming. *IEEE Transactions on Power Apparatus and Systems*, v. PAS-89, n. 7, p. 1688-1697, Sept./Oct. 1970.

GCPS. *Plano Decenal de Expansão: 1999-2008*. Brasília: Eletrobrás, Ministério de Minas e Energia, 1999. 308 p., 30 cm.

GEERT, E. V.; GLENDE, I.; HALBERG, N.; JURESET, R.; MATHIEU, E.; O'riordan, C.; SUNDEL, J. ARAKAMA, F.; BASTOS, L.; BRESSANEM, J. M.; FERN, D.; SHALABI, O.; YERSEVITCH, V.; PETRESCU, A. M.; DOULLIEZ, P. Methods for planning under uncertainty: towards flexibility in power system development. *Electra*, n. 161, p. 143-163, Aug. 1995.

GILLES, M. L. Optimum HVAC transmission expansion planning - a new formulation. *IEEE Transactions on Power Systems*, v. 1, p. 48-56, 1986.

GONEN, T. *Electric power transmission system engineering: analysis and design*. Singapore: John Wiley & Sons, Inc., 1988. Chap. 1, p. 3-24.

GOENSTIN, B. G.; CAMPODÓNICO, N. M.; COSTA, J. P.; PEREIRA, M. V. F. Power system expansion planning under uncertainty. *IEEE Transactions on Power Systems*, v. 8, n. 1, p. 129-136, Feb. 1993.

GRANVILLE, S.; PEREIRA, M. V. F. *Mathematical decomposition techniques for power system expansion planning: decomposition methods and uses*. Stanford University, EPRI-Research Project 2473-6, EPRI EL-5299, v. 1, 1985. (Technical Report)

GRANVILLE, S.; PEREIRA, M. V. F. *Mathematical decomposition techniques for power system expansion planning: analysis of the linearized power flow model in Benders decomposition*. Stanford University, EPRI-Research Project 2473-6, EPRI EL-5299, v. 2, 1985. (Technical Report)

HAFFNER, S.; MONTICELLI, A.; GARCIA, A.; MANTOVANI, J.; ROMERO, R. Branch and bound algorithm for transmission system expansion planning using a transportation model, *IEEE Proceedings - Generation, Transmission and Distribution*, v. 147, n. 3, p. 149-156, May 2000.

HENAULT, P. H.; EASTVEDT, R. B.; PESCHON, J.; HADJU, L. P. Power system long-term planning in the presence of uncertainty. *IEEE Transactions on Power Apparatus and Systems*, v. PAS-89, n. 1, p. 156-163, 1970.

KALTENBACH, J. C.; PESCHON, J.; GEHRIG, E. H. A mathematical optimization technique for the expansion of electric transmission systems. *IEEE Transactions on Power Apparatus and Systems*, v. PAS-90, n. 1, p. 113-119, Jan. 1970.

KIM, K. J.; PARK, Y. M.; LEE, K. Y. Optimal long-term transmission expansion planning based on maximum principle. *IEEE Transactions on Power Systems*, v. 3, p. 1494-1501, Nov. 1988.

LATORE-BAYONA, G.; PEREZ-ARRIAGA, J. I. Chopin: a heuristic model for long term transmission expansion planning. *IEEE Transactions on Power Systems*, v. 9, n. 4, p. 1886-1894, 1994.

LATORRE-BAYONA, G. *Modelos estáticos para la planificación a largo plazo de la red de transporte de energía eléctrica..* Comillas, 1993. Tese (PhD en Ingeniería Eléctrica) - Universidad Pontificia Comillas.

LEE, S. T. Y.; HICKS, K. L.; HNYILICZA, E. Transmission expansion by branch-and-bound integer programming with optimal cost-capacity curves. *IEEE Transactions on Power Apparatus and Systems*, v. PAS-96, n. 2, p. 657-666, 1977.

LEVI, V. A. A new mixed-integer methodology for optimal transmission expansion planning, *Electric Power Systems Research*, v. 32, n. 3, p. 227-238, October 1995.

LEVI, V. A.; CALOVIC, M. S. A new decomposition based method for optimal expansion planning of large transmission networks. *IEEE Transactions on Power Systems*, v. 6, n. 3, p. 937-943, October 1991.

MAMANDUR, K. R.; BERG, G. J. Alternative long-range expansion plans for transmission systems. In: IEEE POWER ENGINEERING SOCIETY WINTER POWER MEETING, 1978. *Proceedings...* 1978. (paper n. A78 042-4)

MANSO, L. A. F. *Um ambiente para planejamento de sistemas de geração e transmissão utilizando o valor da confiabilidade.* Itajubá, 1999. 109 p. Tese (Doutorado em Engenharia Elétrica) - Universidade Federal de Santa Catarina.

MONTICELLI, A.; ROMERO, R. Network planning. In: LEE, K. Y.; EL-SHARKAWI, M. A. (Eds.) *Tutorial on Modern Heuristic Optimization Techniques with Applications to Power Systems*, New York: IEEE Power Engineering Society, 2002. p. 114-129.

MONTICELLI, A.; SANTOS JR., A.; PEREIRA, M. V. F.; CUNHA, S. H.; PRAÇA, J. C. G.; PARK, B. J. Interactive transmission network planning using a least-effort criterion. *IEEE Transactions on Power Apparatus and Systems*, v. PAS-101, n. 10, p. 3919-3925, Oct. 1982.

MONTICELLI, A.; ROMERO, R. *Fundamental of simulated annealing.* In: LEE, K. Y.; EL-SHARKAWI, M. A. (Eds.) *Tutorial on Modern Heuristic Optimization Techniques with Applications to Power Systems*, New York: IEEE Power Engineering Society, 2002. p. 52-66.

NWODO, T. Generation and transmission investment planning of the nigerian power system using a heuristic approach and linear programming. In: POWER SYSTEM COMPUTATION CONFERENCE, 6., Aug. 1978, Darmstadt. *Proceedings...* Darmstadt: v. 1, 1978. p.146-153

OLIVEIRA, G. C.; COSTA, A. P. C.; BINATO, S. Large scale transmission network planning using optimization and heuristic techniques. *IEEE Transactions on Power Systems*, v. 10, n. 4, p. 1828-1833, Nov. 1995.

ORTIZ, J. M. A. *Metodologia da expansão automática da transmissão utilizando um algoritmo de busca tabu*. Florianópolis, 1997. 128 p. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Santa Catarina.

PARK, Y.; LEE, K.; YOUNG, L. New analytical approach for long-term generation expansion planning based on maximum principle and gaussian distribution function. *IEEE Transactions on Power Apparatus and Systems*, v. PAS-104, n. 2, p. 390-397, Feb. 1985.

PEREIRA, M. V. F.; CAMPODÓNICO, N. M.; GORENSTIN, B. G.; COSTA, J. P. *Application of stochastic optimization to power system planning and operation*. In: STOCKHOLM POWER TECH CONFERENCE, June 1995, Stockholm. *Proceedings...* Stockholm: 1999. p. 234-239.

PEREIRA, M. V. F.; PINTO, L. M. V. G.; OLIVEIRA, G. C.; CUNHA, S. H. F. *Composite generation-transmission expansion planning*. Stanford University, EPRI-Research Project 2473-9, 1987. (Technical Report)

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 Transactions on Power Apparatus and Systems*, v. PAS-104, n. 11, p. 3074-3083, Nov. 1985.

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

PLATTS, J. E.; SIGLEY, R. B.; GARVER, L. L. A method for horizon-year transmission planning. In: IEEE POWER ENGINEERING SOCIETY WINTER POWER MEETING, 1972. *Proceedings...* 1972. (paper n. C72 166-2)

PUNTEL, W. R.; REPPEN, N. D.; RINGLE, R. J.; PLATTS, J. E.; RYAN, W. A.; SULLIVAN, P. J. An automated method for long-range planning of transmission networks. In: POWER IND. COMPUT. APPLICATION CONFERENCE, 10., 1973. *Proceedings...* 1973. P. 38-46.

QUIROGA, L. F.; PARRONDO, M.; ROSALES, J. I.; TAMARIT, J. Long term transmission expansion (1974-1993) within a system with a nuclear generation alternative. *CIGRE*, v. 32, n. 7, 1976.

ROMERO, R.; MONTICELLI, A.; GARCIA, A.; HAFFNER, S. Test systems and mathematical models for transmission network expansion planning. *IEE Proceedings - Generation, Transmission and Distribution*, v. 149, n. 1, p. 27-36, Jan. 2002.

ROMERO, R.; MANTOVANI, M.; GALLEGO, R. A.; MONTICELLI, A. Experimental analysis of selection methods in a genetic algorithm applied to the planning electrical transmission. In:

INTERNATIONAL CONFERENCE ON CAD/CAM, ROBOTICS AND FACTORIES AT THE FUTURE, 15., Aug. 1999, Águas de Lindóia. *Proceedings...* Águas de Lindóia: 1999.

ROMERO, R.; GALLEGU, R. A.; MONTICELLI, A. Transmission system expansion planning by simulated annealing. *IEEE Transactions on Power Systems*, v. 11, n. 1, p. 364-369, Feb. 1996.

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

ROMERO, R.; MONTICELLI, A. A zero-one implicit enumeration method for optimizing investments in transmission expansion planning. *IEEE Transactions on Power Systems*, v. 9, n. 3, p. 1385-1391, Aug. 1994.

ROMERO, S.; BINATO, S. Power transmission network expansion planning by a hybrid genetic algorithm. In: SYMPOSIUM OF SPECIALISTS IN ELECTRIC OPERATION AND EXPANSION PLANNING, 6., May 1998, Salvador. *Proceedings...* Salvador: 1998.

SANTOS, A.; FRANÇA, P. M.; SAID, A. An optimization model for long-range transmission expansion planning. *IEEE Transactions on Power Systems*, v. 4, n. 1, p. 94-101, Feb. 1989.

SARVEY, R. M.; ZINN, C. D. A mathematical model for long range expansion planning of generation and transmission in electric utility systems. *IEEE Transactions on Power Apparatus and Systems*, v. PAS-96, n. 2, p. 657-666, 1977.

SEIFU, A.; SALON, S.; LIST, G. Optimization of transmission line planning including security constraints. *IEEE Transactions on Power Systems*, v. 4, n. 4, p. 1507-1513, 1989.

SERNA, C.; DURAN, J.; CAMARGO, A. A model for expansion planning of transmission systems: a practical application example. *IEEE Transactions on Power Apparatus and Systems*, v. PAS-97, n. 2, p. 610-615, 1978.

SHARIFNIA, A.; AASHTIANI, H. Z. Transmission network planning: a method of synthesis of minimum cost secure networks. *IEEE Transactions on Power Apparatus and Systems*, v. PAS-104, n. 8, p. 2026-2034, Aug. 1985.

SJELVGREN, D. V.; BUBENKO, J. A. Nonlinear integer programming for transmission expansion planning. In: IEEE POWER ENGINEERING SOCIETY WINTER POWER MEETING, 1977. *Proceedings...* 1977. (paper n. A77 150-6)

STYCZYNSKI, Z. A. Power network planning using game theory. In: POWER SYSTEM COMPUTATION CONFERENCE, 13., June 28 - July 2nd, 1999, Trondheim. *Proceedings...* Trondheim: 1999. p. 607-613.

TEIVE, R. C. G. *Planejamento da expansão da transmissão de sistemas de energia elétrica utilizando sistemas especialistas*. Florianópolis, 1997. 180 p. Tese (Doutorado em Engenharia de Produção) Universidade Federal de Santa Catarina.

TSAMASPHYROU, P.; REANUD, A.; CARPENTIER, P. Transmission network planning: an efficient Benders decomposition scheme. In: POWER SYSTEM COMPUTATION CONFERENCE, 13., June 28 - July 2nd, 1999, Trondheim. *Proceedings...* Trondheim: 1999. p. 487-494.

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

WEN, F. S.; CHANG, C. S. Transmission network optimal planning using the tabu search method. *Electric Power Systems Research*, v. 42, n. 2, p. 153-163, 1997.

WHYSONG, J. L.; URAM, R.; BROWN, H. E.; KING, C. W.; DE SALVO, C. A. Computer program for automatic transmission planning. *IEEE Transactions on Am. Inst. Electr. Eng.*, v. 81, p. 774-781, 1963.

YOSHIMOTO, K.; YASUDA, K.; YOKOHAMA, R. Transmission expansion planning using neuro-computing hybridized with genetic algorithm. In: INTERNATIONAL CONFERENCE ON EVOLUTIONARY COMPUTING, 1995, Perth. *Proceedings...* Perth: 1995. p. 126-131.

YOUSSEF, H. K.; HACKMAN, R. New transmission planning model. *IEEE Transactions on Power Systems*, v. 4, n. 1, p. 9-18, Feb. 1989.

9.3 “Simulated Annealing”

AARTS, E. H. L.; KORST, J. *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing*. Chichester: John Wiley & Sons - Interscience Series in Discrete Mathematics and Optimization, 1989, 272 p.

AARTS, E. H. L.; VAN LAARHOVEN, P. J. M. Simulated annealing: an introduction. *Statistica Neerlandica*, n. 43, 1988.

AARTS, E. H. L.; KORST, J.; VAN LAARHOVEN, P. J. M. A quantitative analysis of the simulated annealing algorithm: a case study for the traveling salesman problem. *Journal of Statistical Physics*, v. 50, n. 1-2 p. 187-206, Jan. 1988.

AARTS, E. H. L.; VAN LAARHOVEN, P. J. M. Simulated annealing: a pedestrian review of the theory and some applications. In: DEVIJVER, P. A., KITTLER, J. *Pattern Recognition Theory and Applications*. Berlin: Springer-Verlag - NASI Series on Computer and Systems Sciences, n. 30, 1987. p. 179-192.

AARTS, E. H. L.; VAN LAARHOVEN, P. J. M. A new polynomial time cooling schedule. In: INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 1985, Santa Clara. *Proceedings...* Santa Clara: IEEE, 1985. p. 206-208.

ABIDO, M. A. Robust design of multimachine power system stabilisers using simulated annealing. *IEEE Transactions on Energy Conversion*, v. 15, n. 3, p. 297-304, Sep. 2000.

ABRAMSON, D. *Constructing school timetables using simulated annealing: sequential and parallel algorithms*. Melbourne: Royal Melbourne Institute of Technology, 1987. (Technical Report 112069R)

ALFA, A. S.; HERAGU, S. S.; CHEN, M. Y. A 3-OPT based simulated annealing algorithm for vehicle routing problems. *Computers & Industrial Engineering*, v. 21, n. 1-2, p. 635-639, 1991.

ANNAKKAGE, U. D.; NUMNONDA, T.; PAHALAWATHTHA, N. C. Unit commitment by parallel simulated annealing. *IEE Proceedings - Generation, Transmission and Distribution*, v. 142, n. 6, p. 595-600, Nov. 1995.

ARAÚJO, H. A. *Algoritmo Simulated Annealing: uma nova abordagem*. Florianópolis, 2001. 95 p. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal de Santa Catarina.

BANERJEE, P.; JONES, M. A parallel simulated annealing algorithm for standard cell placement on a hypercube computer. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 1986, Santa Clara. *Proceedings...* Santa Clara: IEEE, 1986. p. 34-37.

BEENKER, G. F. M.; CLAASEN, T. A. C. M.; HERMENS, P. W. C. Binary sequences with maximally flat amplitude spectrum. *Philips Journal of Research*, n. 40, p. 289-304, 1985.

- BERNASCONI, J. Low autocorrelation binary sequences: statistics mechanics and configuration space analysis. *Journal de Physique*, n. 48, p. 559-567, 1987.
- BHASKER, J.; SAHNI, S. Optimal linear arrangement of circuit components. *Journal of VLSI and Computer Systems*, n. 2, p. 87-109, 1987.
- BOHACHEVSKY, I.; JOHNSON, M. E.; STEIN, M. L. Generalized simulated annealing for function optimization. *Technometrics*, n. 28, p. 209-217, 1986.
- BONOMI, E.; LUTTON, J. L. The asymptotic behaviour of quadratic sum assignment problems: a statistical mechanics approach. *European Journal of Operational Research*, n. 26, p. 295-300, 1986.
- BONOMI, E.; LUTTON, J. L. The N-city traveling salesman problem: statistical mechanics and the Metropolis algorithm. *SIAM Review*, n. 26, p. 551-568, 1984.
- BURKARD, R. E.; RENDL, F. A thermodynamically motivated simulation procedure for combinatorial optimization problems. *European Journal of Operational Research*, n. 17, p. 169-174, 1984.
- CARNEVALLI, P.; COLETTI, L.; PATARNELLO, S. Image processing by simulated annealing. *IBM Journal of Research and Development*, n. 29, p. 569-579, 1985.
- CASOTTO, A.; ROMEO, F.; SANGIOVANNI-VINCENTELLI, A. L. A parallel simulated annealing for the placement of macro-cells. *IEEE Transactions on Computer-Aided Design*, n. 6, p. 838-847, 1987.
- CASOTTO, A.; ROMEO, F.; SANGIOVANNI-VINCENTELLI, A. L. Placement of standard cells using simulated annealing on the connection machine. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 1987, Santa Clara. *Proceedings...* Santa Clara: IEEE, 1987. p. 350-353.
- ČERNÝ, V. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications*, v. 45, n. 1, p. 41-51, 1985.
- CHAMS, M.; HERTZ, A.; DE WERRA, D. Some experiments with simulated annealing for colouring graphs. *European Journal of Operational Research*, n. 32, p. 260-266, 1987.
- CHANG, H. C.; KUO, C. C. Network reconfiguration in distribution systems using simulated annealing. *Electric Power Systems Research*, v. 29, p. 227-238, 1994.
- CHEN, X. R.; PAHALAWATHTHA, U. D.; KUMBLE, C. S. Design of decentralised output feedback TCSC damping controllers by using simulated annealing. *IEE Proceedings - Generation, Transmission and Distribution*, v. 145, n. 5, p. 553-558, 1998.
- CHEN, W. M.; WONG, Y. X.; PING, X. Flow-shop scheduling by the knowledge of statistical mechanics and annealing. In: IEEE CONFERENCE ON DECISION AND CONTROL, 1., 1987, Los Angeles. *Proceedings...* Los Angeles: IEEE, 1987. p. 642-643.

CHIANG, W. C.; RUSSELL, R. A. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, v. 63, p. 3-27, 1996.

CHIANG, H. D.; JUMEAU, R. J. Optimal network reconfigurations in distribution systems, part I: formulation and a solution methodology. *IEEE Transactions on Power Delivery*, v. 5, n. 4, p. 1902-1909, Oct. 1990.

CHIANG, H. D.; JUMEAU, R. J. Optimal network reconfigurations in distribution systems, part II: solution algorithms and numerical results. *IEEE Transactions on Power Delivery*, v. 5, n. 3, p. 1568-1574, Jul. 1990.

CHIANG, H. D.; WANG, J. C.; COCKINGS, O.; SHIN, H. D. Optimal capacitor placement in distribution systems, part I: a new formulation and the overall problem. *IEEE Transactions on Power Delivery*, v. 5, n. 2, p. 634-642, Apr. 1990.

CHIANG, H. D.; WANG, J. C.; COCKINGS, O.; SHIN, H. D. Optimal capacitor placement in distribution systems, part II: solution algorithms and numerical results. *IEEE Transactions on Power Delivery*, v. 5, n. 2, p. 643-649, Apr. 1990.

COLLINS, N. E.; EGGLESE, R. W.; GOLDEN, B. L. *Simulated annealing - an annotated bibliography*. Cambridge: University Cambridge, 1987.

CONNOLLY, D. An improved annealing scheme for the QAP. *European Journal of Operational Research*, n. 46, p. 93-100, 1990.

CORANA, A. et al. Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm. *ACM Transactions on Mathematical Software*, n. 13, p. 262-280, 1987.

DAREMA-ROGERS, F.; KIRKPATRICK, S.; NORTON, V. A. Parallel algorithms for chip placement by simulated annealing. *IBM Journal of Research and Development*, n. 31, p. 391-402, 1987.

DAVIS, L.; *Genetic algorithms and simulated annealing*. Los Altos: Morgan Kauffmann Publishers, Inc., 1987.

DE BONT, F. M. J.; et al. Placement of shapeable blocks. *Philips Journal of Research*, n. 43, p. 1-19, 1988.

DEVADAS, S.; NEWTON, A. R. Topological optimization of multiple level array logic: on uni and multi-processors. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 1986, Santa Clara. *Proceedings...* Santa Clara: IEEE, 1986. p. 38-41.

DISTANTE, F.; PIURI, V. Optimum behavioral test procedure for VLSI devices. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER DESIGN, 1986, Port Chester. *Proceedings...* Port Chester: IEEE, 1986. p. 31-35.

DRESS, A.; KRÜGER, M. Parsimonious phylogenetic trees in metric spaces and simulated annealing, *Advances in Applied Mathematics*, n. 8, p. 8-37, 1987.

DREXL, A. A simulated annealing approach to the multiconstraint zero-one knapsack problem, *Computing*, n. 40, p. 1-8, 1988.

EGLESE, R. W. A tool for operational research, *European journal of operational research*, n. 46, p. 271-281, 1990.

EGLESE, R.W.; RAND, G.K. Conference seminar timetabling, *Journal of the Operational Research Society*, n. 38, p. 591-598, 1987.

EL GAMAL, A. et al. Using simulated annealing to design good codes. *IEEE Transactions on Information Theory*, n. 33, p. 116-123, 1987.

FELTEN, E.; KARLIN, S.; OTTO, S. W. The traveling salesman problem on a hypercubic, MIMD computer. In: INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING, 1985, St. Charles. *Proceedings...* St. Charles: 1985. p. 6-10.

FLEISHER et al. Simulated annealing as a tool for logic optimization in a CAD environment. In: INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 1985, Santa Clara. *Proceedings...* Santa Clara: IEEE, 1985. p. 203-205.

GEMAN, S. Stochastic relaxation methods for image restoration and expert systems. In: COOPER, D. B., LAUNER, R. L., MCCLURE, D. E. *Automated Image Analysis: Theory and Experiments*. New York: Academic Press, 1987.

GEMAN, S.; GEMAN, D. Bayesian images analysis. In: BIENENSTOCK, E., FOGELMAN SOUTLIE, F., WEISBUCH, G., *Disordered Systems and Biological Organization*. Berlin: Springer-Verlag - NASI Series on Computer and Systems Sciences, n. 20, 1986. p. 301-319.

GEMAN, S., HWANG, C. R. Diffusions for global optimization, *SIAM Journal of Control and Optimization*, n.24, p.1031-1043, 1986.

GEMAN, S.; GEMAN, D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Proceedings of Pattern Analysis and Machine Intelligence*, n. 6, p. 721-741, 1984.

GOLDEN, B. L.; SKISCIM, C. C. Using simulated annealing to solve routing and location problems. *Naval Logistics Research Quarterly*, n. 33, p. 261-279, 1986.

GOLDSTEIN, L.; WATERMAN, M. S. Mapping DNA by stochastic relaxation, *Advances in Applied Mathematics*, n. 8, p. 194-207, 1987.

GONSALVES, G. Logic synthesis using simulated annealing. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER DESIGN, 1986, Port Chester. *Proceedings...* Port Chester: IEEE, 1986. p. 561-564.

GROVER, L. K. A new simulated annealing algorithm for standard cell placement. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 1986, Santa Clara. *Proceedings...* Santa Clara: IEEE, 1986. p. 378-380.

HAJEK, B. Cooling schedules for optimal annealing, *Mathematics of Operations Research*, n. 13, p. 311-329, 1988.

HAJEK, B. A tutorial survey of the theory and applications of simulated annealing. In: IEEE CONFERENCE ON DECISION AND CONTROL, 24., 1985, Ft. Lauderdale. *Proceedings...* Ft. Lauderdale: 1985. p. 755-760.

HSIAO, Y. T.; CHIANG, H. D. Applying network window schema and a simulated annealing technique to optimal VAR planning in large scale power systems. *International Journal of Electrical Power & Energy Systems*, v. 22, n. 1, p. 1-8, Jan. 2000.

IRVING, M. R.; STERLING, M. J. H. Optimal network tearing using simulated annealing. *IEE Proceedings - Generation, Transmission and Distribution*, v. 137, n. 1, p. 69-72, Jan. 1990.

IZQUIERDO, V. B. *Uma proposta de especificação formal e fundamentação Teórica para simulated annealing*. Porto Alegre, 2000. 98 p. Dissertação (Mestrado em Computação) - Universidade Federal do Rio Grande do Sul.

JEPSEN, D. W.; GELATT, C. D. Macro placement by Monte Carlo annealing. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER DESIGN, 1983, Port Chester. *Proceedings...* Port Chester: IEEE, 1983. p. 495-498.

JOHNSON et al. *Optimization by simulated annealing: an experimental evaluation, Part II*. Murray Hill: AT&T Bell Laboratories, 1988. ("preprint")

JOHNSON et al. *Optimization by simulated annealing: an experimental evaluation, Part I*. Murray Hill: AT&T Bell Laboratories, 1987. ("preprint")

KHACHATURYAN, A. G. Statistical mechanics approach in minimizing a multivariable function. *Journal of Mathematical Physics*, n. 27, p. 1834-1838.

KIRKPATRICK, S.; TOULOUSE, G. Configuration space analysis of traveling salesman problems. *Journal of Physique*, n. 46, p. 1277-1292, 1985.

KIRKPATRICK, S. Optimization by simulated annealing: quantitative studies. *Journal of Statistical Physics*, n. 34, p. 975-986, 1984.

KIRKPATRICK, S.; GELATT JR., C. D.; VECCHI, M. Optimization by simulated annealing. *Science*, v. 220, n. 4598, p. 498-516, 1983.

KOULMAS, C.; ANTONY, S. R.; JAEN, R. A survey of simulated annealing applications to operations research problems. *Omega International Journal of Management Science*, v. 22, n. 41-56, Jan. 1994.

- KRAVRITZ, S. A.; RUTENBAR, R. Placement by simulated annealing on a multiprocessor. *IEEE Transactions on Computer-Aided Design*, n.6, p.534-549, 1987.
- LAM, J.; DELOSME, J. M. Logic minimization using simulated annealing. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 1986, Santa Clara. *Proceedings...* Santa Clara: IEEE, 1986. p. 348-351.
- LAURSEN, P. S. Simulated annealing for the QAP - optimal tradeoff between simulation time and solution quality. *European Journal of the Operational Research*, v. 69, n. 2, p. 238-243, Sept. 1993.
- LEE, J. Y.; CHOI, M. Y. Optimization by multicanonical annealing and the traveling salesman problem. *Physical Review E*, v. 50, n. 2, p. 651-654, Aug. 1994.
- LEONG, H. W. A new algorithm for gat matrix layout. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 1986, Santa Clara. *Proceedings...* Santa Clara: IEEE, 1986. p. 316-319.
- LEONG, H. W.; WONG, D.F.; LIU, C.L. A simulated-annealing channel router. In: INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 1985, Santa Clara. *Proceedings...* Santa Clara: IEEE, 1985. p. 226-229.
- LEONG, H. W.; LIU, C. L. Permutation channel routing. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER DESIGN, 1985, Port Chester. *Proceedings...* Port Chester: IEEE, 1985. p. 579-584.
- LIGHTHART, M. M.; AARTS, E. H. L.; BEENKER, F. P. M. A design-for-testability of PLA's using statistical cooling. In: DESIGN AUTOMATION CONFERENCE, 23., 1986, Las Vegas. *Proceedings...* Las Vegas: 1986. p. 339-345.
- LIN, F. T.; KAO, C. Y.; HSU, C. C. Applying the genetic approach to simulated annealing in solving some NP-hard problems. *IEEE Transactions on Systems Man and Cybernetics*, n. 23, 1993, p. 1752-1767.
- LIU, C. W.; JWO, W. S.; LIU, C. C.; HSIAO, Y. T. A fast global optimization approach to VAR planning for the large scale electric power systems. *IEEE Transactions on Power Systems*, v. 12, n. 1, Feb. 1997, p. 437-442.
- LUNDY, S.; MEES, A. Convergence of an annealing algorithm, *Mathematical Programming*, n. 34, p. 111-124, 1986.
- LUNDY, M. Applications of the annealing algorithm to combinatorial problems in statistics, *Biometrika*, n. 72, p. 191-198, 1985.
- LUTTON, J.L.; BONOMI, E. Simulated annealing algorithm for the minimum weighted perfect Euclidean matching problem, *Recherche Opérationnelle*, n. 20, p. 177-197, 1986.

LYBERATOS, A.; WOHLFARTH, P.; CHANTRELL, R.W. Simulated annealing: an application in fine particle magnetism. *IEEE Transactions on Magnetism*, n. 21, p. 1277-1282, 1985.

MANTAWY, A. H.; ABDEL-MAGID, Y. L.; SELIM, S. Z. A simulated annealing algorithm for the unit commitment. *IEEE Transactions on Power Systems*, v. 13, n. 1, p. 197-204, Feb. 1998.

METROPOLIS, N.; ROSENBLUTH, A. W.; ROSENBLUTH, M. N.; TELLER, A. H.; TELLER, E. Equations of state calculations by fast computing machines. *Chem. Phys.*, v. 21, p. 1087-1092, 1958.

MORGENSTERN, C. A.; SHAPIRO, H. D. *Chromatic number approximation using simulated annealing*. Albuquerque: Department of Computer Science, The University of New Mexico, 1986. (Technical Report CS86-1)

MORI, H.; TAKEDA, K. Parallel simulated annealing for power system decomposition. *IEEE Transactions on Power Systems*, v. 9, n. 2, p. 789-795, May 1994.

MURRAY, D. W.; BUXTON, B. F. Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, n. 9, p. 220-226, 1987.

NAHAR, S.; SAHNI, S.; SHRAGOWITZ, E. Simulated annealing and combinatorial optimization. In: DESIGN AUTOMATION CONFERENCE, 23., 1986, Las Vegas. *Proceedings...* Las Vegas: 1986. p. 293-299.

NICHOLSON, D. M., CHOWDDHARY, A., SCHWARTZ, L. Monte Carlo optimization of pair distribution functions - application to the electronic structure of disordered metals. *Physical Review B*, n. 19, p. 1633-1637, 1984.

NOLTE, A.; SCHRADER, R. Simulated annealing and graph coloring. *Combinatorics Probability & Computing*, v. 10, n. 1, p. 29-40, Jan. 2001.

OSMAN, I. H. Heuristics for the generalized assignment problem: simulated annealing and tabu search approaches. *OR Spektrum*, v. 17, n. 4, p. 211-225, 1995.

OTTEN, R. H. J. M.; VAN GINNEKEN, L. P. P. P. Floorplan design using simulated annealing. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 1984, Santa Clara. *Proceedings...* Santa Clara: IEEE, 1984. p. 96-98.

PAXMAN, R. G.; SMITH, W. E., BARRETT, H. H. Two algorithms for use with an orthogonal-view coded-aperture system. *Journal of Nuclear Medicine*, n. 25, p. 700-705, 1984.

PERUSH, M. Simulated annealing applied to a single machine scheduling problem with sequence-dependent setup time and due dates. *Belgian Journal of Operational Research, Statistics, and Computer Science*, n. 27, 1987.

RAM, D. J.; SREENIVAS, T. H.; SUBRAMANIAM, K. G. Parallel simulated annealing algorithms. *Journal of Parallel and Distributed Computing*, v. 37, n. 2, p. 207-212, Sept. 1996.

- RANDELMAN, R. E.; GREY, G. S. N-city traveling salesman problem: optimization by simulated annealing. *Journal of Statistical Physics*, n. 45, p. 885-890, 1986.
- RIPLEY, B. D. Statistics, images and pattern recognition. *Canadian Journal of Statistics*, n. 14, p. 83-102, 1986.
- ROMERO, R.; MONTICELLI, A. Fundamentals of simulated annealing. In: LEE, K. Y.; EL-SHARKAWI, M. A. (Eds.) *Tutorial on Modern Heuristic Optimization Techniques with Applications to Power Systems*, New York: IEEE Power Engineering Society, 2002. p. 52-66.
- ROMERO, R.; GALLEGU, R. A.; MONTICELLI, A. Transmission system expansion planning by simulated annealing. *IEEE Transactions on Power Systems*, v. 11, n. 1, p. 364-369, Feb. 1996.
- ROSE, J. S. et al. Fast, high quality VLSI placement on an MIMD multiprocessor. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 1986, Santa Clara. *Proceedings...* Santa Clara: IEEE, 1986. p. 42-45.
- ROSSIER, Y.; TROYON, M.; LIEBLING, T. M. Probabilistic exchange algorithms and Euclidean traveling salesman problems, *OR Spectrum*, n. 8, p. 151-164, 1986.
- ROTHMAN, D. H. Nonlinear inversion, statistical mechanics, and residual statics estimation, *Geophysics*, n. 50, p. 2784-2796, 1985.
- ROWEN, C.; HENNESSY, J. L. SWAMI: a flexible logic implementation system. In: DESIGN AUTOMATION CONFERENCE, 22., 1985, Las Vegas. *Proceedings...* Las Vegas: 1985. p. 169-175.
- SAIT, S. M.; YOUSSEF, H. *Iterative computer algorithms with applications in engineering*. Los Alamitos: IEEE Computer Society, 1999.
- SASAKI, G. H.; HAJEK, B. The time complexity of maximum matching by simulated annealing, *Journal of the ACM*, n. 35, p. 387-403, 1988.
- SATOH, T.; NARA, K. Maintenance scheduling by using simulated annealing method (for power plants). *IEEE Transactions on Power Systems*, v. 6, n. 2, p. 850-857, May 1991.
- SECHEN, C.; SANGIOVANNI-VICENTELLI, A. L. The Timber-Wolf placement and routing package. *IEEE Journal on Solid State Circuits*, n. 30, p. 510-522, 1985.
- SEMENOVSKAYA, S. V.; KHACHATURYAN, K. A.; KHACHATURYAN, A. G. Statistical mechanics approach to the structure determination of a crystal. *Acta Crystallographica A*, n. 41, p. 268-273, 1985.
- SHARPE, R.; MARKSJO, B. S. Facility layout optimization using the Metropolis algorithm. *Environment and Planning B*, n. 12, p. 443-453, 1985.

- SHARPE, R. et al. An interactive model for the layout of buildings. *Applied Mathematics and Modelling*, n. 9, p. 207-214, 1985.
- SHEILD, J. Partitioning concurrent VLSI simulation programs onto a multiprocessor by simulated annealing, *IEE Proceedings on Computers and Digital Techniques*, n. 134, p. 24-30, 1987.
- SELIM, S. Z.; ALSULTAN, K. A simulated annealing algorithm for the clustering problem. *Pattern Recognition*, v. 24, n. 10, p. 1003-1008, 1985.
- SIARRY, P.; BERGONZI, L., DREYFUS, G. Thermodynamic optimization of block placement. *IEEE Transactions on Computer-Aided Design*, n. 6, p. 211-221, 1987.
- SKISCIM, C. C.; GOLDEN, B. L. Optimization by simulated annealing: a preliminary computational study for the TSP. In: WINTER SIMULATION CONFERENCE, 2., 1983, Arlington. *Proceedings...* Arlington: 1983. p. 523-535.
- SMITH, W. E.; BARRET, H. H.; PAXMAN, R. G. Reconstruction of objects from coded images by simulated annealing. *Optics Letters*, n. 8, p. 199-201, 1983.
- SONTAG, E. D.; SUSSMANN, H. J. Image restoration and segmentation using simulated annealing. In: CONFERENCE ON DECISION AND CONTROL, 24., 1986, Ft. Lauderdale. *Proceedings...* Ft. Lauderdale: 1986. p. 768-773.
- SOURLAS, N. Statistical mechanics and the traveling salesman problem, *Europhysics Letters*, n. 2, p. 919-923, 1986.
- STORER, J. A.; BECKER, J.; NICAS, A. J. Uniform circuit placement. In: BERTOLAZZI, P., LUCCIO, F. (Eds.) *Proceedings of International Workshop on Parallel Computing and VLSI*. Amsterdam: Elsevier Science Publishers, 1985. p. 255-273.
- SYED, P. M.; LUKAS, M. D.; LEE, K. Y.; JUNG, J. W. A Genetic Algorithm Involving a New Hybrid Crossover Method for Constrained Optimal Tuning of Controllers in a Fuel Cell Power Plant. In: INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEM APPLICATIONS TO POWER SYSTEMS, June 2001, Budapest. *Proceedings...* Budapest: 2001. p. 20-24.
- TELLEY, H.; LIEBLING, T. M.; MOCELLIN, A. Reconstruction of polycrystalline structures: a new application of combinatorial optimization, *Computing*, n. 38, p. 1-11, 1987.
- THANGIAH, S. R.; OSMAN, I. H.; SUN, T. *Metaheuristics for vehicle routing problems with the time windows*. Canterbury: Institute of Mathematics and Statistics, University of Kent, 1994. (Working paper UKC/IMS/OR94/8).
- VANDERBILT, D.; LOUIE, S. G., A Monte-Carlo simulated annealing approach to optimization over continuous variables, *Journal of Computational Physics*, n.36, p.259-271, 1984.
- VAN LAARHOVEN, P. J. M. *Theoretical and computational aspects of simulated annealing*. Rotterdam: Erasmus University Rotterdam, 1988. (Thesis, Ph.D.)

- VAN LAARHOVEN, P. J. M.; AARTS, E. H. L.; LENSTRA, J. K. *Solving job shop scheduling problems by simulated annealing*. Eindhoven: Philips Research Laboratories, 1988. ("preprint")
- VAN LAARHOVEN, P. J. M.; AARTS, E. H. L. *Simulated annealing: theory and applications*. Dordrecht: Kluwer Academic Publishers - Mathematics and Its Applications, 1987, 187 p.
- VECCHI, M. P.; KIRKPATRICK, S. Global wiring by simulated annealing. *IEEE Transactions on Computer-Aided Design*, n. 2, p. 215-222, 1983.
- WEBER, M.; LIEBLING, T. M. Euclidean matching problems and the Metropolis algorithm, *Zeitschrift für Operations Research*. n. 30, p. 85-110, 1986.
- WILHELM, M. R.; WARD, T. L. Solving quadratic assignment problems by simulated annealing. *IEEE Transactions on Industrial Engineering Research and Development*, v. 19, n. 1, p. 107-119, Mar. 1987.
- WILLE, L. T. The football pool problem for 6 matches: a new upper bound obtained by simulated annealing. *Journal of Combinatorial Theory A*, n. 45, p. 171-177, 1987.
- WILLE, L. T. Searching potential energy surfaces by simulated annealing, *Nature*, n. 324, p. 46-47, 1986.
- WILLE, L. T. *A bibliography of papers on simulated annealing and related optimization techniques*. Warrington: SERC Daresbury Laboratory, 1986. (Technical Report)
- WILLE, L. T.; VENNIK, J. Electrostatic energy minimization by simulated annealing. *Journal of Physics A*, n. 18, p. 1983-1990, 1985.
- WOLBERG, G.; PAVLIDIS, T. Restoration of binary images using stochastic relaxation with annealing, *Pattern Recognition Letters*, n. 3, p. 375-388, 1985.
- WONG, S. Y. W. An enhanced simulated annealing approach to unit commitment. *Internacional Journal of Electrical Power & Energy Systems*, v. 20, n. 5, p. 359-368, June 1998.
- WONG, K. P.; WONG, Y. W. Short-term hydrothermal scheduling part I: simulated annealing approach. *IEE Proceedings - Generation, Transmission and Distribution*, v. 141, n. 5, p. 497-501, Sept. 1994.
- WONG, K. P.; WONG, Y. W. Short-term hydrothermal scheduling part II: Parallel simulated annealing approach. *IEE Proceedings - Generation, Transmission and Distribution*, v. 141, n. 5, p. 502-506, Sept. 1994.
- WONG, K. P.; FUNG, C. C. Simulated annealing based economic dispatch algorithm. *IEE Proceedings - Generation, Transmission and Distribution*, v. 140, n. 6, p. 509-515, Nov. 1993.
- WONG, D. F.; LEONG, H. W.; LIU, C. L. *Simulated Annealing for VLSI Design*. Boston: Kluwer, 1988.

WONG, D. F.; LIU, C. L. A new algorithm for floorplan design. In: DESIGN AUTOMATION CONFERENCE, 23., 1986, Las Vegas. *Proceedings...* Las Vegas: 1986. p. 101-107.

WOOTEN, F.; WINER, K.; WEAIRE, D. Computer generation of structural models of amorphous Si and Ge. *Physics Review Letters*, n. 54, p. 1392-1395, 1985.

YIP, P. P. C.; PAO, Y. H. A guide evolutionary simulated annealing approach to the quadratic assignment problem. *IEEE Transactions Man and Cybernetics*, v. 24, n. 9, p. 1383-1387, Sept. 1994.

ZHUANG, F.; GALIANA, F. D. Unit commitment by simulated annealing. *IEEE Transactions on Power Systems*, v. 5, n. 1, p. 311-317, Feb. 1990.

ZHU, J. X.; BILBRO, G.; CHOW, M. Y. Phase balancing using simulated annealing. *IEEE Transactions on Power Systems*, v. 14, n. 4, p. 1508-1513, Nov. 1999.

9.4 Algoritmos Genéticos

AARTS, E. H. L.; EIBEN, A. E.; VAN HEE, K. M. A theory of genetic algorithms. *Computing Science Notes*, v. 89, n. 8, 1989.

ABDEL-MAGID, Y. L.; BETTAYEB, M.; DAWOUD, M. M. Simultaneous stabilization of multi-machine power systems using genetic algorithms. *IEEE Transactions on Power Systems*, v. 14, n. 4, p. 1428-1439, Nov. 1999.

ABDEL-MAGID, Y. L.; BETTAYEB, M.; DAWOUD, M. M. Simultaneous stabilization of power systems using genetic algorithms. *IEE Proceedings - Generation, Transmission and Distribution*, v. 144, n. 1, p. 39-44, Jan. 1997.

ABDEL MAGID, Y. L.; DAWOUD, M. M. Tuning of AGC of interconnected reheat thermal systems with genetic algorithms. In: IEEE CONFERENCE PUBLICATION ON MAN AND CYBERNETICS, 3., 1995. *Proceedings...* IEEE, 1995. p. 2622-2627.

ABDEL MAGID, Y. L.; DAWOUD, M. M. Genetic algorithms: applications in load frequency control. In: IEE CONFERENCE PUBLICATION, 1995. *Proceedings...* IEEE, n. 414, 1995. p. 207-213.

ABIDO, M. A.; ABDEL-MAGID, Y. L. Hybridizing rule-based power systems stabilizers with genetic algorithms. *IEEE Transactions on Power Systems*, v. 14, n. 2, p. 600-607, May 1999.

ABRAMSON; ABELA. *A parallel genetic algorithm for solving the school timetabling problem*. Melbourne: Division of Inf. Technology, C.S.I.R.O., Dept. of Communication Electronic Engineering, Royal Melbourne Institute of Technology, April 1991. (Technical Report)

ADELI, H.; CHENG, N. T. Augmented lagrangian genetic algorithm for structural optimization. *Journal of Aerospace Engineering*, v. 7, n. 1, p. 104-, 1994.

AL-DUWAISH, H. N.; AL-HAMOUZ, Z. M. A genetic approach to the selection of the variable structure controller feedback gain. In: IEEE INTERNATIONAL CONFERENCE ON CONTROL APPLICATIONS, Sept. 1-4, 1998, Trieste. *Proceedings...* Trieste: IEEE, 1998. p. 227-231.

ANTONISSE, H. J. A new interpretation of schema notation that overtuns the binary encoding constraint. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 3., 1989. *Proceedings...* Morgan Kaufmann, 1989. p. 86-91.

A systematic approach to performance weights selection in design of robust H_∞ /PSS using genetic algorithms. *IEEE Transactions on Energy Conversion*, v. 11, n. 1, p. 111-117, Mar. 1996.

A computer-aided process planning model based on genetic algorithms. *Computer Op. Research*, v. 22, n. 8, p. 841-856, Mar. 1995.

BAECK, T. Generalized convergence models for tournament and μ , λ selection. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 6. 1995. *Proceedings...* San Francisco: Morgan Kaufmann Publishers, 1995. p. 2-7.

BAGLEY, J. D. *The behavior of adaptive systems which employ genetic and correlation algorithms*. Ann Arbor: University of Michigan, 1967. (Ph.D. Dissertation)

BAKER, J. E.; et al. Reducing bias and inefficiency in the selection algorithm. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS: GENETIC ALGORITHMS AND THEIR APPLICATIONS, 2., 1987. *Proceedings...* Hillsdale: Lawrence Erlbaum Associates, 1987.

BAKER, J. E.; et al. Adaptive selection methods for genetic algorithms. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 1., 1985. *Proceedings...* 1985.

BAKIRTZIS, A.; PETRIDIS, V.; KAZARLIS, S. Genetic algorithm solution to the economic dispatch problem. *IEEE Proceedings - Generation, Transmission and Distribution*, v. 141, n. 4, p. 377-382, July 1994.

BAUER JR., R. *Genetic algorithms and investment strategies*. New York: John Wiley and Sons, 1994.

BEASLEY, D.; BULL, D. R.; MARTIN, R. R. An overview of genetic algorithms: Part 1, fundamentals. *University Computing*, v. 15, n. 2, p. 58-69, 1993.

BEASLEY, D.; BULL, D. R.; MARTIN, R. R. An overview of genetic algorithms: Part 2, research topics. *University Computing*, v. 15, n. 4, p. 170-181, 1993.

BEZDEK, J. C.; BOGGAVAPARU, S.; HALL, L. O.; BENSAID, A. Genetic algorithm guided clustering. In: INTERNATIONAL CONFERENCE ON EVOLUTIONARY COMPUTATION, 1994. *Proceedings...* Piscataway: IEEE Service Center, 1994. p. 34-39.

ROMERO, S. P.; BINATO, S. Genetic Engineering applied to genetic algorithms. In: INTELLIGENT SYSTEM APPLICATIONS TO POWER SYSTEMS, April 4-8 1999, Rio de Janeiro. *Proceedings...* Rio de Janeiro: 1999. p. 42-47.

BLICKLE, T.; THIELE, L. *A comparison of selection schemes used in genetic algorithms*. Swiss Federal Institute of Technology, TIK-Report, n. 11, 2nd version, 1995. (Technical Report)

BOONE, G.; CHIANG, H. D. *Optimal capacitor placement in distribution systems by a genetic algorithm*. IEE, Power Engineering Series, 1992.

CANTU-PAZ, E. Markov chain models of parallel genetic algorithms. *IEEE Transactions on Evolutionary Computation*, v. 4, n. 3, p. 216-226, 2000.

CHANG, H. C.; CHEN, P. H. A novel economic dispatch approach - implementation of genetic algorithm. In: IEEE/KTH STOCKHOLM POER TECH CONFERENCE, June 18-22 1995, Stockholm. *Proceedings...* Stockholm: IEEE/KTH, 1995. p. 48-54.

- CHANG, H. C.; CHEN, P. H. Hydrothermal generation scheduling package: a genetic based approach. *IEE Proceedings - Generation, Transmission and Distribution*, v. 145, n. 4, p. 451-457, July 1998.
- CHEN, P. H.; CHANG, H. C. Genetic aided scheduling of hidraulically coupled plants in hydrothermal coordination. *IEEE Transactions on Power Systems*, v. 11, n. 2, p. 975-981, May 1996.
- CHEN, P. H.; CHANG, H. C. Large-scale economic dispatch by genetic algorithm. *IEEE Transactions on Power Systems*, v. 10, n. 4, p. 1919-1926, 1995.
- CHEN, T. H.; CHERNG, J. T. Optimal phase arrangement of distribution transformes connected to a primary feeder for system unbalance improvement and loss reduction using a genetic algorithm. *IEEE Transactions on Power Systems*, v. 15, n. 3, p. 994-1000, Aug. 2000.
- CHUANG, A. S.; WU, F. An extensible genetic algoritm framework for problem solving in a comom environment. *IEEE Transactions on Power Systems*, v. 15, n. 1, p. 269-275, Feb. 2000.
- COLORNI, A.; DORIGO, M. MANIEZZO, V. *A genetic algorithm to solve the timetable problem*. Milano: Politecnico de Milano, TR n. 90-060, 1990. (Technical Report)
- COLORNI, A.; DORIGO, M. MANIEZZO, V. Genetic algorithms and highly constrained problems: the time-table case. In: INTERNATIONAL WORKSHOP ON PARALLEL PROBLEM SOLVING FROM NATURE, 1., Dortmund, 1990. *Proceedings...* Dortmund: Springer-Verlag, Notes in Computer Science, 1990. p. 55-59.
- CORNE, D.; FANG, H. L.; MELLISH, C. Solving the modular exam scheduling problem with genetic algorithms. In: INTERNATIONAL CONFERENCE ON INDUSTRIAL AND ENGINEERING APPLICATIONS OF ARTIFICIAL INTELLIGENCE & EXPERT SYSTEMS, 6., 1993. *Proceedings...* ISAI, 1993.
- DARREL, W. *A genetic algorithm tutorial*. Colorado State University, Nov. 1993. (Technical Report)
- DA SILVA, E. L.; GIL, H. A.; AREIZA, J. M. Transmission network expansion planning under an improved genetic algorithm. *IEEE Transactions on Power Systems*, v. 15, n. 3, p. 1168-1175, Aug. 2000.
- DACCHILLE, C. M.; DUPOUY, J. F.; VILLASANA, R.; URDANETA, A. J. Short circuit levels adequation by network topology modifications using genetic algorithms. In: IASTED INTERNATIONAL CONFERENCE POWER AND ENERGY SYSTEMS , 2., June 25-28, 2002, Crete. *Proceedings...* Crete: ACTA Press, 2002. p. 311-315. (ISBN 0-88986-336-9)
- DASGUPTA, D.; MCGREGOR, D. R. Thermal unit commitment using genetic algorithms. *IEE Proceedings Part C - Generation, Transmission and Distribution*, v. 141, n. 5, p. 459-465, 1994.
- DAVIS, L.; *Genetic algorithms and simulated annealing*. Los Altos: Morgan Kauffmann Publishers, Inc., 1987.

DAVIS, L. (Ed.) *Handbook of genetic algorithms*. New York: Van Nostrand Reinhold, 1991.

DAVIS, L. Job-Shop scheduling with genetic algorithms. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 1995. *Proceedings...* 1985. p. 136-140.

DE JONG, K. A. Genetic algorithms are NOT function optimizers. In: *Foundations of genetic algorithms 2*. Morgan Kaufman, p. 5-17, 1993.

DE JONG, K. A. An analysis of the behavior of a class of genetic adaptive systems. Ann Arbor: University of Michigan, 1975. (Doctoral Dissertation)

DELFANI, M.; GRANELLI, G. P.; MARANNINO, P.; MONTAGNA, M. Optimal capacitor placement using deterministic and genetic algorithm. *IEEE Transactions on Power Systems*, v. 15, n. 3, p. 1041-1046, Aug. 2000.

DIMEO, R.; LEE, K. Y. Boiler-turbine control system design using a genetic algorithm. *IEEE Transactions on Energy Conversion*, v. 10, n. 4, p. 752-759, Dec. 1995.

DING, H.; EL-KEIB, A. A.; SMITH, R. Optimal clustering of power networks using a genetic algorithms. *Electric Power Systems Research*, v. 30, p. 209-214, 1994.

DING, H.; et al. Optimal clustering of power networks using genetic algorithms. In: BIENNIAL SYMPOSIUM INDUST. ELECT. APPLICATIONS, 3., 1992, Ruston. *Proceedings...* Ruston: LA Tech. University, 1992.

DO BOMFIM, A. L. B.; TARANTO, G. N.; FALCÃO, D. M. Simultaneous tuning of power system damping controllers using genetic algorithms. *IEEE Transactions on Power Systems*, v. 15, n. 1, p. 163-169, Feb. 2000.

DORNELLAS, C. R. R.; FALCÃO, D. M.; BONFIM, A. L. B. Otimização do despacho de reativos utilizando algoritmos genéticos. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, 11., 1996, São Paulo. *Anais...* São Paulo: Sociedade Brasileira de Automática, v. 1, 1996. p. 143-160.

EBERHART, R.; SHI, Y. Comparison between genetic algorithms and particle swarm optimization. In: ANNUAL CONFERENCE ON EVOLUTIONARY PROGRAMMING, 7., March 1998, San Diego. *Proceedings...* Berlin: Springer-Verlag, 1998.

EPRI. *Optimization of the unit commitment problem by a coupled gradient network and a genetic algorithm*. EPRI, TR-103697, 1994. (Technical Report)

ESHELMAN, L. J. Genetic algorithms. In: BACK, T.; FOGEL, D. B. (Eds.) *Handbook of evolutionary computation*. New York: Oxford University Press, 1997.

ESHELMAN, L. J.; SCHAFFER, J. D. Real-coded genetic algorithms and interval-schemata. In: *Foundations of genetic algorithms 2*. Morgan Kaufman, p. 187-202, 1993.

- FALCÃO, D. M.; DO BONFIM, L. B.; DORNELLAS, C. R. R.; TARANTO, G. N. Genetic algorithms in power system optimization. In: SYMPOSIUM OF SPECIALISTS IN ELECTRIC OPERATION AND EXPANSION PLANNING, 5., May 1996, Recife. *Proceedings...* Recife: 1996.
- FALKENAUER, E.; BOUFFOUX, S. A genetic algorithm for job-shop. In: IEEE International Conference on Robotics and Automation, April 1991, Sacramento. *Proceedings...* Sacramento: 1991.
- FANG, H. L.; ROSS, P.; CORNE, D. A promising genetic algorithm approach to job-shop scheduling, rescheduling & open-shop scheduling problems. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 1993. *Proceedings...* 1993. p. 375-382.
- FANG, H. L. *Investigating GAs for scheduling*. Edinburgh: Dept. of Artificial Intelligence, University of Edinburgh, 1992.
- FRASER, A. S. Simulation of genetic systems. *Journal of Theoretical Biology*, v. 2, p. 329-346, 1962.
- FRASER, A. S. Simulation of genetic systems by automatic digital computers: 5-linkage, dominance and epistasis. In: KEMPTHORNE, O. (Ed.) *Biometrical genetics*. New York: Macmillan, 1960. p. 70-83.
- FRASER, A. S. Simulation of genetic systems by automatic digital computers. *Australian Journal of Biological Science*, v. 10, p. 484-499, 1957.
- FUKUYAMA, Y.; CHIANG, H. A parallel genetic algorithm for generation expansion planning. *IEEE Transactions on Power Systems*, v. 11, n. 2, p. 955-961, May 1996.
- GEN, M.; CHENG, R. A survey of penalty techniques in genetic algorithms. In: IEEE CONFERENCE ON EVOLUTIONARY COMPUTATION, 3., 1996. *Proceedings...* IEEE Press, 1996. p. 804-809.
- GLOVER, F.; KELLY, J. P.; LAGUNA, M. Genetic algorithms and tabu search - hybrids for optimization. *Computers & Operations Research*, n. 22, 1995, p. 111.
- GOLDBERG, D. E.; DEB, K. A comparative analysis of selection schemes used in genetic algorithms. In: *Foundations of genetic algorithms*. Morgan kauffman, p. 69-93, 1991.
- GOLDBERG, D. E. *Genetic algorithms in search, optimization and machine learning*. Reading: Addison-Wesley Publishing Company, 1989.
- GOLDBERG, D. E. Computer-aided gas pipeline operation using genetic algorithms and rule learning (doctoral dissertation, University of Michigan). *Dissertation Abstracts International*, v. 44, n. 10, 3174B, 1983.
- GREFENSTETTE, J. J. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 16, n. 1, p. 122-128, 1986.

GREFENSTETTE, J. J. (Ed.) INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS AND THEIR APPLICATIONS, 1985. *Proceedings...* Hillsdale: Lawrence Erlbaum Associates, 1985.

GREFENSTETTE, J. J. GENESIS: a system for using genetic search procedures. In: CONFERENCE ON INTELLIGENT SYSTEMS AND MACHINES, 1984. *Proceedings...* 1984. p. 161-165.

GREFENSTETTE, J. J. *A user's guide to GENESIS*. Nashville: Computer Science Dept., Vanderbilt University, 1984. (Technical Report CS-84-11)

HAIDA, T.; et al. Genetic algorithms approach to voltage optimization. In: INTERNATIONAL FORUM ON APPLICATIONS OF NEURAL NETWORKS TO POWER SYSTEMS, 1991, New York. *Proceedings...* New York: IEEE, 1991.

HAUPT, R.; HAUPT, S. *Practical genetic algorithms*. New York: John Wiley and Sons, 1998.

HOLLAND, J. H. Genetic algorithms. *Scientific American*, v. 267, n. 1, p. 66-73, 1992.

HOLLAND, J. H. *Adaptation in Natural and Artificial Systems*. East Lansing: University of Michigan Press, 1975.

IBA, K. Reactive power optimization by genetic algorithm. *IEEE Transactions on Power Systems*, v. 9, n. 2, p. 685-692, May 1994.

IORNI, A.; DORIGO, M.; MANIEZZO, V. Genetic algorithms: a new approach to the time-table problem. In: AKGUEL, M.; et al. (Eds.) *Combinatorial Optimization*. Springer-Verlag, p. 235-239, 1990.

JIAO, L.; WANG, L. A novel genetic algorithm based in immunity. *IEEE Transactions on Systems, Man, and Cybernetics - Parte A*, v. 30, n. 5, p. 552-561, 2000.

JU, P.; HANDSCHIN, E.; REYER, F. Genetic algorithm aided controller design with application to SVC. *IEE Proceedings - Generation, Transmission and Distribution*, v. 143, n. 3, May 1996, p. 258-262, 1996.

KAGAN, N. *Configuração de redes de distribuição através de algoritmos genéticos e tomada de decisão fuzzy*. São Paulo, 1999. Tese (Livre Docência em Engenharia Elétrica) - Escola Politécnica, Universidade de São Paulo.

KAZARLIS S. A.; BAKIRTZIS, A. G.; PETRIDIS, V. A genetic algorithm solution to the unit commitment problem. *IEEE Transactions on Power Systems*, v. 11, n. 1, p. 83-92, Feb. 1996.

KELAREVA, G.; NEGNEVITSKY, M. Multi-layer genetic algorithm for maintenance scheduling in power systems. In: IASTED INTERNATIONAL CONFERENCE POWER AND ENERGY SYSTEMS, 2., June 25-28, 2002, Crete. *Proceedings...* Crete: ACTA Press, 2002. p. 32-37. (ISBN 0-88986-336-9)

KOZA, J. R. *Genetic programming: on the programming of computers by means of natural selection*. Cambridge: MIT Press, 1992.

KRICHNAKUMAR, K.; GOLDBERG, D. E. Control system optimization using genetic algorithms. *Journal of Guidance, Control, and Dynamics*, v. 15, n. 3, p. 735-740, 1992.

KUNG, C. H.; DEVANEY, M. J.; HUANG, C. M.; KUNG, C. M. Fuzzy-based adaptive digital power metering using a genetic algorithm. *IEEE Transactions on Instrumentation and Measurement*, v. 47, n. 1, p. 183-188, Feb. 1998.

KUO, C. C.; CHANG, H. C. Applying a refined genetic algorithm to network reconfiguration for loss reduction. *IEE of Japan Power & Energy*, 1994.

LAI, L. L.; SICHANIE, A. G.; GWYN, B. J. Comparison between evolutionary programming and a genetic algorithm for fault-section estimation, *IEE Proceedings - Generation, Transmission and Distribution*, v. 145, n. 5, p. 616-620, Sep. 1998.

LAI, L. L.; MA, J. T.; YOKOYAMA, R.; ZHAO, M. Improved genetic algorithm for optimal power flow under both normal and contingent operation states. *Electric Power and Energy Systems*, v. 19, n. 5, p. 287-292, 1997.

LEE, K. Y.; YANG, F. F. Optimal reactive power planning using evolutionary algorithms: a comparative study for evolutionary programming, evolutionary strategy, genetic algorithm, and linear programming. *IEEE Transactions on Power Systems*, v. 13, n. 1, p. 101-108, Feb. 1998.

LEE, K. Y.; BAI, X.; PARK, Y. M. Optimization method for reactive power planning by using a modified simple genetic algorithm. *IEEE Transactions on Power Systems*, v. 10, n. 4, p. 1843-1850, Nov. 1995.

LEGEY, L. F. L.; KAZAY, H. F. Generation expansion planning: a genetic algorithm approach. In: INTELLIGENT SYSTEM APPLICATIONS TO POWER SYSTEMS, April 4-8 1999, Rio de Janeiro. *Proceedings...* Rio de Janeiro: 1999. p. 239-243.

LI, F.; MORGAN, R.; WILLIAMS, D.; SONG, Y. H. Handling constrained power dispatch with genetic algorithms. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS IN ENGINEERING SYSTEMS: INNOVATIONS AND APPLICATIONS, 1., Sept. 12-14 1995, Sheffield. *Proceedings...* Sheffield: IEE/IEEE, 1995. p. 181-187.

LI, F.; MORGAN, R.; WILLIAMS, D.; SONG, Y. H. Constrained economic dispatch scheduling in a real public electricity supply system using a genetic approach. In: UNIVERSITIES POWER ENGINEERING CONFERENCE, 30., 1995, London. *Proceedings...* London: v. 2, 1995. p. 439-442.

LI, F.; SONG, Y. H.; MORGAN, R. Genetic algorithm based optimization approach to power system economic dispatch. In: UNIVERSITIES POWER ENGINEERING CONFERENCE, 29., 1994, Galway. *Proceedings...* Galway: Part 2, 1994. p. 680-683.

- LIEPINS, G. E.; POTTER, W. D. A genetic algorithm approach to multiple-fault diagnosis. In: DAVIS, I. (Ed.) *Handbook of genetic algorithms*. New York: Van Nostrand Reinhold, 1991.
- LIN, W. M.; CHENG, F. S.; TSAY, M. T. Distribution feeder reconfiguration with refined genetic algorithm, *IEEE Proceedings - Generation, Transmission and Distribution*, v. 147, n. 6, p. 348-354, Nov. 2000.
- MA, H.; EL-KEIB, A. A.; SMITH, R. E.; MA, H. Genetic algorithm based approach to thermal unit commitment of electric power systems. *Electric Power Systems Research*, v. 34, n. 1, p. 29-36, July 1995.
- MA, H.; EL-KEIB, A. A.; SMITH, R. E.; MA, H. Genetic algorithm-based approach to economic dispatch of power systems. In: IEEE SOUTHEASTCON, 1994, Miami. *Proceedings...*, Miami: IEEE, 1994. p. 212-216.
- MAFHOUD, S. *Niching methods for genetic algorithms*. Urbana-Champaign: University of Illinois, 1995. (PhD Thesis)
- MAFHOUD, S. *Niching methods for genetic algorithms*. Urbana-Champaign: Illinois Genetic Algorithm Laboratory, University of Illinois, IlliGAL Report n. 95-001, 1995. (Technical Report)
- MAIFELD, T. T.; SHEBLÉ, G. B. Genetic-based unit commitment algorithm. *IEEE Transactions on Power Systems*, v. 11, n. 3, p. 1359-1370, Aug. 1996.
- MAN, K. F.; TANG, K. S.; KWONG, S. Genetic algorithms: concepts and applications. *IEEE Transactions on Industrial Electronics*, v. 43, n. 5, Oct. 1996.
- MANTAWY, A. H.; ABDEL-MAGID, Y. L.; SELIM, S. Z. A new genetic algorithm approach for unit commitment. In: GENETIC ALGORITHMS IN ENGINEERING SYSTEMS: INNOVATIONS AND APLICATIONS, July 2-4 1997. *Proceedings...* 1997. p. 215-220.
- MANTAWY, A. H.; ABDEL-MAGID, Y. L.; SELIM, S. Z. A genetic algorithm with local search for unit commitment. In: INTELLIGENT SYSTEM APPLICATION TO POWER SYSTEMS, July 6-10 1997, Korea. *Proceedings...* Korea: 1997. p. 170-175.
- MEREDITH, D. L.; KUMAR, K. K.; KARR, C. L. The use of genetic algorithms in the design of fuzzy logic controllers. In: WNN-AIND, 1991. *Proceedings...* 1991. p. 695-702.
- MICHALEWICZ, Z. *Genetic algorithms + data structures = evolution programs*. Berlin: Springer, Artificial Intelligence, 1996.
- MILLER, B. L.; GOLDBERG, D. E. *Genetic algorithms, selection schemes, and the varying effects of noise*. Urbana-Champaign: Illinois Genetic Algorithm Laboratory, University of Illinois, IlliGAL Report n. 95-009, 1995. (Technical Report)
- MIRANDA, V.; RANITO, J. V.; PROENÇA, L. M. Genetic algorithms in optimal multistage distribution network planning. *IEEE Transactions on Power Systems*, v. 9, n. 4, p. 1927-1933,

Nov. 1994.

MITCHELL, M. *An introduction to genetic algorithms*. Cambridge: MIT Press, 1996.

MIU, K. N.; CHIANG, H. D.; DARLING, G. Capacitor placement, replacement and control in large-scale distribution systems by a GA-based two-stage algorithm. *IEEE Transactions on Power Systems*, v. 12, n. 3, p. 1160-1166, Aug. 1997.

MORI, H. A Genetic approach to power system topological observability. In: INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, 1991, New York. *Proceedings...* New York, IEEE, 1991.

MORI, H; HORIGUCHI, T. A genetic algorithm based approach to economic load dispatching. In: INTERNATIONAL FORUM ON APPLICATIONS ON NEURAL NETWORKS TO POWER SYSTEMS, 2., 1993, Japan. *Proceedings...* Japan: 1993. p. 145-150.

NAKANO, R. Conventional genetic algorithms for job-shop problems. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 1991. *Proceedings...* 1991. p. 474-479.

NARA, K.; SHIOSE, A.; KITAGAWA, M.; ISHIHARA, T. Implementation of genetic algorithm for distribution systems loss minimum reconfiguration. *IEEE Transactions on Power Systems*, v. 7, n. 3, p. 1044-1051, Aug. 1992.

NEVES, M. R. *Planejamento da expansão de sistemas de distribuição utilizando algoritmos genéticos*. Rio de Janeiro, 1999. 105 p. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Rio de Janeiro.

ORERO, S. O.; IRVING, M. R. A genetic algorithm modelling framework and solution technique for short term optimal hydrothermal scheduling. *IEEE Transactions on Power Systems*, v. 13, n. 2, p. 501-518, May 1998.

ORERO, S. O.; IRVING, M. R. Economic dispatch of generators with prohibited operating zones: a genetic algorithm approach. *IEE Proceedings - Generation, Transmission and Distribution*, v. 143 n. 6, p. 529-534, Nov. 1996.

ORERO, S. O.; IRVING, M. R. Genetic algorithm for generator scheduling in power systems. *International Journal of Electrical Power and Energy System*, v. 18, n. 1, p. 19-26, Jan. 1996.

OZDAMAR, L. A genetic algorithm approach to a general category project scheduling problem. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, v. 29, n. 1, p. 44-59, Feb. 1999.

PARK, J. B.; PARK, Y. M.; WON, J. R.; LEE, K. Y. An improved genetic algorithm for generation expansion planning. *IEEE Transactions on Power Systems*, v. 15, n. 3, p. 916-922, Aug. 2000.

PARK, J. B.; PARK, Y. M.; WON, J. R.; LEE, K. Y. Economic dispatch solutions using an improved genetic algorithm based on the multi-stage method and directional crossover. *Engineering*

Intelligent Systems, v. 7, n. 4, p. 219-225, Dec. 1999.

PARK, J. B.; PARK, Y. M.; WON, J. R.; LEE, K. Y. Least-cost generation expansion planning based on an improved genetic algorithm. In: IEEE POWER ENGINEERING SOCIETY SUMMER MEETING, July 1999. *Proceedings...* July 1999. p. 1043-1047.

PARK, Y. M.; PARK, J. B.; WON, J. R. A genetic algorithms approach for generation expansion planning optimization. In: IFAC SYMPOSIUM ON POWER SYSTEMS AND POWER PLANT CONTROL, 1996, Pergamon. *Proceedings...* Pergamon: IFAC, 1996. p. 257-262.

PATERNI, P.; VITET, S.; BENA, M.; YOKOHAMA, A. Optimal location of phase shifters in the French network by genetic algorithm. *IEEE Transactions on Power Systems*, v. 14, n. 1, p. 37-42, Feb. 1999.

PIMENTEL, W. *Planejamento de rotas aéreas utilizando algoritmos genéticos*. Rio de Janeiro, 2001. 80 p. Dissertação (Mestrado em Sistemas e Computação) - Universidade Federal do Rio de Janeiro.

RAMIREZ-ROSADO, I. J.; BERNAL-AGUSTIN, J. L. Genetic algorithm applied to the design of large power distribution systems. *IEEE Transactions on Power Systems*, v. 13, n. 2, p. 696-703, May. 1998.

REEVES, C. R. *A genetic algorithm for flowshop sequencing*. Coventry: Coventry Polytechnic, 1993. (Working Paper)

REFORMAT, M.; KUFFEL, E.; WOODFORD, D.; PEDRYCZ, W. Application of genetic algorithms for control design in power systems. *IEE Proceedings - Generation, Transmission and Distribution*, v. 145, n. 4, p. 345-354, July 1998.

RIBEIRO FILHO, J. L.; TRELEAVEN, P. C.; ALIPPI, C. Genetic-algorithms programming environments, *Computer*, p. 28-43, June 1994.

RICHARDS, G. G.; YANG, H. D. Distribution system harmonic worst case design using a genetic algorithm. *IEEE Transactions on Power Delivery*, v. 8, n. 3, p. 1484-1491, July 1993.

RICHTER JR., C. W.; SHEBLÉ, G. B. A profit-based unit commitment GA for the competitive environment. *IEEE Transactions on Power Systems*, v. 15, n. 2, p. 715-721, May 2000.

RICHTER JR., C. W.; SHEBLÉ, G. B.; ASLOCK, D. Comprehensive bidding strategies with genetic programming/finite state automata. *IEEE Transactions on Power Systems*, v. 14, n. 4, p. 1207-1212, Nov. 1999.

RICHTER JR., C. W.; SHEBLÉ, G. B. Genetic algorithm evolution of utility bidding strategies for the competitive marketplace. *IEEE Transactions on Power Systems*, v. 13, n. 1, p. 256-261, Feb. 1998.

- ROGERS, A.; PRÜGEL-BENNET, A. Genetic drift in genetic algorithm selection schemes. *IEEE Transactions on Evolutionary Computation*, v. 3, n. 4, p. 298-303, 1999.
- RUDNICK, H.; PALMA, R.; CURA, E; SILVA, C. Economically adapted transmission systems in open access schemes - application of genetic algorithms. *IEEE Transactions on Power Systems*, v. 11, n. 3, p. 1427-1440, 1996.
- RUDOLFH, G. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, v. 5, n. 1, p. 96-101, 1994.
- RUDOLF, A.; BAYRLEITHNER, R. A genetic algorithm for solving the unit commitment problem of a hydro-thermal power systems. *IEEE Transactions on Power Systems*, v. 14, n. 4, p. 1460-1468, Nov. 1999.
- SAITOH, H.; INOUE, K.; TOYODA, J. Genetic algorithm approach to unit commitment problem. In: INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS APPLICATIONS TO POWER SYSTEMS, Sept. 1994, France. *Proceedings...* France: 1994. p. 583-589.
- SALEH, R. A. F.; BOLTON, H. R. Genetic algorithm-aided design of a fuzzy logic stabilizer for a superconducting generator. *IEEE Transactions on Power Systems*, v. 15, n. 4, p. 1329-1335, Nov. 2000.
- SALAMI, M.; CAIN, G. Multiple genetic algorithm processor for the economic power dispatch problems. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS IN ENGINEERING SYSTEMS: INNOVATIONS AND APPLICATIONS, 1., Sept. 12-14 1995, Sheffield. *Proceedings...* Sheffield: IEE/IEEE, 1995. p. 188-193.
- SAVIC, D.; WALTERS, G. Genetic algorithms for least-cost design of water distribution networks. *Journal of Water Resources and Management*, March-April 1997.
- SHEBLÉ, G. B.; MAIFELD, T. T.; BRITTIG, K.; FAHD, G. Unit commitment by genetic algorithm with penalty methods and a comparison of lagrangian search and genetic algorithm-economic dispatch algorithm, *International Journal of Electric Power & Energy Systems*, v. 18, n. 6, p. 339-346, 1996.
- SHEBLÉ, G. B.; BRITTIG, K. Refined genetic algorithm - economic dispatch example. *IEEE Transactions on Power Systems*, v. 10, n. 1, p. 117-124, Feb. 1995.
- SHEBLÉ, G. B.; MAIFELD, T. T. Unit commitment by genetic algorithm and expert system. *Electric Power Systems Research*, v. 30, p. 115-121, 1994.
- SMITH, G. D. Commercial applications of genetic algorithms. In: ADAPTIVE COMPUTING AND INFORMATION PROCESSING, Jan. 25-27, 1994, London. *Proceedings...* Uxbridge: Unicom Seminars, v. 2, 1994. p. 483-506.
- SONG, Y. H.; WANG, G. S.; WANG, P. Y.; JOHNS, A. T. Effective implementation of genetic algorithms on power economic dispatch. *IEE Proceedings - Generation, Transmission and Distribution*,

v. 144, n. 4, p. 377-382, July 1997.

SONG, Y.H.; LI, F.; MORGAN, R.; CHENG, D. T. Y. Effective implementation of genetic algorithms on power economic dispatch. In: INTERNATIONAL POWER ENGINEERING CONFERENCE, March 1995, Singapore. *Proceedings...* Singapore: 1995. p. 268-274.

SRINIVAS, M., PATNAIK, L. M. Genetic algorithms: a survey. *Computer*, p. 17-26, June 1994.

STENDER, J.; HILLEBRAND, E.; KINGDON, J. *Genetic algorithms in optimization, simulation and modeling*. Washington DC: IOS Press, 1994.

SUNDHARARAJAN, S.; PAHWA, A. Optimal selection of capacitors for radial distribution systems using a genetic algorithm. *IEEE Transactions on Power Systems*, v. 9, n. 3, p. 1499-1507, Aug. 1994.

SWARUP, K. S.; YOSHIMI, M.; IZUI, Y.; NAGAI, N. Genetic algorithm approach to environmental constrained economic dispatch. In: INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS APPLICATIONS TO POWER SYSTEMS, Sept.1994, France. *Proceedings...* France: 1994. p. 707-714.

SYSWERDA, G. Uniform crossover in genetic algorithms. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 3., 1989. *Proceedings...* San Mateo: Morgan Kaufmann Publishers, 1989. p. 2-9.

TACKETT, W. A. *Recombination, selection and the genetic construction of computer programs*. April 1994. Thesis (PhD Electric Engineering) - University of Shouthern California.

TANURE, C. Z. *Estudo e aplicação de algoritmo genético para um problema de distribuição dos correios*. Rio de Janeiro, 1999. 100 p. Dissertação (Mestrado em Engenharia de Produção) Pontifícia Universidade Católica do Rio de Janeiro.

TARANTO, G. M.; FALCÃO, D. M. Robust decentralised control design using genetic algorithms in power system damping control. *IEE Proceedings - Generation, Transmission and Distribution*, v. 145, n. 1, p. 1-6, Jan. 1998.

VALENTIM, M. A. X. *Uma metaheurística genética não convencional para uma generalização do problema do caixeiro viajante*. Rio de Janeiro, 1998. 86 p. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal Fluminense.

ZHANG, P.; COONICK, A. H. Coordinated synthesis of PSS parameters in multi-machine power systems using the method of inequalities applied to genetic algorithm. *IEEE Transactions on Power Systems*, v. 15, n. 2, p. 811-816, May 2000.

ZHAO, Y.; EDWARDS, R. M.; LEE, K. Y. Hybrid feedforward and feedback controller design for nuclear steam generators over wide range operation using genetic algorithm. *IEEE Transactions on Energy Conversion*, v. 12, n. 1, p. 100-106, Mar. 1997.

ZHAO, Y. *Hybrid feedforward and feedback control for wide range operation of nuclear steam generators using genetic algorithm*. Pennsylvania State University, May 1996. (Thesis, Master of Science in Electrical Engineering)

WALTERS, D. C.; SHEBLÉ, G. B. Genetic algorithm solution of economic dispatch with valve point loading. *IEEE Transactions on Power Systems*, v. 8, n. 3, p. 1325-1332, Aug. 1993.

WANG, Y.; HANDSCHIM, E. Unit maintenance scheduling in open systems using genetic algorithm. In: IEEE TRANSMISSION AND DISTRIBUTION CONFERENCE, 1999. *Proceedings...* 1999.

WEI, L.; ZENG, Q.; JIANG, T.; YU, J. New heuristic genetic algorithm and its application to electric power system unit combination. *Proceedings of the Chinese Society of Electrical Engineering*, v. 14, p. 67-72, 1994. (In Chinese)

WELSH, M.; MEHTA, P.; DARWISH, M. K. Genetic algorithm and extended analysis optimization techniques for switched capacitor active filters-comparative study. *IEE Proceedings - Electric Power Applications*, v. 147, n. 1, p. 21-26, Jan. 2000.

WEN, J.; CHENG, S.; MALIK, O. P. A synchronous generator fuzzy excitation controller optimally designed with a genetic algorithm. *IEEE Transactions on Power Systems*, v. 13, n. 3, p. 884-889, Aug. 1998.

WEN, F. S.; CHANG, C. S. Probabilistic approach for fault-section estimation in power systems based on a refined genetic algorithm. *IEE Proceedings Generation, Transmission and Distribution*, v. 144, n. 2, p. 160-168, Nov. 1997.

WHITLEY, L. D.; STARKWEATHER, T.; FUQUAY, D. Scheduling problems and traveling salesmen: the genetic edge recombination operator. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 3., 1989. *Proceedings...* San Mateo: Morgan Kaufmann Publishers, 1989. p. 133-140.

WHITLEY, L. D. The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trial is best. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 3., 1989. *Proceedings...* San Mateo: Morgan Kaufmann Publishers, 1989. p. 116-121.

WHITLEY, D. A genetic algorithm tutorial. *Statistics and Computing*, v. 4, p. 65-85, 1994.

WONG, K. P.; LI, A.; LAW, M. Y. Development of constrained-genetic algorithm load-flow method. *IEE Proceedings Generation, Transmission and Distribution*, v. 144, n. 2, p. 91-99, Marh 1997.

WU, Y. G.; HO, C. Y.; WANG, D. Y. A diploid genetic approach to short-term scheduling of hydro-thermal system. *IEEE Transactions on Power Systems*, v. 15, n. 4, p. 1268-1274, Nov. 2000.

XU, J. X.; CHANG, C. S.; WANG, X. W. Constrained multiobjective global optimization of longi-

tudinal interconnected power system by genetic algorithm. *IEE Proceedings - Generation, Transmission and Distribution*, v. 143, n. 5, p. 435-446, Sept. 1996.

YAMADA, T.; NAKANO, R. A genetic algorithm applicable to large-scale job-shop problems. In: PPSN, 1992. *Proceedings...* 1992. p. 281-290.

YANG, H. T.; YANG, P. C.; HUANG, C. L. A parallel genetic algorithm approach to solving the unit commitment problem: implementation on the transputer networks. *IEEE Transactions on Power Systems*, v. 12, n. 2, p. 661-668, May 1997.

YANG, H. T.; YANG, P. C.; HUANG, C. L. Applications of the genetic algorithm to the unit commitment problem in power generation industry. In: IEEE INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS, 4., 1995, Yokohama. *Proceedings...* Yokohama: IEEE, part 1, 1996. p. 267-274.

YIN, X.; GERMANY, N. Investigations on solving the load flow problem by genetic algorithm. *Electric Power Systems Research*, Elsevier, v. 22, 1991.

YOSHIMI, M.; SWARUP, K. S.; IZUI, Y. Optimal economic power dispatch using genetic algorithms. In: INTERNATIONAL FORUM ON APPLICATIONS ON NEURAL NETWORKS TO POWER SYSTEMS, 2., 1993, Japan. *Proceedings...* Japan: 1993. p. 157-162.

9.5 Busca Tabu

ABDINNOUR-HELM, S.; HADLEY, S. W. Tabu search based heuristics for multi-floor facility layout. *International Journal of Production Research*, v. 38, n. 2, p. 365-382, 2000.

ABIDO, M. A.; ABDEL-MAGID, Y. L. Robust design of multimachine power system stabilisers using tabu search algorithm. *IEE Proceedings - Generation, Transmission and Distribution*, v. 147, n. 6, p. 387-394, Nov. 2000.

BAI, X.; SHAHIDEHPOUR, S. M. Hydro-thermal scheduling by tabu search and decomposition method. *IEEE Transactions on Power Systems*, v.11, n. 2, p. 968-974, May 1996.

BARBAROSOGLU, G.; OZGUR, D. A tabu search algorithm for the vehicle routing problem. *Computers & Operations Research*, v. 26, n. 3, p. 255-270, 1999.

BARNES, J. W.; CHAMBERS, J. B. Solving the job shop scheduling problem with tabu search. *IIE Transactions*, v. 27, n. 2, p. 257-263, 1995.

BATTITI, R.; TECCHIOLLI, G. The continuous reactive tabu search: blending combinatorial optimization and stochastic search for global optimization. *Annals of Operational Research*, n. 63, 1996.

BATTITI, R.; TECCHIOLLI, G. The reactive tabu search. *ORSA Journal on Computing*, n. 6, p. 126-140, 1994.

BATTITI, R.; TECCHIOLLI, G. Parallel biased search for combinatorial optimization: genetic algorithms and tabu. *Microprocessors and Microsystems*, n. 16, p. 351-367, 1992.

BAYKASOGLU, A. MOAPPS 1.0: aggregate production planning using the multiple-objective tabu search. *International Journal of Production Research*, V. 39, N. 16, p. 3685-3702, 2001.

CARLTON, W. B.; BARNES, J. W. Solving the traveling salesman problem with time windows using tabu search. *IIE Transactions*, v. 28, n. 8, p. 617-629, 1996.

CHANG, C. S.; LU, L. R.; WEN, F. S. Power system network partitioning using tabu search. *Electric Power Systems Research*, v. 49, n. 1, p. 55-61, 1999.

COSTAMAGNA, E.; FANNI, A.; GIACINTO, G. A tabu search algorithm for the optimization of telecommunication networks. *European Journal of Operational Research*, v. 106, n. 2-3, p. 357-372, 1998.

DANTAS, N. M. *Metaheurística busca tabu aplicada à solução do problema da unidade móvel de pistoneio*. Natal, 2001. 100 p. Dissertação (Mestrado em Sistemas e Computação) Universidade Federal do Rio Grande do Norte.

ECK, B.; PINEDO, M. Good solutions to job scheduling problems via tabu search. In: ORSA/TIMS MEETING, May 10 1989, Vancouver. *Proceedings...* Vancouver: 1989.

- FANNI, A.; MARCHESI, M.; PILO, F.; SERRI, A. Tabu search metaheuristic for designing digital filters, *International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, v. 17, n. 5-6, 1998.
- FIECHTER, C. N. A parallel tabu search algorithm for large traveling salesman problems, *Discrete Applied Mathematics*, v. 51, n. 3, p. 243-267, 1994.
- GAN, D. Q.; QU, Z. H.; CAI, H. Z. Large-scale VAR optimization and planning by tabu search, *Electric Power Systems Research*, v. 39, n. 3, p. 195- 204, 1996.
- GENDREAU, M.; LAPORTE, G.; SEMET, F. A tabu search heuristic for the undirected selective traveling salesman problem, *European Journal of Operational Research*, v. 106, n. 2-3, p. 539-545, 1998.
- GENDREAU, M.; HERTZ, A.; LAPORTE, G. A tabu search heuristic for the vehicle routing problem, *Management Science*, v. 40, n. 10, p. 1276-1290, 1994.
- GLASS, C. A.; POTTS, C. N.; SHADE, P. Unrelated parallel machine scheduling using local search, *Mathematical and Computer Modelling*, n.20, p.41, 1994.
- GLOVER, F.; LAGUNA, M. *Tabu search*. Klüwer Academic Publishers, 1997.
- GLOVER, F. *Tabu search fundamental and uses*. University of Colorado, Graduate School of Business, 1995.
- GLOVER, F.; TAILLARD, E.; DE WERRA, D. A users guide to tabu search. *Annals of Operations Research*, v. 41, p. 3-28, 1993.
- GLOVER, F. Tabu Search - Part II. *ORSA Journal Computing*, v. 2, n. 1, Winter 1990.
- GLOVER, F. Tabu Search - Part I. *ORSA Journal Computing*, v. 1, n. 3, p. 190-206, Summer 1989.
- HANAFI, S.; FREVILLE, A. An efficient tabu search approach for the 0-1 multidimensional knapsack problem. *European Journal of Operational Research*, v. 106, n. 2-3, p. 659-675, 1998.
- HUANG, Y. C.; YANG, H. T.; HUANG, C. L. Solving the capacitor placement problem in a radial distribution system using tabu search approach. *IEEE Transactions on Power Systems*, v. 11, n. 4, p. 1868-1873, Nov. 1996.
- HUSBSCHER, R.; GLOVER, F. Applying tabu search with influential diversification to multiprocessor scheduling. *Computers & Operations Research*, v. 21, 1994.
- JORNSTEN, K.; LOKKETANGEN, A. Tabu search for weighted k-cardinality trees. *Asia-Pacific Journal of Operational Research*, v. 14, n. 2, p. 9-26, 1997.

- KARABOGA, N.; GUNEY, K.; AKDAGLI, A. A new effective patch radius expression obtained by using a modified tabu search algorithm for the resonant frequency of electrically thick circular microstrip antennae. *International Journal of Electronics*, v. 86, n. 7, p. 825-835, 1999.
- KELLY, J. P.; LAGUNA, M.; GLOVER, F. A Study of diversification strategies for the quadratic assignment problem. *Computers & Operations Research*, v. 21, n. 8, p. 885-893, 1994.
- KNOX, J. Tabu search performance on the symmetrical travelling salesman problem. *Computers & Operations Research*, v. 21, n. 8, p. 867-876, 1994.
- KUWABARA, H.; NARA, K.; SATOH, T.; KANEZASHI, M. Distribution systems planning method by tabu search. In: IASTED INTERNATIONAL CONFERENCE POWER AND ENERGY SYSTEMS, 2., June 25-28, 2002, Crete. *Proceedings...* Crete: ACTA Press, 2002. p. 529-535.
- LAGUNA, M. Tabu search tutorial. In: ESCUELA DE VERANO LATINO-AMERICANA DE INVESTIGACIÓN OPERATIVA, 2, 1995. *Tutorial*. 1995. 112 p.
- LAGUNA, M.; KELLY, J. P.; GONZALEZ VELARDE, J. L.; GLOVER, F. Tabu search for the multilevel generalized assignment problem. *European Journal of Operational Research*, v. 82, n. 1, p. 176-189, 1995.
- LAGUNA, M. A guide to implementing tabu search. *Investigacion Operativa*, v. 4, n. 1, Abr. 1994.
- LOKKETANGEN, A.; GLOVER, F. Probabilistic move selection in tabu search for 0-1 mixed integer programming. In: OSMAN, I. H., KELLY, J. P., *Metaheuristics: theory and applications*. Boston: Kluwers Academic Publishers, 1995.
- MANTAWY, A. H.; ABDEL-MAGID, Y. L.; SELIM, S. Z. A new genetic-based tabu search algorithm for unit commitment problem. *Electric Power Systems Research*, v. 49, n. 2, p. 71-78, 1999.
- MANTAWY, A. H.; ABDEL-MAGID, Y. L.; SELIM, S. Z. Unit commitment by tabu search, *IEEE Proceedings Generation, Transmission and Distribution*, v. 145, n. 1, p. 56-64, Jan. 1998.
- MORI, H.; OGITA, Y. A parallel tabu search based method for reconfiguration of distribution systems. In: IEEE POWER ENGINEERING SOCIETY SUMMER MEETING, 2000. *Proceedings...* IEEE PES, Jul. 2000, p. 73-78.
- MOSCATO, P. An introduction to population approaches for optimization and hierarchical objective functions: A discussion on the role of tabu search. *Annals of Operational Research*, n. 41, p. 85-121, 1993.
- NARA, K.; HAYASHI, Y.; MUTO, S.; TUCHIDA, K. A new algorithm for distribution feeder expansion planning in urban area. *Electric Power Systems Research*, v. 46, n. 3, p. 185-193, 1999.
- ONWUBOLU, G. Tabu search-based algorithm for the TOC product mix decision. *International Journal of Production Research*, v. 39, n. 10, p. 2065-2076, 2001.

OSMAN, I. H. Heuristics for the generalized assignment problem: simulated annealing and tabu search approaches. *OR Spektrum*, v. 17, n. 4, p. 211-225, 1995.

OSMAN, I. H. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problems. *Annals of Operations Research*, n. 41, p. 421-451, 1993.

PONNAMBALAM, S. G.; ARAVINDAN, P.; RAJESH, S. V. A tabu search algorithm for job shop scheduling, *International Journal of Advanced Manufacturing Technology*, v. 16, n.10, p. 765-771, 2000.

ROLLAND, E.; PIRKULL, H.; GLOVER, F. Tabu search for graph partitioning. *Annals of Operations Research*, n.63, p. 209-232, 1996.

SAIT, S. M.; YOUSSEF, H. *Iterative computer algorithms with applications in engineering*. Los Alamitos: IEEE Computer Society, 1999.

SCRICH, C. R. *Busca tabu para programação de tarefas em "job-shop" com datas de entrega*. Campinas, 1997. 111 p. Tese (Doutorado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.

SILVA, M. L. *Um algoritmo paralelo assíncrono de busca tabu para o problema de escalonamento de tarefas*. Rio de Janeiro, 1998. 76 p. Dissertação (Mestrado em Computação Aplicada e Automação) - Universidade Federal Fluminense.

SKORINKAPOV, J. Extensions of a tabu search adaptation to the quadratic assignment problem. *Computers & Operations Research*, v. 21, n. 8, p. 855-865, 1994.

SORIANO, P.; GENDREAU, M. Diversification strategies in tabu search algorithm for the maximum clique problem. *Annals of Operations Research*, n. 21, p. 885, 1994.

SUN, M.; ARONSON, J. E.; MCKEOWN, P. G.; DRINKA, D. A tabu search heuristic procedure for the fixed charge transportation problem. *European Journal of Operational Research*, v. 106, n. 2-3, p. 441-456, 1998.

TAYACHI, D.; CHRÉTIENNE, P.; MELLOULI, K. Une méthode tabou pour l'ordonnancement multiprocesseur avec délais de communication. *Operations Research*, v. 34, n. 4, p. 467-485, 2000.

THANGIAH, S. R.; OSMAN, I. H.; SUN, T. *Metaheuristics for vehicle routing problems with the time windows*. Canterbury: Institute of Mathematics and Statistics, University of Kent, 1994. (Working paper UKC/IMS/OR94/8).

VASCONCELOS, M. G. P. *Um Framework Orientado a Objetos para Aplicações Baseadas em Busca Tabu*. Rio de Janeiro, 1998. 83 p. Dissertação (Mestrado em Computação Aplicada e Automação) - Universidade Federal Fluminense.

VEERAMANI, D.; STINNES, A. H.; SANGHI, D. Application of tabu search to process plan optimization for four-axis CNC turning centres. *International Journal of Production Research*, v.

37, n. 16, p. 3803-3822, 1999.

WOODRUFF, D. L.; ZEMEL, Z. Hashing vectors for tabu search. *Annals of Operations Research*, n. 41, p. 123-138, 1993.

WANG, Y. C.; YANG, H. T.; HUANG, C. L. Solving the capacitor placement problem in a radial distribution on power systems. *IEEE Transactions on Power Systems*, v. 11, n. 4, p. 1868-1873, 1996.

WEN, F. S.; CHANG, C. S. Possibilistic-diagnosis theory for fault-section estimation and state identification of unobserved protective relays using tabu-search method. *IEE Proceedings Generation, Transmission and Distribution*, v. 145, n. 6, p. 722-730, Nov. 1998.

WEN, F. S.; CHANG, C. S. Tabu search approach to alarm processing in power systems, *IEE Proceedings Generation, Transmission and Distribution*, v. 144, n. 1, p. 31-38, 1997.

WEN, F. S.; CHANG, C. S. A tabu search approach to fault section estimation in power systems, *Electric Power Systems Research*, v. 40, n. 1, p. 63-73, 1997.

WU, J. F.; KELLY, J. P. A network flow based tabu search heuristic for the vehicle routing problem, *Transportation Science*, v. 30, n. 4, p. 379-392, 1996.

9.6 Algoritmos Híbridos

ABDEL GHANY, A. M. Adaptive discrete variable structure control using neuro-genetic algorithm application to a dc permanent magnet motor. In: IASTED INTERNATIONAL CONFERENCE POWER AND ENERGY SYSTEMS, 2., June 25-28, 2002, Crete. *Proceedings...* Crete: ACTA Press, 2002. p. 699-707.

ABDEL GHANY, A. M. Adaptive discrete-time PI load frequency control controllers for an interconnected multi-area power system using neuro-genetic technique. In: INTERNATIONAL CONFERENCE AI-AZHAR ENGINEERING, 6., Sept. 1-4, 2000, Cairo. *Proceedings...* Cairo: 2000.

BURKE, E. K.; SMITH, A. J. Hybrid evolutionary techniques for the maintenance scheduling problem. *IEEE Transactions on Power Systems*, v. 15, n. 1, p. 122-128, Feb. 2000.

CHENG, C. P.; LIU, C. W.; LIU, C. C. Unit commitment by lagrangian relaxation and genetic algorithm. *IEEE Transactions on Power Systems*, v. 15, n. 2, p. 707-714, May 2000.

GLOVER, F.; KELLY, J. P.; LAGUNA, M. Genetic algorithms and tabu search - hybrids for optimization. *Computers & Operations Research*, v. 22, n. 1, p. 111, 1994.

GLOVER, F. Artificial intelligence, heuristic frameworks and tabu search. *Managerial and Decision Economics*, v. 11, p. 365-375, 1990.

HSIAO, Y. T.; CHIEN, C. Y. Enhancement of restoration service in distribution systems using a combination fuzzy-GA method. *IEEE Transactions on Power Systems*, v. 15, n. 4, p. 1394-1400, Nov. 2000.

HUANG, S. J. Application of genetic based fuzzy systems to hidroelectric generation scheduling through. *IEEE Transactions on Energy Conversion*, v. 14, n. 3, p. 724-730, Aug. 1999.

HUANG, S. J.; HUANG, C. L. Application of genetic based neural networks to thermal unit commitment. In: IEEE-PES WINTER POWER MEETING, Jan. 21-25, 1996, Baltimore. *Proceedings...* Baltimore: IEEE-PES, paper 96 WM 203-0 PWRS, 1996.

JWO, W. S.; LIU, C. W.; LIU, C. C.; HSIAO, Y. T. Hybrid expert system and simulated annealing approach to optimal reactive power planning. *IEE Proceedings - Generation, Transmission and Distribution*, v. 142, n. 4, p. 381-385, July 1995.

KARR, C. L. Applying genetics to fuzzy logic. *AI Expert*, v. 6, n. 3, p. 38-43, 1991.

KARR, C. L. Genetic algorithms for fuzzy controllers. *AI Expert*, v. 6, n. 2, p. 26-33, 1991.

KARR, C. L.; STANLEY, D. A. Fuzzy logic and genetic algorithms in time-varying control problems. In: NAFIPS, 1991. *Proceedings...* 1991. p. 285-290.

KENNEDY, J.; SPEARS, W. Matching algorithm to problems: an experimental test of the particle swarm optimization and some genetic algorithms on the multimodal problem generator. In: IEEE INTERNATIONAL CONFERENCE ON EVOLUTIONARY COMPUTATION, May, 1998, Anchorage. *Proceedings...* Anchorage: IEEE, 1998. p. 78-83.

KIM, H.; HAYASHI, Y.; NARA, K. An algorithm for thermal unit maintenance scheduling through combined use of GA, SA and TS. *IEEE Transactions on Power Systems*, v. 12, n. 1, p. 329-335, Feb. 1997.

MANTAWY, A. H.; ABDEL-MAGID, Y. L.; SELIM, S. Z. Integrating genetic algorithms, tabu search and simulated annealing for the unit commitment problem. *IEEE Transactions on Power Systems*, v. 14, n. 3, p. 829-836, Aug. 1999.

MANTAWY, A. H.; ABDEL-MAGID, Y. L.; SELIM, S. Z. A new simulated annealing-based tabu search algorithm for unit commitment. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS, Oct. 12-15 1997, Orlando. *Proceedings...* Orlando: 1997.

MEDSKER, L. R. *Hybrid intelligent systems*. Boston: Kluwer Academic Pub., 1995.

MEREDITH, D. L.; KUMAR, K. K.; KARR, C. L. The use of genetic algorithms in the design of fuzzy logic controllers. In: WNN-AIND, 1991. *Proceedings...* 1991. p. 695-702.

MIRANDA, V.; NAING, W. O.; FIDALGO, J. N. Experimenting in the optimal capacitor placement and control problem with hybrid mathematical-genetic algorithms. In: INTELLIGENT SYSTEMS APPLICATIONS IN POWER SYSTEMS CONFERENCE, 2001, Budapest. *Proceedings...* Budapest: 2001.

MIRANDA, V.; PUN, H. S. Unit commitment with a genetic algorithm, probabilistic spinning reserve and fuzzy wind dispatch - the CARE project (EU Joule Programme). In: NATIONAL SYMPOSIUM ON SECURITY IN THE OPERATION OF POWER SYSTEMS, 16., Sept. 22-23 1999, Bacau. *Proceedings...* Bacau: ICEMENERG S. A., 1999.

MIRANDA, V.; PROENÇA, L. M. A general methodology for distribution planning under uncertainty, including genetic algorithms and fuzzy models in a multi-criteria environment. In: IEEE STOCKHOLM POWER TECH CONFERENCE, June 18-22, 1995, Stockholm. *Proceedings...* Stockholm: IEEE/KTH, 1995. p. 832-837.

OHTA, T.; MATSUI, T.; MAKATA, T.; KATO, M.; AOYAGI, M.; KUNUGI, M.; SHIMADA, K.; NAGATA, J. Practical approach to unit commitment problem using genetic algorithm and lagrangian relaxation method. In: ISAP, Jan. 29 - Feb. 2 1996, Orlando. *Proceedings...* Orlando: 1996. p. 434-440.

ORERO, S. O.; IRVING, M. R. Scheduling of generators with a hybrid genetic algorithm. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS IN ENGINEERING SYSTEMS: INNOVATIONS AND APPLICATIONS, 1., Sept. 12-14 1995, Sheffield. *Proceedings...* Sheffield: IEE/IEEE, 1995. p. 143-160.

OSMAN, I. H.; CHRISTOFIDES, N. Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operational Research*, n. 1, p. 317-336, 1994.

PARK, D.; KANDEL, A.; LANGHOLZ, G. Genetic-based new fuzzy reasoning models with application to fuzzy control. *IEEE Trans. on SMC*, v. 24, n. 1, p. 39-47, Jan. 1994.

SILVA, M. M. *Uma estratégia metaheurística híbrida genética de agrupamento para o problema bin packing 1-dimensional*. Natal, 2000. 89 p. Dissertação (Mestrado em Sistemas e Computação) - Universidade Federal do Rio Grande do Norte.

SONG, Y. H.; WANG, G. S.; WANG, P. Y.; JOHNS, A. T. Environmental/economic dispatch using fuzzy logic controlled genetic algorithms. *IEE Proceedings - Generation, Transmission and Distribution*, v. 144, n. 4, p. 377-382, July 1997.

URDANETA, A. J.; GÓMEZ, J. F.; SORRENTINO, E.; FLORES, L.; DIAZ, R. A hybrid genetic algorithm for optimal reactive power planning based upon successive linear programming. *IEEE Transactions on Power Systems*, v. 14, n. 4, p. 1292-1298, Nov. 1999.

ZOLFAGHARI, S.; LIANG, M.; Jointly solving the group scheduling and machining speed selection problems: a hybrid tabu search and simulated annealing approach. *International Journal of Production Research*, v. 37, n. 10, p. 2377-2397, 1999.

WERNER, P. V. P. *Uma metaheurística genética não convencional e "ant colony systems" para resolver o problema do caixeiro viajante com agrupamentos*. Rio de Janeiro, 1999. 132 p. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal Fluminense.

WONG, K. P.; WONG, Y. W. Thermal generator scheduling using hybrid genetic/simulated annealing approach. *IEE Proceedings - Generation, Transmission and Distribution*, v. 142, n. 4, p. 372-380, Jul. 1999.

WONG, K. P.; SUZANNAH, Y. W. W. Hybrid genetic/simulated annealing approach to short-term multiple-fuel constraint generation scheduling. *IEEE Transactions on Power Systems*, v. 12, n. 2, p. 776-784, May 1997.

WONG, K. P.; WONG, Y. W. Combined genetic algorithms/simulated annealing/fuzzy set approach to short-term generation scheduling with take-or-pay fuel contract. *IEEE Transactions on Power Systems*, v. 11, n. 1, p. 128-136, Feb. 1996.

WONG, K. P.; WONG, Y. W. Thermal generator scheduling using hybrid genetic/simulated annealing approach. *IEE Proceedings Part C - Generation, Transmission and Distribution*, v. 142, p. 372-380, 1995.

WONG, K. P.; WONG, Y. W. Genetic and genetic/simulated annealing approaches to economic dispatch. *IEE Proceedings Part C - Generation, Transmission and Distribution*, v. 141, n. 5, p. 507-513, Sept. 1994.

9.7 Times Assíncronos

AVILA-ABASCAL, P.; TALUKDAR, S. N. Cooperative Algorithms and abductive causal networks for the automatic generation of intelligent substation alarm processors. In: ISCAS, 1996. *Proceedings...* 1996.

BARÁN, B.; KASZKUREWICZ, E.; BHAYA, A. Distributed asynchronous team algorithms: application to the load flow problem. In: CONFERÊNCIA LATINOAMERICANA DE INFORMÁTICA, 19., 1993, Buenos Aires. *Anais...* Buenos Aires: 1993.

BERNARDI, R. *Aplicação da técnica de times assíncronos na otimização de problemas de empacotamento unidimensional*. São Paulo, 2001. 73 p. Dissertação (Mestrado em Engenharia Elétrica) - Universidade de São Paulo.

CAMPONOGARA, E.; DE SOUZA, P. S. *Modelo matemático de um problema de suprimento de demandas*. Campinas: DCC-IMECC-UNICAMP, 1995, 6 p. (Relatório Técnico)

CAMPONOGARA, E.; DE SOUZA, P. S. *A extensão de um modelo de escalonamento aplicada a um problema de suprimento de demandas*. Campinas: DCC-IMECC-UNICAMP, 1995, 5 p. (Relatório Técnico)

CAVALCANTE, V. F.; DE SOUZA, P. S. *Solving the job shop scheduling problem by asynchronous teams*. Campinas: DCC-IMECC-UNICAMP, 1995, 10 p. (Relatório Técnico)

CHEN, C. L. *Bayesian nets and a-teams for power system fault diagnosis*. Pittsburgh: Department of Electrical and Computer Engineering, Carnegie Mellon University, 1992. (Thesis, Ph.D.)

CHEN, S. Y.; TALUKDAR, S. N. Causal nets for diagnosis. In: SYMPOSIUM ON EXPERT SYSTEMS APPLICATION TO POWER SYSTEMS, 4., Jan. 4-8 1993, Melbourne. *Proceedings...* Aldershot: CRL Publishing, 1993. p. 379-386.

CHEN, S. Y.; TALUKDAR, S. N.; SADEH, N. M. Job-shop-scheduling by a team of asynchronous agents. In: IJCAI - WORKSHOP ON KNOWLEDGE-BASED PRODUCTION, SCHEDULING AND CONTROL, 1993, Chambéry. *Conference Paper*. Chambéry: 1993.

DE ALMEIDA, C. R. T. *Time assíncrono inicializador para o planejamento da expansão da transmissão*. Campinas, 1998. 89 p. Dissertação (Mestrado em Engenharia Elétrica) Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.

DE ALMEIDA, C. R. T. *Times assíncronos e organizações multiagentes - síntese teórica*. Campinas: DSEE-FEEC-UNICAMP, 1996, 16 p. (Relatório Técnico)

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, 12., 1998, Uberlândia. *Proceedings...* Uberlândia: SBA-UFU, v. 3, 1998. p. 1029-1034.

- DE OLIVEIRA, S. A. *Parâmetros ótimos para algoritmo genético aplicado ao planejamento da expansão da transmissão*. Campinas: DSEE/ FEEC/UNICAMP, dezembro/1998, 111 p. (Relatório Técnico)
- DE OLIVEIRA, S. A.; DE ALMEIDA, C. R. T. *Times Assíncronos: protótipos e testes*. Campinas: DSEE/FEEC/UNICAMP, junho/1997, 33 p. (Relatório Técnico)
- DE OLIVEIRA, S. A. *Metaheurísticas - um estudo introdutório*. Campinas: DSEE/ FEEC/UNICAMP, dezembro/1996, 58 p. (Relatório Técnico)
- DE SOUZA, P. S. *Asynchronous organizations for multi-algorithm problems*. Pittsburgh: Department of Electrical and Computer Engineering, Carnegie Mellon University, 1993. (Thesis, Ph.D.)
- DE SOUZA, P. S.; TALUKDAR, S. N. Asynchronous organizations for multi-algorithm problems. In: ACM/SIGAPP SYMPOSIUM ON APPLIED COMPUTING, Feb. 14-16, 1993, Indianapolis. *Proceedings...* New York: ACM, 1993. p. 286-293.
- DE SOUZA, P. S.; TALUKDAR, S. N. Genetic algorithms in asynchronous teams. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS, 4., 1991, Los Altos. *Proceedings...* Los Altos: 1991. p. 392-397.
- FALCÃO, D. M.; BARÁN, B. Team algorithms: uma aplicação para fluxo de potência elétrica. In: CONFERÊNCIA LATINOAMERICANA DE INFORMÁTICA, 18., 1992, Las Palmas de Gran Canárias. *Anais...* Las Palmas de Gran Canárias: 1992.
- GORTI, S. R. et al. Solving constraint satisfaction problems using a-teams. In: AI-EDAM. (to appear).
- HADDAD, E. G. *Times assíncronos para o "job shop scheduling problem": heurísticas de melhoria*. Campinas, 1996. 80 p. Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Campinas.
- KAO, J. H.; HEMMERLE, J. S., PRINZ, F. P. Asynchronous-teams based collision avoidance in PAWS. *EDRC Report*, Carnegie Mellon University, June 1995.
- MEHROTRA, R.; TALUKDAR, S. N. Task scheduling on multiprocessors for power system problems. *IEEE Transactions on Power Systems*, v. PAS-102, n. 11, p. 3590-3597, Nov. 1983.
- MURTHY, S. *Synergy in cooperating agents: designing manipulators from task specifications*. Pittsburgh: Department of Electrical and Computer Engineering, Carnegie Mellon University, 1992. (Thesis, Ph.D.)
- NASCIMENTO, H. A. D. *Uma abordagem para desenho de grafos baseada na utilização de times assíncronos*. Campinas, 1996. 80 p. Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Campinas.

NII, H. P. Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures, Parts I and II, *AI Magazine*, 7:2 and 7:3, 1986.

OSTER, G. F.; WILSON, E. O. *Caste and ecology in the social insects*, Princeton University Press, Princeton, 1968.

PATEL, M. J.; MANIEZZO, V. NN's and GA's evolving co-operative behaviour in adaptive learning agents. In: IEEE CONFERENCE ON EVOLUTIONARY COMPUTATION - IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE, 1., June 27-29, 1994, Orlando. *Proceedings...* New York: IEEE, v. 1, 1994. p. 290-295.

PYO, S.S. *Asynchronous algorithms for distributed processing*. Pittsburgh: Department of Electrical and Computer Engineering, Carnegie Mellon University, 1985. (Thesis, Ph.D.)

QUADREL, R.W. *Asynchronous design environments: architecture and behavior*. Pittsburgh: Department of Architecture, Carnegie Mellon University, 1991. (Thesis, Ph.D.)

QUADREL, R. W. et al. Controlling asynchronous team design environments by simulated annealing. *Research in Engineering Design*, v. 5, n. 2, p. 88-104, 1993.

RABAK, C. S. *Otimização do processo de inserção automática de componentes eletrônicos empregando a técnica de times assíncronos*. São Paulo, 1999. 114 p. Dissertação (Mestrado em Engenharia Elétrica) - Universidade de São Paulo.

RACHLIN, J. al. Forest view: a system for integrated scheduling in complex manufacturing domains. *IBM Report*, 1996.

RAMESH, V. C. *Inertial search and asynchronous decompositions*. Pittsburgh: Department of Electrical and Computer Engineering, Carnegie Mellon University, 1994. (Thesis, Ph.D.)

RAMESH, V. C.; QUADREL, R.; DE SOUZA, P. S.; TALUKDAR, S. N. Asynchronous teams - an organizational model for distributed problem solving. In: SUMMER NATIONAL MEETING, Aug. 1991, Pittsburgh. *Proceedings...* Pittsburgh: American Institute of Chemical Engineers, 1991.

RAMESH, V. C.; TALUKDAR, S.N. A parallel asynchronous decomposition for on-line contingency planning. *IEEE Transactions on Power Systems*, v. 11, n. 1, p. 344-349, Feb. 1996.

RIBEIRO, W. E. *Algoritmos heurísticos para o "prize collecting traveling salesman problem"*. Campinas, 1997. 82 p. Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Campinas.

RODRIGUES, R. F.; DE SOUZA, P. S. *Manipulando conjuntos de soluções multidimensionais através de times assíncronos*. Campinas: DCC-IMECC-UNICAMP, 1996, 6 p. (Relatório Técnico)

RODRIGUES, R. F. *Times assíncronos para a resolução de problemas de otimização combinatória com múltiplas funções objetivo*. Campinas, 1996. 100 p. Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Campinas.

RODRIGUES, R. F.; DE SOUZA, P. S. *Asynchronous teams: a multi-algorithm approach for solving combinatorial optimizations problems*. Campinas: DCC-IMECC-UNICAMP, Nov. 1995, 11p. (Relatório Técnico DCC-95-18)

SAITO JÚNIOR, P. A. *Treinamento de redes neurais artificiais utilizando time assíncrono*. São José dos Campos, 1999. 104 p. Dissertação (Mestrado em Engenharia Eletrônica e Computação) - Instituto Tecnológico da Aeronáutica.

SZCZERBICKI, E. Rule-based integration of autonomous multi-agent systems. *International Journal of Systems Science*, v. 24, n. 11, p. 2117-2134, Nov. 1993.

TALUKDAR, S. N. Asynchronous teams. *EDRC internal document*, 1992.

TALUKDAR, S. N. Asynchronous Teams. In: SYMPOSIUM ON EXPERT SYSTEMS APPLICATION TO POWER SYSTEMS, 4., Jan. 4-8 1993, Melbourne. *Proceedings...* Aldershot: CRL Publishing, 1993. p. 647-655

TALUKDAR, S.N.; CARDOZO, E. Artificial intelligence techniques for power system operations. Some demonstration and an assessment of opportunities. *Electric Power Research Institute - Final Report*, n. 1999-7, Carnegie-Mellon University, Pittsburgh, 1985.

TALUKDAR, S. N.; DE SOUZA, P. S. *Insects, fish and computer-based super-agents - Systems and control theory for power systems*. Berlin: Springer-Verlag, v. 64, 1994.

TALUKDAR, S. N.; DE SOUZA, P. S. Scale efficient organizations. In: INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS, Oct. 18-21, 1992, Chicago. *Proceedings...* New York: IEEE, 1993. p. 1458-1463.

TALUKDAR, S.; DE SOUZA, P. S. Asynchronous teams. In: SIAM CONF. LINEAR ALGEBRA: SIGNAL SYSTEMS AND CONTROL, 2., Nov. 1990, San Francisco. *Proceedings...* San Francisco: 1990.

TALUKDAR, S. N. et al. Cooperation schemes for autonomous agents. In: SEMINÁRIO A-TEAMS - TIMES ASSÍNCRONOS PARA PROBLEMAS INDUSTRIAIS, 1., 20 de agosto de 1996, São Paulo. *Anais...* São Paulo: SOFTEX2000, CTI, IPT, 1996. p. 2-36.

TALUKDAR, S. N. et al. A system for distributed problem solving. *Relatório Técnico*, n. 1999-7, Carnegie Mellon University, Pittsburgh, 1986.

TALUKDAR, S. N.; PYO, S. S.; GYRAS, T. C. Asynchronous procedures for parallel processing. *IEEE Transactions on Power Systems*, v. PAS-102, n. 11, p. 3652-3659, Nov. 1983.

TALUKDAR, S. N.; PYO, S. S.; MEHROTRA, R. Designing algorithms and assignments for distributed processing. *Electric Power Research Institute - Final Report*, Carnegie-Mellon University, Pittsburgh, 1983.

TALUKDAR, S. N.; RAMESH, V. C. A-teams for real-time operations. *International Journal of Electrical Power & Energy Systems*, v.14, n.2-3, p. 138-143, Apr.-Jun. 1992.

TALUKDAR, S. N.; RAMESH, V. C. A parallel global optimization algorithm and its application to the CCOPF problem. In: POWER INDUSTRY COMPUTER APPLICATIONS CONFERENCE, May 1993, Phoenix. *Proceedings...* Phoenix: 1993.

TALUKDAR, S. N.; RAMESH, V. C. A multi-agent technique for contingency constrained optimal power flows. *IEEE Transactions on Power Systems*, v. 9, n. 2, p.855-861, May 1994.

TALUKDAR, S. N.; RAMESH, V. C.; QUADREL, R.; CHRISTIE, R. Multiagent organizations for real-time operations. *Proceedings of the IEEE*, v. 80, n. 5, p. 765-778, May 1992.

TSEN, C. K. *Solving train scheduling problems using a-teams*. Pittsburgh: Department of Electrical and Computer Engineering, Carnegie Mellon University, 1995. (Thesis, Ph.D.)

Apêndice A

Dados dos Sistemas Testes

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

Tabela A.1: Dados de Barras - Garver

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

Tabela A.2: Dados de Linhas - Garver

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

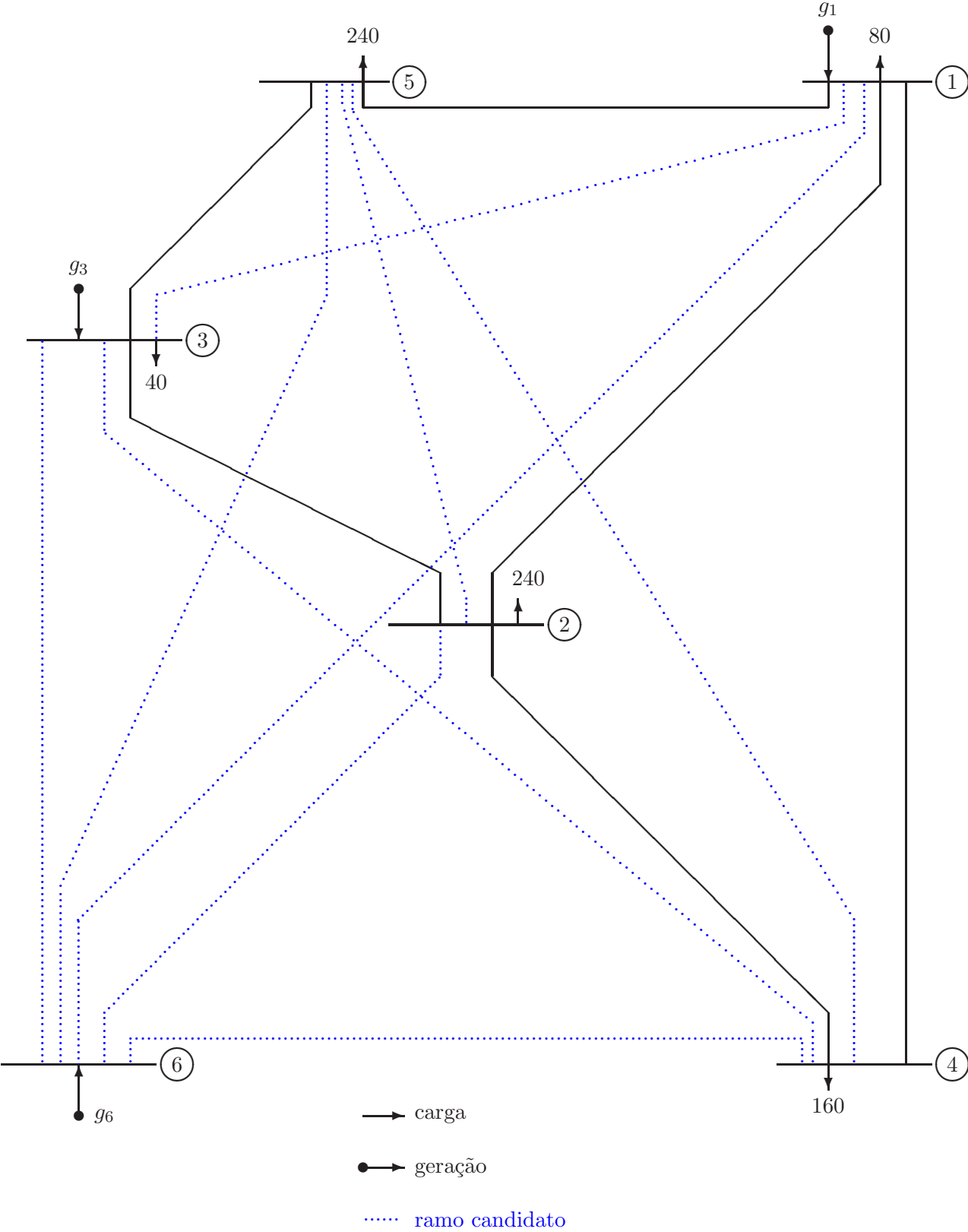


Figura A.1: Sistema Garver - Configuração Inicial.

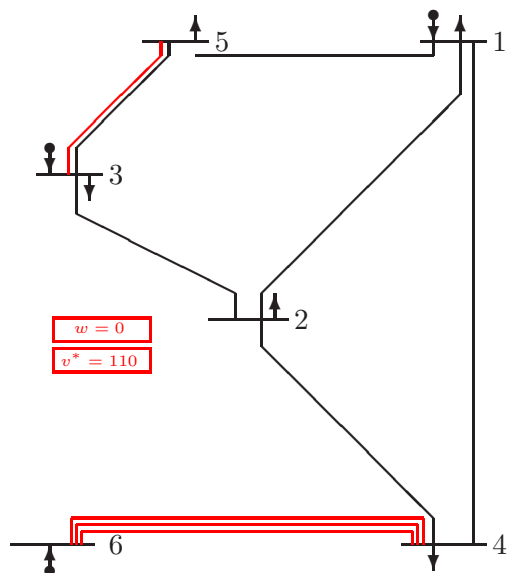


Figura A.2: Sistema Garver - Configuração Ótima com Redespacho

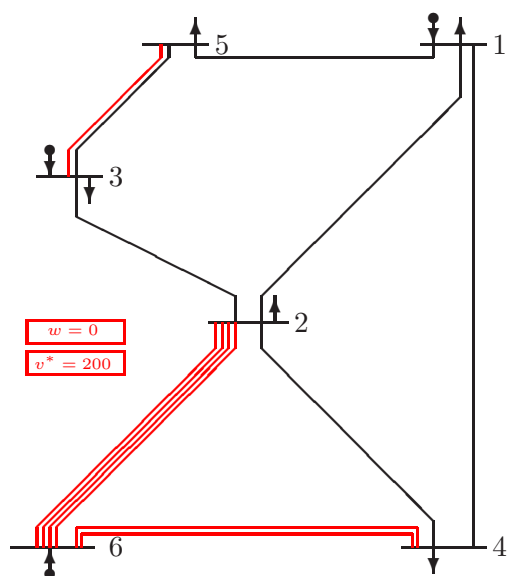


Figura A.3: Sistema Garver - Configuração Ótima sem Redespacho

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

Tabela A.3: Dados de Barras - Sul Brasileiro

Barra	Capacidade de Geração (MW)	Geração Atual (MW)	Carga (MW)
1	0,0	0,0	0,0
2	0,0	0,0	443,1
3	0,0	0,0	0,0
4	0,0	0,0	300,7
5	0,0	0,0	238,0
6	0,0	0,0	0,0
7	0,0	0,0	0,0
8	0,0	0,0	72,2
9	0,0	0,0	0,0
10	0,0	0,0	0,0
11	0,0	0,0	0,0
12	0,0	0,0	511,9
13	0,0	0,0	185,8
14	1.257,0	944,0	0,0
15	0,0	0,0	0,0
16	2.000,0	1.366,0	0,0
17	1.050,0	1.000,0	0,0
18	0,0	0,0	0,0
19	1.670,0	773,0	0,0
20	0,0	0,0	1.091,2
21	0,0	0,0	0,0
22	0,0	0,0	81,9
23	0,0	0,0	458,1
24	0,0	0,0	478,2
25	0,0	0,0	0,0
26	0,0	0,0	231,9
27	220,0	54,0	0,0
28	800,0	730,0	0,0
29	0,0	0,0	0,0
30	0,0	0,0	0,0
31	700,0	310,0	0,0
32	500,0	450,0	0,0
33	0,0	0,0	229,1
34	748,0	221,0	0,0
35	0,0	0,0	216,0
36	0,0	0,0	90,1
37	300,0	212,0	0,0
38	0,0	0,0	216,0
39	600,0	221,0	0,0
40	0,0	0,0	262,1
41	0,0	0,0	0,0
42	0,0	0,0	1.607,9
43	0,0	0,0	0,0
44	0,0	0,0	79,1
45	0,0	0,0	86,7
46	700,0	599,0	0,0
Total	10.545,0	6.880,0	6.880,0

Tabela A.4: Dados de Linhas - Sul Brasileiro

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

continua na próxima página

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

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

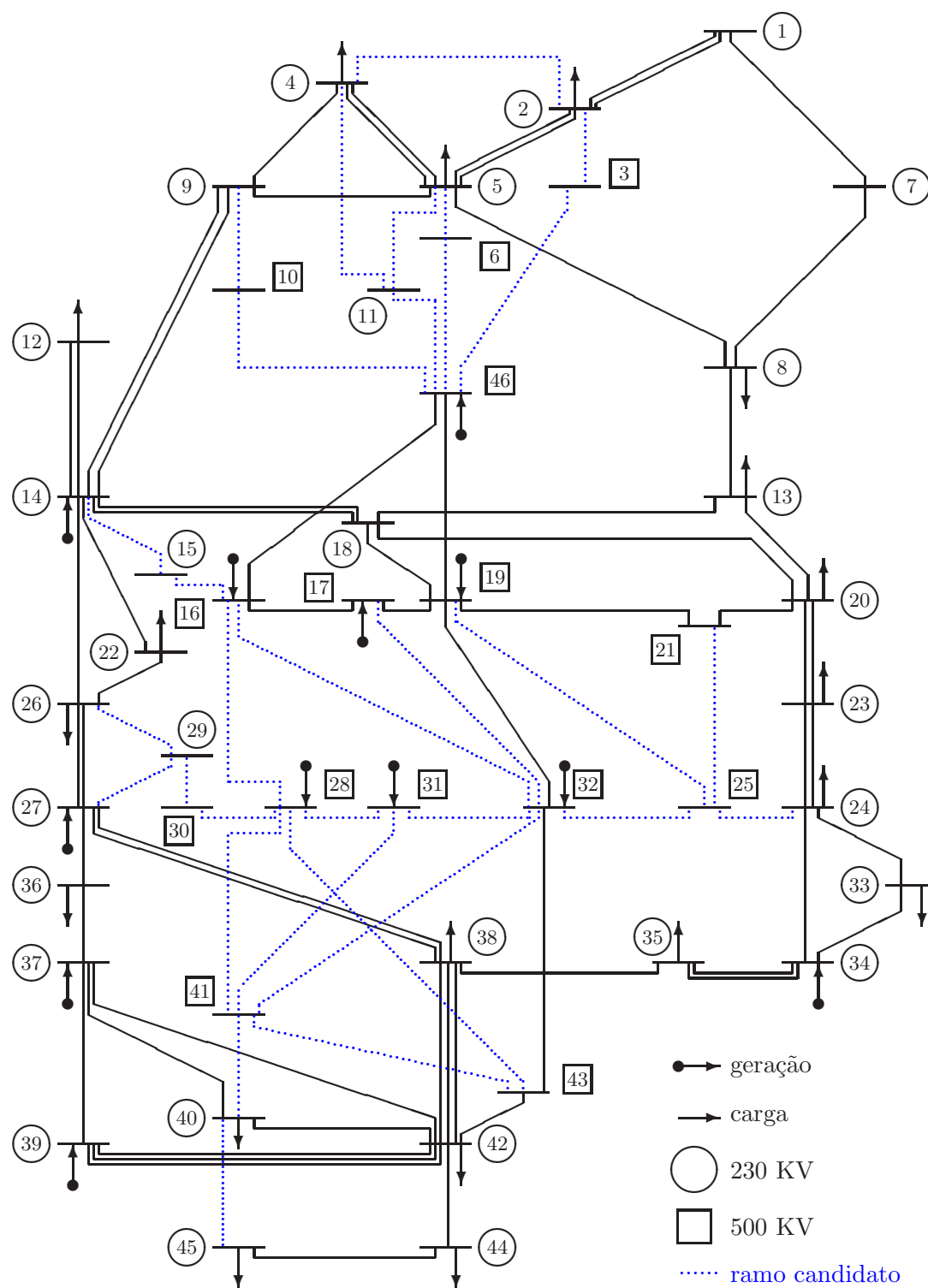


Figura A.4: Sistema Sul brasileiro - Configuração Inicial.

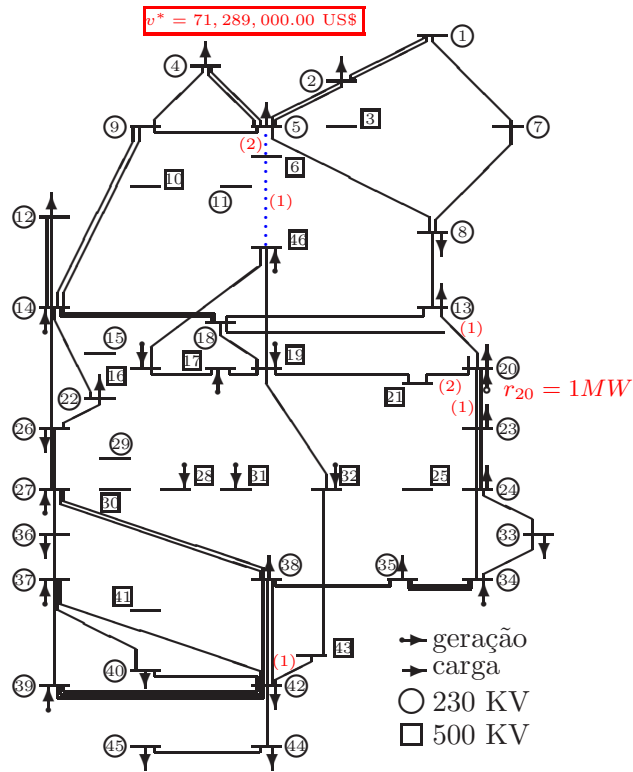


Figura A.5: Sistema Sul Brasileiro - Config. Ótima com Redespatch.

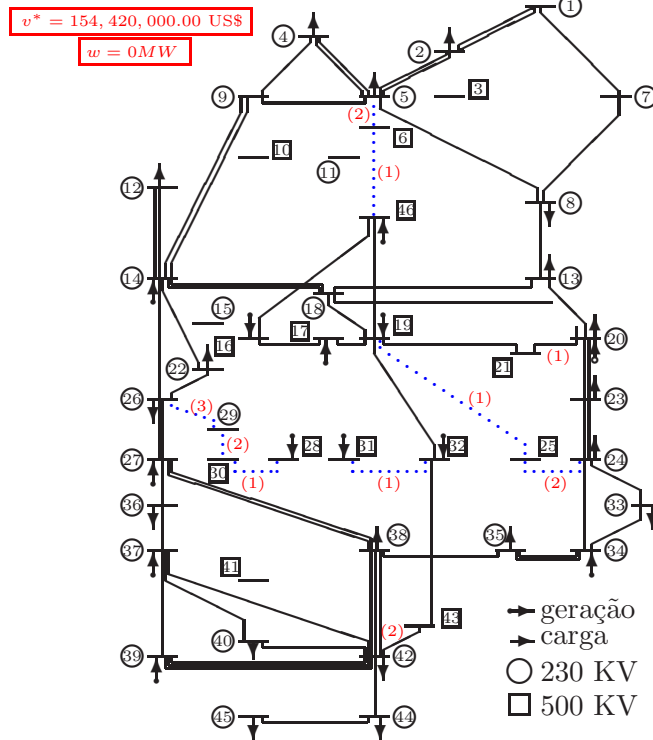


Figura A.6: Sistema Sul Brasileiro - Config. Ótima sem Redespatch.

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

Tabela A.5: Dados de Barras - Norte-Nordeste Brasileiro

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

continua na próxima página

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

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

continua na próxima página

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

Barra	Geração em 2002 (MW)	Carga em 2002 (MW)	Geração em 2008 (MW)	Carga em 2008 (MW)
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 A.6: Dados de Linhas - Norte-Nordeste Brasileiro

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

continua na próxima página

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

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
22	05-68	0	0,0406	1.000	48,880.00
23	05-70	0	0,0464	1.000	55,580.00
24	05-80	0	0,0058	1.200	21,232.00
25	06-07	1	0,0288	1.000	35,212.00
26	06-37	1	0,0233	300	7,510.00
27	06-67	0	0,0464	1.000	55,580.00
28	06-68	0	0,0476	1.000	5,6920.00
29	06-70	0	0,0371	1.000	44,860.00
30	06-75	0	0,0058	1.200	21232.00
31	07-08	1	0,0234	1.000	29,048.00
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

continua na próxima página

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

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
61	15-46	1	0,0117	600	8,926.00
62	15-53	0	0,0423	1.000	50,890.00
63	16-44	4	0,0117	600	8,926.00
64	16-45	0	0,0220	1.200	27,440.00
65	16-61	0	0,0128	1.000	16,720.00
66	16-77	0	0,0058	1.200	21,232.00
67	17-18	2	0,0170	1.200	21,678.00
68	17-59	0	0,0170	1200	21,678.00
69	18-50	4	0,0117	600	8,926.00
70	18-59	1	0,0331	1.200	40,170.00
71	18-74	0	0,0058	1.200	21,232.00
72	19-20	1	0,0934	170	5,885.00
73	19-22	1	0,1877	170	11,165.00
74	20-21	1	0,0715	300	6,960.00
75	20-38	2	0,1382	300	12,840.00
76	20-56	0	0,0117	600	8,926.00
77	20-66	0	0,2064	170	12,210.00
78	21-57	0	0,0117	600	8,926.00
79	22-23	1	0,1514	170	9,130.00
80	22-37	2	0,2015	170	11,935.00
81	22-58	0	0,0233	300	7,510.00
82	23-24	1	0,1651	170	9,900.00
83	24-25	1	0,2153	170	12,705.00
84	24-43	0	0,0233	300	7,510.00
85	25-26	2	0,1073	300	29,636.00
86	25-55	0	0,0117	600	8,926.00
87	26-27	2	0,1404	300	25,500.00
88	26-29	1	0,1081	170	6,710.00
89	26-54	0	0,0117	600	8,926.00
90	27-28	3	0,0826	170	5,335.00
91	27-35	2	0,1367	300	25,000.00
92	27-53	1	0,0117	600	8,926.00
93	28-35	3	0,1671	170	9,900.00
94	29-30	1	0,0688	170	4,510.00
95	30-31	1	0,0639	170	4,235.00
96	30-63	0	0,0233	300	7510.00
97	31-34	1	0,1406	170	8,525.00
98	32-33	0	0,1966	170	11,660.00
99	33-67	0	0,0233	300	7,510.00

continua na próxima página

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

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

continua na próxima página

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

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
139	61-86	0	0,0139	1.000	18,060.00
140	62-67	0	0,0464	1.000	55,580.00
141	62-68	0	0,0557	1.000	66,300.00
142	62-72	0	0,0058	1.200	21,232.00
143	63-64	0	0,0290	1.000	35,480.00
144	65-66	0	0,3146	170	18,260.00
145	65-87	0	0,0233	300	7,510.00
146	67-68	0	0,0290	1.000	35,480.00
147	67-69	0	0,0209	1.000	26,100.00
148	67-71	0	0,0058	1.200	21,232.00
149	68-69	0	0,0139	1.000	18,060.00
150	68-83	0	0,0058	1.200	21,232.00
151	68-87	0	0,0186	1.000	23,240.00
152	69-87	0	0,0139	1.000	18,060.00
153	70-82	0	0,0058	1.200	21,232.00
154	71-72	0	0,0108	3.200	125,253.00
155	71-75	0	0,0108	3.200	125,253.00
156	71-83	0	0,0067	3.200	80,253.00
157	72-73	0	0,0100	3.200	116,253.00
158	72-83	0	0,0130	3.200	149,253.00
159	73-74	0	0,0130	3.200	149,253.00
160	73-75	0	0,0130	3.200	149,253.00
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

continua na próxima página

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

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
178	81-83	0	0,0016	3.200	23,253.00
179	82-84	0	0,0135	3.200	155,253.00

A.4 Sistema Colombiano (93 Barras/155 Ramos)

Tabela A.7: Dados de Barras - Sistema Colombiano.

Barra	Geração em 2005 (MW)	Carga em 2005 (MW)	Geração em 2009 (MW)	Carga em 2009 (MW)	Geração em 2012 (MW)	Carga em 2012 (MW)
1	240,0	0,00	240,0	0,00	240,0	0,00
2	0,0	352,90	165,0	406,53	165,0	486,66
3	0,0	393,00	0,0	490,50	0,0	587,08
4	0,0	0,00	0,0	0,00	0,0	0,00
5	40,0	235,00	40,0	293,56	40,0	351,42
6	34,0	0,00	34,0	0,00	34,0	0,00
7	0,0	300,00	0,0	374,26	136,0	448,03
8	100,0	339,00	230,0	423,00	230,0	505,87
9	0,0	348,00	0,0	434,12	0,0	519,69
10	0,0	60,00	0,0	74,21	0,0	88,84
11	80,0	147,00	108,0	183,90	108,0	220,15
12	47,0	0,00	47,0	0,00	47,0	0,00
13	0,0	174,00	0,0	217,26	0,0	260,08
14	0,0	0,00	0,0	0,00	0,0	0,00
15	0,0	377,00	0,0	470,17	0,0	562,84
16	0,0	236,00	0,0	294,00	0,0	351,90
17	35,0	136,00	35,0	169,57	35,0	203,00
18	480,0	36,20	540,0	45,20	540,0	54,10
19	900,0	19,60	1.340,0	24,46	1.340,0	29,28
20	0,0	202,40	0,0	252,50	45,0	302,27
21	0,0	186,00	0,0	231,70	0,0	277,44
22	200,0	53,00	200,0	66,13	200,0	79,17
23	0,0	203,00	0,0	252,50	0,0	302,27
24	120,0	0,00	150,0	0,00	150,0	0,00
25	86,0	0,00	86,0	0,00	86,0	0,00
26	70,0	0,00	70,0	0,00	70,0	0,00
27	0,0	266,00	0,0	331,40	0,0	396,71
28	0,0	326,00	0,0	406,30	14,0	486,39
29	618,0	339,00	618,0	422,60	618,0	505,96
30	0,0	137,00	0,0	166,70	0,0	199,55
31	189,0	234,00	189,0	327,30	189,0	391,88
32	0,0	126,00	0,0	157,30	0,0	188,33
33	0,0	165,00	0,0	206,53	0,0	247,24
34	0,0	77,50	0,0	96,70	0,0	115,81
35	200,0	172,00	200,0	214,60	200,0	256,86
36	0,0	112,00	0,0	140,00	44,0	167,29
37	138,0	118,00	138,0	147,30	138,0	176,30

continua na próxima página

Tabela A.7: Dados de Barras - Sistema Colombiano (continuação)

Barra	Geração em 2005 (MW)	Carga em 2005 (MW)	Geração em 2009 (MW)	Carga em 2009 (MW)	Geração em 2012 (MW)	Carga em 2012 (MW)
38	0,0	86,00	15,0	108,40	15,0	129,72
39	0,0	180,00	0,0	224,00	15,0	268,19
40	305,0	0,00	305,0	0,00	305,0	0,00
41	70,0	54,80	100,0	68,40	100,0	81,85
42	0,0	102,00	0,0	127,30	0,0	152,39
43	0,0	35,40	0,0	44,20	0,0	52,90
44	23,0	257,00	23,0	321,30	23,0	384,64
45	950,0	0,00	1.208,0	0,00	1.208,0	0,00
46	150,0	121,00	150,0	151,70	150,0	181,62
47	0,0	41,15	0,0	51,50	0,0	61,60
48	775,0	600,00	885,0	750,00	885,0	896,26
49	0,0	130,00	0,0	162,00	0,0	193,27
50	240,0	424,00	240,0	528,00	240,0	632,75
51	0,0	128,00	0,0	159,00	0,0	190,45
52	0,0	38,00	0,0	46,50	0,0	55,60
53	280,0	0,00	320,0	0,00	320,0	0,00
54	0,0	76,00	0,0	95,30	0,0	114,19
55	40,0	223,00	40,0	279,00	40,0	333,59
56	0,0	0,00	0,0	0,00	0,0	0,00
57	0,0	226,00	130,0	281,00	130,0	336,94
58	190,0	0,00	190,0	0,00	190,0	0,00
59	160,0	0,00	160,0	0,00	160,0	0,00
60	1.191,0	0,00	1.216,0	0,00	1.216,0	0,00
61	155,0	0,00	155,0	0,00	155,0	0,00
62	0,0	0,00	0,0	0,00	0,0	0,00
63	900,0	35,00	1.090,0	44,00	1.090,0	52,77
64	0,0	88,00	0,0	110,55	280,0	132,35
65	0,0	132,00	0,0	165,00	0,0	197,58
66	200,0	0,00	300,0	0,00	300,0	0,00
67	474,0	266,00	474,0	332,45	474,0	397,98
68	0,0	0,00	0,0	0,00	0,0	0,00
69	0,0	71,40	0,0	89,00	0,0	106,61
70	30,0	0,00	180,0	0,00	180,0	0,00
71	0,0	315,00	211,0	393,00	424,0	471,21
72	0,0	0,00	0,0	0,00	0,0	0,00
73	0,0	0,00	0,0	0,00	0,0	0,00
74	0,0	0,00	0,0	0,00	0,0	0,00
75	0,0	0,00	0,0	0,00	0,0	0,00
76	40,0	0,00	40,0	0,00	40,0	0,00

continua na próxima página

Tabela A.7: Dados de Barras - Sistema Colombiano (continuação)

Barra	Geração em 2005 (MW)	Carga em 2005 (MW)	Geração em 2009 (MW)	Carga em 2009 (MW)	Geração em 2012 (MW)	Carga em 2012 (MW)
77	0,0	55,00	0,0	70,00	0,0	82,85
78	0,0	36,65	0,0	45,10	0,0	54,07
79	0,0	98,00	0,0	123,00	300,0	146,87
80	0,0	60,00	0,0	72,00	0,0	88,34
81	0,0	0,00	0,0	0,00	0,0	0,00
82	0,0	0,00	0,0	0,00	0,0	0,00
83	0,0	0,00	0,0	0,00	0,0	0,00
84	0,0	0,00	0,0	0,00	500,0	0,00
85	0,0	0,00	0,0	0,00	0,0	0,00
86	0,0	0,00	300,0	0,00	850,0	0,00
87	0,0	0,00	0,0	0,00	0,0	0,00
88	0,0	0,00	0,0	0,00	300,0	0,00
89	0,0	0,00	0,0	0,00	0,0	0,00
90	0,0	0,00	0,0	0,00	0,0	0,00
91	0,0	0,00	0,0	0,00	0,0	0,00
92	0,0	0,00	0,0	0,00	0,0	0,00
93	0,0	0,00	0,0	0,00	0,0	0,00
Total	9.750,0	9.750,00	12.162,00	12.162,00	14.559,0	14.559,00

Tabela A.8: Dados de Linhas - Sistema Colombiano

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
1	52-88	0	0,0980	300,0	34,190.00
2	43-88	0	0,1816	250,0	39,560.00
3	57-81	0	0,0219	550,0	58,890.00
4	73-82	0	0,0374	550,0	97,960.00
5	27-89	0	0,0267	450,0	13,270.00
6	74-89	0	0,0034	550,0	14,570.00
7	73-89	0	0,0246	550,0	66,650.00
8	79-83	0	0,0457	350,0	15,400.00
9	8-67	0	0,2240	250,0	29,200.00
10	39-86	0	0,0545	350,0	9,880.00
11	25-28	1	0,0565	320,0	9,767.00
12	25-29	1	0,0570	320,0	9,882.00
13	13-14	2	0,0009	350,0	3,902.00
14	13-20	1	0,0178	350,0	5,742.00
15	13-23	1	0,0277	350,0	7,007.00

continua na próxima página

Tabela A.8: Dados de Linhas - Colombiano (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
16	14-31	2	0,1307	250,0	18,622.00
17	14-18	2	0,1494	250,0	20,232.00
18	14-60	2	0,1067	300,0	15,977.00
19	2-4	2	0,0271	350,0	6,662.00
20	2-9	1	0,0122	350,0	5,282.00
21	2-83	1	0,0200	570,0	5,972.00
22	9-83	1	0,0200	400,0	5,972.00
23	15-18	1	0,0365	450,0	7,927.00
24	15-17	1	0,0483	320,0	9,422.00
25	15-20	1	0,0513	320,0	9,652.00
26	15-76	1	0,0414	320,0	9,882.00
27	15-24	1	0,0145	350,0	5,282.00
28	37-61	1	0,0139	350,0	4,937.00
29	19-61	2	0,1105	250,0	16,092.00
30	61-68	1	0,0789	250,0	12,412.00
31	37-68	1	0,0544	320,0	9,652.00
32	40-68	1	0,1320	320,0	18,162.00
33	12-75	1	0,0641	320,0	11,492.00
34	24-75	1	0,0161	350,0	5,512.00
35	35-36	1	0,2074	250,0	27,362.00
36	27-35	1	0,1498	250,0	22,072.00
37	35-44	2	0,1358	250,0	20,347.00
38	38-68	1	0,0389	350,0	7,927.00
39	38-39	1	0,0300	350,0	6,317.00
40	27-80	1	0,0242	350,0	7,007.00
41	44-80	1	0,1014	250,0	17,587.00
42	56-81	1	0,0114	550,0	32,858.00
43	45-54	1	0,0946	320,0	13,562.00
44	45-50	2	0,0070	350,0	4,362.00
45	10-78	1	0,0102	350,0	4,937.00
46	7-78	1	0,0043	350,0	4,132.00
47	30-64	1	0,1533	250,0	20,577.00
48	30-65	1	0,0910	250,0	13,677.00
49	30-72	2	0,0173	350,0	5,512.00
50	55-57	1	0,0174	600,0	46,808.00
51	57-84	1	0,0087	600,0	26,658.00
52	55-84	1	0,0087	600,0	26,658.00
53	56-57	2	0,0240	600,0	62,618.00
54	9-77	1	0,0190	350,0	5,857.00
55	77-79	1	0,0097	350,0	5,167.00

continua na próxima página

Tabela A.8: Dados de Linhas - Colombiano (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
56	1-59	2	0,0232	350,0	6,202.00
57	59-67	2	0,1180	250,0	16,667.00
58	8-59	2	0,1056	250,0	15,402.00
59	1-3	1	0,1040	250,0	15,862.00
60	3-71	1	0,0136	450,0	5,167.00
61	3-6	1	0,0497	350,0	9,422.00
62	55-62	1	0,0281	550,0	70,988.00
63	47-52	1	0,0644	350,0	10,572.00
64	51-52	1	0,0859	250,0	12,872.00
65	29-31	2	0,1042	250,0	32,981.00
66	41-42	1	0,0094	350,0	4,707.00
67	40-42	1	0,0153	350,0	5,167.00
68	46-53	2	0,1041	250,0	14,597.00
69	46-51	1	0,1141	250,0	16,322.00
70	69-70	2	0,0228	350,0	6,202.00
71	66-69	2	0,1217	250,0	17,127.00
72	9-69	2	0,1098	350,0	15,747.00
73	60-69	2	0,0906	350,0	13,677.00
74	31-32	1	0,0259	350,0	6,547.00
75	32-34	1	0,0540	350,0	9,767.00
76	16-18	1	0,0625	350,0	10,917.00
77	16-23	1	0,0238	350,0	6,892.00
78	16-21	1	0,0282	350,0	6,892.00
79	31-34	1	0,0792	250,0	12,412.00
80	31-33	2	0,0248	350,0	6,432.00
81	31-60	2	0,1944	250,0	25,982.00
82	31-72	2	0,0244	350,0	6,317.00
83	47-54	2	0,1003	250,0	14,252.00
84	47-49	2	0,0942	250,0	13,562.00
85	18-58	2	0,0212	350,0	5,742.00
86	18-20	1	0,0504	350,0	9,537.00
87	18-66	2	0,0664	350,0	11,377.00
88	18-21	1	0,0348	350,0	7,467.00
89	18-22	1	0,0209	350,0	6,432.00
90	19-22	1	0,0691	350,0	11,722.00
91	4-5	3	0,0049	350,0	4,247.00
92	5-6	2	0,0074	350,0	4,477.00
93	17-23	1	0,0913	250,0	12,987.00
94	17-76	1	0,0020	350,0	3,902.00
95	12-17	1	0,0086	350,0	4,707.00

continua na próxima página

Tabela A.8: Dados de Linhas - Colombiano (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
96	1-71	2	0,0841	250,0	14,367.00
97	1-8	1	0,0810	250,0	13,217.00
98	1-11	1	0,0799	250,0	12,527.00
99	4-36	2	0,0850	250,0	13,562.00
100	19-58	1	0,0826	320,0	11,722.00
101	27-64	1	0,0280	350,0	6,777.00
102	27-28	1	0,0238	350,0	6,202.00
103	27-44	1	0,0893	250,0	16,322.00
104	26-27	1	0,0657	350,0	10,917.00
105	27-29	1	0,0166	350,0	5,052.00
106	19-66	1	0,0516	350,0	9,307.00
107	73-74	1	0,0214	600,0	58,278.00
108	64-65	1	0,0741	350,0	11,837.00
109	29-64	1	0,0063	350,0	4,362.00
110	4-34	2	0,1016	270,0	14,942.00
111	34-70	2	0,0415	350,0	8,272.00
112	33-34	1	0,1139	320,0	16,322.00
113	8-71	1	0,0075	400,0	4,477.00
114	54-63	3	0,0495	320,0	9,077.00
115	48-63	1	0,0238	350,0	6,317.00
116	67-68	2	0,1660	250,0	22,072.00
117	39-68	1	0,0145	350,0	5,282.00
118	8-9	1	0,0168	350,0	5,972.00
119	79-87	1	0,0071	350,0	4,477.00
120	8-87	1	0,0132	350,0	5,167.00
121	39-43	1	0,1163	250,0	16,552.00
122	41-43	1	0,1142	250,0	16,322.00
123	23-24	1	0,0255	350,0	6,317.00
124	21-22	1	0,0549	350,0	9,882.00
125	26-28	1	0,0512	350,0	9,307.00
126	28-29	1	0,0281	350,0	6,777.00
127	6-10	1	0,0337	350,0	7,582.00
128	33-72	1	0,0228	350,0	6,202.00
129	39-40	2	0,1020	250,0	16,207.00
130	12-76	1	0,0081	350,0	4,707.00
131	48-54	3	0,0396	350,0	8,042.00
132	50-54	2	0,0876	250,0	12,872.00
133	62-73	1	0,0272	750,0	73,158.00
134	49-53	2	0,1008	250,0	14,252.00
135	40-41	1	0,0186	350,0	5,742.00

continua na próxima página

Tabela A.8: Dados de Linhas - Colombiano (continuação)

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
136	45-81	1	0,0267	450,0	13,270.00
137	64-74	1	0,0267	500,0	13,270.00
138	54-56	3	0,0267	450,0	13,270.00
139	60-62	3	0,0257	450,0	13,270.00
140	72-73	2	0,0267	500,0	13,270.00
141	19-82	1	0,0267	450,0	13,270.00
142	55-82	1	0,0290	550,0	77,498.00
143	62-82	1	0,0101	600,0	30,998.00
144	83-85	2	0,0267	450,0	13,270.00
145	82-85	1	0,0341	700,0	89,898.00
146	19-86	1	0,1513	300,0	20,922.00
147	68-86	1	0,0404	350,0	8,272.00
148	7-90	2	0,0050	350,0	4,247.00
149	3-90	1	0,0074	350,0	4,592.00
150	90-91	1	0,0267	550,0	13,270.00
151	85-91	1	0,0139	600,0	40,298.00
152	11-92	1	0,0267	450,0	13,270.00
153	1-93	1	0,0267	450,0	13,270.00
154	92-93	1	0,0097	600,0	30,068.00
155	91-92	1	0,0088	600,0	27,588.00

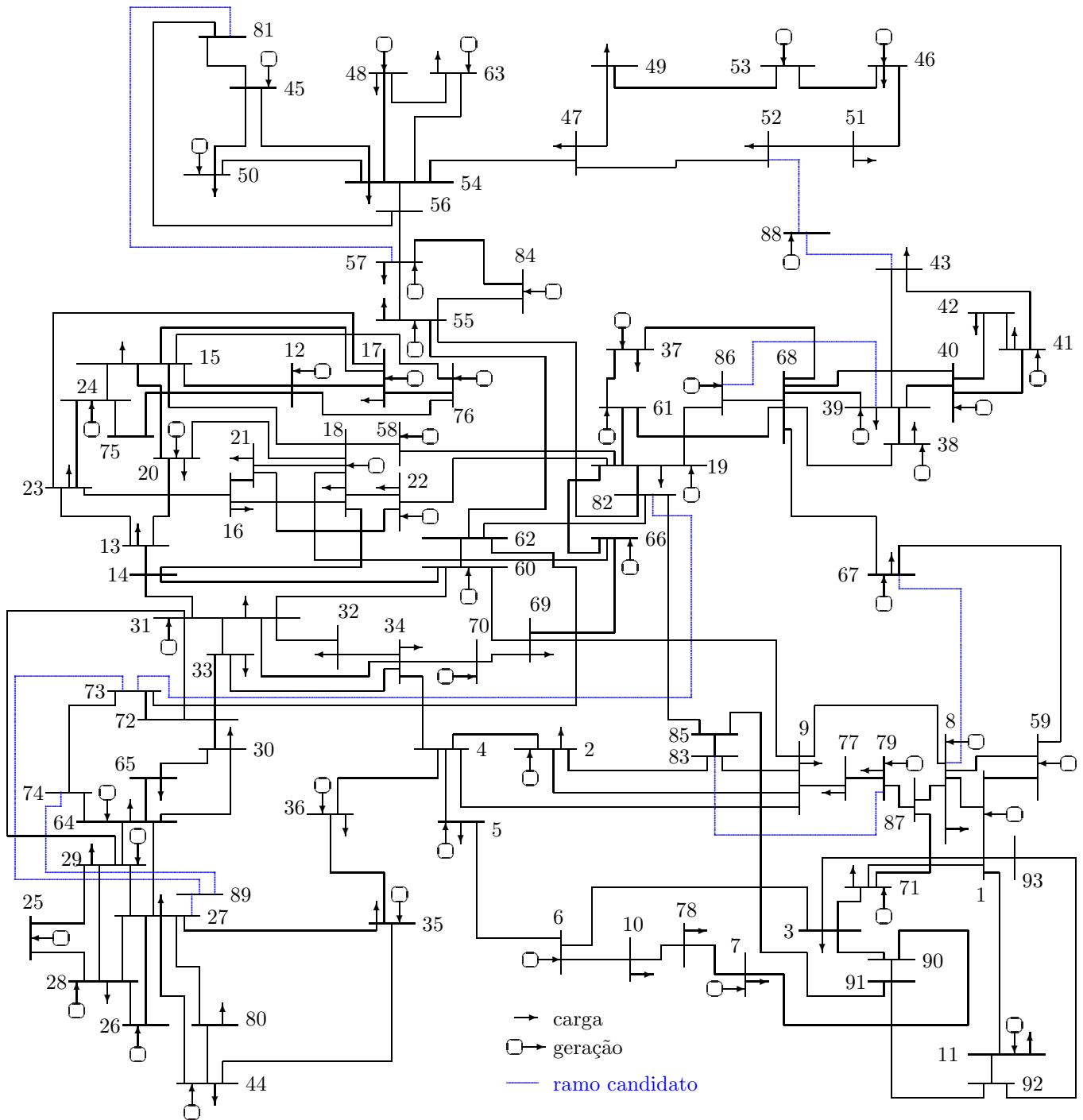


Figura A.7: Sistema Colombiano - Rede Inicial

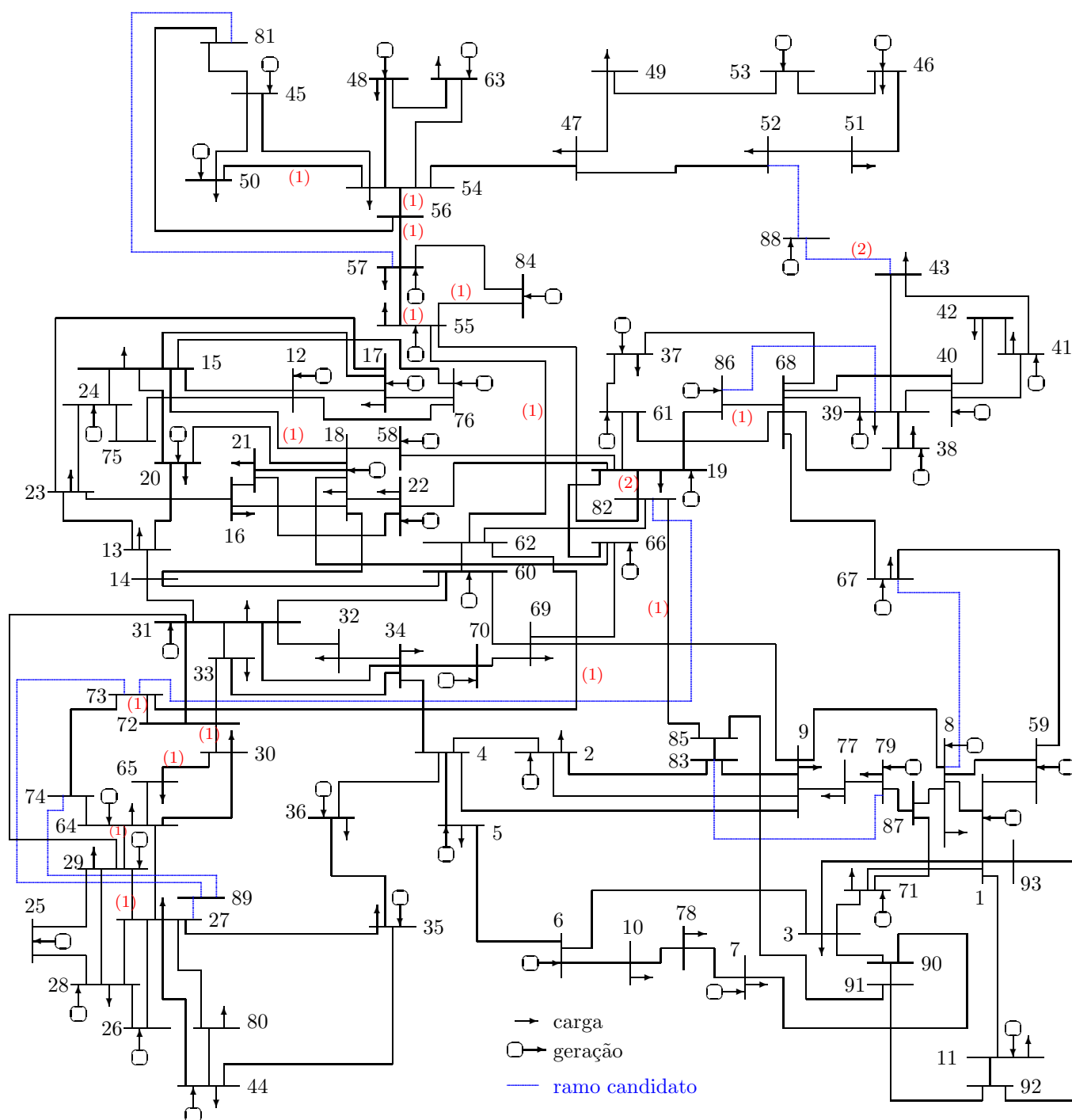


Figura A.8: Sistema Colombiano - Configuração Ótima

Apêndice B

Executando o Time Assíncrono Inicializador

Neste apêndice é apresentado todos os procedimentos necessários para se criar e executar o time assíncrono inicializador dos métodos heurísticos construtivos, programa *INIC*, que calcula soluções iniciais factíveis para serem utilizadas posteriormente nas metaheurísticas combinatórias.

```
*****  
COMO COMPILAR OS PROGRAMAS  
*****
```

1) Primeiramente deve-se criar um "link" na raiz do usuário:

```
> ln -s ~/pvm3/bin/SUN4SOL2 pvm
```

2) Feito isto, basta executar o script "comp" no diretório ~/pvm3/bin/SUN4SOL2/INIC, quando então os programas serão compilados e seus executáveis gerados no diretório ~/pvm3/bin/SUN4SOL2 (diretório de trabalho do PVM).

OBS: todo programa PVM deve ser executado a partir do diretório ~/pvm3/bin/SUN4SOL2

```
*****  
COMO CONFIGURAR O INICIALIZADOR  
*****
```

1) Copiar o arquivo com os dados da rede desejada (p.ex., garv1.dat) como "rede.dat", no diretório: ~/pvm3/bin/SUN4SOL2.

Arquivos de dados disponiveis no diretório ~/pvm3/bin/SUN4SOL2/INIC/dados:

"garv1.dat", "garv2.dat", "sur1.dat", "sur2.dat", "nor2002.dat", "nor2008.dat".

OBS: Pode-se usar o script "monta" no diretório ~/pvm3/bin/SUN4SOL2/INIC para automaticamente: - montar os dados relativos ao sistema a ser simulado,

- criar os arquivos: "opcoes", "param.dat", "nlin.dat" e "nconec.dat".

Sintaxe: monta garv1 gar (para montar todos os dados referentes ao sistema garv1.dat)

Estes arquivos são copiados automaticamente para o diretório ~/pvm3/bin/SUN4SOL2, onde será rodado o programa "INIC".

- 2) Alterar o arquivo "opcoes" de acordo com o sistema elétrico escolhido e com a máquina virtual desejada.

Ex: (opções para o sistema garv1.dat)

```

15 número de linhas do sistema elétrico
05 número de estações da máquina virtual
09 1 estação / agente
08 2 estação / agente
10 4 estação / agente
07 5 estação / agente
06 7 estação / agente
-----ESTAÇÕES / AGENTES-----
01 = tubarao 1 = garverA
02 = badejo 2 = garverB
03 = piranha 3 = garverC
04 = carpa 4 = minD
05 = tuvira 5 = minE
06 = piava 7 = mccF
07 = dourado
08 = lambari
09 = pintado
10 = cascudo

```

- 3) Alterar o arquivo "param.dat", alterando-se apenas os valores referentes a:
 - 3.1) número de configurações a serem geradas
 - 3.2) mínimo fluxo de sobrecarga permitido
 - 3.3) número de alternativas de caminhos

Ex: ("param.dat" para o sistema garv1.dat)

```

8      (nsem)      número de configurações a serem geradas
4531   (kseed)     número aleatório
0.4    (fraclinha) fração de linha adicionada
0      (fluxo)     fluxo de sobrecarga permitido (Garver=10,Sul:40,Norte:300)
100    (nadição)   tamanho da rede p/ o qual se faz correções adicionais (100)
3      (ncami)     número de alternativas de caminhos (Garver/Sul=3,Norte=5)

```

- 4) Copiar arquivo "nconec.gar", correspondente ao sistema Garver, para "nconec.dat".

OBS: Se foi utilizado o script "monta", o mesmo já fez isto.

- 5) Copiar arquivo "nlin.gar", correspondente ao sistema Garver, para "nlin.dat".

OBS: Se foi utilizado o script "monta", o mesmo já faz isto.

```
*****
EXECUTANDO O INICIALIZADOR
*****
```

1) inicializar PVM com as estações determinadas em "opcoes".

```
> pvm hosts (OBS: colocar no arq. "hosts" as máquinas que aparecem em "opcoes")
```

2) a partir do diretório ~/pvm3/bin/SUN4SOL2 executar o programa "inic".

```
> inic
```

OBS: pode-se usar o processamento remoto, via "at".

```
Ex: > at now
    at> inic
    at> CTRL_D
```

os dados relativos ao processamento do programa "INIC" serão enviados via 'mail'.

```
*****
SAIDA DO INICIALIZADOR
*****
```

Após a finalização de todos os agentes, o programa "INIC" gera um arquivo de saída ("mpop.dat") com as configurações iniciais factíveis que poderão ser usadas como configurações iniciais para os algoritmos combinatórios: SA, GA ou TS.

```
*****
COMO ADICIONAR AGENTES
*****
```

Basta executar o agente diretamente (garverA, por ex.) do "prompt" da estação, a partir do diretório ~/pvm3/bin/SUN4SOL2 (desde que a estação pertença a máquina virtual paralela). Senão, deve-se adicionar primeiro a estação, via "prompt" do PVM (comando "add"), para depois adicionar o agente ao time.

```
Ex: pvm> add nome_nova_estação
    pvm> quit
```

```
~/pvm3/bin/SUN4SOL2> garverA
```

```
*****
COMO REMOVER AGENTES
*****
```

Quando executado, o Inicializador cria um arquivo denominado "maquina_virtual", que contém cada agente inicialmente desovado, seu respectivo TID e máquina onde foi desovado.

Por ex.: badejo:pvm3/bin/SUN4SOL2 [183]more maquina_virtual

Desovando na MAQ = pintado 1 garverA TID = 786434 INFO = 1

Desovando na MAQ = lambari 1 garverB TID = 1048578 INFO = 1

Desovando na MAQ = cascudo 1 minD TID = 1572866 INFO = 1

Desovando na MAQ = dourado 1 minE TID = 1310722 INFO = 1

Desovando na MAQ = piava 1 mccG TID = 1835011 INFO = 1

Ao mesmo tempo, para cada agente é gerado um arquivo que contém o seu TID e o seu valor correspondente de "rank". Este valor de "rank" corresponde a ordem na qual os agentes foram gerados. Este arquivo possui um nome específico relacionado com o agente e com a ordem na qual foi desovado. Sendo assim, este nome então é composto de duas partes:

a primeira determina o tipo de agente :

```
gar    ---> garverA, garverB ou garverC ;
minesf ---> minD ou minE;
mincor ---> mccG.
```

a segunda, uma letra maiúscula do alfabeto, determina a ordem (que corresponde ao "rank") na qual o mesmo foi desovado por "INIC" (ou por adição manual).

Temos então, por ex., o arquivo "minesfC", que corresponde a um agente do tipo mínimo esforço (minD ou minE) e que foi o terceiro agente a ser desovado ("C" é a terceira letra do alfabeto).

```
badejo:pvm3/bin/SUN4SOL2 [185]more minesfC
D- tid = 1572866 rank = 3
```

Com estes dois arquivos, é possível a remoção de um ou mais agentes do time.

Por ex., no caso anterior, se o objetivo é remover o agente "minD" da estação cascudo, primeiramente deve-se identificar o TID do mesmo, no arquivo "maquina_virtual". Ali, nota-se que o agente possui TID igual a 1572866. Com este valor em mãos, e sabendo-se que é um agente de mínimo esforço, pode-se descobrir o seu "rank" (com ajuda do comando "grep"):

```
badejo:pvm3/bin/SUN4SOL2 [186]grep '1572866' minesf*
minesfC: D- tid = 1572866 rank = 3
```

Portanto, o agente determinado possui "rank" igual a 3.

Para remover o agente, executa-se o programa "remv" a partir do diretório ~/pvm3/bin/SUN4SOL2. O mesmo solicitará o rank do agente a ser removido.

Faz-se então o seguinte :

badejo:pvm3/bin/SUN4SOL2 [179]remv
 RANK DO PROCESSO (formato = XX): 03

Enviando sinal de encerramento para o processo de rank = 1 e TID = 1572866 INFO = 0

 USANDO OS RESULTADOS DO "INIC" NOS ALGORITMOS COMBINATÓRIOS

Para usar as configurações encontradas com a simulação via "INIC", deve-se efetuar os seguintes passos:

- 1) Alterar o arquivo "mpop.dat" (do diretório ~/pvm3/bin/SUN4SOL2), retirando os comentários e deixando simplesmente as configurações encontradas.
- 2) Copiar o arquivo "mpop.dat" para o diretório ~/mheur/blocos.
- 3) Leia as instruções constantes do arquivo README do diretório ~/mheur/blocos e realize-as, obtendo um arquivo denominado "mbloco.dat".
- 4) Transferir os arquivos 'mpop.dat' e 'mbloco.dat' para o diretório do programa de planejamento que se pretende rodar, renomeando-os respectivamente como 'mpop8.ga1' e 'mbloco8.ga1'. (Isto quando se estiver simulando o sistema garv1.dat).

Ou colocá-los no diretório ~/dados, para usá-los em todos algoritmos combinatórios através de um 'link'.

- 5) Alterar os parâmetros correspondentes ao número de configurações iniciais, nos arquivos de parâmetros dos algoritmos combinatórios.
- 6) Rodar o algoritmo combinatório.

-----FIM DAS INSTRUÇÕES-----

Apêndice C

Artigos Publicados

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, 12., 1998, Uberlândia. *Proceedings...* Uberlândia: SBA/UFU, v.III, 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, 15., 1999, Foz do Iguaçu. *Anais...* Foz do Iguaçu: CIGRÉ - Itaipu Binacional, 1999. Grupo VII-GPL/04.

DE OLIVEIRA, S.A.; ROMERO, R.A.; GALLEG0, R.A.; MONTICELLI, A. Algoritmo genético paralelo aplicado ao planejamento da expansão da transmissão. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA - CBA2000, 13., 2000, Florianópolis. *Anais...* Florianópolis: SBA/UFSC, 2000. p.1790-1795. 1 CD

DE OLIVEIRA, S.A.; GALLEG0, R.A.; ESCOBAR, A.; RESTREPO, R. Parallel combinatorial algorithm for statical planning of a transmission system. In: INTERNATIONAL CONFERENCE ON CAD/CAM, ROBOTICS AND FACTORIES AT THE FUTURE - 2001, 17., 2001, Durban. *Proceedings...* Durban: 2001. p.1183-1191.

GALLEG0, R.A.; ESCOBAR, A.; ROMERO, R.A.; DE OLIVEIRA, S.A. Hibrid model analysis and constructive algorithms in the performance of a genetic algorithm for the planning of an electric power transmission system. In: INTERNATIONAL CONFERENCE ON CAD/CAM, ROBOTICS AND FACTORIES AT THE FUTURE - 2001, 17., 2001, Durban. *Proceedings...* Durban: 2001. p.850-859.

DE OLIVEIRA, S.A.; ROMERO, R.A.; GALLEG0, R.A.; MONTICELLI, A. Metaheurísticas aplicadas ao planejamento da expansão da transmissão em ambiente de processamento distribuído. In: SIMPÓSIO DE ESPECIALISTAS EM PLANEJAMENTO DA OPERAÇÃO E EXPANSÃO ELÉTRICA - SEPOPE, 9., 23 a 27 de maio 2004, Rio de Janeiro. *Anais...* Rio de Janeiro: CIGRÉ - FURNAS, 2004.

Proceedings of XII Brazilian Automatic Control Conference - XII CBA
Vol. III, pp. 1029-1034 - September 14 - 18, 1998 - Uberlândia, MG, Brazil

TIMES ASSÍNCRONOS APLICADOS A MÉTODOS HEURÍSTICOS CONSTRUTIVOS DE PLANEJAMENTO DA EXPANSÃO DA TRANSMISSÃO

SÉRGIO A. DE OLIVEIRA*

CLÁUDIO R. T. DE ALMEIDA, ALCIR MONTICELLI**

(*) DEE. FEIS. UNESP - Cr. Postal 31. 15385-000 Ilha Solteira, SP, BRASIL

(**) DSEE. FEEC. UNICAMP - Cr. Postal 6101. 13081-970 Campinas, SP, BRASIL

E-mails: grilo@dee.feis.unesp.br, thomas@mpc.com.br, alcir@dsee.fee.unicamp.br

Resumo— Este artigo trata da aplicação de *Times Assíncronos (Times-A)* ao planejamento da expansão de sistemas de transmissão de energia elétrica. Uma parte importante desse processo é a determinação de famílias de soluções iniciais contendo características atrativas, isto é, configurações com linhas e transformadores e conjuntos de linhas e transformadores (blocos construtivos) que podem formar parte das soluções ótimas, ou quase ótimas obtidas via métodos como os Algoritmos Genéticos e Busca Tabu. *Times-A* foram desenvolvidos para gerar estas configurações iniciais. Esses times utilizam diferentes métodos heurísticos construtivos disponíveis na literatura, os quais são distribuídos em uma rede heterogênea de computadores e executados de modo concorrente. Testes foram feitos para sistemas de pequeno, médio e grande porte.

Abstract— This paper deals with the application of Asynchronous Teams (A-Teams) to the power network transmission expansion planning problem. An important part of this process is the determination of families of initial configurations containing attractive characteristics, that is, configurations with lines, transformers and sets of lines and transformers (building blocks) that may form part of the optimal, or near-optimal, solutions yielded by methods such as Genetic Algorithms and Tabu Search. The A-Teams methodology of this paper has been developed to generate these initial configurations. The A-teams use a different group of heuristic methods available in literature which run concurrently in a heterogeneous network of workstations. Tests has been carried out for small, medium and large systems.

Key Words— Asynchronous Teams; Distributed Processing; Heuristics Methods; Transmission Planning; Transmission Systems.

1 Introdução

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

Este trabalho trata especificamente da fase *estática* do planejamento da transmissão. Métodos tipo Algoritmos Genéticos e Busca Tabu normalmente necessitam da determinação prévia de blocos de ramos atraentes (conjuntos coerentes de linhas e transformadores). Isto exige uma inicialização adequada desses métodos. A parte principal tanto dos Algoritmos Genéticos como dos métodos baseados na Busca Tabu são facilmente paralelizáveis. Acredita-se que uma paralelização via *Times Assíncronos* poderá ser altamente eficaz. Assim, neste trabalho buscou-se caminhar nessa direção. Como um primeiro passo, foi implementada uma versão de *Times-A* para resolver o problema de inicialização do problema de planejamento estático.

Quando o sistema é modelado pelo fluxo de carga DC, o problema de planejamento estático é um problema de programação não linear inteira mista (PNLIM). Dadas as dimensões que o problema assume para casos práticos em geral

observa-se o fenômeno da explosão combinatorial (problema NP-completo). Para uma alternativa de investimento, o problema se reduz a um problema de programação linear cujo objetivo é verificar a factibilidade da alternativa.

Em uma época na qual os recursos computacionais eram mais limitados, foram desenvolvidos métodos heurísticos como por exemplo os métodos de Garver (Garver, 1970), Mínimo Esforço (Monticelli *et al.*, 1982) e Mínimo Corte de Carga (Pereira *et al.*, 1985). Essas metodologias ainda são utilizadas por concessionárias como parte de procedimentos interativos que exigem uma participação ativa dos planejadores. Este trabalho incorpora essas metodologias na construção dos *Times-A* inicializadores.

Utilizou-se um modelo de programação multiagente conhecido como *Times Assíncronos (Times-A)*. Um *Time-A* consiste de uma rede computacional fortemente cíclica, onde agentes autônomos cooperam, trabalhando todo o tempo em paralelo, com a partilha permanente de resultados entre os membros do time. Inicialmente desenvolveu-se um *Time-A* com variantes dos algoritmos baseados no método de Garver, e, posteriormente, um outro time acrescido de variantes dos algoritmos de Mínimo Esforço e Mínimo Corte de Carga.

Foram feitos testes para três sistemas: Garver (6 barras/15 linhas), Sul brasileiro (46 barras/79 linhas) e Norte-Nordeste brasileiro (89 barras/179 linhas), em uma rede heterogênea de estações SUN, sistema operacional Solaris 2.5.1, com a biblioteca para processamento paralelo PVM 3.3.5.

2 Times Assíncronos

Em (Talukdar e De Souza, 1993) define-se *Time-A* como qualquer super-agente cujos agentes são autônomos, cujas comunicações são assíncronas e cujo fluxo de dados é cíclico. Em outras palavras, *Time-A* é um conjunto de processos independentes que trabalham em paralelo todo o tempo, onde as soluções geradas por um processo podem ser reutilizadas por outros.

O agente faz sua própria escolha sobre sua seleção de entrada, escalonamento e política de alocação de recursos. Agentes podem ler e escrever informações continuamente nas memórias compartilhadas sem qualquer sincronização entre elas, de maneira que um fluxo contínuo de modificações possa ser realizado tornando possível iterações e "feedback" entre os mesmos.

Pode-se detalhar melhor a idéia contida num *Time-A* com as seguintes considerações (De Souza, 1993):

- (1) Resultados acumulam nas memórias do fluxo de dados (como nos "blackboards") para formar populações (como nos algoritmos genéticos).
- (2) Estas populações são continuamente modificadas por dois tipos de agentes:
 - *Agentes de Construção* (que adiciona elementos às populações).
 - *Agentes de Destruição* (que elimina elementos das populações, mantendo fixo o número de elementos das mesmas).
- (3) As habilidades em resolver problemas de um fluxo de dados pode ser arbitrariamente repartida entre construção e destruição.
- (4) Não existe um controle centralizado, mas novos agentes autônomos pode ser facilmente adicionados ou retirados.
- (5) Os fluxos de dados destes *Times-A* não somente produzem boas soluções, mas eles parecem ser "scale-efficient", isto é, os mesmos podem ser montados para produzirem sempre as melhores soluções pela adição de certos agentes e memórias.

É importante observar que, após a fase de inicialização, cada algoritmo roda em seu próprio ritmo usando os dados disponíveis na(s) memória(s) compartilhada(s). Isto mostra o caráter assíncrono do *Time-A* que pode facilmente ser executado em paralelo; cada agente e memória podem ser alocados nas diferentes máquinas de uma rede de computadores.

3 Métodos Heurísticos Construtivos

3.1 Método de Garver

O modelo formulado em (Garver, 1970), também conhecido como *modelo de transportes*, foi a primeira proposta para planejamento de redes de

transmissão que usou programação linear. Esta metodologia consiste basicamente em resolver de maneira aproximada uma versão relaxada do modelo DC. Neste modelo somente se leva em conta a Primeira Lei de Kirchhoff e a capacidade de transmissão das linhas, não sendo levado em conta, portanto, a Segunda Lei de Kirchhoff.

Na metodologia de Garver, todo fluxo que não puder ser transportado pelas ligações normais, fluirão pelas ligações de sobrecarga, pois estas tem capacidades ilimitadas, e só passarão através das ligações de sobrecarga quando for impossível transportá-los pelas ligações normais, já que estas tem custos muito inferiores. Em cada estágio do processo de planejamento, deve-se resolver um problema de programação linear e logo adicionar um circuito na trajetória de maior sobrecarga. O processo é repetido até eliminar todas as sobrecargas.

A vantagem da metodologia de Garver é a simplicidade na implementação do algoritmo pois ela exige somente soluções sucessivas de programação linear. A maior limitação da metodologia é que ela não garante a obtenção da solução ótima do sistema planejado. Portanto, esta metodologia, em essência, é de natureza heurística.

A metodologia foi usada em três versões (agentes): *GarverA*, *GarverB* e *GarverC*.

Os algoritmos utilizados (Gallego, 1997), são descritos a seguir:

GarverA

- (1) Faz-se a análise DC (MINOS) do sistema (o banco de dados da rede já considera a presença das barras e linhas fictícias), penalizando-se a configuração pelo custo de um ramo predeterminado.
- (2) Com os resultados de (1), calcula-se o fluxo para cada ramo do sistema.
- (3) Ordena-se os ramos pelo valor dos respectivos fluxos, em ordem decrescente.
- (4) Se o fluxo do ramo mais sobrecarregado for menor que um limite preestabelecido no início do programa, então vá para (6). Caso contrário, o programa oferece duas opções de adição de linhas, dependendo do sistema em questão. Na primeira, pode-se simplesmente adicionar uma linha ao ramo mais sobrecarregado. Na segunda, é feita uma identificação da necessidade da criação de um caminho completo ou somente um ramo, a partir da adição do ramo mais sobrecarregado, a fim de evitar a presença de elementos desconexos no sistema.
- (5) Voltar para (1).
- (6) Efetua-se o *refinamento* do sistema, que consiste na verificação da possibilidade de se remover linhas adicionadas desnecessariamente.

GarverB

O algoritmo *GarverB* é semelhante ao *GarverA*, com a diferença que agora a ordenação é realizada baseada no número de linhas requeridas em cada ramo e não no fluxo. Da mesma forma, o critério de parada considera agora uma parcela mínima de linhas permitida.

GarverC

O algoritmo *GarverC* apresenta a mesma estrutura básica que o *GarverB*, exceto pela introdução das seguintes correções:

- correção de caminhos conectando linhas penduradas;
- correção de caminhos entre barras fictícias, para que a capacidade de transmissão das linhas em série seja suficiente;
- adição de linhas cuja fração seja superior a um valor de entrada.

3.2 Método de Mínimo Esforço

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

$$\Delta z_i = \psi_i^2 \Delta \gamma_i \quad (1)$$

onde:

ψ_i é a diferença angular do ramo *i* antes da adição.

$\Delta \gamma_i$ é a susceptância do circuito adicionado.

A metodologia foi usada em duas versões (agentes): *minD* e *minE*.

minD

O programa *minD* possui algoritmo semelhante aos anteriores com diferenças, relacionadas ao novo critério, somente nos seguintes itens:

- (1) Penaliza-se a configuração pela reatância de um ramo predeterminado.
- (2) Calcula-se o somatório das sobrecargas do sistema e o índice de mínimo esforço para cada ramo do sistema.
- (3) Ordena-se os ramos pelo valor dos respectivos índices de mínimo esforço.
- (4) Se o somatório das sobrecargas do sistema for menor que um limite preestabelecido no início do programa, então vá para (6). Caso contrário, escolhe-se aleatoriamente um dos três primeiros ramos da lista ordenada para a adição de uma linha ou de um caminho, com as mesmas considerações anteriores.

minE

A única diferença deste programa em relação ao *minD* consiste na introdução de uma normalização pelo custo de cada ramo, quando do cálculo do índice de mínimo esforço.

3.3 Método de Mínimo Corte de Carga

De maneira semelhante ao Método de Mínimo Esforço, este método (Pereira *et al.*, 1985) realiza a adição de linhas selecionadas de acordo com um índice de sensibilidade que permite encontrar linhas mais atrativas. O índice utilizado é:

$$IS_{mcc} = (\pi_i - \pi_j)(\theta_i - \theta_j) \quad (2)$$

onde:

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

θ_j são os ângulos de tensão de barra obtidos na análise DC do sistema.

Neste método elaborou-se uma única versão (agente): *mccF*.

mccF

O programa *mccF* também mantém a mesma estrutura dos anteriores com diferenças, associadas ao novo índice, somente nos itens:

- (1) Penaliza-se a configuração pela reatância de um ramo predeterminado.
- (2) Calcula-se o corte de carga do sistema (via MINOS), e o índice de mínimo corte de carga, aqui já normalizado pelo custo, para cada ramo do sistema.
- (3) Ordena-se os ramos pelo valor dos respectivos índices de corte de carga.
- (4) Se o corte de carga do sistema for menor que um limite preestabelecido no início do programa, então vá para (6). Caso contrário, escolhe-se aleatoriamente um dos três primeiros ramos da lista ordenada para a adição de uma linha ou de um caminho, com as mesmas considerações anteriores.

4 Modelo de Time-A proposto

A estrutura do *Time-A* implementada é composta pelos seguintes objetos:

- *memória central* - memória compartilhada pelos agentes, a qual é formada por uma população de configurações. O tamanho desta população é constante e determinado previamente pelo usuário.
- *agentes de construção* - são os diversos algoritmos heurísticos construtivos implementados (*garverA*, *garverB*, etc.).
- *agente destruidor* - mantém constante o tamanho da população.

Um exemplo de *Time-A*, dentro da estrutura elaborada, é mostrado na Fig. 1. Os agentes são representados por setas que indicam um fluxo cíclico em relação à memória central, ou seja, os mesmos lêem os dados da memória, alteram esses dados, e os reescrevem na mesma. A memória central, representada por um retângulo, contém a

população de configurações. Os algoritmos implementados na elaboração do time assíncrono são representados por linhas tracejadas, definindo processos distintos. Daí, observa-se que o agente destruidor e a memória central compõem o processo *inic*. Note que o agente destruidor não é um processo independente como os outros agentes, mas sim uma das funções do processo *inic* que age para manter constante o tamanho da população.

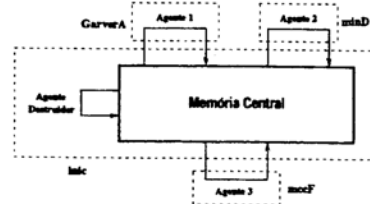


Figura 1. Exemplo de *Time-A* inicializador com três agentes.

A implementação paralela usa o modelo *Mestre/Escravo*, pois o *inic* dá início ao processamento do time, partindo um grupo de agentes predeterminados pelo usuário. Porém, é possível a adição/remoção dos agentes de forma manual, o que reforça a característica de autonomia dos mesmos (agentes entram ou saem do time sem interferir no funcionamento uns dos outros). A remoção manual dos agentes é feita por um programa auxiliar, enquanto que a adição dos mesmos é feita executando-os diretamente no "prompt" do sistema operacional.

O programa *inic* simula uma memória compartilhada e utiliza um controle de acesso aos dados para evitar conflitos ao se tentar acessá-la. Dessa forma, dois agentes não podem ler ou escrever ao mesmo tempo. Sendo assim, quando um agente estiver acessando a memória central, os outros necessariamente esperam numa fila do tipo FIFO ("First-In-First-Out"). Outra característica importante é o fato de que, a cada instante do processamento do *Time-A*, cada configuração (elemento da população) é atualizada somente por um agente, isto é, dois agentes não podem trabalhar sobre os mesmos dados simultaneamente. Dessa forma, o número de membros do time nunca deve exceder o número de configurações da população. Se assim o for, o *inic* automaticamente se encarrega de eliminar gradativamente os agentes excedentes. A Fig. 2 dá uma idéia geral do funcionamento do processo *inic*.

Quanto aos agentes de construção, de uma maneira geral, servem para efetuar os cálculos sobre os dados da memória central, segundo uma heurística própria, atualizando-os até se satisfazer um determinado critério de parada. A Fig. 3 apresenta a estrutura de funcionamento dos agentes em geral. Eles trabalham sinergicamente para obter configurações iniciais diversificadas em menor tempo, não visando necessariamente a obtenção

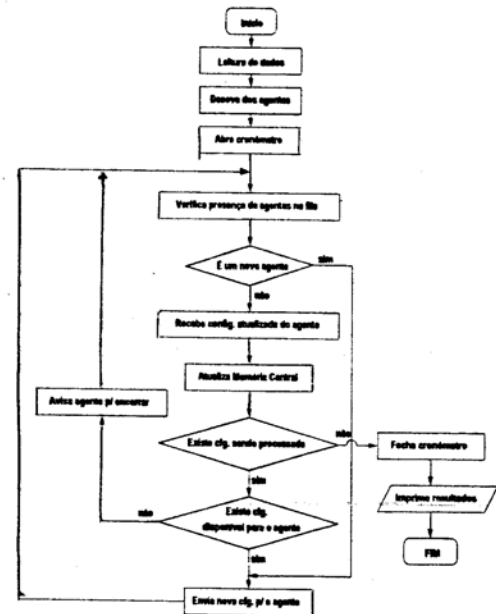


Figura 2. Estrutura básica do programa *inic*.

de um ótimo global, função esta do problema do planejamento como um todo. Os agentes podem ser combinados em quantidades diferentes entre as máquinas da rede.

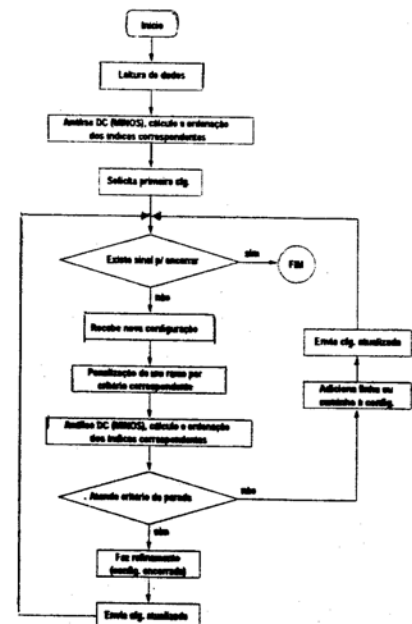


Figura 3. Estrutura básica dos agentes de construção.

5 Testes e Resultados

A partir da elaboração do *Time-A* inicializador, foram definidas duas categorias mais específicas denominadas: *Time-A Garver* que combina os algoritmos de Garver e *Time-A Misto* formado pela combinação de quaisquer dos algoritmos apresen-

tados. Os resultados dos testes realizados nas diferentes simulações são apresentados a seguir.

5.1 Time-A Garver

Foram realizadas diversas simulações para o sistema Garver (6 Barras - 15 Linhas) e algumas para o sistema Sul brasileiro (46 Barras - 79 Linhas). Os resultados destas simulações encontram-se resumidos nas Tabs. 1 a 4 que se seguem. A Tab. 1 mostra os tempos obtidos em dez simulações, para cada agente isoladamente, e serve para comparação com os tempos obtidos nas outras simulações. Para mostrar a característica "scale-efficient" do time, apresenta-se a Tab. 2 e a Fig. 4, referentes ao GarverA. As Tabs. 3 e 4 apresentam simulações com várias combinações diferentes, bem como com diferentes números de agentes, para os dois sistemas testados, respectivamente.

Tabela 1 - Tempos para cada agente de Garver

Sistema Garver (6 Barras - 15 Linhas)				
SIMUL.	COMBINAÇÕES (TEMPOS EM SEGS.)			
	1-A	1-B	1-C	
1	12.917 *	12.717 *	30.059	
2	12.578 *	12.533 *	26.069	
3	12.580 *	12.533 *	30.013	
4	12.629 *	12.630 *	26.477	
5	12.542 *	12.551 *	28.326	
6	12.619 *	12.517 *	28.359	
7	12.580 *	12.626 *	26.175	
8	12.622 *	12.671 *	26.272	
9	12.627 *	12.741 *	31.723	
10	12.594 *	12.643 *	26.630	
Médias	12.629	12.610	28.410	

OBS: População: 8 configurações
(*) Obteve-se a configuração ótima.

Tabela 2 - Tempos para diferentes números do GarverA

Sistema Garver (6 Barras - 15 Linhas)							
SIMUL.	COMBINAÇÕES (TEMPOS EM SEGUNDOS)						
	1-A	2-A	3-A	4-A	5-A	6-A	7-A
1	12.537	7.328	5.580	4.572	3.599	3.532	2.704
2	12.609	6.798	4.928	4.116	3.529	3.147	2.938
3	12.621	6.715	4.946	4.051	3.529	3.217	3.010
4	12.604	6.870	4.978	4.276	3.642	3.219	2.901
5	12.620	7.014	4.851	5.057	3.451	3.377	2.760
6	12.626	6.704	4.972	3.945	3.595	3.384	2.745
7	12.656	6.711	4.728	4.033	3.558	3.215	2.876
8	12.621	6.925	4.755	3.982	3.742	3.289	2.825
9	12.579	6.997	4.961	4.070	3.429	3.549	3.018
10	12.599	7.070	4.825	3.911	3.375	3.134	2.830
Médias	12.592	6.913	4.950	4.091	3.613	3.206	2.801

OBS: População: 8 configurações
Obteve-se a configuração ótima em todos os casos.

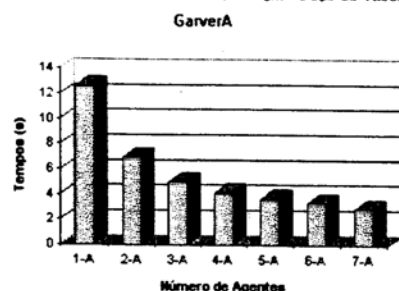


Fig. 4 - Tempos para diferentes números do GarverA

Tabela 3 - Tempos para combinações de agentes Garver

Sistema Garver (6 Barras - 15 Linhas)						
SIMUL.	COMBINAÇÕES (TEMPOS EM SEGUNDOS)					
	1A/1B/1C	1A/1C	2A/2B	3A/2C	3A/3B/1C	
1	6.096 *	11.358 *	4.852 *	7.220 *	4.671 *	
2	7.090 *	9.393	4.242 *	4.066 *	4.296 *	
3	5.777 *	10.367	4.022 *	4.118 *	4.161 *	
4	5.899 *	10.650 *	3.986 *	4.111 *	4.091 *	
5	7.053 *	8.967 *	3.766 *	5.712	4.295 *	
6	5.874 *	9.141 *	4.066 *	4.035	5.919 *	
7	5.964 *	10.353	4.199 *	5.638	4.074 *	
8	7.208 *	11.692 *	4.013 *	7.388 *	2.262 *	
9	7.138 *	9.722 *	3.931 *	5.613 *	4.170 *	
10	6.267 *	8.927 *	4.285 *	5.481	2.982 *	
Médias	6.439	10.057	4.156	5.336	4.093	

OBS: População: 8 configurações
(*) Obteve-se a configuração ótima. (Ver Fig. 5)

Tabela 4 - Tempos para combinações de agentes Garver

Sistema Sul brasileiro (46 Barras - 79 Linhas)				
SIMUL.	COMBINAÇÕES (TEMPOS EM SEGUNDOS)			
	2A/2B/2C	3A/3B	6A	5A/2C
1	71.430	47.927	44.559	72.690
2	52.065	45.949	44.540	95.035
3	49.040	43.183	44.859	75.567
4	54.461	42.686	45.157	75.565
5	93.982	47.935	49.435	46.589
6	71.307	41.218	47.308	46.732
7	68.822	43.222	48.965	71.798
8	71.515	44.406	47.972	73.654
9	47.652	47.284	43.309	71.195
10	72.472	44.597	42.922	56.249
Médias	65.275	44.632	45.776	72.587

OBS: População: 8 configurações

A configuração ótima do sistema Garver, ou seja uma linha adicionada no ramo 11 (3-5) e três linhas no ramo 14 (4-6), encontrada na maioria das simulações do Time-A Garver, é apresentada na Fig. 5.

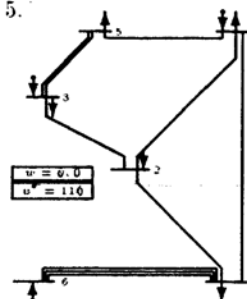


Fig. 5. Sistema Garver - Config. Ótima

A Fig. 6, na sequência, apresenta uma solução bem próxima da ótima, obtida para o sistema Sul brasileiro, faltando somente a adição de uma linha no ramo 14 (13-20). Na figura, os números entre parenteses correspondem à quantidade de linhas adicionadas. Nota-se que, para este sistema, em nenhuma configuração obteve-se o ótimo, mas em todas simulações obteve-se pelo menos uma configuração bem próxima do ótimo, o que pode ser verificado comparando-se com a apresentada no Apêndice A.

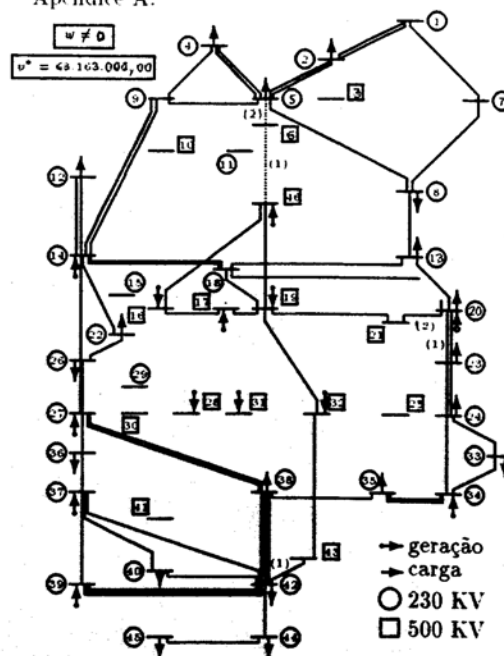


Fig. 6. Sistema Sul brasileiro - Melhor config. obtida.

5.2 Time-A Misto

Foram realizadas diversas simulações para os dois sistemas citados anteriormente, bem como para o sistema Norte-Nordeste brasileiro (89 Barras - 179 Linhas). Porém, são apresentados apenas os resultados para este último, pois trata-se de um sistema de maior complexidade para o planejamento da expansão. Estes resultados são exibidos na Tab. 5 em termos da porcentagem de acertos em relação a melhor configuração que se tem para este sistema. Ali, os mesmos representam a quantidade de linhas adicionadas que coincidem, em cada ramo, com aquelas da solução encontrada em (Gallego, 1997).

Tabela 5 - Porcentagem de acertos

Sistema Norte-Nordeste brasileiro (89 Barras - 179 Linhas)			
Agentes	Nor2008 (%)	Agentes	Nor2008 (%)
A	73.46	C/D	72.91
B	74.30	C/E	65.92
C	68.16	C/F	52.23
D	74.30	D/E	74.30
E	68.44	A/B/C	69.27
F	56.98	A/B/E	59.22
A/B	79.69	A/B/C/D/E	72.07
A/C	74.86	A/A/B/B/C/C	70.39
A/D	71.28	A/A/B/B/C/D/E	66.44
B/C	72.91		

Nota-se que, apesar do aumento do número de agentes, não se obtém necessariamente resultados mais próximos do ótimo, porém os tempos sempre são menores e as configurações iniciais resultantes são mais diversificadas, o que é desejável num programa inicializador.

6 Conclusões

Com a tendência cada vez maior do uso de redes de computadores nas concessionárias de energia elétrica, aliado a disponibilidade de bibliotecas para processamento paralelo, tem-se o ambiente propício para a utilização de modelos de programação distribuída, como times assíncronos, nos problemas complexos e de natureza combinatorial como o problema de planejamento da expansão da transmissão de energia elétrica.

Particularmente, neste trabalho, aplicou-se times assíncronos aos algoritmos heurísticos construtivos (Garver, Mínimo Esforço e Mínimo Corte de Carga), para a implementação de programas inicializadores. Estes, por sua vez, são indispensáveis para o uso de algoritmos combinatoriais quando se trabalha com sistemas de grande porte, pois a eficácia da busca no espaço de soluções é função das soluções iniciais factíveis.

Dos testes realizados, concluiu-se que é possível implementar o modelo de times assíncronos a fim de se obter um desempenho melhor que o apresentado por algoritmos de maneira isolada. Os testes comprovam que é possível se obter soluções em tempos menores. É clara também a perspectiva de melhores resultados, do ponto de vista do planejamento da expansão como um todo,

quando da aplicação do modelo para os algoritmos combinatoriais tais como Algoritmos Genéticos, "Simulated Annealing", Busca Tabu e suas combinações.

Referências Bibliográficas

- De Souza, P.S. (1993). *Asynchronous organizations for multi-algorithm problems*. Pittsburgh: Department of Electrical and Computer Engineering. Carnegie Mellon University. (Dissertation. Ph.D.)
- Gallego, R.A. (1997). *Planejamento a longo prazo de sistemas de transmissão usando técnicas de otimização combinatorial*. Campinas: Faculdade de Eng. Elétrica e de Computação da UNICAMP. 243p. (Tese, Doutorado em Eng. Elétrica)
- Garver, L.L. (1970). Transmission network estimation using linear programming. *IEEE Transactions on Power Apparatus and Systems*, v.PAS-89, pp. 1688-1697.
- Monticelli, A. et al. (1982). Interactive transmission network planning using a least-effort criterion. *IEEE Transactions on Power Apparatus and Systems*, v.PAS-101, n.10, pp. 3919-3925.
- Pereira, M.V.F. e Pinto, L.M.V.G. (1985). Application of sensibility analysis of load supplying capability to interactive transmission expansion planning. *IEEE Transactions on Power Apparatus and Systems*, v.PAS-104, n.2, p.381-389, Feb.
- Talukdar, S.N. e De Souza, P.S. (1993). Scale efficient organizations. *Proceedings of the International Conference on Systems, Man and Cybernetics*. New York: IEEE. pp. 1458-1463.

Apêndice

Configurações ótimas

Sist. Garver (6 Barras - 15 Linhas) linhas adicionadas: $n_{03-05} = 1, n_{04-06} = 3$. Custo da Linhas = 110. corte de carga = 0.0
Sist. Sul brasileiro (46 Barras - 79 Linhas) linhas adicionadas: $n_{13-20} = 1, n_{20-23} = 1, n_{20-21} = 2, n_{42-43} = 1, n_{46-06} = 1, n_{05-06} = 2$. Custo das Linhas = 70289. corte de carga = 1 MW
Sist. Norte-Nordeste brasileiro (89 Barras - 179 Linhas) (Melhor Configuração Obtida em (Gallego, 1997)) linhas adicionadas: $n_{01-02} = 1, n_{02-04} = 1, n_{04-05} = 3, n_{04-06} = 2, n_{04-81} = 3, n_{15-16} = 1, n_{05-56} = 1, n_{05-58} = 3, n_{06-70} = 1, n_{12-13} = 1, n_{14-59} = 5, n_{15-46} = 4, n_{16-44} = 2, n_{16-61} = 0, n_{18-50} = 1, n_{13-15} = 11, n_{16-74} = 5, n_{00-66} = 2, n_{22-37} = 2, n_{22-58} = 1, n_{24-25} = 1, n_{25-26} = 3, n_{26-27} = 1, n_{26-27} = 1, n_{27-53} = 1, n_{28-35} = 2, n_{34-39} = 1, n_{35-46} = 1, n_{35-47} = 1, n_{35-51} = 3, n_{36-39} = 1, n_{36-46} = 1, n_{39-42} = 2, n_{39-86} = 1, n_{40-46} = 2, n_{41-64} = 2, n_{47-48} = 5, n_{48-50} = 2, n_{52-59} = 1, n_{53-76} = 1, n_{53-86} = 1, n_{54-58} = 1, n_{56-57} = 1, n_{58-76} = 2, n_{62-72} = 1, n_{63-64} = 1, n_{63-66} = 3, n_{68-83} = 1, n_{70-82} = 1, n_{71-75} = 2, n_{71-83} = 1, n_{73-75} = 1. Custo das Linhas = 2625663, corte de carga = 0.0 $



**XV SNTPEE
SEMINÁRIO NACIONAL
DE PRODUÇÃO E
TRANSMISSÃO DE
ENERGIA ELÉTRICA**

GPL/04

**17 à 22 de outubro de 1999
Foz do Iguaçu – Paraná - Brasil**

**GRUPO VII
GRUPO DE ESTUDO DE PLANEJAMENTO DE SISTEMAS ELÉTRICOS (GPL)**

**TIME ASSÍNCRONO INICIALIZADOR DE MÉTODOS COMBINATORIAIS PARA
PLANEJAMENTO DA EXPANSÃO DA TRANSMISSÃO**

Sérgio Azevedo de Oliveira* Cláudio R. Thomas de Almeida Alcir Monticelli

UNESP-FEIS-DEE

UNICAMP-FEEC-DSEE

RESUMO

Este artigo trata da aplicação de *Times Assíncronos* (*Times-A*) ao planejamento da expansão de sistemas de transmissão de energia elétrica. Uma parte importante desse processo é a determinação de famílias de soluções iniciais contendo características atrativas, isto é, configurações com linhas e transformadores e conjuntos de linhas e transformadores (blocos construtivos) que podem formar parte das soluções ótimas, ou quase ótimas, obtidas via métodos como os Algoritmos Genéticos e Busca Tabu. *Times-A* foram desenvolvidos para gerar estas configurações iniciais. Esses times utilizam diferentes métodos heurísticos construtivos disponíveis na literatura, os quais são distribuídos em uma rede heterogênea de computadores e executados de modo concorrente. Testes foram feitos para sistemas de pequeno, médio e grande porte.

PALAVRAS-CHAVE

Times Assíncronos; Processamento Paralelo e Distribuído; Métodos Heurísticos Construtivos; Planejamento da Expansão da Transmissão.

1.0 INTRODUÇÃO

O planejamento da expansão de redes de transmissão de sistemas de energia elétrica determina *quando, onde e que* tipos de linhas e/ou transformadores devem ser instalados na rede a fim de que o sistema opere adequadamente, para uma demanda futura predeterminada, com o menor investimento possível. O planejamento *dinâmico* (*quando*) em geral é decomposto em subproblemas *estáticos* que tratam das

questões *onde e que tipo* (planejamento em um estágio, de um ano inicial a um ano final, dados). Quando o sistema elétrico é modelado pelo fluxo de carga DC, o problema de planejamento estático é um problema de programação não linear inteira mista (PNLIM). Dadas as dimensões que o problema assume para casos práticos em geral observa-se o fenômeno da explosão combinatória (problema NP-completo). Para uma alternativa de investimento, o problema se reduz a um problema de programação linear cujo objetivo é verificar a factibilidade da alternativa. Soluções deste problema são obtidas via métodos combinatoriais, tais como Algoritmos Genéticos, Busca Tabu, entre outros. Sendo que uma parte importante desse processo é a determinação de famílias de soluções iniciais contendo características atrativas, isto é, configurações com linhas e transformadores e conjuntos de linhas e transformadores (blocos construtivos) que podem formar parte das soluções ótimas, ou quase ótimas.

Anteriormente, em uma época na qual os recursos computacionais eram mais limitados, foram desenvolvidos métodos heurísticos como por exemplo os métodos de Garver, Mínimo Esforço e Mínimo Corte de Carga. Essas metodologias ainda são utilizadas por concessionárias como parte de procedimentos interativos que exigem uma participação ativa dos planejadores. E atualmente, as concessionárias têm em seu parque computacional inúmeras estações de trabalho que podem ser utilizadas, com o auxílio de bibliotecas para processamento paralelo, como uma máquina paralela virtual, onde é possível o processamento concorrente de diversos programas computacionais. Neste trabalho, que trata especificamente da fase estática do planejamento, foi implementado um modelo

de programação multi-agente conhecido como Times Assíncronos (*Times-A*) para resolver o problema de inicialização do problema de planejamento da expansão da transmissão. Um *Time-A* consiste de uma rede computacional fortemente cíclica, onde agentes autônomos (aqui, os métodos heurísticos construtivos) cooperam, trabalhando todo o tempo em paralelo, com a partilha permanente de resultados (configurações candidatas) entre os membros do time. Em outras palavras, *Time-A* é um conjunto de processos independentes que trabalham em paralelo todo o tempo, onde as soluções geradas por um processo podem ser reutilizada por outros.

Foram desenvolvidos *Times-A* com variantes dos algoritmos baseados no método de Garver e variantes dos algoritmos de Mínimo Esforço e Mínimo Corte de Carga. Foram feitos testes para três sistemas: Garver (6 barras/15 linhas), Sul brasileiro (46 barras/79 linhas) e Norte-Nordeste brasileiro (89 barras/179 linhas), em uma rede heterogênea de estações SUN, sistema operacional Solaris 2.5.1, com a biblioteca para processamento paralelo PVM 3.3.5. Os resultados foram aplicados aos Algoritmos Genéticos e Busca Tabu, como soluções iniciais factíveis, obtendo-se assim soluções ótimas e quase-ótimas para os sistemas testados. A diversidade e qualidade das soluções obtidas com o uso dos times assíncronos inicializadores, comprovam sua eficácia em gerar soluções iniciais factíveis para os algoritmos combinatoriais.

2.0 TIMES ASSÍNCRONOS

Em (Talukdar e De Souza, 1993) define-se *Time-A* como qualquer super-agente cujos agentes são autônomos, cujas comunicações são assíncronas e cujo fluxo de dados é cíclico. Em outras palavras, *Time-A* é um conjunto de processos independentes que trabalham em paralelo todo o tempo, onde as soluções geradas por um processo podem ser reutilizada por outros. O agente faz sua própria escolha sobre sua seleção de entrada, escalonamento e política de alocação de recursos. Agentes podem ler e escrever informações continuamente nas memórias compartilhadas sem qualquer sincronização entre elas, de maneira que um fluxo contínuo de modificações possa ser realizado tornando possível iterações e "feedback" entre os mesmos.

Pode-se detalhar melhor a idéia contida num *Time-A* com as seguintes considerações (De Souza, 1993):

1. Resultados acumulam nas memórias do fluxo de dados (como nos "blackboards") para formar populações (como nos algoritmos genéticos).
2. Estas populações são continuamente modificadas por dois tipos de agentes:

- *Agentes de Construção* (que adiciona elementos às populações), e
 - *Agentes de Destruição* (que elimina elementos das populações, mantendo fixo o número de elementos das mesmas).
3. As habilidades em resolver problemas de um fluxo de dados pode ser arbitrariamente repartida entre construção e destruição.
 4. Não existe um controle centralizado, mas novos agentes autônomos pode ser facilmente adicionados ou retirados.
 5. Os fluxos de dados destes *Times-A* não somente produzem boas soluções, mas eles parecem ser "scale-efficient", isto é, os mesmos podem ser montados para produzirem sempre as melhores soluções pela adição de certos agentes e memórias.

É importante observar que, após a fase de inicialização, cada algoritmo roda em seu próprio ritmo usando os dados disponíveis na(s) memória(s) compartilhada(s). Isto mostra o caráter assíncrono do *Time-A* que pode facilmente ser executado em paralelo; cada agente e memória podem ser alocados nas diferentes máquinas de uma rede de computadores.

3.0 MÉTODOS HEURÍSTICOS CONSTRUTIVOS

3.1 Método de Garver

O modelo formulado em (Garver, 1970), também conhecido como *modelo de transportes*, foi a primeira proposta para planejamento de redes de transmissão que usou programação linear. Esta metodologia consiste basicamente em resolver de maneira aproximada uma versão relaxada do modelo DC. Neste modelo somente se leva em conta a Primeira Lei de Kirchhoff e a capacidade de transmissão das linhas, não sendo levado em conta, portanto, a Segunda Lei de Kirchhoff.

Na metodologia de Garver, todo fluxo que não puder ser transportado pelas ligações normais, fluirão pelas ligações de sobrecarga, pois estas têm capacidades ilimitadas, e só passarão através das ligações de sobrecarga quando for impossível transportá-los pelas ligações normais, já que estas tem custos muito inferiores. Em cada estágio do processo de planejamento, deve-se resolver um problema de programação linear e logo adicionar um circuito na trajetória de maior sobrecarga. O processo é repetido até eliminar todas as sobrecargas.

A vantagem da metodologia de Garver é a simplicidade na implementação do algoritmo pois ela exige somente soluções sucessivas de programação linear. A maior limitação da metodologia é que ela não garante a obtenção da solução ótima do sistema

planejado. Portanto, esta metodologia, em essência, é de natureza heurística.

A metodologia foi usada em três versões (agentes): *GarverA*, *GarverB* e *GarverC*.

Os algoritmos utilizados, são descritos a seguir:

3.1.1 GarverA

1. Faz-se a análise DC (MINOS) do sistema (o banco de dados da rede já considera a presença das barras e linhas fictícias), penalizando-se a configuração pelo custo de um ramo predeterminado.
2. Com os resultados de (1), calcula-se o fluxo para cada ramo do sistema.
3. Ordena-se os ramos pelo valor dos respectivos fluxos, em ordem decrescente.
4. Se o fluxo do ramo mais sobrecarregado for menor que um limite preestabelecido no início do programa, então vá para (6). Caso contrário, o programa oferece duas opções de adição de linhas, dependendo do sistema em questão. Na primeira, pode-se simplesmente adicionar uma linha ao ramo mais sobrecarregado. Na segunda, é feita uma identificação da necessidade da criação de um caminho completo ou somente um ramo, a partir da adição do ramo mais sobrecarregado, a fim de evitar a presença de elementos desconexos no sistema.
5. Voltar para (1).
6. Efetua-se o *refinamento* do sistema, que consiste na verificação da possibilidade de se remover linhas adicionadas desnecessariamente.

3.1.2 GarverB

O algoritmo *GarverB* é semelhante ao *GarverA*, com a diferença que agora a ordenação é realizada baseada no número de linhas requeridas em cada ramo e não no fluxo. Da mesma forma, o critério de parada considera agora uma parcela mínima de linhas permitida.

3.1.3 GarverC

O algoritmo *GarverC* apresenta a mesma estrutura básica que o *GarverB*, exceto pela introdução das seguintes correções:

- correção de caminhos conectando linhas penduradas;
- correção de caminhos entre barras fictícias, para que a capacidade de transmissão das linhas em série seja suficiente;
- adição de linhas cuja fração seja superior a um valor de entrada.

3.2 Método de mínimo esforço

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

$$Dz_i = y_i^2 Dg_i$$

onde:

y_i é a diferença angular do ramo i antes da adição,

Dg_i é a susceptância do circuito adicionado.

A metodologia foi usada em duas versões (agentes): *minD* e *minE*.

3.2.1 minD

O programa *minD* possui algoritmo semelhante aos anteriores com diferenças, relacionadas ao novo critério, somente nos seguintes itens:

1. Penaliza-se a configuração pela reatância de um ramo predeterminado.
2. Calcula-se o somatório das sobrecargas do sistema e o índice de mínimo esforço para cada ramo do sistema.
3. Ordena-se os ramos pelo valor dos respectivos índices de mínimo esforço.
4. Se o somatório das sobrecargas do sistema for menor que um limite preestabelecido no início do programa, então vá para (6). Caso contrário, escolhe-se aleatoriamente um dos três primeiros ramos da lista ordenada para a adição de uma linha ou de um caminho, com as mesmas considerações anteriores.

3.2.2 minE

A única diferença deste programa em relação ao *minD* consiste na introdução de uma normalização pelo custo de cada ramo, quando do cálculo do índice de mínimo esforço.

3.3 Método de mínimo corte de carga

De maneira semelhante ao Método de Mínimo Esforço, este método (Pereira *et al.*, 1985) realiza a adição de linhas selecionadas de acordo com um índice de sensibilidade que permite encontrar linhas mais atrativas. O índice utilizado é:

$$IS_{mcc} = (p_i - p_j)(q_i - q_j)$$

onde:

p_j é o multiplicador de Lagrange da j -ésima restrição

$$(Bq + g + r = d),$$

q_j são os ângulos de tensão de barra obtidos na análise DC do sistema.

Neste método fez-se uma única versão (agente): *mccF*.

3.3.1 *mccF*

O programa *mccF* também mantém a mesma estrutura dos anteriores com diferenças, associadas ao novo índice, somente nos itens:

1. Penaliza-se a configuração pela reatância de um ramo predeterminado.
2. Calcula-se o corte de carga do sistema (via MINOS), e o índice de mínimo corte de carga, aqui já normalizado pelo custo, para cada ramo do sistema.
3. Ordena-se os ramos pelo valor dos respectivos índices de corte de carga.
4. Se o corte de carga do sistema for menor que um limite preestabelecido no início do programa, então vá para (6). Caso contrário, escolhe-se aleatoriamente um dos três primeiros ramos da lista ordenada para a adição de uma linha ou de um caminho, com as mesmas considerações anteriores.

4.0 MODELO DE *TIME-A* PROPOSTO

A estrutura do *Time-A* implementada é composta pelos seguintes objetos:

- *memória central* - memória compartilhada pelos agentes, a qual é formada por uma população de configurações. O tamanho desta população é constante e determinado previamente pelo usuário.
- *agentes de construção*- são os diversos algoritmos heurísticos construtivos implementados (*GarverA*, *GarverB*, etc.).
- *agente destruidor* - mantém constante o tamanho da população.

Um exemplo de *Time-A*, dentro da estrutura elaborada, é mostrado na Figura 1. Os agentes são representados por setas que indicam um fluxo cíclico em relação à memória central, ou seja, os mesmos lêem os dados da memória, alteram esses dados, e os reescrevem na mesma. A memória central, representada por um retângulo, contém a população de configurações. Os algoritmos implementados na elaboração do time assíncrono são representados por linhas tracejadas, definindo processos distintos. Daí, observa-se que o

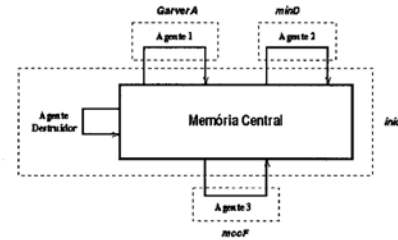


FIGURA 1 - Exemplo de *Time-A* com três agentes.

agente destruidor e a memória central compõem o processo *inic*. Note que o agente destruidor não é um processo independente como os outros agentes, mas sim uma das funções do processo *inic* que age para manter constante o tamanho da população.

A implementação paralela usa o modelo *Mestre/Escravo*, pois o *inic* dá início ao processamento do time, partindo um grupo de agentes predeterminados pelo usuário. Porém, é possível a adição/remoção dos agentes de forma manual, o que reforça a característica de autonomia dos mesmos (agentes entram ou saem do time sem interferir no funcionamento uns dos outros). A remoção manual dos agentes é feita por um programa auxiliar, enquanto que a adição dos mesmos é feita executando-os diretamente no “prompt” do sistema operacional.

O programa *inic* simula uma memória compartilhada e utiliza um controle de acesso aos dados para evitar conflitos ao se tentar acessá-la. Dessa forma, dois agentes não podem ler ou escrever ao mesmo tempo. Sendo assim, quando um agente estiver acessando a memória central, os outros necessariamente esperam numa fila do tipo FIFO (“First-In-First-Out”). Outra característica importante é o fato de que, a cada instante do processamento do *Time-A*, cada configuração (elemento da população) é atualizada somente por um agente, isto é, dois agentes não podem trabalhar sobre os mesmos dados simultaneamente. Dessa forma, o número de membros do time nunca deve exceder o número de configurações da população. Se assim o for, o *inic* automaticamente se encarrega de eliminar gradativamente os agentes excedentes. A Figura 2 dá uma idéia geral do funcionamento do processo *inic*.

Quanto aos agentes de construção, de uma maneira geral, servem para efetuar os cálculos sobre os dados da memória central, segundo uma heurística própria, atualizando-os até se satisfazer um determinado critério de parada. A Figura 3 apresenta a estrutura de funcionamento dos agentes em geral. Eles trabalham sinergicamente para obter configurações iniciais diversificadas em menor tempo, não visando necessa-

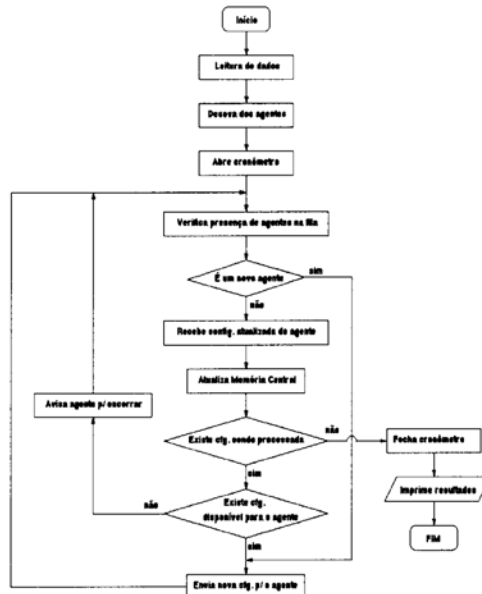
FIGURA 2 - Estrutura básica do programa *inic*.

FIGURA 3 - Estrutura básica dos agentes de construção.

riamente a obtenção de um ótimo global, função esta do problema do planejamento como um todo. Os agentes podem ser combinados em quantidades diferentes entre as máquinas da rede.

5.0 TESTES E RESULTADOS

Foram realizadas diversas simulações para sistemas de pequeno, médio e grande porte, respectivamente: Garver (6 Barras – 15 Linhas), Sul brasileiro (46

Barras – 79 Linhas) e Norte-Nordeste brasileiro (89 Barras – 179 Linhas). Tabelas, gráficos e configurações ótimas referentes as simulações estão disponíveis em (De Oliveira *et al.*, 1998).

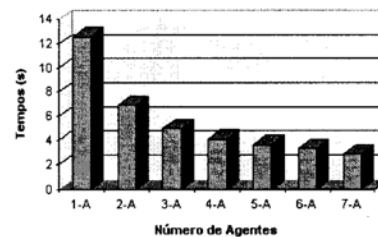
Para mostrar a característica “scale-efficient” do time, reapresenta-se aqui a Tabela 1 e a Figura 4, referentes ao agente *GarverA*.

TABELA 1 – Tempos para diferentes números do *garverA*

Simulação	Sistema Garver (6 Barras – 15 Linhas)						
	Combinações (Tempos em segundos)						
	1-A	2-A	3-A	4-A	5-A	6-A	7-A
1	12,54	7,33	5,58	4,57	4,00	3,53	2,70
2	12,61	6,80	4,93	4,12	3,53	3,15	2,93
3	12,62	6,72	4,94	4,05	3,81	3,22	3,01
4	12,60	6,87	4,98	4,28	3,64	3,22	2,90
5	12,62	7,01	4,85	3,96	3,45	3,38	2,76
6	12,63	6,70	4,97	3,95	3,60	3,38	2,75
7	12,56	6,71	4,73	4,03	3,56	3,22	2,88
8	12,62	6,93	4,74	3,98	3,74	3,29	2,83
9	12,58	7,00	4,96	4,07	3,43	3,55	3,02
10	12,55	7,07	4,83	3,91	3,38	3,13	2,83
Méd.	12,59	6,91	4,95	4,09	3,61	3,31	2,86

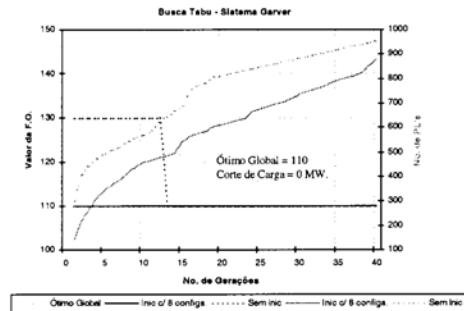
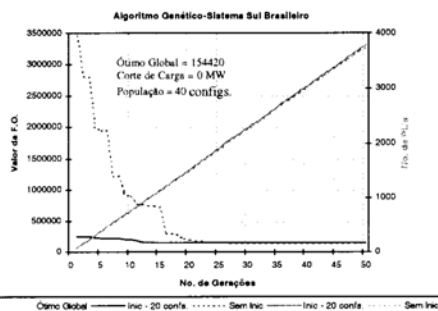
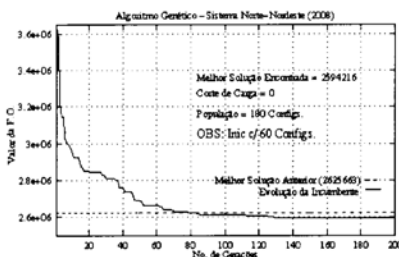
NOTAS: População: 8 configurações

Obteve-se a configuração ótima em todos os casos.

FIGURA 4 – Tempos para diferentes números do *garverA*

Neste trabalho, foram realizadas algumas simulações com os algoritmos Busca Tabu e Algoritmos Genéticos (Gallego, 1997) para verificar a eficiência no uso dos *Times-A* como um algoritmo inicializador. Ou seja, foram geradas configurações iniciais factíveis através de um *Time-A* composto com todas as variantes dos métodos heurísticos construtivos descritos anteriormente. Em seguida, utilizou-se estas configurações para inicializar os algoritmos citados, comparando os resultados com aqueles obtidos, sem o uso do time inicializador.

As Figuras 5 e 6, apresentam alguns dos resultados obtidos para os sistemas Garver e Sul brasileiro, simulados sem e com o auxílio do *Time-A*, para os algoritmos Busca Tabu e Algoritmos Genéticos, respectivamente; enquanto que a Figura 7 apresenta os resultados obtidos para o sistema Norte-Nordeste brasileiro, para os Algoritmos Genéticos, com 60 configurações iniciais geradas previamente.

FIGURA 5—Resultado sem e com *Inic* para Garver/BTFIGURA 6 – Resultado sem e com *Inic* para Sul/AG
Veja, nas figuras anteriores, que o uso do *Time-A*, possibilitou a obtenção dos ótimos globais, para ambos sistemas, em um número menor de ciclos ou gerações (ou seja, com menos resoluções de PL's).FIGURA 7 – Resultado com *Inic* para Norte-Nord/AG

Na Figura acima, selecionou-se 60 melhores configurações de diferentes simulações prévias para o sistema Norte-Nordeste brasileiro, simulado aqui com os Algoritmos Genéticos. Verifica-se que com o auxílio destas configurações iniciais, obteve-se um resultado melhor que o apresentado em (Gallego, 1997).

6.0 CONCLUSÕES

Neste trabalho, aplicou-se times assíncronos aos algoritmos heurísticos construtivos (Garver, Mínimo Esforço e Mínimo Corte de Carga) de planejamento da expansão da transmissão, para a implementação de programa inicializador. Este, por sua vez, é indispensável para o uso de algoritmos combinatoriais

como Busca Tabu e Algoritmos Genéticos, quando se trabalha com sistemas de grande porte, pois a eficácia da busca no espaço de soluções é função das soluções iniciais factíveis.

Dos testes realizados, concluiu-se que é possível implementar o modelo de times assíncronos a fim de se obter um desempenho melhor que o apresentado por algoritmos de maneira isolada. Os testes comprovam também a eficácia no uso das configurações resultantes dos *Times-A* nos algoritmos combinatoriais tais como Algoritmos Genéticos e Busca Tabu.

Com a tendência cada vez maior do uso de redes de computadores nas concessionárias de energia elétrica, aliado a disponibilidade de bibliotecas para processamento paralelo, tem-se o ambiente propício para a utilização de modelos de programação distribuída, como times assíncronos, nos problemas complexos e de natureza combinatoria como o problema de planejamento da expansão da transmissão de energia elétrica.

7.0 REFERÊNCIAS BIBLIOGRÁFICAS

- TALUKDAR, S.N. e DE SOUZA, P.S. (1993). Scale efficient organizations, *Procs. of the Int. Conference on Systems, Man and Cybernetics*, New Y.:IEEE, pp.1458-1463.
- DE SOUZA, P.S. (1993). *Asynchronous organization for multi-algorithm problems*, Pittsburgh, Department of Electrical and Computer Engineering, Carnegie Mellon University. (Dissertation, Ph.D.)
- GARVER, L.L. (1970). Transmission network estimation using linear programming, *IEEE Trans. on PAS*, v.PAS-89, pp. 1688-1697.
- MONTICELLI, A. et al. (1982). Interactive transmission network planning using a least-effort criterion, *IEEE Trans. on PAS*, v.PAS-101, n.10, pp. 3919-3925.
- PEREIRA, M.V.F. e PINTO, L.M.V.G. (1985). Application of sensibility analysis of load supplying capability to interactive transmission expansion planning, *IEEE GALLEGO, R.A. (1997). Planejamento a longo prazo de sistemas de transmissão usando técnicas de otimização combinatorial*, Campinas: Faculdade de Eng. Elétrica e de Computação da UNICAMP, 243p. (Tese, Dr. Eng. Elétrica)

DE OLIVEIRA, S.A. et al. (1998). Times assíncronos aplicados a métodos heurísticos construtivos de planejamento da expansão da transmissão.

(SEPARATA)

DADOS BIOGRÁFICOS

Sérgio Azevedo de Oliveira – Nascido em Murutinga do Sul-SP, em 1958, Engenheiro Eletricista pela UNESP/Iilha Solteira em 1981. Mestre em Eng. Elétrica pela UFSC em 1989. Atualmente fazendo doutorado na UNICAMP. Professor junto ao Depto. de Engenharia Elétrica, da FEIS/UNESP, desde 1982. *Student Member* do IEEE, Membro da SBA. Áreas de atuação: Planejamento da Expansão da Transmissão, Processamento Paralelo e Distribuído, Meta-Heurísticas, Times Assíncronos.



Anais do XIII Congresso Brasileiro de Automática – CBA 2000
11 a 14 de setembro de 2000 – Florianópolis – SC – Brasil

ALGORITMO GENÉTICO PARALELO APLICADO AO PLANEJAMENTO DA EXPANSÃO DA TRANSMISSÃO

SÉRGIO A. DE OLIVEIRA, RUBÉN A. ROMERO

*Depto. de Engenharia Elétrica, Universidade Estadual Paulista
C. P. 31, 15385-000 Ilha Solteira, SP, BRASIL*

E-mails: grilo@dee.feis.unesp.br, ruben@dee.feis.unesp.br

RAMÓN A. GALLEGO

*Fac. de Ingeniería Eléctrica, Universidad Tecnológica de Pereira
A. P. 097, Pereira, RI, COLÔMBIA*

E-mails: ralfonso@andromeda.utp.edu.co

ALCIR MONTICELLI

*Depto. de Sistemas de Energia Elétrica, Universidade Estadual de Campinas
C. P. 6101, 13081-970 Campinas, SP, BRASIL*

E-mail: alcir@dsee.fee.unicamp.br

Resumo— Dentre os métodos de otimização combinatorial para a resolução de problemas complexos e de grande porte, como o problema do planejamento da expansão da transmissão de energia elétrica, destacam-se os Algoritmos Genéticos (GA) que têm em seu funcionamento um paralelismo intrínseco. Este trabalho investiga este fato, apresentando diferentes versões paralelas para a resolução do citado problema. São apresentados a formulação matemática do problema, detalhes dos algoritmos, bem como resultados para os sistemas Garver (6 barras / 15 linhas) e o Sul Brasileiro (46 barras / 79 Linhas).

Abstract— Amongst combinatorial optimization methods for the resolution of complex and very large problems, like the network static expansion planning problem, one distinguish the Genetic Algorithms method because there is an intrinsic parallelism involved. This work investigates this fact and presents parallel versions for the resolution of the mentioned problem. The mathematical formulation of the problem and details of the algorithms are presented, as well as results for the Garver's System (6 buses / 15 branches) and Southern Brazilian System (46 buses / 79 branches).

Key Words— network static expansion planning; combinatorial optimization; genetic algorithms; parallel algorithms; distributed computing.

1 Introdução

O planejamento da expansão de redes de transmissão de sistemas de energia elétrica determina *quando, onde e que* tipos de linhas e/ou transformadores devem ser instalados na rede a fim de que o sistema opere adequadamente para uma demanda futura predeterminada e realizando o menor investimento possível. O planejamento *dinâmico (quando)* em geral é decomposto em subproblemas *estáticos* que tratam das questões *onde e que tipo* (planejamento em um estágio, de um ano inicial a um ano final, dados). Este trabalho trata especificamente da fase estática do planejamento da transmissão.

Quando a rede de transmissão é modelada pelo fluxo de carga DC, o problema de planejamento estático é um problema de programação não linear inteiro misto (PNLIM) (Eq. 1). Dadas as dimensões que o problema assume para casos práticos em geral observa-se o fenômeno da explosão combinatorial (problema *NP-completo*). Para uma alternativa de investimento conhecida, o problema se reduz a um problema de programação linear (Eq. 2) cujo objetivo é verificar a factibilidade da mesma.

dade da mesma.

Atualmente, este tipo de problema é resolvido pelos chamados métodos combinatoriais (Gallego et al., 1997; Gallego, 1997), cujas características fundamentais são as de serem aplicáveis a sistemas de grande porte, chegar a soluções ótimas globais ou próximas destas e obter soluções em razoáveis tempos de processamento. Dentre esses métodos destaca-se os *Algoritmos Genéticos (GA)*, técnica de busca baseada nos mecanismos da seleção natural e na genética, onde indivíduos melhor dotados têm maior chance de sobrevivência e capacidade de adaptação às alterações do meio ambiente.

Por outro lado, devido aos avanços tecnológicos dos últimos anos, as concessionárias de energia possuem em seu parque computacional um grande número de estações de trabalho que, com o auxílio de uma biblioteca de domínio público para processamento paralelo e distribuído (Geist et al., 1994), podem ser utilizadas como uma máquina paralela virtual.

Neste contexto, o artigo propõe versões paralelas de GA para a resolução do problema do planejamento da expansão da transmissão.



O artigo, inicialmente apresenta a formulação matemática do problema, seguida dos conceitos básicos dos algoritmos genéticos. Na sequência, são apresentados detalhes dos algoritmos propostos, com a apresentação de alguns testes e resultados dos mesmos. Os testes foram realizados em uma rede heterogênea de estações SUN, sistema operacional Solaris 2.5.1, utilizando-se a biblioteca PVM 3.3.5.

2 Formulação Matemática do Problema

O problema geral para o planejamento da transmissão em sistemas de energia elétrica, usando o modelo DC, pode ser escrito na seguinte forma:

$$\begin{aligned} \text{Min } v &= \sum_{(i,j) \in \Omega} c_{ij} n_{ij} + \sum_i \alpha_i r_i \quad (1) \\ \text{s.a. } B(x + \gamma^o)\theta + g + r &= d \\ (x_{ij} + \gamma_{ij}^o)|\theta_i - \theta_j| &\leq (x_{ij} + \gamma_{ij}^o)\bar{\phi}_{ij} \\ 0 &\leq g \leq \bar{g} \\ 0 &\leq r \leq d \\ 0 &\leq n_{ij} \leq \bar{n}_{ij} \\ \forall (i, j) &\in \Omega \end{aligned}$$

Em que:

- c_{ij} Custo do ramo (i, j) .
- n_{ij} Número de linhas adicionadas, $n_{ij} = \frac{x_{ij}}{\gamma_{ij}}$, sendo γ_{ij} a susceptância nominal de uma linha entre as barras i - j .
- α Parâmetro adequado de transformação de unidades.
- r Vetor de geradores fictícios ou artificiais.
- $B(\cdot)$ Matriz de susceptâncias.
- γ_{ij}^o Susceptância inicial no ramo $(i, j) \in \Omega$.
- θ Ângulos das tensões nodais.
- g Vetor de gerações.
- d Vetor de cargas.
- x_{ij} Nova susceptância a ser instalada no ramo $(i, j) \in \Omega$.
- $\bar{\phi}_{ij}$ Definida pela relação: $\bar{\phi}_{ij} = \frac{\bar{I}_{ij}}{\gamma_{ij}^o}$, sendo $\bar{I}_{ij}$ o fluxo máximo de potência permitido que passa pela linha i - j .
- $\bar{g}$ Vetor de limites de geração.
- $\bar{n}_{ij}$ Máximo número de linhas adicionadas.
- Ω Conjunto de todos os ramos definidos pelas linhas existentes e as alternativas de expansão.

O problema (1), chamado de modelo DC, leva em conta as duas leis de Kirchhoff para o sistema elétrico, a capacidade de transmissão das linhas existentes e das candidatas, assim como os limites de operação dos geradores.

O modelo apresentado está modificado em relação ao modelo DC convencional pela adição do termo $\sum_i \alpha_i r_i$ na função objetivo. Com isso,

este problema é sempre factível, pois as sobrecargas no sistema, que eventualmente apareceriam no processo de resolução, são compensadas pelos geradores artificiais alocados em todas as barras de carga.

Para uma determinada proposta de investimento x^k (uma configuração) o problema (1) se torna um problema de programação linear (PPL), como a seguir.

$$\begin{aligned} \text{Min } w &= \sum_i \alpha_i r_i \quad (2) \\ \text{s.a. } B(x + \gamma^o)\theta + g + r &= d \\ (x_{ij}^k + \gamma_{ij}^o)|\theta_i - \theta_j| &\leq (x_{ij}^k + \gamma_{ij}^o)\bar{\phi}_{ij} \\ 0 &\leq g \leq \bar{g} \\ 0 &\leq r \leq d \end{aligned}$$

O sistema (2) é um PPL com os x_{ij}^k conhecidos ou determinados para uma alternativa de investimento. Portanto, para um x^k escolhido, a solução obtida por um algoritmo de PL aplicado ao sistema 2 indicará simplesmente se o sistema apresenta corte de carga (infactível) ou não (factível) em relação a formulação original. Neste trabalho utilizou-se o MINOS 5.1 (Murtagh e Saunders, 1987) na resolução do PPL em questão.

3 Algoritmos Genéticos no Planejamento da Transmissão

A aplicação dos algoritmos genéticos ao problema do planejamento da expansão da transmissão, apresenta de maneira geral as seguintes etapas:

Etapa 1 - parte de uma configuração inicial factível (obtida aleatoriamente ou através de um algoritmo inicializador - como o proposto em (de Oliveira et al., 1998; de Oliveira et al., 1999)),

Etapa 2 - faz uma seleção das melhores configurações que irão gerar novos descendentes (através do uso de uma função "fitness" avalia-se a soma dos investimentos mais o custo do corte de carga referente a cada configuração, selecionando as de menor valor),

Etapa 3 - efetua a recombinação entre as configurações selecionadas (através do operador "crossover", que troca parte do código genético entre duas configurações gerando duas novas configurações),

Etapa 4 - faz um processo de mutação (que tem a finalidade de adicionar uma diversidade genética ao processo) e,

Etapa 5 - determina os elementos da nova população (cujo tamanho é mantido fixo pela ordenação das configurações através da função "fitness" e exclusão das piores configurações).

O processo volta à **Etapa 2** até ser atendido um critério de parada.



4 Os Algoritmos Propostos

A teoria dos algoritmos genéticos mostra que existe um paralelismo intrínseco dentro do funcionamento dos GA padrão. Entretanto, pode-se implementar versões paralelas do mesmo de acordo com duas propostas de caráter geral:

Avaliação paralela das funções objetivo da população - Neste caso, os processadores são usados para avaliar a função objetivo de cada configuração da população. Por exemplo, supor que o problema de planejamento da expansão de sistemas de transmissão está sendo resolvido com uma população de n configurações. Um algoritmo genético sequencial deve resolver n problemas de programação linear (PPL) independentes para determinar a função objetivo de cada um dos elementos (configurações) da população. Deve-se lembrar que a resolução de um simples PPL requer maior esforço computacional que o restante do processo sequencial do algoritmo genético, isto é, da implementação da seleção-recombinação-mutação. Portanto, o ganho é muito significativo simplesmente paralelizando a resolução dos PPL's. Neste caso, um algoritmo paralelo simples (versão GAP1A) implementa a paralelização da resolução dos PPL's, isto é, resolve os n PPL's usando os n processadores (um PPL por processador) e o resto do processo é equivalente ao algoritmo sequencial. Um exemplo prático seria utilizar uma máquina paralela com 64 processadores (nCUBE, por exemplo), para uma população de 64 configurações. A bibliografia geral de GA (por exemplo, (Goldberg, 1989)) sugere este tipo de algoritmo mas menciona que pode acontecer problemas de comunicação entre os processadores. Entretanto, este tipo de problema é mais típico naqueles problemas em que avaliar a função objetivo de cada configuração requer um esforço relativamente pequeno comparado com o esforço integral dos GA; fato que não ocorre com o problema de planejamento da expansão do sistema de transmissão. Uma outra alternativa para a resolução do problema (versão GAP1B), agora considerando uma rede de estações de trabalho, seria dividir o cálculo dos n PPL's entre p processadores (estações), cabendo assim a cada processador: n/p PPL's.

A Fig. 1 representa o esquema de funcionamento da versão GAP1B.

Alocação de subpopulações de configurações em cada processador - Neste caso a população total é dividida em subpopulações e cada uma delas trabalha de maneira independente em cada processador. Este caso é praticamente equivalente ao algoritmo da divisão e ao algoritmo híbrido implementado para "simulated annealing" paralelo (SA) (Gallego, 1997). A vantagem adicional do GA em relação aos algoritmos



Figura 1. Esquema de funcionamento do algoritmo GAP1B

mos de SA paralelo é que GA paralelos podem ser implementados de maneira assíncrona. Em um SA paralelo, em cada nível de temperatura, cada processador (*escravo*) realiza a busca de um determinado número de configurações candidatas e depois informa para o programa *mestre* a melhor configuração encontrada. Para que cada processador retome o trabalho de busca, o mestre deve retornar a cada processador o novo valor da temperatura e a melhor configuração encontrada. Nos GA paralelos, cada processador realiza uma busca praticamente sem espera de nova informação pois simplesmente informa ao mestre a melhor configuração encontrada e recebe do mesmo a informação da melhor configuração do processo (*incumbente*). Para o exemplo mencionado anteriormente, a população de n configurações pode ser dividida em p subpopulações de n/p configurações cada uma e seria usado p processadores. Cada processador implementa um ciclo geracional (seleção - recombinação - mutação - avaliação da função objetivo) e envia para o mestre a melhor configuração encontrada. Simultaneamente recebe do mestre a melhor configuração (ou as melhores configurações do processo integral) incorporando-a(s) na sua subpopulação para melhorar o processo de seleção. Logicamente, um processador não precisaria esperar que os outros processadores terminem seu correspondente ciclo geracional. Também a troca de informações não precisa ser a cada ciclo geracional, podendo ser realizada após certo número de gerações. Para avaliar todas estas opções, pode-se implementar as seguintes versões:

- Uma primeira versão (GAP2A) onde,



após um número razoavelmente grande de gerações (k), verifica-se a incumbente relativa a cada processador e escolhe-se a melhor dentre elas.

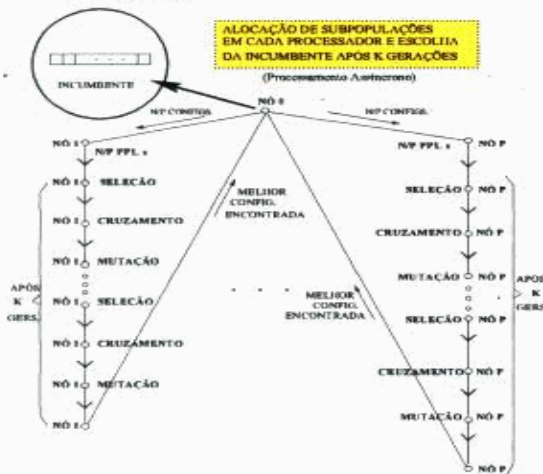


Figura 2. Esquema de funcionamento do algoritmo GAP2A

- Uma segunda versão (GAP2B) aonde, após um número pequeno de gerações ($k1$), o mestre recebe as melhores configurações encontradas, mantém o valor da melhor entre elas como uma referência e continua o processamento até um critério de parada ser satisfeito.

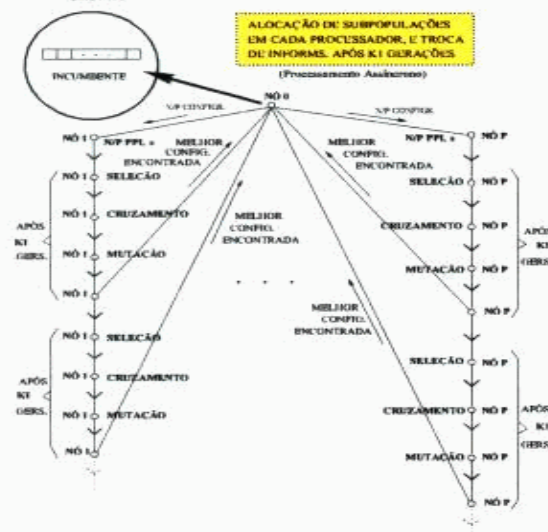


Figura 3. Esquema de funcionamento do algoritmo GAP2B

- Uma outra versão (GAP3A) onde, após um número pequeno de gerações $k1$, o mestre

armazena m configurações elites, diversificadas. Repassa-as novamente a cada processador, que as incorporam à sua subpopulação. Repete-se, até que seja satisfeito um critério de parada.

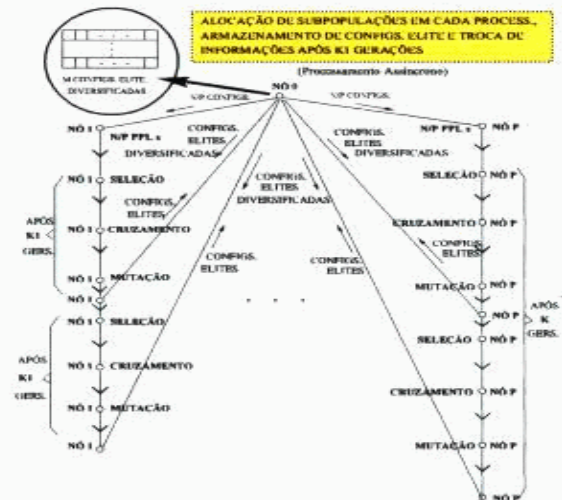


Figura 4. Esquema de funcionamento do algoritmo GAP3A

- E, finalmente, uma versão (GAP3B) onde após um número pequeno de gerações ($k1$), o mestre recebe as configurações elites de cada processador, armazena as elites das elites e as repassa novamente a cada processador, que por sua vez as incorporam em sua subpopulação até um critério de parada ser satisfeito.

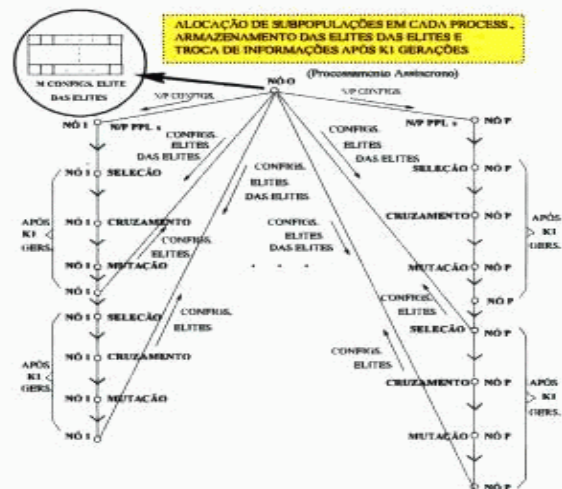


Figura 5. Esquema de funcionamento do algoritmo GAP3B



5 Testes e Resultados

A Tabela 1 a seguir apresenta os resultados obtidos com o algoritmo genético serial para os sistemas Garver (6 barras - 15 linhas) e Sul Brasileiro (46 barras - 79 linhas).

Tabela 1. Resultados do algoritmo genético serial

itens	sistemas	Garver	Sul
n.de gerações obt. ótimo		4	34
número de PPL's		409	6410
tempo p/ solução ótima		23,6s	22,2min
tempo médio por PPL		0,0577 s	0,2077 s

A Tabela 2 apresenta alguns dos resultados obtidos com os algoritmos GAP1B, GAP2A e GAP2B, para o sistema Garver (6 barras - 15 linhas). E a Fig. 6 dá uma idéia da evolução da incumbente para o primeiro algoritmo.

Tabela 2. GA paralelos - Sistema Garver

itens	algoritmos	GAP1B	GAP2A	GAP2B
n.de gerações obt. ótimo		11	7	7
número de PPL's		1015	876	679
n.de estações		3	3	3
troca de dados a cada		-	-	5 gers.
tempo p/ solução ótima		20,5 s	16,6 s	15,3 s
tempo médio por PPL		0,0202s	0,0189 s	0,0226s

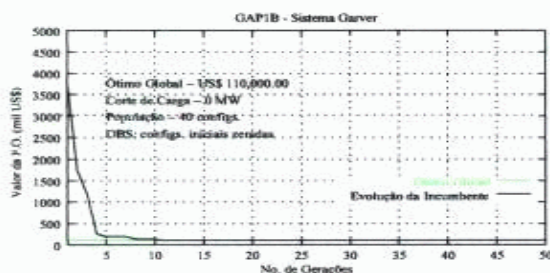


Figura 6. Evolução da incumbente com o algoritmo GAP1B

A Tabela 3 apresenta alguns dos resultados obtidos com os supra citados algoritmos para o sistema Sul Brasileiro (46 barras - 79 linhas), e as Figs. 7 e 8 apresentam, respectivamente, o comportamento dos algoritmos GAP2A e GAP2B para este sistema. Em todos os casos foram estabelecidos previamente um número máximo de gerações.

Tabela 3. GA paralelos - Sistema Sul

itens	algoritmos	GAP1B	GAP2A	GAP2B
n.de gerações obt. ótimo		41	14	34
número de PPL's		16668	2761	8070
n.de estações		3	3	3
troca de dados a cada		-	-	5
tempo p/ solução ótima		35,8min	12,7min	20,1min
tempo médio por PPL		0,1289 s	0,2750 s	0,1496 s

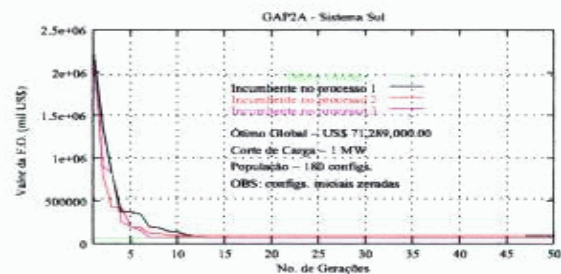


Figura 7. Evolução da incumbente com o algoritmo GAP2A

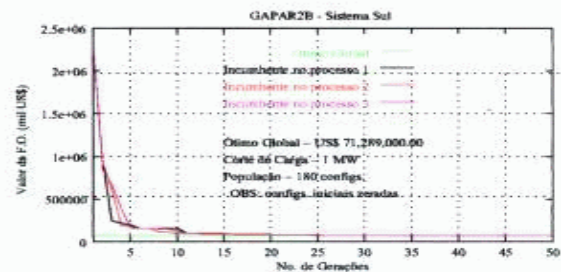


Figura 8. Evolução da incumbente com o algoritmo GAP2B

Comparando os resultados apresentados para o sistema Garver na Tabela 1 com os da Tabela 2, bem como os resultados para o sistema Sul Brasileiro da Tabela 1 com os da Tabela 3 verifica-se que em todos os casos os algoritmos paralelos propostos encontram eficientemente o ótimo global (Apêndice A) em um tempo inferior aos da versão sequencial, exceção feita para o GAP1B para o sistema Sul Brasileiro. Para os testes realizados o algoritmo GAP2A mostrou-se o mais eficiente.

6 Conclusões

Com a tendência cada vez maior do uso de redes de computadores nas concessionárias de energia elétrica, aliado a disponibilidade de bibliotecas para processamento paralelo, tem-se o ambiente propício para a utilização de modelos de programação distribuída nos problemas complexos e



de natureza combinatória como o problema de planejamento da expansão da transmissão de energia elétrica.

Particularmente, neste trabalho, foram desenvolvidos alguns algoritmos genéticos paralelos para esse problema, que além de mostrarem sua eficiência na obtenção do ótimo global para os sistemas testados, o fizeram em um tempo menor quando comparado com um algoritmo genético seqüencial.

Agradecimentos

Este trabalho contou com o apoio parcial da **FA-PESP**, e do **CNPq**, como também suporte financeiro da **UNESP** e da **CAPES**.

Referências Bibliográficas

- de Oliveira, S. A., de Almeida, C. R. T. e Monticelli, A. (1998). Times assíncronos aplicados a métodos heurísticos construtivos de planejamento da expansão da transmissão, *XII Congresso Brasileiro de Automática*, Vol. 3, SBA-UFU, Uberlândia-MG, pp. 1029-1034.
- de Oliveira, S. A., de Almeida, C. R. T. e Monticelli, A. (1999). Time assíncrono inicializador de métodos combinatoriais para planejamento da expansão da transmissão, *XV SNPTEE Seminário Nacional de Produção e Transmissão de Energia Elétrica*, Grupo VII, Foz do Iguaçu-PR. (IT04).
- Gallego, R. A. (1997). *Planejamento a longo prazo de sistemas de transmissão usando técnicas de otimização combinatorial*, PhD thesis, Faculdade de Engenharia Elétrica da UNICAMP, Campinas-SP. (Tese, Dr. Eng. Elétrica).
- Gallego, R. A., Monticelli, A. e Romero, R. A. (1997). Comparative studies on no-convex optimization methods for transmission network expansion planning, *Power Industry Computer Application Conference*, Vol. 1, IEEE, Columbus, pp. 1-7.
- Goldberg, D. E. (1989). *Genetics algorithms in search, optimization and machine learning*, Addison Wesley, Reading, Mass.
- Geist, A. e et al. (1994). *PVM3 users's guide and reference manual*, ORNL, Oak Ridge.
- Murtagh, B. e Saunders, M. (1987). Minos 5.1 user's guide, *Technical Report SOL 83-20R*, Stanford University, Stanford. 125p.

Apêndice

Configurações Ótimas

Sistema Garver (6 Barras / 15 Linhas)

Linhas adicionadas: $n_{03-05} = 1$, $n_{04-06} = 3$.

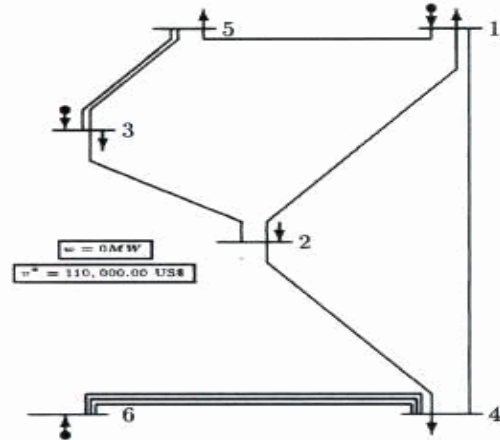


Figura 8 - Sistema Garver - Config. Ótima

Sistema Sul Brasileiro (46 Barras / 79 Linhas)

Linhas adicionadas: $n_{13-20} = 1$, $n_{20-23} = 1$, $n_{20-21} = 2$, $n_{42-43} = 1$, $n_{46-06} = 1$, $n_{05-06} = 2$.

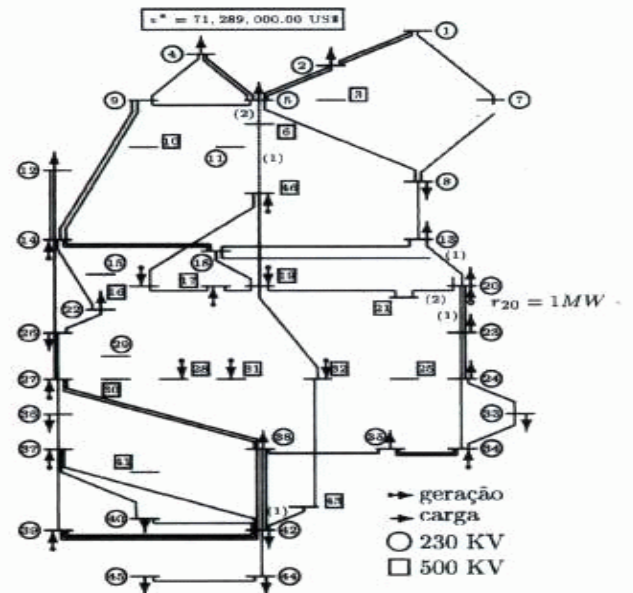


Figura 9 - Sistema Sul Brasileiro - Config. Ótima.

CARS & FOF 2001 – Durban SurAfrica

HYBRID MODEL ANALYSIS AND CONSTRUCTIVE ALGORITHMS IN THE PERFORMANCE OF A GENETIC ALGORITHM FOR THE PLANNING OF AN ELECTRIC POWER TRANSMISSION SYSTEM

ANTONIO ESCOBAR Z.

Ing. Electricista

Profesor Asociado

aescobar@utp.edu.co

RAMON A. GALLEGU R.

Ph.D. Ing. Eléctrica

Profesor Titular

ralfonso@utp.edu.co

UNIVERSIDAD TECNOLÓGICA DE PEREIRA-Pereira Colombia

RUBEN A. ROMERO L.

Ph.D. Ing. Eléctrica

ruben@dee.feis.unesp.br

SÉRGIO A. DE OLIVEIRA

M.Sc. Ing. Eléctrica

grilo@dee.feis.unesp.br

Professors at DEE-FEIS-UNESP-Ilha Solteira-Brasil

ABSTRACT

It has become more evident that in complex problems the performance of a genetic algorithm depends on the initial population, specially on its quality and diversity. In this paper four strategies are presented which, used concurrently allow to find quality and diversified topologies to set up an initial population to be used in a genetic algorithm to solve the problem of planning and expansion of electrical energy transmission systems. Also, experimental tests on real systems are presented to show the proposal's performance. This strategy to generate quality and diversified initial solutions can also be used in other algorithms such as simulated annealing, tabu search, GRASP, etc.

KEYWORDS: Planning, transmission system, hybrid model, genetic algorithm.

1. INTRODUCTION

Optimization algorithms such as simulated annealing, genetic algorithms, tabu search, GRASP, etc. usually require a solution or a set of solutions to start the search process.

It can be shown in a theoretical analysis of these algorithms that the final solution does not generally depend on the initial solution used to start the process. This hypotheses is particularly useful in the theoretical form of the genetic algorithm. However, when an initial population is generated at random, the processing effort is prohibitive especially in real problems with a high degree of complexity.

Recent research on the application of genetic algorithms in complex problems show that an initial population generated by using efficient strategies considerably reduces the computational effort and improves the quality of the solutions found. It has also been verified that the best strategy to generate good initial solutions is to employ heuristic algorithms, usually of constructive type. One of the many reasons for this choice is that there are many heuristic algorithms available in the specialized literature to tackle complex problems. In this case, the proposal consists in finding a good quality and diversified population using

constructive heuristic algorithms of relaxed models to solve the planning problem in electrical transmission systems. The initial population found is used to start optimization using a genetic algorithm.

Relaxed models and heuristic techniques were widely used in the planning problem since the 60s and are still used. In this case two relaxed models called transport and hybrid are analyzed. With the hybrid model it is still possible to use three kinds of constructive heuristic algorithms. In this paper, a genetic algorithm with a new proposal to generate the initial population is used to carry out the planning of the Colombian system.

2. MATHEMATICAL FORMULATION

In the following, the static transmission expansion problem is formulated as a mixed integer nonlinear programming problem in which the power network is represented by a DC power flow model:

$$\begin{aligned}
 \text{Min} \quad & v = \sum c_{ij} n_{ij} \\
 \text{s.a.} \quad & Sf + g = d \\
 & f_{ij} - \gamma_{ij} (\theta_i - \theta_j)(n_{ij} + n_{ij}^o) = 0 \\
 & |f_{ij}| \leq f_{ij \max} (n_{ij} + n_{ij}^o) \\
 & 0 \leq g \leq g_{\max} \\
 & 0 \leq n_{ij} \leq n_{ij \max}
 \end{aligned} \tag{1}$$

$$n_{ij} \text{ integer, } f_{ij} \text{ and } \theta_j \text{ unbounded}$$

where c_{ij} is the cost of the addition of a circuit in branch i - j , n_{ij} is the number of the circuits added in branch i - j , n_{ij}^o is the number of circuits in the initial system, S is the system's bus-branch incidence matrix, γ_{ij} is the susceptance of the new circuits, f_{ij} is the flow in the circuit i - j , $f_{ij \max}$ is the maximum flow in the circuit i - j , d is the demand vector, g is the generation vector, $g_{\max}$ is the maximum generation capacity vector.

A complication arises from the fact that the simple addition of a new circuit may not be enough to guarantee network connectivity. The resultant network can be disconnected.

The DC model must obey Kirchhoff's two laws and the operation and investment constraints. Thus, the first set of restrictions in (1) represents Kirchhoff's first law and the second set, Kirchhoff's second law. The rest are operational restrictions. Problem (1) is a non-linear programming integer mixed problem (PNLIM). It is hard to solve due to its non-convexity and to the fact that it shows the combinatorial explosion phenomenon. One of the most appropriate ways to solve (1) is by means of a genetic algorithm.

3. RELAXED MODELS AND HEURISTIC ALGORITHMS

CARS & FOF 2001 – Durban SurAfrica

The DC model can be substituted by relaxed models where a set of restrictions are not taken into account, making it possible to solve the problem more easily. The most widely used relaxed mathematical models are the transport models and the hybrid models. The latter can have several forms and it can be linear or non-linear.

3.1 TRANSPORT MODELS

When the restrictions corresponding to Kirchhoff's second law in the DC model are eliminated, a relaxed model called "Transport model" is obtained. It is governed by the following equations:

$$\begin{aligned}
 \text{Min} \quad & v = \sum c_{ij} n_{ij} \\
 \text{s.a.} \quad & Sf + g = d \\
 & |f_{ij}| \leq f_{ij \max} (n_{ij} + n_{ij}^{\circ}) \\
 & 0 \leq g \leq g_{\max} \\
 & 0 \leq n_{ij} \leq n_{ij \max}
 \end{aligned} \tag{2}$$

$$n_{ij} \text{ integer; } f_{ij} \text{ unbounded}$$

The transport model corresponds to a linear integer mixed problem (PLIM). It is not easy to find the optimal solution to the transport model. However, there is a constructive heuristic algorithm that finds a good solution to the transport model with a small computing effort. A good solution found with the heuristic algorithm is an excellent topology to build an initial population because it generally finds the most attractive circuits to be incorporated into the system.

3.2 LINEAR HYBRID MODEL

This is also a relaxation of the DC model. In the linear hybrid model all existing circuits in the base topology must fulfill both of Kirchhoff's laws and the added circuits must only comply Kirchhoff's first law. In other words, Kirchhoff's first law must be obeyed in all the system's buses but the second law must only be obeyed in the meshes of the already existing circuits in the base configuration. A mathematical formulation for the linear hybrid model is:

$$\begin{aligned}
 \text{min} \quad & v = \sum c_{ij} n_{ij} \\
 \text{s.a.} \quad & Sf + S^{\circ} f^{\circ} + g = d \\
 & f_{ij}^{\circ} - \gamma_{ij} n_{ij}^{\circ} (\theta_i - \theta_j) = 0 \\
 & |f_{ij}^{\circ}| \leq f_{ij \max} n_{ij}^{\circ} \\
 & |f_{ij}| \leq f_{ij \max} n_{ij} \\
 & 0 \leq g \leq g_{\max} \\
 & 0 \leq n_{ij} \leq n_{ij \max}
 \end{aligned} \tag{3}$$

$$n_{ij} \text{ integer; } f^{\circ}_{ij}, f_{ij} \text{ y } \theta_j \text{ unbounded}$$

Where S° is the system's bus-branch incidence matrix conformed by the existing buses and branches in the initial topology, f° is the power flow vector of the existing branches in the initial topology, f is the flow vector of the newly added circuits whose elements are the flows f_{ij} .

3.3 HEURISTIC ALGORITHMS

Heuristic constructive algorithms exist for each of the presented models. Those algorithms add a circuit in every step and the circuit is selected by using a sensitivity indicator. It is important to note that if in the transport and linear hybrid models the integrality conditions are relaxed, that is, the values n_{ij} are allowed to take non-integer values, then both models reduce to solving a simple linear programming problem (PL). This kind of solution has no practical value. However, it can be used as a strategy to identify the most attractive circuits, that is, it can be used as a sensitivity indicator. Thus, for the transport and hybrid models the following heuristic constructive method can be used:

1. Consider the initial topology as the present topology.
2. Solve the mathematical model for the present topology using a PL algorithm, relaxing the integrality of values n_{ij} . If all values $n_{ij} = 0$ and $v=0$ then stop the process since the system operates adequately. Else, go to step 3.
3. Identify the most attractive path as the added circuit that takes the greatest power flow ($n_{ij}>0$). Add this circuit to the simulated system and update the present topology.

This algorithm finds a feasible topology for the analyzed model. In large sized systems heuristic algorithms do not find the optimal solution. Also, the found topologies, although optimal, aren't usually feasible for the DC model.

However, the algorithm finds good solutions and, even more important, it identifies the most attractive circuits. Therefore, the fundamental strategy consists of finding a set of topologies using relaxed models and then employing those topologies as an initial population to find optimal and sub-optimal solutions for the DC model combined with a specialized genetic algorithm.

3.4 FOUR HEURISTIC ALGORITHMS

Using the already discussed proposal of a constructive heuristic algorithm it is possible to implement an algorithm for the transport model and three algorithms for the hybrid model. With small modifications in the basic algorithm it is possible to find a set of good quality topologies. The figure 1 shows the solution trajectories for the hybrid and transport models.

3.4.1 GARVER'S ALGORITHM

CARS & FOF 2001 – Durban SurAfrica

The heuristic algorithm developed in 3.3 for the transport model is Garver's proposal [6] and it was widely used in the past for planning. Garver's algorithm finds a feasible topology for the transport model. However, many topologies can be found using Garver's algorithm. For example, suppose that Garver's algorithm finds a topology where circuits were added in 40 paths and 25 from those paths are considered important. In this context, 25 further topologies can be found repeating Garver's algorithm and discarding in each case one of the 25 paths considered important. This implementation can be made easily by changing the cost of each circuit in every case.

The foregoing process can be repeated discarding several circuits in each case. It is important to note that many good quality topologies can be found and they can be significantly different.

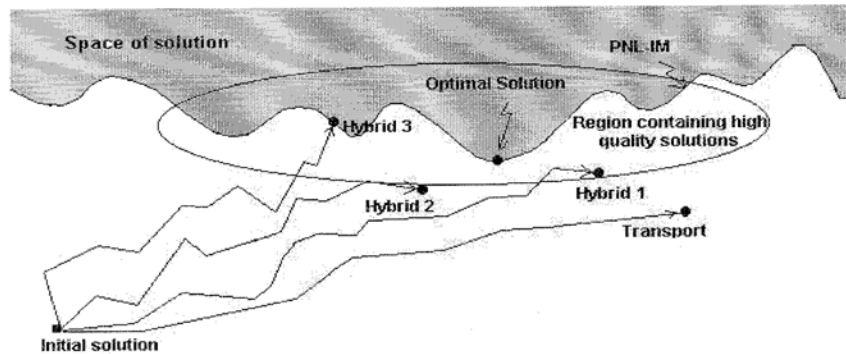


Figure 1. Solution trajectories of the hybrid and transport models

3.4.2 HYBRID ALGORITHM I

It is possible to employ the heuristic algorithm of 3.3 for the hybrid model where the final topology is feasible for the hybrid model. In this case, in step 2 the hybrid model (3) is solved applying the following changes:

1. Relation $Sf + S^o f^o + g = d$ (Kirchhoff's first law is substituted by relation $Sf + S^o f^o + S' f' + g = d$, where S' is the bus-branch incidence matrix formed by the circuits added during the heuristic process and the terminal buses in those circuits. f' is the flow vector with elements f'_{ij} which represent the circuit flow through path $i - j$ that was added during the iterative process.

2. The set of relations $|f'_{ij}| \leq f'_{ijmax} n'_{ij}$ must be added, where n'_{ij} represents the number of circuits added during the iterative process.

It is important to note that every time that the modified problem (3) is solved, there is a flow vector for circuits in the base topology and another flow vector for circuits added in the iterative process and yet another flow vector for circuits with values $n_{ij} \neq 0$ in the PL solution.

It is also important to note that the modified problem (3) changes little between 2 successive iterations since vector n' scarcely varies with the addition of a circuit in element n'_{ij} on the most attractive path i-j. Matrix S' also changes.

3.4.3 HYBRID ALGORITHM II

It is possible to use the heuristic algorithm shown in 3.3 for the hybrid model where the final topology has characteristics that are slightly different from those found in the last proposal. In this proposal, every circuit added during the iterative process must show the following behavior: (1) if it belongs to a path where a circuit already exists in the base configuration, then the newly added circuit must obey Kirchhoff's second law together with the parallel circuits that may have existed on that path in the base topology and (2) if it belongs to a new path then that circuit must only obey Kirchhoff's first law. Thus, the topology found under this proposal is different from the one found in the last proposal. Suppose for example that a circuit exists between buses 25-32 in the base topology and during the iterative process a new circuit on that path was added. In the hybrid algorithm I it is possible that the flow through each of the two circuits be different (one of them fulfills only Kirchhoff's first law, the other fulfills both laws). In the hybrid algorithm II both circuits must have equal flows since the two must fulfill both Kirchhoff's laws.

In this case, in step 2 the hybrid model (3) is solved with the following changes (is used the model of hybrid I) :

On each iteration it is identified if the circuit to be added belongs to a path on which at least a circuit of the base topology exists. That being the case, element n°_{ij} is updated adding a circuit. Else, element n'_{ij} is updated adding a circuit. An update of matrix S' may also be necessary. It is important to note that the PL structure between two successive iterations changes little; only an element from n°_{ij} or an element from n'_{ij} (and also S').

3.4.4 HYBRID ALGORITHM III

It is possible to use the heuristic algorithm of 3.3 for the hybrid model where the final topology is feasible for the DC model. This proposal is known as the Villasana-Garver algorithm. In this case, step 2 of the hybrid model (3) is solved as follows:

1. During each step, relaxed problem (3) is solved using a PL algorithm.
2. On each iteration, the most attractive path i-j must be identified and element n°_{ij} must be updated adding a circuit. It may also be necessary to update matrix S° .

It is important to note that on each step of the iterative process, the circuits existing in the base topology and also those added in the iterative process obey Kirchhoff's second law. Therefore, when the iterative process ends, the final topology found is feasible for the DC model.

3.5 GENERATION OF THE INITIAL POPULATION

CARS & FOF 2001 – Durban SurAfrica

The initial population, conformed by excellent-quality and diversified topologies, can be found using the four constructive heuristic algorithms shown above. It is also observed that constructive heuristic algorithms tend to add many irrelevant circuits in the final phase of the iterative process. Therefore, some topologies can be generated by stopping the process when the investment v found by the PL algorithm is small enough without reaching $v=0$. In this work, topologies to find an initial population were generated according to the following steps:

- For each heuristic algorithm an additional set of new topologies can be found by repeating the process and forbidding in each case an important circuit identified in the initial process with no prohibitions.
- The forementioned strategy can be repeated, but the process stops when the pending investment v is $v \leq v_0$, where v_0 is a value previously fixed.

With this proposal it is possible to find a high number of topologies, usually much higher than the necessary to set up the initial population of a genetic algorithm. However, new good-quality topologies can be found by employing the recombination operator on the found topologies.

4. GENETIC ALGORITHMS

The methodology described for a genetic algorithm is taken from reference [3]. Several methodologies of initializing algorithms are tested using the same base algorithm.

A simplified representation of a Genetic Algorithm Problem requires the following steps:

- Definition of the objective function
- Coding
- Definition of the selection scheme
- Definition of the crossover and mutation mechanisms
- Definition of population size and stopping criterion
- Implementation of the special features associated to the transmission planning problem

In order to develop a genetic algorithm capable of solving large-scale practical cases it is necessary to implement the following functions: Limited selection, elitism, building blocks and unconnected networks, see reference [3].

Control parameters: The parameters used in the analysis of the colombian electrical system are: population size between 140-200, crossover rate between 0.7-0.9, mutation rate between 0.005-0.05. The selection type used is proportional selection with stochastic remainder.

Stop criterion: The process stops whenever the incumbent solution (least cost configuration) does not improve after a specified number of generations (from 50 to 100 generations). The maximum number of generations used is between 500 and 1000.

5. TEST AND RESULTS

Using the Minos program (running in a SUN ultra5 station) a genetic algorithm was implemented, with an initial population generated with the three versions of the hybrid model and the transport model. As test systems the Garver system (6 buses/15 branches) [6] and the Colombian electrical system projected into year 2012 (171 buses/ 223 existing transmission lines and 196 candidates) were used. All the relevant data about this system are found in reference [1].

In the Garver system [6] hybrid initializers complemented with the constructive algorithm described in 3.3 find the optimal solution without resorting to a genetic algorithm. It was found however that each hybrid version reaches the solution through different solution trajectories as shown in figure 2.

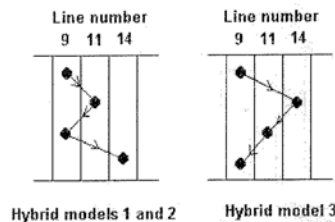


Figure 2. Garver solutions using hybrid models

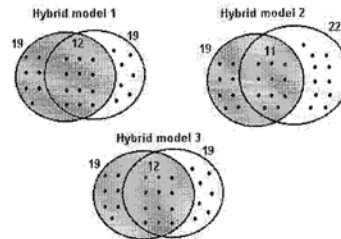


Figure 3. Comparison among the best and the hybrid solutions

The constructive heuristic techniques used in this paper allow the obtention of an optimal solution for small systems.

For the 2012 colombian system, the obtained solutions were confronted using the constructive heuristics described in 3.3 based on the three hybrid versions with the best solution found to date for this system [5]. Figure 3 shows the circuits common and not common to the solutions found with hybrid versions and the best solution in reference [5].

Based on hybrid and transport models an initial population was set up and then used with a genetic algorithm. The following solution was obtained:

OCAN-CESA : 2,	CART-SAB2: 1,	CGVC-TLUA: 1,	CGVC-VIRG: 1,
CMA5-CHI5: 1,	CMA5-SJOG: 1,	SAB5-CHI5: 1,	CMA5-SAC5: 1,
GTPE-MIRF: 1,	JUTO-YUMB: 1,	YUMB-MARC: 1,	SAC5-VIR5: 1,
VIRG-VIR5: 1,	SAB2-SAB5: 2,	MALE-MAL5: 2,	MAL5-FAC5:
1, NDE2-SOG2: 1.			

With a total investment of US M562.3 and a loss of load of 0 MW, the same quality as that reported in [1,5] was reached. Also, other good solutions were found, one of this with a total investment of US M504.37 and a loss of load of 50 Mw.

6. CONCLUSIONS

CARS & FOF 2001 – Durban SurAfrica

In this paper, four constructive algorithms are analyzed: the transport model and 3 versions of the hybrid model. This paper's objective was to conform initial populations with the forementioned algorithms and to use those populations as initializers for combinatorial algorithms such as genetic, memetic, evolutive, tabu search, simulated annealing and GRASP algorithms. The population obtained was used as initializer for a genetic algorithm. The solution obtained was substantially better as the one achieved with the employment of other initializers using the same genetic algorithm and was as good as that obtained using a tabu search.

It can be shown mathematically that it is possible to find very high quality solutions in large electrical systems, but the computational effort is prohibitive in terms of time. In practice, with a random or some other kind of initializer, usually very poor quality solutions are obtained and still the computational effort is high. When the methods described in this paper to find an initial population are used, it is guaranteed that the process starts in a region that contains very good quality solutions, among which the optimal solution may be included. The computational effort is moderate with respect to that made under some other initializer.

7. ACKNOWLEDGEMENTS

The authors wish to thank to the Universidad Tecnológica de Pereira for the support to the electric planning work group.

8. BIBLIOGRAPHY

- [1] Da Silva E.L., Gil H.A., Areiza J.M. (2000). Transmission Network Expansion Planning Under an Improved Genetic Algorithm, *IEEE Trans. On Power Syst.*, 15, 1168-1175.
- [2] De Oliveira, S.A., de Almeida, C.R.T. Monticelli, A. (1998). Times assíncronos aplicados a métodos heurísticos constructivos de planejamento da expansão da transmissão, *XII Congresso Brasileiro de Automática*, 3, 1029-1034.
- [3] Gallego, R.A., Monticelli, A., Romero, R. Transmission System Expansion Planning by extended Genetic Algorithm, *IEE Proc. Part c*, 145-3, 329-335.
- [4] Gallego, R.A, Monticelli, A., Romero, R. (2000). Tabu Search Algorithm for Network Synthesis, *IEEE Transactions on Power Systems*, 15, 490-495.
- [5] Gallego, R.A., Romero, R.A., Escobar, A. (2000). Statical Planning of Colombia's Transmission System Using Genetics Algorithm, *16th International Conference on CAD/CAM Robotic & Factories of the Future*, Trinidad y Tobago.
- [6] Garver, L.L. (1970). Transmission Network Estimation Using Linear Programming. *IEEE. Transactions on Power Apparatus and Systems*, 89, 1688-1697.
- [7] Michalewicz Z. (1992), Genetic Algorithms + Data Structures= Evolution Programs, Artificial Intelligence, *Springer*, Berlin.
- [8] Monticelli, A., Santos, A. Jr., Pereira, M.V.F., Cunha, S., Praça J. G., Park, B. (1982). Interactive Transmission Network Planning Using a Least-Effort Criterion, *IEEE Transactions*, 101-10, 3919-3925.

CARS & FOF 2001 – Durban SurAfrica

- [9] Romero, R., Mantovani, M., Gallego, R.A., Monticelli, A. (1999). Experimental Analysis of Selection Methods in a Genetic Algorithm Applied to the Planning Electrical Transmission. *15th International Conference on CAD/CAM Robotic & Factories of the Future*, Aguas de Lindoia, Brasil.

CARS & FOF 2001 - Durban SurAfrica

PARALLEL COMBINATORIAL ALGORITHM FOR STATICAL PLANNING OF A TRANSMISSION SYSTEM

RAMON A. GALLEGU R.

Ph.D. Ing. Eléctrica

Profesor Titular

ralfonso@utp.edu.co

ANTONIO ESCOBAR Z.

Ing. Electricista

Profesor Asociado

aescobar@utp.edu.co

RICARDO RESTREPO ANGEL

Ing. Electrónico

Profesor Asistente

rangel@utp.edu.co

Professors at UNIVERSIDAD TECNOLOGICA DE PEREIRA-Pereira Colombia

SÉRGIO A. DE OLIVEIRA

M.Sc. Ing. Eléctrica

grilo@dee.feis.unesp.br

Professor at DEE-FEIS-UNESP-Ilha Solteira-Brasil

ABSTRACT

There are several combinatorial optimization methods for the solution of complex and very large problems, such as the network static expansion transmission planning problem. Some of these methods are: genetic, memetic and evolutive algorithms, Tabu Search, simulated annealing, GRASP and hybrid methods. These methods are intrinsically parallelizable.

In this paper, a genetic algorithm will be used as the chosen combinatorial algorithm. This work presents parallel versions for the solution of the above-mentioned problem of planning. The mathematical formulation of the algorithms is presented, as well as results for the Garver's system (6 buses/15 branches) and the Pereira (Colombia) system (24 buses/35 branches).

KEYWORDS: Parallel, genetic algorithm, planning, electrical system, optimization.

1. INTRODUCTION

In this paper several parallelization schemes for genetic algorithms are presented. The algorithms will be applied to the problem of planning the expansion of transmission networks. This work considers only the problem of static planning in which, given an initial configuration and the projected generation/demand data for a time horizon, the minimum investment in lines/ transformers required to satisfy demand on every point. The problem of static planning is a part of the so-called dynamic planning in which the amount of elements (lines/transformers) through time is determined. In this way, statical planning defines how much and where, and dynamic planning defines when.

The transmission network is modeled through a non linear integer mixed model, described by equation (1). This model corresponds to a complex combinatorial problem that is difficult to solve. Given the dimensions in practical cases, usually a combinatorial explosion phenomenon is observed, see [9,10].

This problem has been solved through different methodologies like constructive algorithms [7,9,10], methods based on exact optimization [11] and combinatorial optimization [3,6]. The latter methods have been until now the most effective on every kind of system, from small to large sized. Among the combinatorial optimization methods applied to the solution of the planning problem one can find: simulated annealing, genetic and evolutive algorithms, tabu search, GRASP and hybrid algorithms. From these, genetic and evolutive algorithms have proved to be very effective [3,12]. This fact has fueled a deeper investigation to apply them in the problem of transmission planning.

Among the most important features of genetic algorithms is that they lend themselves to a parallel formulation. This paper will explore the parallelization of the algorithm. The objective of parallelization is to reduce computational time, which in fact is already high for the serial algorithm, making it the greatest weakness of combinatorial methods. On the other side, we seek to improve the quality of the answer obtained with a serial version.

Several parallelization schemes have been proposed to establish a comparison. Also, a comparative analysis of parallel versions against the serial version will be made.

We pretend to take advantage of the progress made in computer technology. The large electrical energy supplying companies own in their inventories many workstations (PCs, SPARCs, Macs and others). With the aid of a public domain library for parallel and distributed processing these machines can be configured as a single virtual parallel machine.

This paper delves initially into the basic concepts of the parallel virtual machine (PVM). Then, a mathematical formulation of the problem is developed. The basic genetic algorithm for transmission planning is discussed next, together with the parallelization algorithms. In the last step, results and conclusions are presented.

2. MATHEMATICAL FORMULATION

In the following, the static transmission expansion problem is formulated as a mixed integer nonlinear programming problem in which the power network is modelled as a DC power flow:

$$\begin{aligned}
 \text{Min} \quad & v = \sum c_{ij} n_{ij} + \sum \alpha r_i \\
 \text{s.a.} \quad & B(x + \gamma^o) \theta + g + r = d \\
 & (x_{ij} + \gamma_{ij}^o) |\theta_i - \theta_j| \leq (x_{ij} + \gamma_{ij}^o) \phi_{ij \max} \\
 & 0 \leq g \leq g_{\max} \\
 & 0 \leq r \leq d \\
 & 0 \leq n_{ij} \leq n_{ij \max}
 \end{aligned} \tag{1}$$

where c_{ij} is the cost of the addition of a circuit in branch i - j , x_{ij} is the total susceptance in branch i - j , $B(\cdot)$ is the susceptance matrix, γ^o is the vector of initial susceptances, whose

CARS & FOF 2001 - Durban SurAfrica

elements are γ_{ij}^o , i.e., the summation of the susceptances in branch i-j at the beginning of the optimization. n_{ij} is the number of the circuits added in branch i-j, $n_{ij} = x_{ij} / \gamma_{ij}$, where γ_{ij} is the susceptance of the new circuits, $\phi_{ij \max}$ is defined as the ratio $f_{ij \max} / \gamma_{ij}$, where $f_{ij \max}$ is the maximum flow in circuit i-j. d is the demand vector, g is the generation vector, $g_{\max}$ is the vector of maximum generation capacity, r is the vector of fictitious generation and α a penalty parameter associated with loss of load caused by lack of transmission capacity.

A complication may arise with the fact that the simple addition of a new circuit may not be enough to guarantee network connectivity. The resultant network can be a disconnected network.

3. THE PARALLEL VIRTUAL MACHINE (PVM)

The Parallel Virtual Machine (PVM) is a free software package that allows data interchange and process spawning among remote machines connected in a computer network that uses TCP/IP protocols. The PVM is available for several platforms. It is then possible to involve in machines the PVM that work under different operating systems.

The PVM's features allow the implementation of parallel algorithms. There are several parallelization schemes. One of them is the Master/Slave strategy used in this document.

The parallel version of the algorithm allows large time savings with respect to the serial version. If the work load of a single machine is partitioned into n machines, the time spent in processing corresponds roughly to T/n , where T is the time used up by a single machine and n is the number of machines working in parallel.

In this work, SUN Netra stations under the operating system SUN OS 2.7 were employed. By the time being, the Master Program's laboratory has the setup for parallel processing shown in figure 1.

To install and run the PVM the following steps are followed:

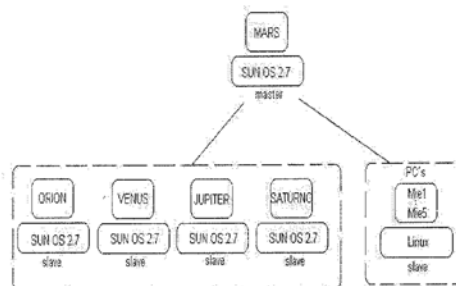


Figure 1. Computer setup for parallel processing

- Obtention of program sources, uncompressing, setting of environment variables and compilation of the PVM.

- Installation of a remote shell (rsh or ssh) in every machine that takes part in the PVM, to allow connection between them.

In the next step, the software that implements the parallel algorithm is installed in every machine. A compilation of this software may be necessary. The PVM is launched and then the process that manages the computation is started. This process spawns the applications on the other computers enrolled in the PVM and manages the data interchange.

4. GENETIC ALGORITHM OVERVIEW

The genetic algorithm used in this work is in the reference [3].

Objective Function: This function has two components: investment costs and penalties associated with the loss of load. Investment costs are functions of the decision variables (integer variables that represent the addition of new transmission equipment); penalties are functions of the continuous operation variables (power flows).

Conventional genetic algorithms are formulated as maximization problems. Minimization problems are transformed into equivalent maximization problems. This is a critical step since the selectivity of the genetic algorithm may be affected by a bad choice of the transformed objective function. Two alternative transformations have been implemented and tested. The first is similar to the one presented in [3]: $\text{Min } v \Leftrightarrow \text{Max } [K - v]$, where K is a parameter which can be adjusted during the solution process. The second type of transformation is presented in [3]: $\text{Min } v \Leftrightarrow \text{Max } [1/v]$. The transformation may or not be used.

Selection-Crossover-Mutation: From the current generation a new generation is produced by three basic mechanisms: selection, crossover and mutation.

Selection models the concept of survival of the fittest in the natural world. Based on the value of the objective function, each population configuration is allowed a certain number of offsprings. In the simplest selection criterion, each configuration is allowed to participate in reproduction proportionally to its objective function. Thus, configurations with high values in the objective function will have their *genes* represented in a larger number of offsprings.

There are three basic types of *crossover*: single-point crossover, two-point crossover and multipoint crossover. In the first case (single-point crossover), the crossover point is randomly chosen: if n is the number of elements in a configuration, a random number np (crossover point) is selected between 1 and $n-1$. Then the two configurations selected for crossover swap their first np positions. Two-point crossover is a generalization of the previous case in which the strings are swapped between two points chosen at random. In the multipoint (or continuous) crossover each element of the two matching configurations is independently tested.

The last operation in a cycle is *mutation*. When a binary representation is used, mutation simply means changing a *gene* (an element of a string representing a configuration) from 0 to 1 and vice versa. The mutation rate is controlled by a parameter pm . In conventional genetic

CARS & FOF 2001 - Durban SurAfrica

algorithms mutation restores part of the variety of the genetic material which is destroyed by selection and crossover operations (we can say that while selection and crossover tend to homogenize the population based on the survival-of-the-fittest principle, mutation recovers part of the genes which disappear in the process and which may be essential for avoiding local optimal solutions).

Implementation Details:

Initial Population: May be generated randomly or by using constructive algorithms.

Control Parameters: The range of the values used in the Pereira 2020 case are: population size between 40 and 60; crossover rate p_c between 0.7 and 0.9 and mutation rate p_m in the range from 0.005 to 0.06.

Stop Criterion: The process stops whenever the incumbent solution (least cost configuration) does not improve after a specified number of generations (typically from 10 to 80 generations) or when the maximum number of generations is reached (typically from 50 to 100 generations).

In addition to the modifications described in the previous section, another extensions such as limited selection, building blocks, unconnected networks and elitism, were required in order to develop a genetic algorithm capable of solving large scale practical cases. See reference [3].

5. PROPOSED ALGORITHMS

As mentioned, one of the important features of genetic algorithms is that they can be intrinsically parallelizable. Making use of this fact, parallel versions will be implemented according to the following two proposals: distribution of the computation of the population's objective functions among processors and assignment of subpopulations to each independent processor.

- Parallel computation of the population's objective functions.

In this case the processors are employed to compute the objective function of the different population configurations. As an example, suppose that the expansion planning problem in transmission systems is solved with a population of n configurations. A sequential genetic algorithm must solve n linear programming problems (PLs) to determine the objective function of each population individual. In this first version, we have a sequential algorithm in which only one processor runs the genetic algorithm: P0. The other processors (P1 to Pn) assist the main processor at the time of computing the objective functions of the population members. From the total computational time, the solution of PLs takes roughly 98%. In this case a parallel algorithm consists of distributing the solution of PLs among the n processors and the rest of the algorithm is equivalent to the sequential one. So, if we had a population with n individuals and if n processors were available, each one would compute one PL. Figure 2 shows a diagram of the parallel version to compute objective functions.

In problems where the computing of the objective function for each configuration implies a relatively small computational effort compared to the whole effort in the Genetic Algorithms (GA), communication problems among processors may arise. This is not the case with the expansion planning problem of the transmission system since approximately 98% of computing time is spent in the calculation of the objective function for the population individuals.

- Assignment of subpopulation configurations to each processor.

In this case the total population is divided into subpopulations and each of them is processed independently in each processor. This case is similar to the algorithm division as implemented in the parallel Simulated Annealing algorithm [4], in which each processor works independently during a fixed number of iterations. The additional advantage of the GA with respect to the parallel SA is that it can be implemented to work asynchronously.

In parallel GAs each processor performs a search practically without waiting for new information since it simply sends the best found configuration to the master and receives from the master the information about the best configuration (incumbent) found by the others until that moment. In the forementioned example, a population of n configurations can be divided into p subpopulations with n/p configurations each, using p processors. On each processor a generational cycle (selection, crossover, mutation) is implemented and the best configuration found after a fixed number of generations is sent to the master.

Simultaneously, it receives from the master the best configurations stored by it and the slave incorporates them to its subpopulation. Obviously, a processor does not have to wait for the rest of the processors to end their generational cycle. Also, the information interchange does not have to happen after every generational cycle because it can be made after a certain number of generations. To evaluate all these options, the following versions can be implemented:

- A first version where, after a number k of generations, each processor's incumbent is verified and the best is selected as the final solution. See figure 3.

CARS & FOF 2001 - Durban SurAfrica

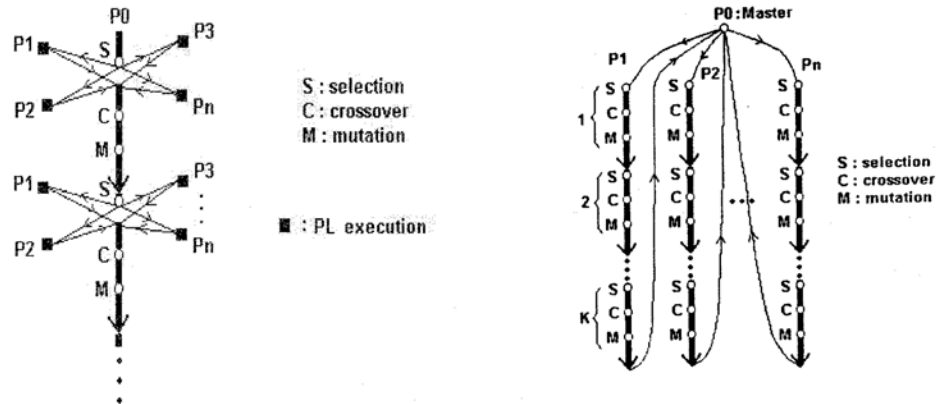


Figura 2. Objective functions parallelization. Figure 3. Subpopulation parallelization without information interchange.

- A second version where, after a number k_1 of generations, each slave processor sends the best solution found. The master receives and stores this solution if it is better than the previously stored. Then the master selects at random one or several configurations and sends them to the slave, which interchanges them with the worst configurations in its subpopulation. This process is repeated by every processor each k_1 generations until a stopping criterion is met. See figure 4.

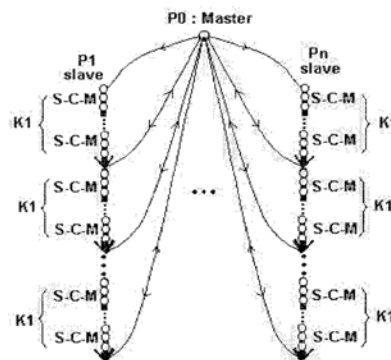


Figure 4. Subpopulation parallelization with information interchange.

- A third version is similar to the last one. The difference is that the configurations selected by the master to be sent to the slave correspond to elite solutions.

6. TESTS AND RESULTS

The implemented parallel genetic algorithm was applied to a Garver's system (6 buses/15 branches) and to Pereira's electrical system projected to year 2020 starting from year 2000 (25 buses/35 branches). Data employed in the analysis of Pereira's system correspond to case 3 of reference [5] and those of Garver's system are found in reference [7].

The basic parameters employed in the genetic algorithm were:

Garver System: Population size 20, crossover rate 0.96, mutation rate 0.13, maximum offsprings number 1. The selection type used is proportional selection with determinist remainder.

Pereira 2020 System: Population size 60, crossover rate 0.9, mutation rate 0.06, maximum offsprings number 2. The selection type used is proportional selection with determinist remainder.

An optimal solution was obtained for Garver's system in 7 generations (679 PLs) using parallel version two, and in 11 generations (1015 PLs) using parallel version one. In both cases, data interchange was performed after 5 generations. 3 workstations were used.

35 generations (2386 PLs) were required in Pereira's case using parallel version one and 22 generations (1351 PLs) using version two. 3 workstations were used.

An optimization in the number of generations to carry out data interchange was performed in the analysis of Pereira's system. Figure 5 shows that the best result was obtained when the information interchange was carried out every 6 generations.

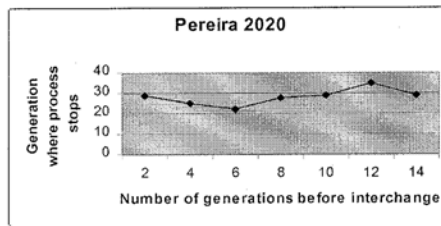


Figure 5. Information interchange parallel version two.

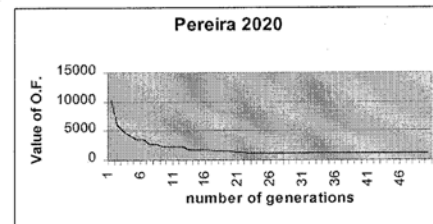


Figure 6. Evolution of the genetic algorithm.

When the information interchange is performed in a smaller or larger number of generations, more computational effort is required to obtain an optimal solution.

Figure 6 shows the evolution of the parallel genetic algorithm using 3 workstations and interchanging information every 6 generations.

7. CONCLUSIONS

CARS & FOF 2001 - Durban SurAfrica

Several parallel versions to solve the energy transmission planning problem were implemented. Advantage was taken from the facts that genetic algorithms can be intrinsically parallelized and that good computing resources and a software library to implement parallel processing were available. Excellent results were attained with regard to the value of the objective function and computing time.

The parallel versions showed better performance in computing time than the serial versions.

The transmission planning problem can also be solved using other combinatorial methods such as simulated annealing, GRASP, tabu search, hybrid, etc. They produce similar results in comparison to genetic algorithms with respect to the solution quality. They are also parallelizable. Thus, a wide range of parallel techniques can be applied to this problem.

8. ACKNOWLEDGEMENTS

The authors wish to thank to the Universidad Tecnológica de Pereira for the support to the electric planning work group.

9. BIBLIOGRAPHY

- [1] De Oliveira, S.A., de Almeida, C.R.T., Monticelli, A. (1998). Times assíncronos aplicados a métodos heurísticos construtivos de planejamento da expansão da transmissão, *XII Congresso Brasileiro de Automática*, 3, 1029-1034.
- [2] De Oliveira, S.A., De Almeida, C.R.T., Monticelli, A. (1999). Time assíncrono inicializador de métodos combinatoriais para planejamento da expansão da transmissão, *XV SNPTEE Seminário Nacional de Produção e Transmissão de Energia Elétrica*, Grupo VII, Foz do Uguacu-PR. (IT04).
- [3] Gallego, R.A., Monticelli, A., Romero, R. Transmission System Expansion Planning by extended Genetic Algorithm, *IEE Proc. Part c*, 145-3, 329-335.
- [4] Gallego, R.A., Alves, B., Monticelli, A., Romero, R. (1997). Parallel Simulated Annealing Applied to Long Term Transmission Network Expansion Planning. *IEEE Transactions on Power Systems*, 12, 181 - 188.
- [5] Gallego, R.A., Romero, R., Escobar, A. (1999). Planeacion del Sistema Eléctrico de Pereira Usando un Algoritmo Genético Eficiente, *CIER, Congreso internacional de generación, transmisión y distribución*, Bogotá, Colombia.
- [6] Gallego, R.A., Monticelli, A., Romero, R. (2000). Tabu Search Algorithm for Network Synthesis, *IEEE Transactions on Power Systems*, 15, 490-495.
- [7] Garver, L.L. (1970). Transmission Network Estimation Using Linear Programming. *IEEE Transactions on Power Apparatus and Systems*, 89, 1688-1697.
- [8] Geist, A. E et al. (1994). *PVMS users's guide and reference manual*, ORNL, Oak Ridge.

- [9] Monticelli, A., Santos, A. Jr., Pereira, M.V.F., Cunha, S., Praça, J.G., Park, B. (1982). Interactive Transmission Network Planning Using a Least-Effort Criterion, *IEEE Transactions PAS*, 101, 3919-3925.
- [10] Pereira, M.V.F., Pinto, L.M.V.G., Oliveira, G.C., Cunha, S.H.F. (1987). Composite Generation Transmission Expansion Planning, *EPRI Research Project* 2473-9.
- [11] Romero, R., Monticelli, A. (1994). A hierarchical Decomposition Approach for Transmission Network Expansion Planning. *IEEE Transactions on Power Systems*, 9, 373-380.
- [12] Romero, R., Mantovani, M., Gallego, R.A., Monticelli, A. (1999). Experimental Analysis of Selection Methods in a Genetic Algorithm Applied to the Planning Electrical Transmission. *15th International Conference on CAD/CAM Robotic & Factories of the Future*, Águas de Lindóia, Brasil.



IX SIMPÓSIO DE ESPECIALISTAS EM PLANEJAMENTO
DA OPERAÇÃO E EXPANSÃO ELÉTRICA

IX SYMPOSIUM OF SPECIALISTS IN ELECTRIC
OPERATIONAL AND EXPANSION PLANNING

**METAHEURÍSTICAS APLICADAS AO
PLANEJAMENTO DA EXPANSÃO DA TRANSMISSÃO
EM AMBIENTE DE PROCESSAMENTO DISTRIBUÍDO**

S.A. DE OLIVEIRA*, R.A. ROMERO
UNESP – Câmpus de Ilha Solteira
Brasil

R.A. GALLEG0
Univ. Tecnológica de Pereira
Colômbia

A. MONTICELLI (in memoriam)
UNICAMP/FEEC/DSEE
Brasil

Resumo: Neste trabalho são apresentados diversos algoritmos paralelos das metaheurísticas “simulated annealing” (SA) e algoritmos genéticos (AG), além da metaheurística time assíncrono (Time-A), aplicados à resolução do problema do planejamento da expansão da transmissão. Os algoritmos das primeiras usam o paradigma de programação SPMD (“single program, multiple data”) e o da última, utiliza o paradigma *Mestre-Escravo*, em que um programa mestre dá início ao processamento partindo um grupo de agentes predeterminado pelo usuário, sendo possível a inclusão e/ou exclusão dos agentes durante o processamento. Foram feitos diversos testes para sistemas elétricos de pequeno, médio e grande porte, em uma rede heterogênea de estações SUN, sistema operacional Solaris, com biblioteca para processamento paralelo PVM. Os resultados comprovam a eficácia dos algoritmos propostos quando comparados com os resultados das versões seriais de cada metaheurística isoladamente, bem como mostram uma redução significativa nos tempos de processamento.

Palavras-Chave: Planejamento da Expansão da Transmissão, “Simulated Annealing” Paralelo, Algoritmo Genético Paralelo, Times Assíncronos, Processamento Distribuído.

1. INTRODUÇÃO

O problema de planejamento da expansão da transmissão (PPET) dos sistemas de energia elétrica determina *quando*, *onde* e *quanto* de linhas e/ou transformadores devem ser instalados na rede a fim de que o sistema opere adequadamente, para uma demanda futura predeterminada, com o menor investimento possível. O planejamento *dinâmico* (*quando*), por sua vez, pode ser decomposto em subproblemas *estáticos* que tratam das questões *onde* e *quanto* (planejamento em um estágio, de um ano inicial a um ano final, preestabelecidos). Com o sistema elétrico modelado pelo fluxo de carga DC, o planejamento estático é um problema de programação não linear inteira mista (PNLIM), e dadas as dimensões que o problema assume para os casos práticos em geral observa-se o fenômeno da explosão

combinatória. Sendo que, para uma alternativa de investimento (uma dada configuração), o problema se reduz a um problema de programação linear (PPL) cujo objetivo é verificar a factibilidade desta alternativa.

Nas décadas passadas, quando os recursos computacionais eram mais limitados, foram desenvolvidos métodos heurísticos (os chamados métodos *construtivos*) para a resolução do problema, como por exemplo os métodos de Garver [01], Mínimo Esforço [02], Villasana-Garver [03] e Mínimo Corte de Carga [04]. Essas metodologias ainda são utilizadas por concessionárias como parte de procedimentos interativos que exigem uma participação ativa dos planejadores. Posteriormente, foram desenvolvidas metodologias baseadas na decomposição de Benders (os chamados métodos *clássicos*), as quais exploram a decomposição natural do problema em dois subproblemas: *de investimento* e *de operação* [05]. Sendo que mais recentemente, para resolver aqueles problemas de grande porte não resolvidos pelas metodologias anteriores, apareceram os chamados métodos *combinatórios* [08, 10] cujas características fundamentais são as de serem aplicáveis a sistemas de grande porte, chegar a soluções próximas ao ótimo global e obter soluções em razoáveis tempos de processamento. Tem-se então, uma nova classe de heurísticas chamadas *metaheurísticas*; classe de métodos que são projetados para atacar problemas “hard” de otimização combinatória onde heurísticas clássicas tem falhado em eficiência. Dentre eles se destacam: “simulated annealing” [07], algoritmos genéticos [09, 12, 20] e busca tabu [11].

Por outro lado, devido aos avanços tecnológicos dos últimos anos, as concessionárias de energia já possuem em seu parque computacional um grande número de estações de trabalho que podem ser utilizadas como uma máquina paralela virtual com o auxílio de biblioteca de domínio público para processamento

* Caixa Postal 31 – CEP 15.385-000 – Ilha Solteira-SP- Tel.: 0xx18-3743-1237 – Fax: 0xx18-3742-2735 – E_mail: grilo@dee.feis.unesp.br

paralelo e distribuído [06]. Sendo possível, portanto, o processamento concorrente de diversos programas computacionais num ambiente computacional heterogêneo. Observa-se também, o surgimento de uma nova metaheurística para a modelagem de problemas de grande porte via programação multi-agente, os chamados *Times Assíncronos* [13-16], onde agentes autônomos (algoritmos computacionais) cooperam, trabalhando todo o tempo em paralelo, com a partilha permanente de resultados (por exemplo, configurações factíveis do sistema em estudo) entre os membros do time.

Neste trabalho são apresentadas diferentes versões de algoritmos paralelos das metaheurísticas: "simulated annealing" [17] e algoritmos genéticos [18, 19], bem como um *Time-A* com agentes baseados nestas metaheurísticas, para a resolução do PPET. Foram feitos diversos testes para os sistemas: Garver (6 barras, 15 linhas), Sul brasileiro (46 barras, 79 linhas), Norte-Nordeste brasileiro (87 barras, 179 linhas) e Colombiano (93 barras, 155 linhas).

2. METAHEURÍSTICAS

Problemas de difícil tratamento, como o PPET aplicado aos sistemas elétricos de grande porte, são atualmente solucionados com maior eficiência através das chamadas metaheurísticas [07-12], entre outras; sendo aquelas de interesse particular deste trabalho apresentadas, resumidamente, a seguir.

3.1. "Simulated Annealing"

Procedimento que simula o comportamento físico de sólidos que são reaquecidos à alta temperatura, com um resfriamento controlado para que as suas partículas se arranjam no chamado estado fundamental do sólido. Em cada nível de temperatura o sólido atinge o equilíbrio térmico se a velocidade de resfriamento é suficientemente lenta, ou caso contrário, apresentará defeitos em sua estrutura cristalina. Este procedimento é aplicado ao PPET através da especificação de três aspectos distintos: uma adequada representação do problema (a função objetivo é representada pelo custo das linhas adicionadas mais uma penalidade devido ao corte de carga), o estabelecimento de um mecanismo de transição (que permite partir de uma configuração factível inicial e gerar configurações candidatas que são aceitas ou não através de um critério probabilístico) e um programa de resfriamento (escolha de parâmetros que determina a estratégia geral de controle do processo). Foi desenvolvida uma versão sequencial desta metaheurística denominada SANOR.

3.2. Algoritmos Genéticos

Técnica de busca baseada nos mecanismos da seleção natural e na genética, onde indivíduos melhor dotados têm maior chance de sobrevivência e capacidade de adaptação às alterações do meio ambiente. No PPET, o algoritmo genético parte de uma configuração inicial

(obtida aleatoriamente ou via algoritmo inicializador), faz uma seleção das melhores configurações que irão gerar novos descendentes (através da avaliação de uma função "fitness"), faz a recombinação entre estas configurações (através do operador "crossover") e num processo de mutação (com a finalidade de restaurar a perda de material genético), determina os elementos da nova população. O processo se repete até ser atendido um critério de parada. Foi desenvolvida uma versão sequencial desta metaheurística denominada GANOR, que usa uma mutação variável baseada na heurística do SA.

3.2. Times Assíncronos

Um *Time-A* consiste em uma organização de *agentes autônomos* (de maneira que novos agentes podem ser inseridos sem necessidade de gerenciamento) que se comunicam assincronamente (todos os agentes podem trabalhar em paralelo todo o tempo) através de memórias compartilhadas, e contém um *fluxo de dados cíclico* (possibilitando iteração e "feedback"). No PPET, os agentes fornecem soluções (configurações factíveis obtidas conforme a heurística de cada um) que são compartilhadas por todos, melhorando a solução global do problema.

3. TIME ASSÍNCRONO INICIALIZADOR

Em trabalhos anteriores [15,16], desenvolveu-se um *Time-A* inicializador, denominado INIC, composto de variantes dos métodos heurísticos construtivos [01-03], que permite a obtenção de configurações iniciais factíveis para inicializar, posteriormente, os algoritmos combinatórios tais como: "simulated annealing", algoritmos genéticos e busca tabu, entre outros.

Em [13], define-se *Time-A* como qualquer super-agente, cujos agentes são autônomos, cujas comunicações são assíncronas e cujo fluxo de dados é cíclico. É importante observar que, após a fase de inicialização, cada algoritmo roda em seu próprio ritmo usando os dados disponíveis na(s) memória(s) compartilhada(s). Isto mostra o caráter assíncrono do *Time-A* que pode facilmente ser executado em paralelo; cada agente e memória podem ser alocados em diferentes máquinas de uma rede de computadores. As soluções iniciais factíveis utilizadas pelas metaheurísticas descritas neste trabalho, foram obtidas, previamente, via o *Time-A* INIC.

4. OS ALGORITMOS PROPOSTOS

Com o objetivo de avaliar o desempenho de versões paralelas das citadas metaheurísticas para a resolução do PPET, em um ambiente de processamento distribuído, foram desenvolvidos algoritmos que são detalhados a seguir.

4.1 "Simulated Annealing" Paralelo

Esta versão do algoritmo, implementada para ambiente distribuído com o auxílio da biblioteca PVM

[08], foi desenvolvida a partir da versão do algoritmo paralelo da divisão [7], inicialmente desenvolvida para uma máquina paralela *nCUBE* com 64 processadores.

Algoritmo paralelo da divisão - Se N_k é o número de tentativas de busca de novas configurações realizadas na temperatura T_k e n_p o número de processadores, então, em cada nível de temperatura, cada processador realiza N_k/n_p tentativas, sendo que cada processador gera sua própria sequência de configurações. Em outras palavras, a tarefa de busca de novas configurações é dividida entre todos os processadores para cada nível de temperatura. Permite-se ainda, haver possibilidade ou não de comunicações entre os processadores, escolhendo-se a melhor configuração das n_p configurações disponíveis, que por sua vez será usada como configuração inicial no nível de temperatura seguinte (algoritmo SAPARNOR).

Um exemplo de resultado obtido com este algoritmo para o sistema Norte-Nordeste brasileiro (Plano 2008), é apresentado na Fig. 1, a seguir.

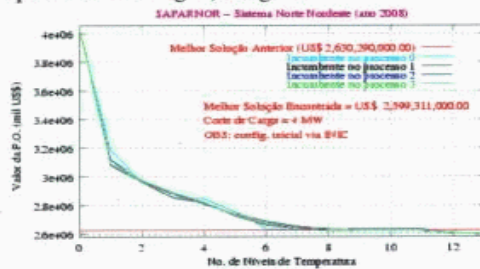


Fig. 1 – Resultado com o SAPARNOR.

4.2 Algoritmos Genéticos Paralelo

A teoria dos algoritmos genéticos mostra que existe um paralelismo intrínseco dentro do funcionamento de um algoritmo genético padrão (AG). Entretanto, pode-se implementar versões paralelas do mesmo de acordo com duas propostas de caráter geral:

Avaliação paralela das funções objetivo da população - Neste caso, os processadores são usados para avaliar a função objetivo de cada configuração da população. Por exemplo, supor que o PPET está sendo resolvido com uma população de n configurações. Um AG sequencial deve resolver n PPL's para determinar a função objetivo de cada um dos elementos (configurações) da população. Deve-se lembrar que a resolução dos PPL's utiliza mais de 95% do tempo de processamento de um AG sequencial. Neste caso, um algoritmo paralelo simples (GAPAR1A) paraleliza a resolução dos PPL's, isto é, resolve os n PPL's usando os n processadores (um PPL por processador) e o resto do processo é equivalente a um AG sequencial. Um exemplo prático é utilizar uma máquina paralela com 64 processadores (*nCUBE*, por exemplo), para uma população de 64 configurações. Uma outra alternativa para a resolução do problema, agora considerando uma rede de estações de trabalho, é dividir o cálculo

dos n PPL's entre p processadores (estações), cabendo assim a cada processador: n/p PPL's (algoritmo GAPAR1B).

A Fig. 2, a seguir, apresenta um exemplo de resultado obtido com este algoritmo para o sistema Norte-Nordeste brasileiro (Plano 2002).

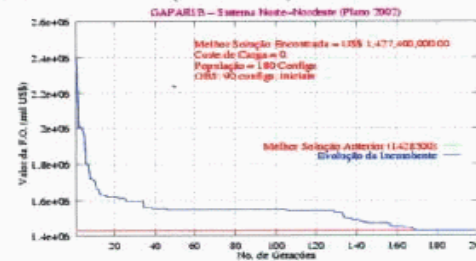


Fig. 2 – Resultado com o GAPAR1B.

Alocação de sub-populações em cada processador - Neste caso a população total é dividida em sub-populações e cada sub-população trabalha de maneira independente em cada processador. Este caso é praticamente equivalente ao algoritmo da divisão e ao algoritmo híbrido implementado para SA [7]. A vantagem adicional dos AG em relação aos algoritmos de SA paralelos é que um AG pode ser implementado de maneira assíncrona. Uma população de n configurações pode ser dividida em p sub-populações de n/p configurações cada uma e seria usado p processadores. Cada processador implementaria um ciclo geracional (seleção-recombinação-mutação-avaliação da função objetivo) e enviaria para o processo mestre a melhor configuração encontrada. Simultaneamente receberia do mestre a melhor configuração (ou as melhores configurações do processo integral) que o processador incorporaria na sub-população para melhorar o processo de seleção. Logicamente, um processador não precisaria esperar que os outros processadores terminem seu correspondente ciclo geracional. Também a troca de informações não precisa ser a cada ciclo geracional, podendo ser realizada após cada certo número de gerações. Para avaliar todas estas opções, fez-se as seguintes versões de algoritmos:

- Uma primeira versão (GAPAR2A) onde, após um número razoavelmente grande de gerações (k), verifica-se a incumbente relativa a cada processador e escolhe-se a melhor dentre elas.
- Uma segunda versão (GAPAR2B) aonde, após um número pequeno de gerações (k_1), o mestre recebe as melhores configurações encontradas, mantendo o valor da melhor dentre elas como uma referência e continua o processamento até um critério de parada ser satisfeito.
- Uma outra versão (GAPAR3A) onde, após um número pequeno de gerações k_1 , o mestre armazena m configurações elites, diversificadas; repassando-as novamente a cada processador que

as incorporam à sua sub-população, satisfazendo um determinado critério de parada.

- E finalmente, uma versão (GAPAR3B) onde, após um número pequeno de gerações (k_1), o mestre recebe as configurações elites de cada processador, armazena as elites das elites, e as repassa a cada processador que as incorporam à sua sub-população, até um critério de parada ser satisfeito.

Na Fig. 3, apresenta-se um exemplo de resultado obtido com o algoritmo GAPAR2B, para o sistema Colombiano (Plano 2012).

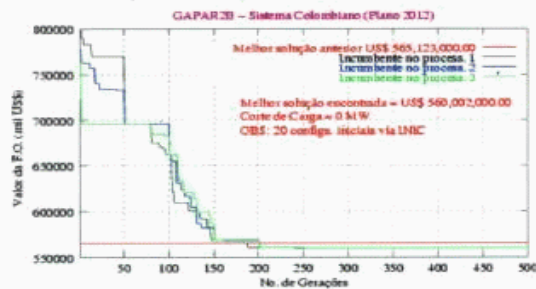


Fig. 3 – Resultado com o GAPAR2B.

4.3 Time Assíncrono de Metaheurísticas

Os algoritmos mencionados, bem como outros possíveis, podem ser incorporados a um *Time-A* de metaheurísticas de diferentes maneiras; como, por exemplo, o apresentado a seguir.

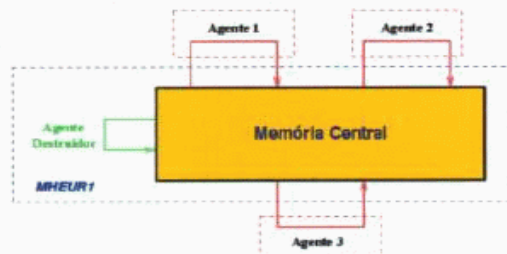


Figura 4 – Exemplo de Time-A de metaheurísticas

Na Fig. 4, o *Time-A* é composto, por exemplo, por três agentes independentes que trabalham segundo suas respectivas heurísticas compartilhando uma *memória central* composta por uma população de configurações. Nesta memória, os agentes tomam uma configuração ou uma sub-população e procuram melhorá-las. O agente destruidor é na prática uma função intrínseca do *Time-A* que mantém fixo o tamanho da população na memória central.

O *Time-A* apresentado neste trabalho é da forma da Fig. 4, sendo a memória central representada pelo programa MHEUR e seus agentes são versões paralelas derivadas dos algoritmos SANOR e GANOR; denominados SAM e GAM respectivamente. O agente SAM recebe do MHEUR a pior configuração

e após cada ciclo de temperatura devolve a configuração incumbente que substitui aleatoriamente uma configuração da população; enquanto que o agente GAM trabalha durante k gerações sobre a população recebida previamente, devolvendo a incumbente que é colocada no lugar da pior configuração da população atual.

A Fig. 5 apresenta um exemplo de resultado obtido para o sistema Norte-Nordeste (Plano 2002), com três agentes SAM.

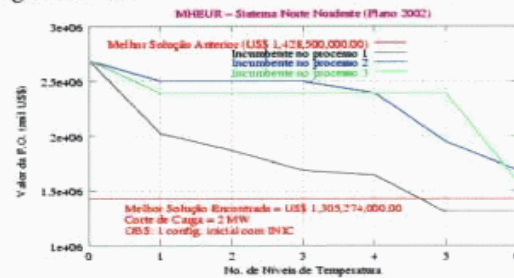


Fig. 5 – Resultado com o MHEUR

5. AMBIENTE DISTRIBUÍDO UTILIZADO

Todas as simulações apresentadas na seção seguinte foram efetuadas em um ambiente de processamento distribuído constituído de uma rede de estações SUN Sparc Ultra-1, sistema operacional Solaris, com biblioteca para processamento paralelo PVM 3.3.5, disponíveis no DSEE, da FEEC-Unicamp.

6. RESULTADOS E CONCLUSÕES

Na Tab. 1, são apresentados resultados de simulações com os algoritmos seriais das metaheurísticas *SA* e *AG* para todos os sistemas citados. Em todas as simulações para os sistemas Garver e Sul, os dados apresentados correspondem aos valores ótimos globais, já conhecidos na literatura. Para os demais sistemas, os valores obtidos são destacados.

Nas Tabs. 2 a 5, são apresentados os resultados das versões paralelas das metaheurísticas, simuladas para os sistemas Garver, Sul Brasileiro, Norte-Nordeste e Colombiano. E finalmente, na Tab. 6 são apresentados os resultados das simulações do *Time-A* MHEUR, para os sistemas Garver e Sul brasileiro.

Para uma melhor visualização de algumas das configurações ótimas obtidas, as mesmas são apresentadas no Apêndice A nos diagramas unifilares dos sistemas Garver, Sul brasileiro e Colombiano, respectivamente nas Figs. A1 e A2, A3 e A4, e A5.

Uma análise mais detalhada das tabelas quanto aos tempos de processamento de cada algoritmo, comprovam a eficácia dos algoritmos propostos na obtenção de soluções, com uma redução significativa do tempo, quando faz-se a comparação com os tempos dos algoritmos seriais. Outra conclusão importante é o benefício obtido no uso de boas configurações iniciais para a inicialização das metaheurísticas, que acelera, na maioria dos casos, a convergência dos algoritmos.

Tab. 1 – Resultados das versões seriais SANOR/GANOR

algoritmos		SANOR		GANOR	
itens	sistema	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.
Garver c/ redespaço					
níveis temper./gerações		5	2	4	2
número de PPL's		401	185	409	130
tempo médio por PPL		0,0588	0,0575	0,0577	0,0594
tempo p/solução ótima		23,6 s	10,6 s	23,6 s	7,5 s
Garver s/ redespaço					
nível temper./gerações		7	4	15	6
número de PPL's		559	208	588	200
tempo médio por PPL		0,0593	0,0565	0,0583	0,0627
tempo p/solução ótima		33,1 s	11,8 s	34,3 s	12,5 s
Sul brasileiro c/ redespaço					
nível temper./gerações		3	5	34	16
número de PPL's		1.617	4.814	6.410	2.556
tempo médio por PPL		0,2490	0,2416	0,2077	0,2322
tempo p/solução ótima		6,7 min	19,4min	22,2min	9,9 min
Sul brasileiro s/ redespaço					
níveis temper./gerações		7	6	37	22
número de PPL's		6.071	4.426	2.856	3.295
tempo médio por PPL		0,2750	0,2328	0,2208	0,2213
tempo p/solução ótima		27,8 min	17,2min	10,5min	12,2min
c/ Inic. e/ Inic.					
Norte-Nordeste (2002)					
níveis temper./gerações		10		779	
número de PPL's		55.134		277.390	
tempo médio por PPL		0,5334		0,6774	
tempo p/solução ótima		8,2 h ^A		52,2 h ^B	
Norte-Nordeste (2008)					
níveis temper./gerações		9		156	
número de PPL's		99.039		81.525	
tempo médio por PPL		0,6980		1,0849	
tempo p/solução ótima		19,2 h ^C		24,6 h ^B	
Colombiano (2012)					
níveis temper./gerações		19		481	
número de PPL's		301.086		278.196	
tempo médio por PPL		0,8951		0,8619	
tempo p/solução ótima		74,9 h ^E		66,6 h ^F	

^A ótimo local US\$ 1.453,205,000.00 – corte de carga 0 MW^B ótimo local US\$ 1.429,808,000.00 – corte de carga 0 MW^C ótimo local US\$ 2,602,361,000.00 – corte de carga 0 MW^D ótimo local US\$ 2,594,167,000.00 – corte de carga 2 MW^E ótimo local US\$ 560,692,000.00 – corte de carga 0 MW^F ótimo local US\$ 560,002,000.00 – corte de carga 0 MW

7. REFERÊNCIAS BIBLIOGRÁFICAS

- [01] L. L. Garver, "Transmission network estimation using linear programming", IEEE Transactions on Power Apparatus and Systems, vol. PAS-89, no. 7, Sep./Oct. 1970, pp. 1688-1697.
- [02] A. Monticelli, A. Santos Jr., M.V.F. Pereira, S.H. Cunha, B.J. Parker, and J.C.G. Praça, "Interactive transmission network planning using a least-effort criterion", IEEE Transactions on Power Apparatus and Systems, vol. PAS-101, no. 10, October 1982, pp. 3919-3925.
- [03] R. Villasana, L.L. Garver, and S.J. Salon, "Transmission network planning using linear programming", IEEE Transactions on Power Apparatus and Systems, vol. PAS-104, no. 2, February 1985, pp. 349-356.

[04] M.V.F. Pereira, and L.M.V.G. Pinto, "Application of sensibility analysis of load supplying capability to interactive transmission expansion planning", IEEE Transactions on Power Apparatus and Systems, vol. PAS-104, no. 2, February 1985, pp. 381-389.

[05] R. Romero, and A. Monticelli, "A hierarchical decomposition approach for transmission network expansion planning", IEEE Transactions on Power Systems, vol. 9, no. 1, February 1994.

[06] A. Geist, et al., "PVM parallel virtual machine: a users' guide and tutorial for networked parallel computing", Cambridge. The MIT Press, 1994, 279 p.

[07] R. Romero, R. Gallego, and A. Monticelli, "Transmission system expansion planning by simulated annealing", IEEE Transactions on Power Systems, vol. 11, no. 1, February 1996, pp. 364-369.

[08] R.A. Gallego, "Planejamento a longo prazo de sistemas de transmissão usando técnicas de otimização combinatorial", Campinas, Unicamp, 1997, 249 p. (Thesis, Ph. D.)

[09] R.A. Gallego, R. Romero, and A. Monticelli, "Transmission system expansion planning by extended genetic algorithm", IEEE Proceedings – Generation, Transmission and Distribution, vol. 3, no. 145, May 1998, pp. 329-335.

[10] R.A. Gallego, A. Monticelli, and R. Romero, "Comparative studies of non-convex optimization methods for transmission network expansion planning", IEEE Transactions on Power Systems, vol. 13, no. 3, Aug. 1998, pp. 822-828.

[11] R.A. Gallego, A. Monticelli, and R. Romero, "Tabu search algorithm for network synthesis", IEEE Transactions on Power Systems, vol. 15, no. 2, May 2000, pp. 490-495.

[12] E.L. da Silva, H.A. Gil, and J.M. Areiza, "Transmission network expansion planning under an improved genetic algorithm", IEEE Transactions on Power Systems, vol. 15, no. 4, November 2000, pp. 1168-1175.

[13] S.N. Talukdar, and P. de Souza, "Scale efficient organizations", Proceedings of 1992 International Conference on Systems, Man and Cybernetics, New York, IEEE, pp. 1458-1463.

[14] P. de Souza, "Asynchronous organizations for multi-algorithm problems", Pittsburgh, Carnegie Mellon University, 1993. (Thesis, Ph.D.)

[15] S.A. de Oliveira, C.R.T. de Almeida, and A. Monticelli, "Times Assíncronos aplicados a métodos heurísticos construtivos de planejamento da expansão da transmissão", Proceedings of CBA-1998, Uberlândia, SBA/UFU, vol. III, pp. 1029-1034.

[16] S.A. de Oliveira, C.R.T. de Almeida, and A. Monticelli, "Time Assíncrono inicializador de métodos combinatoriais para planejamento da expansão da transmissão", Proceedings of 1999 SNPTTE, Foz do Iguaçu, CIGRÉ-Itaipu Binacional, Grupo VII-GPL/04.

[17] R.A. Gallego, A.B. Alves, A. Monticelli, and R. Romero, "Parallel simulated annealing applied to long term transmission network expansion planning", IEEE Transactions on Power Systems, vol. 12, no. 1, February 1997, pp. 181-188.

[18] S.A. de Oliveira, R.A. Romero, R.A. Gallego, and A. Monticelli, "Algoritmo genético paralelo aplicado ao planejamento da expansão da transmissão", Proceedings of CBA-2000, Florianópolis, SBA/UFSC, pp. 1790-1795.

[19] S.A. de Oliveira, R.A. Gallego, A. Escobar, and R. Restrepo, "Parallel combinatorial algorithm for static planning of a genetic algorithm for the planning of an electric power transmission system", Proceedings of CARS & FOF – 2001, Durban, pp. 1183-1191.

[20] R.A. Gallego, A. Escobar, R.A. Romero, and S.A. de Oliveira, "Híbrido model analysis and constructive algorithms in the performance of a genetic algorithm for the planning of an electric power transmission system", Proceedings of CARS & FOF – 2001, Durban, pp. 850-859.

Tab. 2 – Resultados para o algoritmo SAPARNOR

itens	Garver				Sul brasileiro				Norte-Nordeste		Colombiano
	c/ redespaço		s/ redespaço		c/ redespaço		s/ redespaço		2002	2008	2012
	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.	c/ Inic.	c/ Inic.	c/ Inic.
níveis temperatura	2	2	2	2	5	6	8	6	18	13	16
número de PPL's	90	25	67	43	2.413	3.879	2563	3.488	95.770	52.099	76.046
número de máquinas	4	4	4	4	4	4	4	4	4	4	4
configs. iniciais	-	1	-	1	-	1	-	1	1	1	1
tempo médio por PPL	0,0215	0,0224	0,0313	0,0243	0,1214	0,1223	0,1354	0,1358	0,3742	0,2786	0,5837
tempo p/ solução ótima	1,9 s	0,6 s	2,1 s	1,0 s	4,9 min	7,9 min	5,8 min	7,9min	9,9 h ^A	4 h ^B	12,3 h ^C

^A US\$ 1,402,716,000.00 – 0 MW^B US\$ 2,599,311,000.00 – 0 MW^C US\$ 564,142,000.00 – 0 MW

Tab. 3 – Resultados para os Algoritmos Genéticos Paralelos - sistema Garver

sistemas algoritmos	Garver c/ redespacho						Garver s/ redespacho					
	GP1B		GP2A		GP2B		GP1B		GP2A		GP2B	
	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.
tamanho da população	40	40	120	120	120	120	20	20	120	120	60	60
níveis temper./gerações	11	2	5	2	7	2	6	7	14	5	28	12
número de PPL's	1.015	100	394	99	679	136	286	240	1.040	320	876	358
número de máquinas	3	3	3	3	3	3	3	3	3	3	3	3
configs. iniciais	-	8	-	8	-	8	-	8	-	8	-	8
troca de inform.(gers.)	-	-	-	-	5	5	-	-	-	-	5	5
tempo médio por PPL	0,0202	0,0194	0,0208	0,0209	0,0226	0,0204	0,0207	0,0207	0,0204	0,0202	0,0219	0,022
tempo p/ solução ótima	20,5 s	1,9 s	8,2 s	2,1 s	15,3 s	3,3 s	5,9 s	4,1 s	21,2 s	6,4 s	19,2 s	8,1 s

Tab. 4 – Resultados para os Algoritmos Genéticos Paralelos - sistema Sul brasileiro

sistemas algoritmos	Sul brasileiro c/ redespacho						Sul brasileiro s/ redespacho					
	GP1B		GP2A		GP2B		GP1B		GP2A		GP2B	
	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.
tamanho da população	40	40	180	180	60	60	120	120	120	120	180	180
níveis temper./gerações	11	16	28	17	37	19	23	18	25	20	27	15
número de PPL's	1.505	2.556	5.612	2.387	2.071	1.193	9.308	8.101	3.331	1.520	7.395	2.141
número de máquinas	3	3	3	3	3	3	3	3	3	3	3	3
configs. iniciais	-	20	-	20	-	10	-	20	-	20	-	20
troca de inform.(gers.)	-	-	-	-	5	10	-	-	-	-	10	5
tempo médio por PPL	0,2234	0,2322	0,1194	0,1150	0,1198	0,1142	0,1343	0,1356	0,1354	0,1385	0,2797	0,127
tempo p/ solução ótima	5,6 min	9,9 min	11,2min	4,6min	4,1min	2,3min	20,8min	18,3min	7,5 min	3,5min	34,5min	4,5min

Tab. 5 – Resultados para os Algoritmos Genéticos Paralelos - sistema Norte-Nordeste (2002/2008) e Colombiano (2012)

sistemas algoritmos	Norte-Nordeste (2002)			Norte-Nordeste (2008)			Colombiano (2012)		
	GP1B	GP2A	GP2B	GP1B	GP2A	GP2B	GP1B	GP2A	GP2B
	c/ Inic.	c/ Inic.	c/ Inic.	c/ Inic.	c/ Inic.	c/ Inic.	c/ Inic.	c/ Inic.	c/ Inic.
tamanho da população	180	180	800	180	200	180	180	200	180
níveis temper./gerações	163	177	4	83	201	138	339	432	240
número de PPL's	76.616	27.742	3.708	29.398	24.002	18.970	192.642	100.853	21.929
número de máquinas	3	3	3	3	3	3	3	3	3
configs. iniciais	60	90	400	60	60	60	20	20	10
troca de inform.(gers.)	-	-	50	-	-	100	-	-	50
tempo médio por PPL	0,3696	0,3529	0,3029	0,2817	0,3988	0,3841	0,6371	0,6013	0,5518
tempo p/ solução ótima	7,9 h	2,7 h	18,7min	2,3 h	2,7 h	2,3 h	34,1 h	16,8 h	3,4 h
ótimo (mil US\$)	1,543,264	1,440,537	1,356,178	2,653,226	2,624,816	2,680,958	560,692	560,002	560,002
corte de carga (MW)	0	1	0	0	0	0	0	0	0

Tab. 6 – Resultados para o Time-A MHEUR

sistemas	Garver				Sul brasileiro			
	c/ redespacho		s/ redespacho		c/ redespacho		s/ redespacho	
	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.	s/ Inic.	c/ Inic.
tamanho da população	16	8	8	16	40	40	40	40
níveis temper./gers.	3	5	3	4	7	4	3	5
número de PPL's	170	236	204	227	2010	2.773	2.547	4662
número de máquinas	4	4	4	4	4	4	3	4
configs. iniciais	-	1	-	8	-	1	-	1
agentes	4SAM	4SAM	4 SAM	4SAM	4 SAM	4SAM	3 SAM	4SAM
tempo médio por PPL	0,0059	0,0030	0,0211	0,0270	0,1167	0,1242	0,1304	0,1219
tempo p/ solução ótima	1,0 s	0,7 s	4,3 s	6,1 s	3,9 min	5,7 min	5,5 min	9,5 min

APÊNDICE A

A1. ALGUMAS SOLUÇÕES ÓTIMAS OBTIDAS

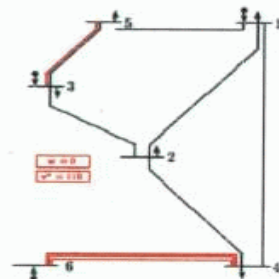


Figura A1 – Solução ótima do sistema Garver com redespacho

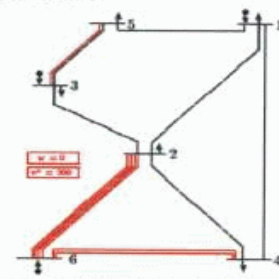


Figura A2 – Solução ótima do sistema Garver sem redespacho

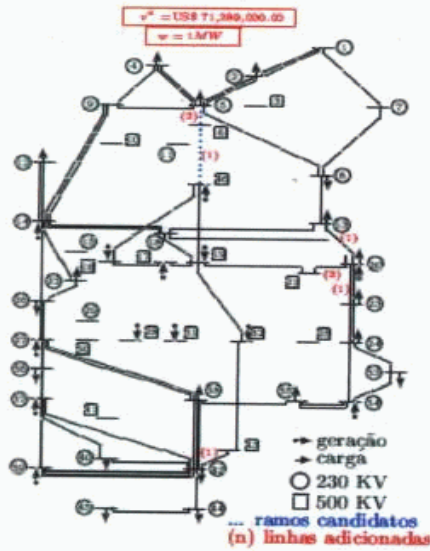


Fig. A3 – Solução ótima do sistema Sul com redesenho

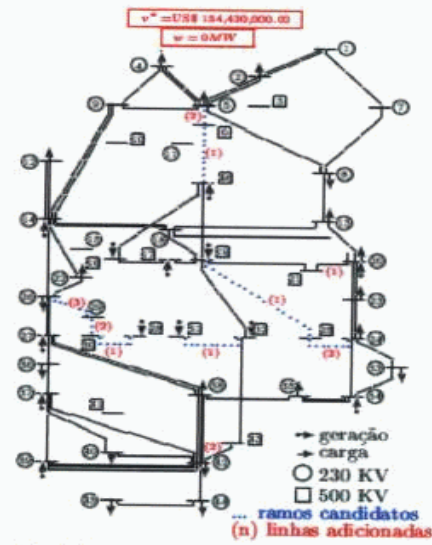


Fig. A4 – Solução ótima do sistema Sul sem redesenho

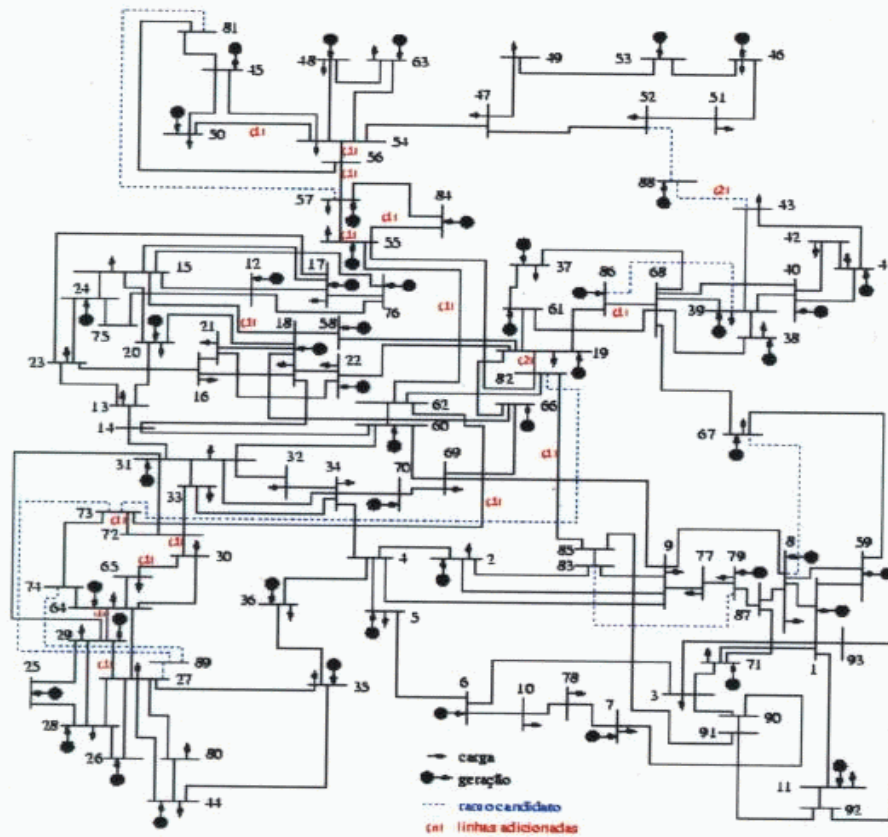


Figura A5 – Solução ótima do sistema Colombiano (Plano 2012)

Índice Remissivo

- a-teams
 - ver* times assíncronos, 36
- Abstract, i
- agente GarverA, 48
- agente GarverB, 49
- agente GarverC, 49
- agente mccF, 51
- agente minD, 50
- agente minE, 50
- algoritmo INIC, 51
- algoritmo MHEUR, 82, 83
- algoritmos combinatórios
 - algoritmos genéticos, 1
 - busca tabu, 1
 - simulated annealing, 1
- algoritmos de otimização clássica
 - métodos de enumeração
 - branch and bound, 13
 - técnica de decomposição de Benders, 12
- algoritmos genéticos, 18
 - algoritmo GANORD, 61
 - descrição, 63
 - estruturas básicas, 61
- aplicações, 24
- bloco de construção, 21
- comprimento de definição, 21
- descrição, 19, 149
- esquema, 20
- instância, 20
- mecanismos de dimensionamento
 - linear, 22
 - truncamento sigma, 22
- operadores, 19
 - crossover, 19
 - inversão, 19
 - mutação, 19
- ordem, 21
- seleção
 - baseada na ordenação, 22
 - Breeding Poll, 23
 - Roulette Whell, 23
- algoritmos genéticos paralelos, 67
 - algoritmo GAP2BCIR, 77
 - algoritmo GAP2BDIM, 77
 - algoritmo GAPAR1A, 68
 - descrição, 68
 - algoritmo GAPAR1B, 70
 - descrição, 70
 - algoritmo GAPAR2A, 73
 - descrição, 73
 - algoritmo GAPAR2B, 75
 - descrição, 75
 - algoritmo GAPAR3A, 77
 - descrição, 77
 - algoritmo GAPAR3B, 79
 - descrição, 79
- algoritmos heurísticos construtivos
 - índices de sensibilidade, 11
 - algoritmo heurístico de Garver, 9
 - Garver, 48
 - método de mínimo corte de carga, 11, 50
 - método de mínimo esforço, 10, 49
 - método de Villanasa-Garver, 9
- ambiente de processamento distribuído, 58
- asynchronous team
 - ver* times assíncronos, 36
- bibliografia, 159
 - algoritmos genéticos, 182
 - algoritmos híbridos, 201
 - busca tabu, 196
 - gerais, 159

- planejamento da expansão da transmissão,
 - 163
 - simulated annealing, 171
 - times assíncronos, 204
- busca tabu, 30
 - aplicações, 34
 - descrição, 30
 - diversificação, 34
 - exploração sensível, 30
 - intensificação, 32
 - medidas de residência, 32
 - medidas de transição, 32
 - memória
 - de curto prazo, 32
 - de longo prazo, 32
 - memória adaptativa, 30
 - ocupação tabu, 32
 - vizinhança, 31
- conclusões, 149
- corte de carga, 7
- desempenho, 150
- eficiência, 150
- heurísticas
 - clássicas, 16
 - definição, 16
- leis de Kirchhoff, 5, 48
- metaheurísticas, 15
 - algoritmos genéticos, 18, 149
 - algoritmos seriais, 59
 - busca tabu, 30
 - definição, 16
 - simulated annealing, 24, 149
 - times assíncronos, 36, 149
- otimização combinatória, 15
 - conceitos básicos, 16
 - algoritmo de busca local, 17
 - instância, 16
 - mecanismo de geração, 17
 - otimalidade local, 18
 - vizinhança, 17, 18
 - definição, 16
 - problema do caixeiro viajante, 12
 - problemas NP-completo, 12
- planejamento da expansão da transmissão
 - inicialização, 13
 - modelagem
 - modelo de transportes, 7
 - modelo híbrido, 8
 - técnicas de resolução
 - algoritmos de otimização clássica, 11
 - algoritmos de otimização combinatória, 12
 - algoritmos heurísticos construtivos, 8
- planejamento estático da expansão da transmissão, 3
 - diagrama de blocos, 4
 - modelagem
 - modelo DC, 5
- problema do caixeiro viajante, 12, 39
- referências bibliográficas, 153
- resultados
 - eficiência, 150
 - GANORD, 89, 97, 106, 114, 121, 129, 135, 144
 - GAPAR1B, 91, 99, 108, 116, 131, 137, 146, 150
 - GAPAR2A, 92, 100, 109, 117, 123, 132, 138, 146, 150
 - GAPAR2B, 93, 101, 110, 118, 124, 133, 139, 147, 150
 - GAPAR3A, 94, 102, 111, 119, 127, 141, 147, 150
 - MHEUR, 95, 103, 112, 120, 129, 142, 148, 150
 - SANORD, 88, 96, 104, 113, 121, 129, 134, 136, 144
 - SAPARNOR, 90, 98, 107, 115, 122, 131, 145, 150
 - sistema Colombiano
 - plano 2012, 134
 - sistema Garver
 - com redespacho, 88
 - sem redespacho, 96

- sistema Norte-Nordeste
 - plano 2002, 121
 - plano 2008, 129
- sistema Sul Brasileiro
 - com redespacho, 104
 - sem redespacho, 113
- speedup, 150
- Resumo, i
- simulated annealing, 24
 - algoritmo de Metropolis, 25
 - algoritmo SANORD, 59
 - descrição, 60
 - estruturas básicas, 59
 - algoritmo SAPARNOR
 - descrição, 65
 - aplicações, 28
 - critério de aceitação, 26
 - definição, 25
 - descrição, 26, 149
 - transição, 26, 28
- simulated annealing paralelo, 63
 - algoritmo da divisão, 66
- sistemas testes
 - Colombiano
 - configuração inicial, 234
 - dados de barras, 225
 - dados de linhas, 227
 - melhor configuração (2012), 234
 - Garver, 209
 - config. ótima c/ redespacho, 209
 - config. ótima s/ redespacho, 209
 - configuração inicial, 209
 - dados de barras, 209
 - dados de linhas, 209
 - Norte-Nordeste Brasileiro
 - dados de barras, 217
 - dados de linhas, 219
 - Sul Brasileiro
 - config. ótima c/ redespacho, 214
 - config. ótima s/ redespacho, 214
 - configuração inicial, 214
 - dados de barras, 212
 - dados de linhas, 213
- speedup, 150
- sugestões para futuros trabalhos, 149
- time assíncrono de metaheurísticas, 57, 82
 - agente GAM
 - descrição, 85
 - agente SAM
 - descrição, 85
 - algoritmo MHEUR
 - descrição, 84
 - algoritmo proposto
 - estrutura do programa, 83
 - estrutura dos agentes, 85
 - esquema de funcionamento, 84
 - exemplos, 82, 83
- time assíncrono inicializador
 - algoritmo proposto, 51
 - estrutura do programa, 54
 - estrutura dos agentes, 55
 - exemplo, 52
- times assíncronos, 36
 - aplicações, 45
 - comportamento
 - diversidade, 44
 - dualidade, 44
 - escala, 44
 - expansão, 44
 - paralelismo, 44
 - tamanho da população, 44
- conceitos básicos
 - agentes autônomos, 37
 - agentes de construção, 39
 - agentes de destruição, 39
 - arcos, 37
 - comunicações assíncronas, 37
 - fluxo de dados, 37
 - fluxo de dados cíclico, 37
 - nós, 37
 - objetos, 37
 - objetos de dados, 37
 - super-agente, 38
- definição, 37, 38
- descrição, 38, 149
- inic
 - execução, 235
- problema do caixeiro viajante

roteiro, 40

traveling salesman problem

ver problema do caixeiro viajante, 39

TSP

ver problema do caixeiro viajante, 39

Classificação de Segurança Livre		Documento no.	
Data (mês e ano) Outubro/2004		Projeto no.	
Título e subtítulo Metaheurísticas aplicadas ao planejamento da expansão da transmissão de energia elétrica em ambiente de processamento distribuído		No. do volume Único	
		No. da parte	
Título do projeto			
Entidade Executora (autor coletivo)		Autor(es) Sérgio Azevedo de Oliveira	
Entidade patrocinada (cliente ou destinatário principal) Faculdade de Engenharia Elétrica e de Computação - UNICAMP			
Resumo Neste trabalho foram desenvolvidas diversas metaheurísticas combinatórias para a resolução do problema do planejamento da expansão da transmissão dos sistemas de energia elétrica, analisado do ponto de vista estático e à longo prazo, dentre as quais uma versão paralela da metodologia “simulated annealing” e diversas versões paralelas de algoritmos genéticos; além de um time assíncrono cujos agentes são variantes destas metaheurísticas. Todas estas versões são inicializadas por um time assíncrono de algoritmos heurísticos construtivos e executadas em um ambiente de processamento distribuído composto por uma rede heterogênea de estações SUN, sistema operacional SunOS, com biblioteca para processamento paralelo PVM. Foram feitos diversos testes para os sistemas: Garver (6 barras/15 ramos), Sul brasileiro (46 barras/79 ramos), Norte-Nordeste brasileiro (87 barras/179 ramos) e sistema colombiano (93 barras/155 ramos), e os resultados comprovam a eficácia das metodologias propostas quando comparados com os resultados das versões seriais de cada metaheurística isoladamente, bem como mostram uma redução significativa nos tempos de processamento.			
Palavras-chave Planejamento à Longo Prazo da Expansão da Transmissão, Planejamento Estático da Expansão da Transmissão, Metaheurísticas, Otimização Combinatória, Simulated Annealing, Algoritmos Genéticos, Times Assíncronos, Processamento Distribuído.			
No. da edição	No. de páginas 315	ISSN (para relatórios publicados)	Classificação (CDC ou CDD)
Distribuidor		Número de exemplares 10 (dez)	Preço -
Observações Tese apresentadas à Faculdade de Engenharia Elétrica e de Computação, UNICAMP, em 01/10/2004, como parte dos requisitos para a obtenção do título de Doutor em Engenharia Elétrica.			