ALDIR SILVA SOUSA

**Desenvolvimento de Modelos e Algoritmos Sequenciais e Paralelos
para o Planejamento da Expansão de Sistemas de Transmissão de
Energia Elétrica**

Tese apresentada à Escola de Engenharia de São Carlos, da
Universidade de São Paulo, como parte dos requisitos para
obtenção do título de Doutor em Ciências, Programa de
Engenharia Elétrica

Área de Concentração: Sistemas Elétricos de Potência
Orientador: Prof. Dr. Eduardo Nobuhiro Asada

São Carlos - SP
2012

¹Trata-se da versão corrigida da tese. A versão original se encontra disponível na EESC/USP que aloja o Programa de Pós-Graduação de Engenharia Elétrica.

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca - EESC/USP

S725d	<p>Sousa, Aldir Silva</p> <p>Desenvolvimento de modelos e algoritmos sequenciais e paralelos para o planejamento da expansão de sistemas de transmissão de energia elétrica. / Aldir Silva Sousa ; orientador Eduardo Nobuhiro Asada. São Carlos, 2012.</p> <p>Tese (Doutorado - Programa de Pós-Graduação em Engenharia Elétrica e Área de Concentração em Sistemas Elétricos de Potência)-- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2012.</p> <p>1. Planejamento de sistemas de transmissão. 2. Otimização. 3. Metaheurísticas. 4. Algoritmos multiobjetivos. 5. Múltiplos cenários de geração. 6. Programação paralela. I. Título.</p>
-------	--

FOLHA DE JULGAMENTO

Candidato: Bacharel **ALDIR SILVA SOUSA**.

Título da tese: "Desenvolvimento de modelos e algoritmos sequenciais e paralelos para o planejamento da expansão de sistemas de transmissão de energia elétrica".

Data da defesa: 16/03/2012

Comissão Julgadora:

Resultado:

Prof. Dr. **Eduardo Nobuhiro Asada (Orientador)**
(Escola de Engenharia de São Carlos/EESC)

APROVADO

Prof. Titular **Djalma Mosqueira Falcão**
(Universidade Federal do Rio de Janeiro/UFRJ)

APROVADO

Prof. Dr. **Rubén Augusto Romero Lázaro**
(Universidade Estadual Paulista Júlio de Mesquita Filho/UNESP - campus Ilha Solteira)

APROVADO

Prof. Dr. **Sérgio Luís Haffner**
(Universidade Federal do Rio Grande do Sul/UFRGS)

APROVADO

Prof. Dr. **Alysson Machado Costa**
(Instituto de Ciências Matemáticas e de Computação/USP)

APROVADO

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica:
Prof. Titular **Denis Vinicius Coury**

Presidente da Comissão de Pós-Graduação:
Prof. Associado **Paulo Cesar Lima Segantine**

À minha querida mãe e ao meu saudoso mas sempre querido pai. Aos meus irmãos Jocimar, Adim e Assis pela amizade e amor.

A Lidianne Lima Santos Sousa por ser presente em minha vida e ao meu querido filhinho Caio.

Agradecimentos

- Ao Professor Dr. Eduardo Nobuhiro Asada pela excelente orientação, compreensão e ajuda durante a elaboração deste trabalho.
- Ao pessoal do LASEE: Todos os Filipes, Eduardo, Wellington, Guilherme, pelas trocas de ideias, pelo apoio e principalmente pela amizade.
- Aos inestimados amigos Ricardo de Andrade L. Rabêlo e Guilherme Enéas pela amizade e incontáveis ajudas.
- Aos amigos Hermes, José Valdemir, Igo Feliciano, Ronildo e outros da colônia piauiense em São Carlos que muito me ajudaram a suportar a saudade de casa e de minha família.
- Aos meus amigos Etienne, Patrick, Gustavo, Alexandre e Dudu Silvestre pela amizade e pelos muitos agradabilíssimos churrascos.
- A minha amada e muito querida esposa e grande amiga Lidiane por ter sido forte para suportar os problemas e ainda me dar suporte de que tanto preciso.
- A todos os colegas, professores e funcionários do Departamento de Engenharia Elétrica da EESC/USP pela colaboração.
- A todas as comunidades de software livre que de alguma forma contribuíram para o desenvolvimento, implementação e documentação deste trabalho. Para citar apenas algumas: Linux, Fedora, KDE, GNOME, Kate, GCC, MPICH, Scilab, TeX Live, Dia, Inkscape.
- À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES , e à Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP pelo apoio financeiro, equipamentos e materiais concedidos.

Resumo

SOUSA, Aldir Silva. **Desenvolvimento de Modelos e Algoritmos Sequenciais e Paralelos para o Planejamento da Expansão de Sistemas de Transmissão de Energia Elétrica**. 115 f. Tese (Doutorado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Paulo, 2012.

O principal objetivo deste estudo é propor uma nova metodologia para lidar com o problema de Planejamento da Expansão de Redes de Transmissão de Energia Elétrica com Múltiplos Cenários de Geração (PERTEEG). Com a metodologia proposta neste trabalho almeja-se construir planos de expansão de redes de transmissão de energia elétrica que sejam capazes de, no menor custo de investimento possível, satisfazer às novas exigências dos sistemas elétricos modernos, tais como construção de redes de transmissão livres de congestionamento e robustas à incerteza em relação aos cenários de geração futuros. Através de estudos realizados na literatura do problema, verificou-se que novos modelos e metodologias de abordagem do PERTEEG se fazem necessários. Ao se modelar o PERTEEG visando construir redes de transmissão que contornem as incertezas em relação aos cenários de geração futuros e concomitantemente minimizar o custo de investimento para a expansão do sistema, o planejador se depara com um problema de otimização multiobjetivo. Existem na literatura da pesquisa operacional diversos algoritmos que visam lidar com problemas multiobjetivos. Nesta tese, foram aplicados dois desses algoritmos: *Nondominated Sorting Genetic Algorithms-II* (NSGA-II) e *SPEA2:Strength Pareto Evolutionary Algorithm* (SPEA2).

Em primeira análise, se destacou uma das maiores dificuldade de lidar com o PERTEEG, a saber, o esforço computacional elevado. Por isso, vislumbrou-se que uma possível solução para contornar esta dificuldade esteja na computação paralela. Para se confirmar esta suspeita, nesta tese foram implementadas versões paralelas dos algoritmos sequenciais testados. A qualidade das soluções encontradas pelos algoritmos foram bastante superiores às soluções encontradas pelos algoritmos sequenciais.

Neste trabalho também será mostrado que as soluções ótimas clássicas considerando somente o objetivo de mínimo custo são incapazes de atender às novas necessidades dos sistemas elétricos de potência.

Testes computacionais foram realizados e analisados neste trabalho. Considerando as metodologias conhecidas na literatura para medição da qualidade das soluções encontradas por algoritmos multiobjetivo, se pode afirmar de que a proposta de abordagem do problema de PERTEEG pode ser viável tanto do ponto de vista de engenharia como do ponto de vista da computação

matemática.

Palavras-chave: Planejamento de sistemas de transmissão, Otimização, Metaheurísticas, Algoritmos Multiobjetivos, Múltiplos cenários de geração, Programação paralela.

Abstract

SOUSA, Aldir Silva. **Development of Mathematical Models, Sequential and Parallel Algorithms for Transmission Expansion Planning** . 115 p. Thesis (Doctorate Degree) - Engineering School of São Carlos, University of São Paulo, São Paulo, 2010.

The main objective of this study is to propose a new methodology to deal with the long-term transmission system expansion planning with multiple generation dispatch scenarios problem (TEP-MDG). With the methodology proposed in this thesis we aim to build expansion plans with minimum investment cost and also capable of meeting the new demands of modern electrical systems, such as uncertainty about the future generation scenarios and congestion in the transmission systems. By modeling the TEP-MDG aiming to build transmission networks that circumvent the uncertainties regarding the future generation scenarios and simultaneously minimize the cost of investment for transmission networks expansion, the planner faces a multiobjective optimization problem. One can find various algorithms that aim to deal with multiobjective problems in the literature of operations research. In this thesis, we apply two of these algorithms: *Nondominated Sorting Genetic Algorithms-II* (NSGA-II) and *SPEA2:Strength Pareto Evolutionary Algorithm* (SPEA2).

In a first analysis, we have found that the most critical issue with the TEP-MOG is the high computational demand. Therefore, in order to circumvent this difficulty we have implemented parallel versions of the sequential algorithms tested. In performed tests, the parallel algorithms have found solutions of superior quality than the solutions found by the sequential algorithms.

In this thesis we also show that optimal solutions considering only the classical least cost objective are unable to meet the electric power systems new demands.

Tests have been performed and analyzed in this work. By considering the methods known in the literature convinced to measure the quality of solutions found by multiobjective algorithms, we concluded that the proposed approach to TEP-MDG may be feasible from the point of view of both engineering and computational mathematics.

Palavras-chave: Transmission systems expansion planning, Optimization, Metaheuristics, Evolutionary Algorithms, Multiple Generation Dispatch Scenarios, Parallel Programming.

Sumário

Lista de Símbolos e Abreviaturas	xvii
Lista de Figuras	xix
Lista de Tabelas	xxi
1 Introdução	1
1.1 Importância do Planejamento dos Sistemas Elétricos	1
2 Revisão Bibliográfica	5
2.1 Modelos Matemáticos	8
2.1.1 Modelos Estáticos	8
2.1.2 Modelo Híbrido Não-Linear	11
2.1.3 Modelos Para o Planejamento Multiestágio	11
3 Planejamento da Expansão Considerando Múltiplos Cenários de Geração	15
3.1 Metodologia Proposta	16
3.2 Modelagem Com Múltiplos Cenários de Geração	19
3.3 Geração dos Cenários	20
3.3.1 Sistema de Seis Barras	20
3.3.2 Sistema IEEE de 24 barras	21
3.3.3 Sistema Sul Brasileiro	22
4 Otimização Multiobjetivo	23
4.1 Conceitos e Definições Básicas	23
4.1.1 Dominância	23
4.1.2 Otimalidade de Pareto	24
4.1.3 Ponto Ideal	25
4.1.4 Níveis de Não-Dominância	26
4.2 Algoritmos Para Otimização Multiobjetivo	27
4.3 Algoritmos Genéticos	28
4.3.1 Terminologia	28
4.3.2 <i>Vector Evaluated Genetic Algorithm</i>	29
4.3.3 <i>Multi-Objective Genetic Algorithm</i>	29
4.3.4 <i>Niched Pareto Genetic Algorithm</i>	30
4.3.5 <i>Nondominated Sorting Genetic Algorithms</i>	30
4.3.6 <i>Strength Pareto Evolutionary Algorithm</i>	31

4.3.7	<i>SPEA2:Strength Pareto Evolutionary Algorithm</i>	31
4.3.8	<i>Pareto Archived Evolution Strategy</i>	31
4.3.9	<i>Nondominated Sorting Genetic Algorithms-II</i>	32
4.4	Busca Tabu Para Otimização Multiobjetivo	32
5	Introdução à Computação Paralela	35
5.1	Classificação de Arquiteturas Paralelas	35
5.1.1	Taxonomia de Flynn	35
5.1.2	Arquiteturas SMP	37
5.1.3	Acesso Não-Uniforme à Memória	38
5.1.4	Aglomerado de Computadores	38
5.2	Modelos de Paralelização de Algoritmos Evolutivos Multiobjetivos	38
5.2.1	Cooperação Paralela Autônoma	39
5.2.2	Paralelização Intra-Algoritmo Independente do Problema	41
5.2.3	Paralelização Intra-Algoritmo Dependente do Problema	42
6	Aplicação de Algoritmos Multiobjetivo ao PERTEE	47
6.1	Detalhamento dos AEMOs Sequenciais	47
6.1.1	NSGA-II Aplicado ao PERTEE Com Múltiplos Cenários de Geração	47
6.1.2	SPEA2 Aplicado ao PERTEE Multicenário	53
6.2	Paralelização dos Algoritmos NSGA-II e SPEA2	56
6.2.1	Paralelização do Algoritmo NSGA-II	56
6.2.2	Paralelização do Algoritmo SPEA2	57
6.3	Técnicas para Avaliação de Desempenho	57
6.3.1	Diminuição de Tempo de Resposta e Eficiência	57
6.3.2	Métricas Para Avaliar a Aproximação da Fronteira Pareto-Ótima	59
7	Testes e Resultados Considerando Algoritmos Sequenciais	63
7.1	Testes Com o Algoritmo NSGA-II	63
7.1.1	Sistema de (Garver, 1970)	63
7.1.2	Sistema IEEE 24 Barras	64
7.1.3	Sistema Sul Brasileiro	65
7.2	Testes Com o Algoritmo SPEA2	65
7.2.1	Sistema IEEE 24 Barras	65
7.2.2	Sistema Sul Brasileiro	66
7.3	Análise dos Resultados	67
7.3.1	Sistema de Garver	67
7.3.2	Sistema IEEE de 24 barras	68
7.3.3	Sistema Sul Brasileiro	68
7.3.4	Análise Comparativa Entre os Algoritmos Testados	69
8	Testes e Resultados Considerando os Algoritmos Paralelos	73
8.1	Resultados Computacionais Considerando o Algoritmo NSGA-II	73
8.1.1	Sistema IEEE 24 Barras	73

8.1.2	Sistema Sul Brasileiro	74
8.2	Resultados Computacionais Considerando o Algoritmo SPEA2	76
8.2.1	Sistema IEEE 24 Barras	76
8.2.2	Sistema Sul Brasileiro	77
8.3	Análise dos Resultados	78
8.4	Conclusões Parciais	83
9	Conclusões e Trabalhos Futuros	85
A	Cluster Utilizado	87
A.1	Interface de Passagem de Mensagens	87
A.1.1	Processos	88
A.1.2	Comunicadores	88
A.1.3	Mensagens	88
A.1.4	Tipos de Dados	88
A.1.5	Comunicações Ponto-a-ponto	89
A.1.6	Comunicações Coletivas	90
B	Técnica Usada Para Simulação das Variáveis Probabilísticas Contínuas	95
B.1	Método da Transformada Inversa	95
B.2	Método da Aceitação-Rejeição	95
B.3	Variável Com Distribuição <i>Gamma</i>	96
B.4	Variável Com Distribuição <i>Beta</i>	97
C	Dados dos Sistemas Elétricos Testados	101
C.1	Sistema de Garver 6 Barras	101
C.2	Dados do Sistema Sul Brasileiro	102
C.3	Dados do Sistema IEEE 24 Barras	104
	Referências Bibliográficas	107

Lista de Símbolos e Abreviaturas

SEP Sistema Elétrico de Potência

AHCs Algoritmos Heurísticos Construtivos

B&B *Branch-and-Bound*

CC Corrente Contínua

IS Índice de Sensibilidade

LCK Lei das Correntes de Kirchhoff

LTK Lei das Tensões de Kirchhoff

PL Programação Linear

PPL Problema de Programação Linear

PPLs Problemas de Programação Linear

PLIM Programação Linear Inteiro Misto

PNLIM Programação Não-Linear Inteiro Misto

PERTEE Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica

PERTEEG Planejamento da Expansão de Redes de Transmissão de Energia Elétrica com Múltiplos Cenários de Geração

CNPE Conselho Nacional de Política Energética

MME Ministério de Minas e Energia

ANEEL Agência Nacional de Energia Elétrica

EPE Empresa de Pesquisa Energética

CCEE Câmara de Comercialização de Energia Elétrica

CMSE Comitê de Monitoramento do Setor Elétrico

ONS Operador Nacional do Sistema Elétrico

AGs Algoritmos Genéticos

AG Algoritmo Genético

PMOs Problemas Multiobjetivo

VEGA *Vector Evaluated Genetic Algorithm*

AEMO Algoritmo Evolutivo Multiobjetivo

MOGA *Multi-Objective Genetic Algorithm*

NPGA *Niched Pareto Genetic Algorithm*

NSGA *Nondominated Sorting Genetic Algorithms*

NSGA-II *Nondominated Sorting Genetic Algorithms-II*

pNSGA-II NSGA-II Paralelo

SPEA *Strength Pareto Evolutionary Algorithm*

SPEA2 *SPEA2:Strength Pareto Evolutionary Algorithm*

pSPEA2 *SPEA2 Paralelo*

PAES *Pareto Archived Evolution Strategy*

IEEE24Bar Sistema Teste IEEE de 24 Barras

SULBr Sistema Sul Brasileiro

Garver Sistema de Seis Barras com Redespacho da Geração

MPI Message Passing Interface

Lista de Figuras

3.1	Função de distribuição de probabilidade beta	18
4.1	Fronteira Pareto-ótima	25
4.2	Ilustração da distribuição das soluções na fronteira Pareto-ótima.	26
4.3	Classificação das soluções em níveis de não-dominância.	27
5.1	Modelo SISD	36
5.2	Modelo SIMD	36
5.3	Modelo MIMD	37
5.4	Arquitetura SMP.	37
5.5	Arquitetura NUMA.	38
5.6	Subpopulações cooperativas.	40
5.7	Abordagem com múltiplos inícios.	40
5.8	Modelo administrador - trabalhador.	41
5.9	Ganho de desempenho em relação ao número de ilhas.	45
5.10	Diminuição do desempenho do modelo hibridizado à medida que aumenta n_{ipm}	45
5.11	Diminuição do número de ilhas em que ocorre o menor esforço esperado à medida que aumenta n_{ipm}	46
6.1	Ilustração do algoritmo NSGA-II.	48
6.2	Cálculo de distância de aglomeração.	49
6.3	Um sistema fictício de quatro barras.	50
6.4	Ilustração da recombinação de um ponto.	52
6.5	Soluções má distribuídas ao longo da fronteira Pareto-ótima.	54
6.6	Indivíduo a ser transferido entre processos.	56
6.7	Vetor representando o indivíduo a ser transferido entre processos.	56
6.8	Hipervolume formado pelas soluções não-dominadas.	60
6.9	Ilustração dos hipercubos.	61
7.1	NSGA-II: Fronteira Pareto-ótima obtida para o sistema Sistema de Seis Barras com Redespacho da Geração (Garver).	64
7.2	NSGA-II: Fronteira Pareto-ótima obtida para o sistema Sistema Teste IEEE de 24 Barras (IEEE24Bar).	64
7.3	NSGA-II: Fronteira Pareto-ótima obtida para o sistema Sistema Sul Brasileiro (SULBr).	65
7.4	SPEA2: Fronteira Pareto-ótima obtida para o sistema IEEE24Bar.	66

7.5	SPEA2: Fronteira Pareto-ótima obtida para o sistema SulBr.	67
7.6	Comparação entre NSGA-II e SPEA2 para o sistema IEEE24Bar.	69
7.7	Comparação entre NSGA-II e SPEA2 para o sistema SULBr.	70
8.1	NSGA-II Paralelo (pNSGA-II): Fronteiras do sistema IEEE24Bar.	73
8.2	pNSGA-II: relação entre o aumento do número de processadores e a diminuição do valor do limitante superior para o sistema IEEE24Bar.	74
8.3	pNSGA-II: Fronteiras do sistema SULBr.	75
8.4	pNSGA-II: relação entre o aumento do número de processadores e a diminuição do valor do limitante superior para o sistema SULBr.	76
8.5	<i>SPEA2 Paralelo</i> (pSPEA2): Fronteiras do sistema IEEE24Bar.	76
8.6	pSPEA2: relação entre o aumento do número de processadores e a diminuição do valor do limitante superior para o sistema IEEE24Bar.	77
8.7	pSPEA2: Fronteiras do sistema SULBr.	77
8.8	pSPEA2: relação entre o aumento do número de processadores e a diminuição do valor do limitante superior para o sistema SULBr.	78
8.9	Soluções não-dominadas encontradas para o sistema IEEE24Bar.	79
8.10	Soluções não-dominadas encontradas para o sistema SULBr.	79
8.11	Convergência da métrica distância geracional em relação ao número de processadores para o sistema IEEE24Bar.	80
8.12	Convergência da métrica espalhamento em relação ao número de processadores para o sistema IEEE24Bar.	80
8.13	Convergência da métrica hipervolume em relação ao número de processadores para o sistema IEEE24Bar.	81
8.14	Convergência da métrica distância geracional em relação ao número de processadores para o sistema SULBr.	82
8.15	Convergência da métrica espalhamento em relação ao número de processadores para o sistema SULBr.	82
8.16	Convergência da métrica hipervolume em relação ao número de processadores para o sistema SULBr.	83
A.1	Ilustração da operação de coleta.	91
A.2	Ilustração da operação de distribuição.	92
B.1	<i>Geração de variáveis aleatórias com função de distribuição de probabilidade beta.</i> .	98
B.2	Sobreposição das figuras B.1(a) e B.1(b).	99

Lista de Tabelas

3.1	Distribuição dos cenários para o sistema Garver com $N = 573$ e $N = 1.000.000$. . .	21
3.2	Distribuição dos cenários para o sistema IEEE24Bar com $N = 1.689$ e $N = 1.000.000$. . .	22
3.3	Distribuição dos cenários para o sistema SULBr com $N = 2.087$ e $N = 1.000.000$. . .	22
7.1	Cenários não atendidos pela configuração \$110,00 com $N = 573$ e $N = 1.000.000$. . .	67
7.2	Cenários não atendidos pela configuração \$220,00 com $N = 573$ e $N = 1.000.000$. . .	67
7.3	Cenários não atendidos pela configuração \$ 152,00 $\times 10^4$ com $N = 1.689$ e $N = 1.000.000$	68
7.4	SPEA2: Cenários não atendidos pela configuração \$1.062,00 $\times 10^4$ com $N = 1.689$ e $N = 1.000.000$	68
7.5	NSGA-II: Cenários não atendidos pela configuração \$1.057,00 $\times 10^4$ com $N = 1.689$ e $N = 1.000.000$	68
7.6	Cenários não atendidos pela configuração U\$ 70.289,00 $\times 10^3$ com $N = 2.087$ e $N = 1.000.000$	68
7.7	SPEA2: Cenários não atendidos pela configuração U\$235.681,00 $\times 10^3$ com $N = 2.087$ e $N = 1.000.000$	69
7.8	Dados a respeito do desempenho do algoritmo SPEA2	70
7.9	Dados a respeito do desempenho do algoritmo NSGA-II	70
8.1	pNSGA-II: Diminuição do valor do limitante superior à medida que aumenta o número de processadores para o sistema IEEE24Bar.	74
8.2	pNSGA-II: Diminuição do valor do limitante superior à medida que aumenta o número de processadores para o sistema SULBr.	75
8.3	pSPEA2: Diminuição do valor do limitante superior à medida que aumenta o número de processadores para o sistema IEEE24Bar.	77
8.4	pSPEA2: Diminuição do valor do limitante superior à medida que aumenta o número de processadores para o sistema SULBr.	78
8.5	Métricas para o sistema IEEE24Bar.	80
8.6	Métricas para o sistema SULBr.	81
A.1	Tipos Message Passing Interface (MPI) predefinidos e seus respectivos correspondentes em C e Fortran.	89
A.2	Operadores de redução pré-definidos pelo padrão MPI	93
C.1	Garver - Dados de barras	101
C.2	Garver - Dados das linhas	101

C.3 SULBr - Dados de barras 102

C.4 SULBr - Dados das linhas 103

C.5 IEEE24Bar - Dados de barras 104

C.6 IEEE24Bar - Dados das linhas 105

Lista de Algoritmos

1	Definindo a amostra	20
2	Esboço de um algoritmo genético	29
3	Principais passos da busca tabu	32
4	Algoritmo NSGA-II	48
5	Função de atribuição de distância de aglomeração	49
6	Gera população factível	52
7	Esboço do algoritmo SPEA2	55
8	Esboço do algoritmo pSPEA2	58

Capítulo 1

Introdução

O avanço tecnológico contínuo e a crescente necessidade de uma maior oferta de energia elétrica têm impulsionado uma expansão progressiva do Sistema Elétrico de Potência (SEP). O SEP é composto basicamente de três grandes setores: a geração, a transmissão e a distribuição de energia elétrica, os quais são planejados e operados no intuito de proporcionar qualidade no fornecimento e suprir a demanda de energia elétrica aos seus consumidores.

Os geradores são responsáveis pela produção da energia elétrica, os consumidores demandam a energia elétrica e os sistemas de transmissão e distribuição permitem transferir a energia elétrica dos geradores aos consumidores. Para que os geradores e consumidores de energia possam exercer suas atividades de compra e venda de energia, existem agentes incumbidos de controlar o adequado funcionamento e expansão do sistema elétrico. No Brasil, são sete os agentes institucionais do setor elétrico: Conselho Nacional de Política Energética (CNPE), Ministério de Minas e Energia (MME), Agência Nacional de Energia Elétrica (ANEEL), Empresa de Pesquisa Energética (EPE), Câmara de Comercialização de Energia Elétrica (CCEE), Comitê de Monitoramento do Setor Elétrico (CMSE) e Operador Nacional do Sistema Elétrico (ONS).

Considerando que algumas atividades do setor elétrico podem ser desenvolvidas em ambientes competitivos, na década de 1970 foi iniciado nos Estados Unidos o movimento geral em favor da reestruturação do setor elétrico. Neste momento, foram definidas duas reformas: a introdução da competição e a imposição de preços que refletissem os custos reais de acordo com o uso. Estas reformas tinham como principal objetivo o estabelecimento da concorrência entre os diversos provedores dos mesmos serviços no intuito de estimular a existência de um fornecimento eficiente com os menores preços possíveis.

O Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica (PERTEE) tem um papel importante em prover acesso a todos os participantes de um mercado competitivo para geração e distribuição de energia elétrica. Neste trabalho, lidaremos com o Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica. O PERTEE visa expandir e fortalecer uma rede de transmissão para que esta possa suportar de maneira eficiente a demanda crescente de energia elétrica, sujeito a restrições econômicas, técnicas e ambientais.

1.1 Importância do Planejamento dos Sistemas Elétricos

O planejamento do setor elétrico é fundamental para garantir a continuidade do fornecimento da energia elétrica necessária para o desenvolvimento de um país. A falta de planejamento eficiente do setor elétrico pode acarretar diversas consequências negativas e imediatas, como elevação dos

custos de produção, transmissão e distribuição da energia elétrica e diminuição na qualidade dos serviços e confiabilidade do sistema elétrico. Com a recente preocupação por entidades governamentais e não-governamentais com a degradação de recursos naturais, o setor elétrico se vê desafiado em proporcionar energia de qualidade com os menores preços e menor impacto ambiental possível. Esses são objetivos impossíveis de serem alcançados sem um planejamento eficiente do setor elétrico.

Nos sistemas elétricos tradicionais, a operação e o planejamento da rede elétrica são realizados por entidades organizadas numa estrutura vertical. O processo de desverticalização por que passaram alguns sistemas elétricos teve como pontos principais a implantação de modelos comerciais competitivos, a garantia do livre acesso à rede e a redução do papel do Estado no setor. Um movimento comum entre os sistemas modernos descentralizados é a criação de instituições especializadas para executar as funções de regulação, planejamento da expansão, operação e financiamento do setor.

Tradicionalmente, o planejamento da expansão das redes de transmissão tem lidado apenas com as restrições técnicas, de confiabilidade e o objetivo de investimento de mínimo custo. Com a descentralização do setor elétrico, muitas outras restrições e objetivos têm sido abordados, aumentando consideravelmente a complexidade do problema. Os sistemas modernos estão cada vez mais voltados para a implantação de um mercado de energia. Porém, redes de transmissão não planejadas com o fim de promover a competição entre os participantes do mercado de energia podem não satisfazer as exigências do mercado devido ao congestionamento que pode surgir e não permitir a programação de alguns despachos de geração. Como os despachos de geração são decorrentes do processo de negociação entre vendedores e compradores de energia elétrica, os despachos horários e diários possuem grande incerteza, uma vez que existe um número muito grande de cenários factíveis.

O congestionamento da rede de transmissão é um entrave à livre concorrência do mercado de energia. Desta forma, providências no planejamento de curto prazo devem ser tomadas para eliminar as restrições que afetem a livre concorrência no mercado de energia atual cabendo ao planejamento de longo prazo a incumbência de eliminar ou mitigar a possibilidade de congestionamento da rede de transmissão, dado o conjunto de todos os cenários de geração possíveis.

O estudo do PERTEE considerando as exigências dos mercados de energia requer uma nova forma de abordagem do problema no intuito de fornecer uma adequação tanto nos modelos matemáticos, como nos algoritmos de solução, conforme observado nas referências (Baldick; Kahn, 1993, Bresesti et al., 2009, Buygi et al., 2004a, Fan; Cheng; Yao, 2009, Fang; Hill, 2002, Haffner, 2000, Latorre et al., 2003, Léautier; Thelen, 2009).

Uma vez que as restrições de transmissão afetam a livre concorrência e, portanto, a eficiência do mercado de energia (as restrições de transmissão afetam o desenvolvimento deste mercado por afetar a livre concorrência), é importante determinar o custo da eliminação completa do congestionamento, para se ter uma medida do custo de adequação da rede atual às condições ideais de livre concorrência no futuro. Ressaltando que estudos atuais indicam que os custos com a rede de transmissão representam apenas entre 6% a 10% do custo de geração da maioria dos sistemas (Léautier; Thelen, 2009), um investimento em expansão da transmissão pode ser justificado se a economia resultante da redução do custo total do congestionamento for maior ou

igual ao custo do investimento com a expansão da rede.

Planos de expansão que impeçam o congestionamento da rede de transmissão para qualquer cenário de geração são geralmente muito caros, e podem ser economicamente inviáveis. Do ponto de vista computacional, é muito custoso encontrar planos de expansão que atendam todo e qualquer cenário de geração, uma vez que o número de cenários factíveis pode ser infinito. Dada a importância do tema para o mercado de energia elétrica, propõe-se nesta tese uma nova abordagem do PERTEE visando o atendimento de múltiplos cenários de geração, porém sem o encargo computacional de gerar configurações factíveis para todos os cenários possíveis. Além disso, nesta abordagem visa-se diminuir as chances de congestionamento da rede de transmissão sem um aumento excessivo nos custos de expansão da rede de transmissão.

De maneira geral se divide o planejamento da expansão de redes de transmissão de energia elétrica em três escalas temporais distintas, conforme explicado por (Barros, 2004):

- *Horizonte de Longo Prazo (de 6 a 25 anos)*: Parte do princípio de que as decisões referentes à execução de grandes obras de geração só acarretarão resultados efetivos em um prazo de seis a oito anos. Isto implica a necessidade de se analisar alternativas de geração e os respectivos sistemas de transmissão;
- *Horizonte de Médio Prazo (de 6 a 10 anos)*: Dentro deste horizonte são estudadas possíveis alternativas, cujos prazos de maturação dos investimentos são da ordem de três a cinco anos, eventuais alterações no cronograma de construção de determinadas usinas e/ou modificações na previsão de mercado;
- *Horizonte de Curto Prazo (até 6 anos)*: Neste horizonte o problema a resolver consiste em tomar decisões concernentes ao comissionamento e à construção de obras de transmissão. Leva-se em conta que é necessário definir com maior precisão todos os parâmetros elétricos pertinentes à rede de transmissão, bem como as datas adequadas para implantação das novas obras.

Nesta tese, lida-se com o planejamento a longo prazo.

Lidar com o planejamento da expansão das redes de transmissão considerando múltiplos cenários de geração incorre em um desafio multiobjetivo. Nesta tese são almejados dois objetivos: minimizar os custos de expansão da rede de transmissão e maximizar o atendimento aos incontáveis cenários de geração possíveis decorrentes da livre concorrência do setor energético. Teorias e técnicas para lidar com problemas com mais de um objetivo serão detalhadas e exploradas no intuito de lidar com o problema objeto desta tese. Em uma primeira análise, nos convencemos que um dos maiores gargalos para lidar com o problema aqui estudado é a demanda por esforço computacional muito elevado. Assim se justifica a aplicação de teorias e técnicas da computação paralela no intuito de contornar o problema levantado.

Além deste capítulo, este trabalho está organizado da seguinte forma: no Capítulo 2, será apresentado o estado da arte do problema aqui estudado e os principais modelos matemáticos utilizados pelos pesquisadores. No Capítulo 3 serão apresentadas as definições e justificativas relacionadas ao planejamento da expansão de redes de transmissão de energia elétrica considerando múltiplos cenários de geração. Além disso, definir-se-á a metodologia proposta neste trabalho para

lidar com o referido problema. No Capítulo 4, serão introduzidos os conceitos e principais algoritmos multiobjetivos. No Capítulo 5, serão apresentadas as principais arquiteturas de computadores paralelos e alguns dos mais modelos de paralelização de algoritmos evolutivos multiobjetivos. No Capítulo 6, serão detalhadas as adequações realizadas nos algoritmos multiobjetivos NSGA-II e SPEA2 no intuito de aplicá-los ao planejamento considerando múltiplos cenários de geração. Além disso, definir-se-ão algumas métricas a serem utilizadas para comparações entre as soluções encontradas pelos algoritmos sequenciais e paralelos. No Capítulo 7, serão detalhados os testes realizados considerando a computação sequencial. Além disso, serão realizadas comparações entre os algoritmos implementados e será realizada uma análise da metodologia proposta neste trabalho. No Capítulo 8, serão detalhados os testes realizados considerando os algoritmos paralelos implementados. Além disso, serão realizadas comparações entre os algoritmos implementados. No Capítulo 9, serão apresentadas algumas considerações finais e sugeridas propostas para continuação deste trabalho.

No Apêndice A, será detalhado o computador paralelo utilizado durante o desenvolvimento deste trabalho, e será realizada uma introdução à biblioteca MPI. No Apêndice B, serão descritos os métodos usados para gerar as variáveis aleatórias usadas ao longo deste trabalho. No Apêndice C mostramos os dados dos sistemas utilizados nesta tese.

Capítulo 2

Revisão Bibliográfica

Este levantamento bibliográfico apresenta alguns dos trabalhos em destaque sobre a temática proposta nesse projeto de pesquisa.

O pesquisador Knight (1960) foi um dos precursores no estudo do PERTEE. A grande contribuição do seu trabalho foi mostrar a importância do planejamento da expansão das redes elétricas. Dez anos após a publicação deste trabalho, surgiu uma das primeiras heurísticas bem sucedidas para solução do PERTEE. Em (Garver, 1970), foi mostrado que o PERTEE poderia ser modelado como um problema de transportes ao se considerar apenas a Lei das Correntes de Kirchhoff (LCK). O método consiste em, partindo da configuração inicial do sistema, adicionar linhas ao sistema corrente guiado por um Índice de Sensibilidade (IS) até se encontrar uma solução factível ao modelo proposto naquele trabalho. O trabalho de Garver (1970) teve basicamente três grandes contribuições, listadas abaixo:

- A prova de que ao se considerar somente a LCK, o problema poderia ser modelado como um problema de transportes.
- O índice de sensibilidade proposto neste trabalho leva a uma solução factível muito rapidamente (em trabalhos posteriores, verificou-se que isso ocorre mesmo quando se consideram sistemas de grande porte). Isto é de grande relevância tanto para a época, quando o desempenho dos computadores digitais era muito aquém em relação aos hodiernos; como para os dias atuais, uma vez que índices de sensibilidade são recorrentemente utilizados nas meta-heurísticas hoje aplicadas ao problema.
- A prova de que ao se estar diante de uma solução factível, a solução do Problema de Programação Linear (PPL) terá função objetivo igual a zero.

Em (Villasana; Garver; Salon, 1985) foi realizado um pequeno, mas muito importante aperfeiçoamento do trabalho de Garver (1970). Villasana; Garver; Salon (1985) sugeriram aplicar a Lei das Tensões de Kirchhoff (LTK) somente às linhas já existentes na configuração base do sistema. O método consiste em modelar a configuração inicial do sistema considerando as duas leis de Kirchhoff, e adicionar novas linhas de forma construtiva até a obtenção de uma solução factível. Fazendo isso, e ditando que as novas linhas adicionadas seguissem a LTK, quando da convergência do algoritmo, obter-se-ia uma configuração factível ao modelo de Corrente Contínua (CC), doravante chamado apenas de modelo cc. Assim, resolvendo apenas Problemas de Programação Linear (PPLs), com este método pode-se obter uma solução factível para um modelo não-linear.

Os trabalhos (Monticelli et al., 1982) e (Pereira; Pinto, 1985) apresentam novos índices de sensibilidade e uma ligeira modificação no modelo de Garver (1970). Desta modificação, surge um modelo com menos restrições e, por consequência, de resolução mais rápida. Este modelo é o mais usado nas abordagens do PERTEE por meio de meta-heurísticas.

Lee; Hicks; Hnyilicza (1974) estão entre os primeiros pesquisadores a lidar com o PERTEE por meio de métodos exatos. No artigo (Lee; Hicks; Hnyilicza, 1974), o algoritmo *Branch-and-Bound*, inicialmente publicado em (Land; Doig, 1960), foi aplicado para resolver uma série de programas inteiros zero-um. O trabalho de Lee; Hicks; Hnyilicza (1974) foi seguido pelos trabalhos (Haffner et al., 2000, Levi, 1995, Romero; Monticelli, 1994a) e (Asada et al., 2005), dentre outros.

Com a publicação do artigo de Benders (1962), surgiu um novo método de solução de problemas de programação linear com restrições de integralidade. O trabalho de Pereira et al. (1985) foi um dos primeiros a aplicar o método proposto por Benders (1962) ao PERTEE. Cabe ressaltar que no trabalho de Pereira et al. (1985), a decomposição é empregada para separar o problema de seleção dos investimentos (inteiro) do problema de operação da rede (linear, pois os investimentos já foram selecionados no problema de investimentos). Portanto, sob uma perspectiva de sistemas centralizados, que era comum à época. Seguindo esta linha de pesquisa, Romero; Monticelli (1994b) encontraram pela primeira vez a solução ótima do modelo cc para o sistema sul brasileiro através de um método exato ao aplicar o método de decomposição de Benders a este sistema.

No entanto, os métodos exatos apresentam grande dificuldade de convergência quando aplicados a sistemas de grande porte. Daí, a aplicação das meta-heurísticas teve grande impacto na linha de pesquisa tratada neste trabalho. Com as meta-heurísticas, restrições antes relaxadas pelos métodos heurísticos e exatos, passaram a ser consideradas de forma bem-sucedida.

Uma vez que os métodos heurísticos apresentam baixo custo computacional, porém configurações muito longe da solução ótima, e os métodos exatos apresentam altíssimos custos computacionais e dificuldade de convergência quando se lida com sistemas de grande porte, a aplicação das meta-heurísticas passou a ser um rumo natural para os pesquisadores do PERTEE. Um dos primeiros trabalhos aplicando meta-heurísticas ao PERTEE foi o artigo (Romero; Gallego; Monticelli, 1996). Naquele trabalho, a meta-heurística *simulated annealing* foi abordada para o planejamento estático. Outras meta-heurísticas logo foram aplicadas: algoritmos genéticos (Ceciliano; Nieva, 1999, Gallego; Monticelli; Romero, 1998); busca tabu (Da Silva et al., 2001, Gallego; Romero; Monticelli, 2000); e meta-heurísticas baseadas em inteligência coletiva (Jin et al., 2007).

Além disso, ferramentas de inteligência artificial também têm sido aplicadas pelos pesquisadores do PERTEE. Sistemas *fuzzy* foram aplicados para resolver o PERTEE em (Chung et al., 2003) onde um tomador de decisão *fuzzy* é aplicado para selecionar as topologias geradas por um algoritmo genético em um contexto multiobjetivo onde o custo de investimento, a segurança e o impacto ambiental são os objetivos. Em (Choi; El-Keib; Tran, 2005) o PERTEE é modelado como um problema de programação inteiro-*fuzzy* considerando investimento e segurança. Em (Buygi et al., 2004b) sistemas *fuzzy* são usados para gerenciar riscos em um cenário competitivo do PERTEE. Em (Sousa; Asada, 2011) um tomador de decisão *fuzzy* é usado para evitar a convergência prematura dos Algoritmos Heurísticos Construtivos (AHCs) convencionais.

Pode-se lidar com o PERTEE considerando um planejamento estático ou um planejamento

dinâmico (multiestágio). No problema estático, o planejamento especifica quais equipamentos devem ser adicionados à rede para que esta possa operar adequadamente dentro de um horizonte de planejamento específico. Assim, pode-se dizer que o planejamento estático é um planejamento de um único estágio, ou um planejamento que considera apenas um horizonte de planejamento, e o investimento ocorre no início do horizonte de planejamento. O planejamento multiestágio especifica além dos equipamentos que devem ser adicionados ao sistema, quando estes equipamentos devem ser adicionados. O horizonte de planejamento é dividido em estágios e os equipamentos devem ser adicionados ao sistema em cada estágio do horizonte de planejamento. O investimento é realizado no início de cada estágio (Cedeno; Arora, 2011, Escobar; Gallego; Romero, 2004, Latorre et al., 2003). No planejamento multiestágio, o objetivo é a minimização do valor presente da soma de todos os investimentos realizados durante os anos correspondentes aos períodos simulados (Escobar; Gallego; Romero, 2004, Garcia; Mahey; LeBlanc, 1998, Sum-Im et al., 2009). Os métodos e abordagens supracitados lidaram numa primeira instância com o problema estático.

Na década de 1970, o movimento geral em favor da reestruturação do setor elétrico foi iniciado nos Estados Unidos. Naquele momento, foram definidas duas reformas: a introdução da competição e a imposição de preços que refletissem os custos reais de acordo com o uso (Haffner, 2000, Leme, 2009). Diante deste novo ambiente, há um aumento de interesse sobre questões do planejamento da transmissão em um ambiente competitivo. Os artigos (Buygi et al., 2004b, Choi; El-Keib; Tran, 2005, Choi et al., 2005, Chung et al., 2003, Fang; Hill, 2002) consideraram aspectos importantes resultantes da reestruturação do setor elétrico.

Com a introdução dos mercados de compra e venda de energia baseados em bolsas de valores e contratos bilaterais, novos modelos, restrições e objetivos têm sido abordados. Considerações importantes como a maximização do benefício social (Shrestha; Fonseca, 2004, Torre; Conejo; Contreras, 2008), a abordagem considerando diferentes cenários de geração e demanda (Escobar; Romero; Gallego, 2008, Fang; Hill, 2002) e as condições de confiabilidade e segurança (Choi; Mount; Thomas, 2007, Güler; Gross, 2010, Shariati et al., 2008) têm sido frequentemente abordadas pelos pesquisadores do PERTEE.

Dentre os aspectos críticos que podem afetar a livre concorrência nos mercados de energia, o congestionamento produzido pelas restrições da rede (Bjorndal; Jörnsten, 2007, Fang; David, 1999, Harvey; Hogan; Pope, 1996, Lu et al., 2006) e a incerteza na carga e na geração (Silva Junior et al., 2006, Tor; Guven; Shahidehpour, 2008) são os temas mais abordados. Em (Kumar; Srivastava; Singh, 2005) é apresentada uma revisão bibliográfica sobre o planejamento da expansão da transmissão considerando o congestionamento da rede de transmissão.

Do ponto de vista matemático, a evolução dos aspectos abordados pelos pesquisadores do Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica tem resultado em modelos cada vez mais complexos. O primeiro modelo proposto para PERTEE vastamente utilizado foi o modelo de transportes proposto por Garver (1970). Este modelo foi aperfeiçoado em (Villasana; Garver; Salon, 1985). Tanto o modelo de transportes como o modelo de Villasana; Garver; Salon (1985) são modelos de Programação Linear Inteiro Misto (PLIM). O modelo proposto por Monticelli et al. (1982) é um modelo mais próximo das condições reais de operação do sistema elétrico e trata-se de um modelo de Programação Não-Linear Inteiro Misto (PNLIM). Cada um destes modelos será detalhado na Seção 2.1.

Com a desregulamentação do setor de energia, restrições não determinísticas foram adiciona-

das ao problema (Buygi et al., 2004a, Ehrenmann; Smeers, 2006, Neufville, 2003):

- Incentivar e facilitar a competição entre os participantes do mercado de energia;
- Prover acesso não-discriminatório à geração barata a todos os consumidores e acesso igualitário às cargas a todos geradores;
- Custo esperado de carga não suprida;
- Regras de mercado e políticas governamentais;
- Disponibilidade de geradores, linhas de transmissão e outros equipamentos do sistema;
- Prover uma rede de transmissão robusta a todas as incertezas;

Ao se abordar o PERTEE com essas incertezas, o problema passa a se caracterizar como um problema de programação não-linear inteiro misto e estocástico.

Esta seção não referenciou todos os trabalhos e pesquisadores do PERTEE. Porém, procurou-se listar os principais artigos e métodos de abordagens desta linha de pesquisa destacando sua importância histórica e ordem cronológica.

2.1 Modelos Matemáticos

A solução de problemas reais utilizando as ferramentas computacionais e matemáticas existentes nos dias de hoje parte da modelagem matemática das características físicas e/ou lógicas do problema a ser abordado. A princípio, quanto mais próximo das características reais do problema alvo, maior é a dificuldade de solução do modelo matemático. Nesta seção apresentam-se os principais modelos matemáticos aplicados ao PERTEE, estático e multiestágio.

2.1.1 Modelos Estáticos

Nesta seção serão detalhados os principais modelos estáticos usados na literatura do Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica.

Modelo CC

O modelo cc é considerado o mais apropriado para equacionar problemas de planejamento de sistemas de energia elétrica (Monticelli, 1983, Romero et al., 2007), sendo representado matematicamente como segue:

$$\min v = \sum_{(i,j) \in \Omega} c_{ij} n_{ij} \quad (2.1)$$

sujeito a

$$Sf + g = d \quad (2.2)$$

$$f_{ij} - \gamma_{ij}(n_{ij}^0 + n_{ij})(\theta_i - \theta_j) = 0 \quad \forall (i, j) \in \Omega \quad (2.3)$$

$$|f_{ij}| \leq (n_{ij}^0 + n_{ij}) \bar{f}_{ij} \quad \forall (i, j) \in \Omega \quad (2.4)$$

$$0 \leq g \leq \bar{g} \quad (2.5)$$

$$0 \leq n \leq \bar{n}_{ij} \quad (2.6)$$

$$n_{ij} \text{ inteiro}; f_{ij} \text{ irrestrito } \forall (i, j) \in \Omega; \quad (2.7)$$

$$\theta_j \text{ irrestrito } \forall j \quad (2.8)$$

onde

- v é o custo do investimento;
- c_{ij} é o custo de adição de uma linha no ramo $i - j$;
- S é a matriz de incidência nó-ramo transposta;
- θ_j é o ângulo de tensão na barra j ;
- γ_{ij} é a susceptância da linha adicionada em $i - j$;
- f_{ij} é o fluxo do ramo $i - j$;
- $\bar{f}_{ij}$ é o limite superior de f_{ij} ;
- g é o vetor de gerações;
- $\bar{g}$ é o vetor com o limite superior das gerações;
- d é o vetor com o valor das cargas (demandas) do sistema;
- n_{ij} é o número de linhas que podem ser adicionadas no ramo $i - j$;
- $\bar{n}_{ij}$ é o número máximo de linhas que pode ser adicionadas no ramo $i - j$;
- n_{ij}^0 é número de linhas presentes na topologia base no ramo $i - j$;
- Ω é o conjunto de todos os ramos.

A equação (2.2) representa a conservação de potência em cada nó. Esta restrição corresponde à Lei das Correntes de Kirchhoff. A equação (2.3) modela a Lei das Tensões de Kirchhoff. A equação (2.3) é não-linear devido ao produto $(n_{ij}^0 + n_{ij})(\theta_i - \theta_j)$. As demais restrições estão relacionadas aos limites das variáveis. Este modelo é um modelo de PNLIM.

Modelo de Transportes

O modelo (2.1-2.8) teve as restrições (2.3) relaxadas em (Garver, 1970). Naquele trabalho, Garver (1970) propôs modelar o PERTEE como um modelo de transportes, removendo a não-linearidade do modelo cc, ao se considerar apenas a LCK em todas as linhas. O modelo de transportes pode ser representado pelas seguintes equações.

$$\min v = \sum_{(i,j) \in \Omega} c_{ij} n_{ij} \quad (2.9)$$

sujeito a

$$Sf + g = d \quad (2.10)$$

$$|f_{ij}| \leq (n_{ij}^0 + n_{ij}) \bar{f}_{ij} \quad \forall (i,j) \in \Omega \quad (2.11)$$

$$0 \leq g \leq \bar{g} \quad (2.12)$$

$$0 \leq n \leq \bar{n}_{ij} \quad (2.13)$$

$$n_{ij} \text{ inteiro}; f_{ij} \text{ irrestrito } \forall (i,j) \in \Omega; \quad (2.14)$$

Note que a relaxação proposta por Garver (1970) consiste em apenas remover as restrições relacionadas à LTK (Eq. (2.3) do modelo cc). O modelo (2.9-2.14) é um modelo de PLIM.

O modelo de transportes tem menor complexidade do que o modelo cc, sendo, portanto, de mais fácil resolução. Porém, a solução ótima do modelo de transportes pode não ser factível ao modelo cc, devido à relaxação imposta. Por isso, em (Villasana; Garver; Salon, 1985) foi realizada uma pequena modificação e sugerida uma heurística construtiva para que quando da convergência da heurística, obtivesse-se uma solução factível ao modelo cc. O modelo proposto em (Villasana; Garver; Salon, 1985) é chamado de modelo híbrido, e será detalhado na próxima seção.

Modelo Híbrido

Villasana; Garver; Salon (1985) consideraram que ao se modelar a configuração inicial do sistema (linhas já existentes no sistema) obedecendo às duas leis de Kirchhoff, o modelo seria menos relaxado do que o modelo de transportes e consequentemente a solução ótima do modelo seria menos infactível em relação ao modelo cc. Além disso, considerou-se uma heurística construtiva, onde o sistema com as linhas iniciais mais as linhas adicionadas durante o processo iterativo fosse tomado como a configuração base. Ao se fazer com que todas as linhas da configuração base obedecessem às duas leis de Kirchhoff, quando da convergência desse algoritmo, todas as linhas estariam obedecendo às duas leis de Kirchhoff o que, por definição, corresponde a uma configuração factível ao modelo cc. A grande vantagem desta abordagem é a possibilidade de se obter uma configuração de modelo cc, que é não-linear, por meio da solução de um número determinado, e geralmente pequeno, de PPLs. As Eqs. (2.15-2.23) formulam o modelo híbrido de Villasana; Garver; Salon (1985).

$$\min v = \sum_{(i,j) \in \Omega} c_{ij} n_{ij} \quad (2.15)$$

sujeito a

$$Sf + S_0 f_0 + g = d \quad (2.16)$$

$$f_{ij}^0 - \gamma_{ij} n_{ij}^0 (\theta_i - \theta_j) = 0 \quad \forall (i, j) \in \Omega_1 \quad (2.17)$$

$$|f_{ij}^0| \leq n_{ij}^0 \bar{f}_{ij} \quad \forall (i, j) \in \Omega_1 \quad (2.18)$$

$$|f'_{ij}| \leq n_{ij} \bar{f}_{ij} \quad \forall (i, j) \in \Omega \quad (2.19)$$

$$0 \leq g \leq \bar{g} \quad (2.20)$$

$$0 \leq n_{ij} \leq \bar{n}_{ij} \quad (2.21)$$

$$n_{ij} \text{ inteiro } \forall (i, j); f_{ij} \text{ irrestrito } \forall (i, j); \quad (2.22)$$

$$\theta_j \text{ irrestrito } \forall j \quad (2.23)$$

onde

- S representa a matriz de incidência nó-ramo transposta do sistema elétrico inteiro;
- S_0 representa a matriz de incidência nó-ramo transposta das linhas existentes na configuração base;

- f_0 representa o vetor de fluxos nas linhas existentes na configuração base, cujos elementos são f_{ij}^0 ;
- n_{ij}^0 representa o número de linhas existentes no ramo $i - j$ na configuração base;
- θ_j representa o ângulo de fase da barra j que está ligada ao sistema elétrico na configuração base;
- Ω é o conjunto de todas as linhas possíveis de serem adicionadas ao sistemas;
- Ω_1 é o conjunto das linhas pertencentes à configuração base.

Note que a única modificação em relação ao modelo cc é a remoção da variável n_{ij} da equação (2.3).

2.1.2 Modelo Híbrido Não-Linear

Uma outra forma de se modelar o PERTEE é considerar as restrições relacionadas à LTK (Eq. (2.3) do modelo cc) somente para as linhas existentes na configuração base, e para as linhas candidatas a adição que sejam paralelas a estas. O artigo (Romero; Monticelli, 1994b) foi o primeiro a modelar o PERTEE desta forma. Este modelo é conhecido como modelo híbrido não-linear. O modelo híbrido não-linear pode ser representado matematicamente como segue:

$$\min v = \sum_{(i,j) \in \Omega} c_{ij} n_{ij} \quad (2.24)$$

sujeito a

$$Sf + g = d \quad (2.25)$$

$$f_{ij} - \gamma_{ij}(n_{ij}^0 + n_{ij})(\theta_i - \theta_j) = 0 \quad \forall (i, j) \in \Omega_1 \quad (2.26)$$

$$|f_{ij}| \leq (n_{ij}^0 + n_{ij}) \bar{f}_{ij} \quad \forall (i, j) \in \Omega \quad (2.27)$$

$$0 \leq g \leq \bar{g} \quad (2.28)$$

$$0 \leq n \leq \bar{n}_{ij} \quad (2.29)$$

$$n_{ij} \text{ inteiro}; f_{ij} \text{ irrestrito } \forall (i, j) \in \Omega; \quad (2.30)$$

$$\theta_j \text{ irrestrito } \forall j \quad (2.31)$$

Neste modelo, Ω_1 representa o conjunto de caminhos onde existem linhas na configuração base. O conjunto de restrições Eq. (2.25) e Eq. (2.26) indicam, respectivamente, que o modelo híbrido não-linear deve satisfazer a LCK em todas as barras do sistema e a LTK somente nos ramos existentes na configuração inicial.

Este modelo tem sido pouco utilizado pelos pesquisadores porque resolver o modelo híbrido não-linear tem quase a mesma complexidade que resolver o modelo cc.

2.1.3 Modelos Para o Planejamento Multiestágio

No planejamento multiestágio, além de se determinar a adição de novas linhas ao sistema, deve-se também determinar o melhor momento de adição destas linhas de tal forma que a soma de todos os investimentos realizados ao longo dos anos de planejamento seja a menor possível.

Assim, o planejamento multiestágio pode ser visto como a divisão de um horizonte de planejamento longo em fases de execução. A vantagem de se fazer esta divisão está em eximir o planejador de realizar todo o investimento no plano de expansão de uma só vez. Diante disto, podem-se vislumbrar dois cenários: 1) o planejador não dispõe de todo o investimento necessário para realizar o planejamento a longo prazo (o que é o mais comum). Nesta situação só há necessidade da realização de empréstimos ao início de cada estágio de planejamento, minimizando os juros totais para a realização da expansão. 2) o planejador dispõe de todo o investimento necessário para a realização do planejamento a longo prazo. Neste cenário, é mais vantajoso aplicar o capital necessário para a realização do estágio atual e o restante aplicar em outros rendimentos. Desta forma, maximizam-se os ganhos pelo investimento do capital ainda não aplicado ao plano de expansão.

É importante destacar que cada estágio do planejamento está sujeito às restrições de operação do sistema. Portanto, a cada estágio busca-se obter o menor custo de investimento para manter o sistema em operação, com o objetivo final de minimizar a soma de todos os investimentos realizados ao longo dos anos de planejamento.

Considerando uma taxa anual de desconto I , o cálculo dos valores presentes dos custos de investimentos, para o ano de referência t_0 , com ano inicial t_1 , um horizonte de $t_T - t_1$ anos e T estágios, pode ser realizado através da Eq. (2.32), conforme (Escobar; Gallego; Romero, 2004).

$$c(x) = \frac{c_1(x)}{(1+I)^{t_1-t_0}} + \frac{c_2(x)}{(1+I)^{t_2-t_0}} + \dots + \frac{c_T(x)}{(1+I)^{t_T-t_0}} \quad (2.32)$$

ou

$$c(x) = \delta_{inv}^1 c_1(x) + \delta_{inv}^2 c_2(x) + \dots + \delta_{inv}^T c_T(x) \quad (2.33)$$

onde

- $\delta_{inv}^t = 1/(1+I)^{t-t_0}$;
- x representa as variáveis de investimento (linhas a serem construídas);
- $c_t(x)$ representa o valor do investimento no estágio t .

Modelo CC Multiestágio

O modelo cc multiestágio pode ser formulado como segue.

$$\min v = \sum_{t=1}^T \left[\delta_{inv}^t \sum_{(i,j) \in \Omega} c_{ij} n_{ij}^t + \alpha \sum_{k \in \Gamma} r_k^t \right] \quad (2.34)$$

sujeito a

(2.35)

$$S^t f^t + g^t + r^t = d^t \quad (2.36)$$

$$f_{ij}^t - \gamma_{ij} \left(n_{ij}^0 + \sum_{m=1}^t n_{ij}^m \right) (\theta_i^t - \theta_j^t) = 0 \quad (2.37)$$

$$|f_{ij}^t| \leq \left(n_{ij}^0 + \sum_{m=1}^t n_{ij}^m \right) \bar{f}_{ij} \quad (2.38)$$

$$\underline{g}_j^t \leq g_j^t \leq \bar{g}_j^t \quad (2.39)$$

$$0 \leq r^t \leq d^t \quad (2.40)$$

$$\underline{n}_{ij}^t \leq n_{ij}^t \leq \bar{n}_{ij}^t \quad (2.41)$$

$$\sum_{t=1}^T n_{ij}^t \leq \bar{n}_{ij} \quad (2.42)$$

$$n_{ij}^t \text{ inteiro}; \theta_i^t, f_{ij}^t \text{ irrestrito} \quad (2.43)$$

$$t = 1, 2, \dots, T \quad (2.44)$$

onde T é o número de estágios; r^t é o conjunto de geradores artificiais no estágio t e seu somatório indica o corte de carga existente em um determinado estágio do plano de expansão; $\underline{g}_j^t$ é o limite inferior para o gerador j no estágio t ; $\underline{n}_{ij}^t$ é o limite inferior para a variável n_{ij} no período t ; Γ é o conjunto de barras de carga. As demais variáveis são as mesmas do planejamento estático com apenas o acréscimo do índice t que define o estágio do planejamento.

Como se pode ver pelo modelo representado pelas Eqs. (2.34)-(2.44), o modelo multiestágio apresenta maior dificuldade de resolução em relação ao modelo estático (representado pelas Eqs. (2.1)-(2.8)), uma vez que o conjunto de restrições (2.37) apresenta maior número de produto de variáveis do que o conjunto de restrições (2.3) e o número de variáveis inteiras aumenta em T vezes.

Modelo Híbrido Multiestágio

Em (Romero et al., 2007) é sugerido um modelo híbrido multiestágio, formulado da seguinte forma:

$$\min v = \sum_{t=1}^T \left[\delta_{inv}^t \sum_{(i,j) \in \Omega} c_{ij} n_{ij}^t + \alpha \sum_{k \in \Gamma} r_k^t \right] \quad (2.45)$$

$$\text{sujeito a} \quad (2.46)$$

$$S^t f^t + S^0 f^0 + g^t + r^t = d^t \quad (2.47)$$

$$f_{ij}^{0t} - \gamma_{ij} n_{ij}^0 (\theta_i^t - \theta_j^t) = 0 \quad (2.48)$$

$$f_{ij}^{0t} \leq n_{ij}^0 \bar{f}_{ij} \quad (2.49)$$

$$f_{ij}^t \leq (\sum_{m=1}^t n_{ij}^m) \bar{f}_{ij} \quad (2.50)$$

$$\underline{g}_j^t \leq g_j^t \leq \bar{g}_j^t \quad (2.51)$$

$$0 \leq r^t \leq d^t \quad (2.52)$$

$$0 \leq n_{ij}^t \leq \bar{n}_{ij}^t \quad (2.53)$$

$$\sum_{t=1}^T n_{ij}^t \leq \bar{n}_{ij} \quad (2.54)$$

$$n_{ij}^t \text{ inteiro}; \theta_j^t, f_{ij}^{0t} \text{ e } f_{ij}^t \text{ inteiro} \quad (2.55)$$

$$\forall (i, j) \in \Omega, t = 1, 2, \dots, T \quad (2.56)$$

As variáveis são as mesmas do planeamento estático com apenas o acréscimo do índice t que define o estágio do planeamento.

É importante destacar que este modelo pode ser linear ou não-linear dependendo de como se considera a variável n_{ij}^0 . Ao se considerar n_{ij}^0 como sendo a configuração inicial do sistema, para todo t , como é feito no modelo híbrido tradicional, o modelo (2.45)-(2.56) trata-se de um problema de PLIM. Porém, esta relaxação pode apresentar resultados muito ineficazes em relação ao modelo cc multiestágio. Se, em vez disso, se considerar n_{ij}^0 somente para o estágio 1, e para os demais estágios considerar a Eq. (2.57),

$$n_{ij}^{0t} = \sum_{k=1}^{t-1} n_{ij}^{0k} \quad (2.57)$$

o modelo passa a se tratar de um PNLIM, de quase mesma complexidade que o modelo formado pelas Eqs. (2.34)-(2.44). Provavelmente devido a estas desvantagens que o modelo híbrido não tem sido muito considerado para lidar com o PERTEE multiestágio.

Capítulo 3

Planejamento da Expansão Considerando Múltiplos Cenários de Geração

O Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica tradicionalmente tem sido abordado no intuito de indicar onde, quando e quais equipamentos devem ser adicionados a uma rede de transmissão para que esta possibilite que a previsão de crescimento futuro da demanda seja atendida pela geração futura de energia. Considerando este problema, muitas propostas de solução foram publicadas (Romero et al., 2007, Sousa; Asada, 2011). Porém, as recentes alterações ocorridas na indústria de eletricidade influenciaram de forma impactante o problema de Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica, principalmente no que se concerne à abertura do acesso ao sistema de transmissão e ao incentivo à livre competição.

Anteriormente, os sistemas de energia eram centralizados, e as entidades detentoras do poder sobre a geração, transmissão e distribuição de energia elétrica eram organizadas em estruturas verticalmente integradas. Neste ambiente, as atividades de planejamento e operação podiam ser executadas de forma centralizada seguindo critérios estabelecidos por uma empresa, geralmente pública, detentora do monopólio do serviço. Desta forma, o planejador devia se preocupar somente com as restrições técnicas, com restrições de segurança e com o objetivo de construir uma rede com menor investimento possível.

Com o advento da descentralização do sistema elétrico de potência, os agentes responsáveis pela geração, transmissão e distribuição passaram a ser diferentes, com interesses diversos e geralmente conflitantes. O propósito da descentralização é permitir a livre concorrência e dar aos consumidores liberdade de escolha sobre que agente contratar, conseguindo com isso aumento na qualidade do serviço obtido e preços mais baixos.

As redes de transmissão têm papel crítico no intuito de prover acesso a todos os participantes do mercado competitivo de energia, uma vez que cabe à rede de transmissão possibilitar as diversas programações de despacho de geração, decorrentes da interação entre as ofertas dos agentes geradores e as necessidades de potência dos consumidores. Em consequência dessas interações, muitos cenários de geração podem existir e os despachos de geração horários ou diários possuem grande incerteza.

O congestionamento da rede de transmissão para algum dos cenários possíveis inibe a livre concorrência do mercado de energia. Portanto, para que um ambiente competitivo nos mercados de energia possa existir, faz-se imperioso que a rede de transmissão seja planejada de tal forma que estes cenários possíveis de geração possam ser atendidos pela rede sem que haja congestionamento.

Portanto, deve-se promover o acesso livre às redes de transmissão tanto para os geradores como para os consumidores de energia elétrica. Do lado dos geradores, o acesso livre possibilita que estes agentes compitam entre si, uma vez que a rede suporta os diversos cenários de despacho de geração, possibilitando a existência dos mercados de energia elétrica. Cabe ressaltar que no momento em que o planejamento é realizado, pouco ou nada se sabe sobre quais cenários terão maior probabilidade de ocorrer, uma vez que estes mudarão de acordo com as condições de mercado futuro. Ressalta-se ainda que uma modelagem matemática que atenda a todos os cenários de geração possíveis pode resultar em um plano de expansão muito caro, inviabilizando o propósito de atendimento a múltiplos cenários, já que o investimento com a expansão não pode ser maior do que os custos de congestionamento da rede.

Em consequência disto, o PERTEE deve ser realizado levando em conta que o futuro sistema terá diversos padrões de fluxo de potência resultantes das variações do mercado de energia elétrica. Diante deste novo ambiente e, segundo Latorre et al. (2003), a experiência obtida pelos sistemas inglês e chileno promoveu o interesse sobre questões do planejamento da transmissão em um ambiente competitivo resultando em um aumento de publicações na área (Baldick; Kahn, 1993).

Nesse contexto, há a necessidade de se rever o processo de planejamento da expansão das redes de transmissão. Esta revisão deve levar em conta tanto a idealização de novos modelos e algoritmos de solução como os problemas consequentes do ambiente competitivo em que estão os sistemas de potência modernos.

Este trabalho foca na síntese da topologia da rede de transmissão e visa explorar o impacto no custo de atendimento aos diversos cenários futuros de geração pela rede de transmissão. Devido ao acesso livre e ao despacho baseado em oferta, os padrões de geração e demanda estão propensos a mudanças mais rápidas e significativas (David; Wen, 2002). Como resultado, algumas partes da rede de transmissão existente serão usadas de formas diferentes daquelas originalmente planejadas ou usadas historicamente. Novos gargalos podem ocorrer e algumas restrições existentes podem ser violadas mais frequentemente e com maiores impactos econômicos. Portanto, novos métodos para o desenvolvimento de uma rede de transmissão que melhore a eficiência de um mercado competitivo de energia se fazem necessários.

Existem poucos trabalhos na literatura a respeito da aplicação de métodos computacionais para solução do planejamento da expansão considerando a restrição de possibilitar um ambiente de acesso livre à rede de transmissão. Informações sobre o planejamento considerando um grupo seletivo de geração podem ser encontradas em (Fang; Hill, 2002). Silva Junior (2005) faz uma análise do PERTEE com restrições de segurança e que, na ausência de contingência, opere adequadamente para vários planos de programação da geração. Em (Zuluaga; Romero; Gallego, 2008) o problema é explorado no intuito de encontrar soluções de expansão que atendam a qualquer plano de geração possível. Nos artigos (Baldick; Kahn, 1993, Buygi et al., 2004a, Latorre et al., 2003) é apresentada uma revisão dos novos desafios do problema tratado neste trabalho.

3.1 Metodologia Proposta

Com o novo paradigma, um sistema elétrico deve ter a capacidade de operar em várias propostas de programação da geração determinadas pelas exigências e volatilidade do mercado de energia elétrica. Os seguintes critérios devem ser satisfeitos por um sistema de energia robusto às necessidades dos mercados de energia elétrica (Silva Junior, 2005, Zuluaga, 2008):

- Uma distribuidora deve ser atendida em sua demanda especificada;
- Qualquer geradora pode operar em qualquer ponto de operação respeitando seus limites;
- O sistema não deve apresentar problemas de congestionamento para nenhum plano de programação factível;

Cabe ressaltar que um plano de programação é considerado factível quando a soma de potência ativa de todos os geradores do sistema seja suficiente para atender à demanda de pico do sistema.

Desta forma, faz-se necessário dispor de uma modelagem matemática adequada que leve em conta todos os cenários factíveis e futuros de geração, o que será chamado neste trabalho, assim como em (Zuluaga, 2008), planejamento considerando múltiplos cenários de geração.

Em (Zuluaga; Romero; Gallego, 2008) é mostrada uma forma adequada para se determinar o número máximo de planos de programação diferentes que podem existir em um sistema elétrico. A metodologia considera a lógica do órgão operador do sistema que afirma que se em um sistema existem k geradores, então $(k - 1)$ geradores operam em sua capacidade máxima ou mínima e o gerador que sobra faz o balanço de potência. Estes cenários são chamados na literatura de cenários extremos. Considerando esta metodologia, o número de cenários extremos é $k \times 2^{k-1}$. Deve-se observar que considerando esta lógica, alguns cenários extremos serão infactíveis.

Como já mencionado neste trabalho, os esforços para realizar o PERTEE para atender aos diversos cenários de geração não podem implicar em um custo de investimento que seja maior do que o custo de congestionamento da rede. Porém, as abordagens do PERTEE realizadas até o momento indicam que esta restrição econômica pode ser violada. Em (Silva Junior, 2005), a proposta para a expansão considerando apenas 5 cenários de geração para o sistema de teste IEEE de 24 barras sugere um incremento de 250,00% em relação ao ótimo clássico. Para o sistema sul brasileiro com redespatch, houve um incremento de 192,49% em relação ao ótimo clássico para atender a 11 cenários de geração. Zuluaga (2008) fez uma abordagem considerando os cenários extremos de geração. Em sua abordagem, houve um incremento de 775,00% em relação ao ótimo clássico do sistema de teste IEEE de 24 barras.

O incremento significativo no custo de investimento do planejamento se dá devido ao fato de se estar considerando que todo e qualquer cenário possível deve ser atendido. Desta forma, novas metodologias devem ser estudadas e desenvolvidas no intuito de diminuir o custo de investimento do planejamento que atenda a múltiplos cenários de geração. Uma alternativa para diminuir o custo de investimento do planejamento, é o desenvolvimento de uma metodologia que visa atender aos cenários mais susceptivos de ocorrência, diminuindo o custo do investimento à medida que apresenta margens de confiança em relação à robustez da rede para o atendimento dos diversos cenários de geração.

Propõe-se neste trabalho uma nova abordagem do planejamento da expansão da transmissão com o intuito de possibilitar que a rede de transmissão suporte os múltiplos cenários de geração e possibilite o livre acesso à rede. Para tanto, considerou-se o modelo de despacho em *pool* (*pool-dispatch model*) em que os geradores de energia elétrica vendem a energia para o *pool* e os consumidores a compram do *pool* (Fang; Hill, 2002, Murray, 1998).

O objetivo deste trabalho é fortalecer a rede de transmissão de tal forma que esta seja capaz de atender ao crescimento futuro da demanda e que seja capaz de suportar os diferentes cenários de

geração decorrentes do padrão de fluxo de potência baseado em mercado de energia elétrica. Para tanto, os cenários de geração serão simulados, em uma primeira fase do planejamento, de tal forma que sigam uma distribuição de probabilidade de ocorrência. Albuquerque; Almeida; Cavalcante (2009), com base em dados históricos e análises estatísticas e matemáticas, modela o Sistema Nordeste Brasileiro de acordo com uma função de distribuição de probabilidade semelhante à distribuição *beta*. Para os sistemas testados nesta tese não dispomos de tais análises ou dados, por isso será suposto neste trabalho que os geradores dos sistemas variam sua geração de energia de acordo com a função *beta* de distribuição de probabilidade, conforme a Figura 3.1.

Figura 3.1: Função de distribuição de probabilidade beta

Neste trabalho, será simulada a geração de N cenários em que cada gerador terá seu limite de geração seguindo a distribuição de probabilidade supracitada. Desta forma, dada a formulação DC tradicional do problema, a única modificação será no limite dos geradores. Assim, o limite do gerador i em cada cenário q será formulado de acordo com a Eq. (3.1).

$$0 \leq g_i^q \leq \bar{g}_i^q \quad (3.1)$$

onde $\bar{g}_i^q \sim \beta(6, 2, 0, \bar{g}_i)$. Em (Ross, 2009) e no Apêndice B são apresentadas mais informações sobre a distribuição de probabilidade *beta*.

Um cenário q será factível se $\sum_{i \in \Omega_g} \bar{g}_i^q \geq \sum_{i \in \Gamma} d_i$. Onde Ω_g é o conjunto das barras de geração e Γ é o conjunto de barras de carga.

Note que com esta distribuição há 84,14% de probabilidade de um determinado gerador estar entre 60% a 100% de sua capacidade máxima e, conseqüentemente, probabilidade de 15,86% de um gerador estar abaixo de 60% de sua capacidade máxima. Esta distribuição de probabilidade foi utilizada partindo da suposição de que os geradores estarão na maioria dos cenários de geração na faixa 60% – 100% de sua capacidade. O uso desta metodologia é flexível, e o planejador pode alterar os parâmetros da distribuição *beta* ou até mesmo usar outra distribuição de probabilidade que se adeque mais ao sistema de transmissão específico, baseado em uma análise histórica do sistema e do mercado em análise ou em outra metodologia.

É importante destacar que, por questão de simplicidade, considerou-se que todos os geradores se comportam de acordo com a mesma distribuição de probabilidade. Porém, pode ocorrer que em

um sistema em particular cada gerador se comporte de forma diferente. Assim, um estudo para definir qual distribuição de probabilidade se adéqua a cada gerador poderá se fazer necessário para a aplicação deste modelo a um sistema real. Porém, a única modificação nesta metodologia se dará na definição da função de distribuição de probabilidade dos geradores.

Na abordagem aqui proposta, uma vez gerados os N cenários, aplica-se uma metodologia de solução multiobjetivo que gere configurações que atendam ao objetivo de satisfazer o máximo número de cenários com o menor custo de investimento. Desta forma o plano de expansão conseguirá atender a múltiplos cenários de geração, porém sendo o custo de adição dos reforços na rede o menor possível.

3.2 Modelagem Com Múltiplos Cenários de Geração

Como foi comentado anteriormente, abordagem proposta nesta tese é multiobjetivo. Portanto, propõe-se neste trabalho, o seguinte modelo multiobjetivo para solução do PERTEE com múltiplos cenários de geração.

$$\begin{cases} \min \varphi &= \sum_{(i,j) \in \Omega} c_{ij} n_{ij} \\ \max \psi &= \frac{\sum_{q=1}^N l^q}{N} \end{cases} \quad (3.2)$$

$$(3.3)$$

sujeito a

$$Sf^q + g^q + r^q = d \quad (3.4)$$

$$f_{ij}^q - \gamma_{ij}(n_{ij}^0 + n_{ij})(\theta_i^q - \theta_j^q) = 0 \quad (3.5)$$

$$|f_{ij}^q| \leq (n_{ij}^0 + n_{ij})\bar{f}_{ij} \quad (3.6)$$

$$0 \leq g^q \leq \bar{g}^q \quad (3.7)$$

$$\bar{g}^q \sim \beta(6, 2, 0, \bar{g}) \quad (3.8)$$

$$0 \leq r_i^q \leq d_i \quad (3.9)$$

$$0 \leq n \leq \bar{n}_{ij} \quad (3.10)$$

$$n_{ij} \text{ inteiro}; f_{ij}^q \text{ irrestrito}; \quad (3.11)$$

$$q = 1, 2, 3, \dots, N \quad (3.12)$$

$$\theta_j^q \text{ irrestrito } \forall j \quad (3.13)$$

onde l^q é definido de acordo com a Eq. (3.14);

$$l^q = \begin{cases} 1, & \text{se } \sum_{i \in \Gamma} r_i^q = 0 \\ 0, & \text{caso contrário} \end{cases} \quad (3.14)$$

f_{ij}^q , θ_j^q e r_i^q representam as variáveis de operação para o cenário de geração q dos N cenários gerados; Γ é o conjunto de barras de carga.

Na Eq. (3.2), o objetivo φ visa encontrar o plano de expansão de menor custo e o objetivo ψ representa a porcentagem de cenários que uma determinada configuração atende dos N cenários

gerados e visa encontrar o plano de expansão que atenda ao maior número de cenários de geração. É fácil ver que se $N = 1$ e $\bar{g}^p = \bar{g}$, o modelo (3.2-3.13) se equivale ao modelo clássico (2.1-2.8).

3.3 Geração dos Cenários

A metodologia proposta neste trabalho pode ser dividida em duas fases. A primeira, consiste em gerar N cenários factíveis de geração. A segunda, em encontrar planos de expansão factíveis atendendo aos objetivos de menor custo e atendimento ao maior número de cenários possíveis. Como os geradores podem, a priori, estar em qualquer ponto contínuo entre 0 e sua capacidade máxima de geração, o conjunto com todos os cenários factíveis de geração pode ser infinito. Por isso, na metodologia proposta neste trabalho, sugere-se a utilização de apenas uma amostragem dos cenários factíveis possíveis.

Porém, deve-se definir um valor para N que de fato represente o conjunto de todos os cenários factíveis possíveis. Em nossa abordagem, usaremos o conceito de coeficiente de variação para definir o valor de N . O coeficiente de variação (ϑ) é uma medida de dispersão usada para comparação de distribuições diferentes e é dado pela divisão do desvio padrão (σ) da amostra pela sua média ($\bar{X}$). O Algoritmo 1 foi usado para definir a amostra.

É importante destacar que a amostra X contém os diversos cenários de geração para cada gerador, e que todos os cenários em X devem ser factíveis, ou seja, a soma dos limites dos geradores tem que ser maior ou igual à demanda de pico do sistema. Quando o Algoritmo 1 convergir, usamos $N = |X|$.

Na linha 11 do Algoritmo 1 faz-se um teste para verificar se o coeficiente de variação de todos os geradores se estabilizou. Quando isto ocorrer, temos uma amostra estável do conjunto de cenários factíveis possíveis.

Algoritmo 1: Definindo a amostra

1	Gere uma amostra X com 100 cenários.
2	Atribua $i = 0$ e $N = 100$.
3	Calcule ϑ_o^b para cada barra b considerando a amostra X .
4	Repita
5	Atribua $i = i + 1$.
6	Atribua $k = \lceil 0.1 \times N \rceil$.
7	Gere uma amostra X_i com k cenários.
8	Atribua $X \leftarrow X_i \cup X$.
9	Calcule ϑ_i^b para cada barra b considerando a amostra X .
10	Atribua $N = N + k$.
11	até ($ norma(\vartheta_i) - norma(\vartheta_{i-1}) \leq \epsilon$) ;

Para todos os sistemas, o parâmetro ϵ foi definido em 1.0×10^{-4} . Para verificar a estabilidade do Algoritmo 1, testes foram realizados com três sistemas usados na literatura do PERTEE, conforme serão descritos nas próximas seções.

3.3.1 Sistema de Seis Barras

O Sistema de Seis Barras com Redespacho da Geração (Garver) tem 6 barras e 15 ramos. A demanda de pico é de 760,00 MW e a capacidade geração instalada é de 1.110,00 MW. Para este

sistema, o Algoritmo 1 convergiu com uma amostra com 573 cenários.

Para demonstrar que a geração dos N cenários para este sistema é uma amostragem estável, primeiramente dividimos os cenários da amostra em três perfis:

- Cenários com baixa folga de geração: nestes cenários, o sistema está operando com capacidade de geração muito próxima da demanda de pico.
- Cenários com alta folga de geração: nestes cenários, o sistema está operando com capacidade de geração muito próxima da geração total do sistema. Em outras palavras, $\bar{g}_i^p$ está próximo de $\bar{g}_i$ para a maioria dos cenários p da amostra.
- Cenários com folga moderada de geração: nestes cenários, o sistema está operando com capacidade de geração entre os cenários com baixa folga de geração e os cenários com alta folga de geração.

Para cada sistema, foram considerados três perfis de geração. Considere $\mathcal{L}$ como sendo a porcentagem do excedente de geração do sistema em relação a sua demanda de pico, e C_k a porcentagem do excedente de geração de um cenário k em relação à demanda de pico do sistema. Desta forma, os cenários são divididos como segue.

1. Cenários com baixa folga de geração: $0 < C_k \leq \|\mathcal{L}/3\|$.
2. Cenários com folga de geração moderada: $\|\mathcal{L}/3\| < C_k \leq \|2\mathcal{L}/3\|$.
3. Cenários com alta folga de geração: $\|2\mathcal{L}/3\| < C_k \leq \mathcal{L}$.

onde: considerando x um número real, $\|x\|$ é o número inteiro mais próximo de x .

Para constatar que a amostra gerada pelo Algoritmo 1 é uma amostra significativa dos cenários possíveis, mostraremos os perfis de cenários para uma amostra com os N cenários gerados pelo algoritmo e para uma amostra com 1.000.000 de cenários.

O sistema Garver tem 32% de folga de geração. Desta forma, definimos três classes de cenários: cenários com baixa folga de geração ($0\% \leq folga < 11\%$); cenários de folga moderada de geração ($11\% \leq folga \leq 21\%$) e cenários com alta folga de geração ($21\% < folga \leq 32\%$). Para este sistema, como já mencionado, foi gerada uma amostra com 573 cenários. A Tabela 3.1 mostra o perfil dos cenários de geração são próximos para $N = 573$ e $N = 1.000.000$, o que demonstra que o Algoritmo 1 consegue gerar uma amostragem representativa do conjunto de cenários factíveis para este sistema.

Tabela 3.1: Distribuição dos cenários para o sistema Garver com $N = 573$ e $N = 1.000.000$.

N	Folga %		
	Baixa	Média	Alta
573	53,32	37,86	8,82
1.000.000	53,17	37,61	9,22

3.3.2 Sistema IEEE de 24 barras

O IEEE24Bar testado tem 24 barras e 41 ramos. A demanda de pico é de 8.550,00 MW e a capacidade geração instalada é de 1.0215,00 MW. Para este sistema, o Algoritmo 1 gerou $N = 1.689$ amostras.

O sistema IEEE24Bar tem 16% de folga de geração. Desta forma, definimos três classes de cenários: cenários com baixa folga de geração ($0\% \leq folga < 5\%$); cenários de folga moderada de geração ($5\% \leq folga \leq 11\%$) e cenários com alta folga de geração ($11\% < folga \leq 16\%$). Para este sistema, como já mencionado, foi gerada uma amostra com 1.689 cenários. A Tabela 3.2 mostra o perfil dos cenários de geração são próximos para $N = 1.689$ e $N = 1.000.000$ o que demonstra que o Algoritmo 1 consegue gerar uma amostragem representativa do conjunto de cenários factíveis para este sistema.

Tabela 3.2: Distribuição dos cenários para o sistema IEEE24Bar com $N = 1.689$ e $N = 1.000.000$.

N	Folga %		
	Baixa	Média	Alta
1.689	96,63	3,32	0,06
1.000.000	96,75	3,25	0,01

3.3.3 Sistema Sul Brasileiro

O sistema SULBr tem 46 barras e 79 ramos. A demanda de pico é de 6.880,00 MW e a capacidade geração instalada é de 10.545,00 MW.

O sistema SULBr tem 35% de folga de geração. Desta forma, definimos três classes de cenários: cenários com baixa folga de geração ($0\% \leq folga < 12\%$); cenários de folga moderada de geração ($12\% \leq folga \leq 23\%$) e cenários com alta folga de geração ($23\% < folga \leq 35\%$). Para este sistema, foi gerada uma amostra com 2.087 cenários. A Tabela 3.3 mostra o perfil dos cenários de geração são próximos para $N = 2.087$ e $N = 1.000.000$ o que demonstra que o Algoritmo 1 também consegue gerar uma amostragem representativa do conjunto de cenários factíveis para este sistema.

Tabela 3.3: Distribuição dos cenários para o sistema SULBr com $N = 2.087$ e $N = 1.000.000$.

N	Folga %		
	Baixa	Média	Alta
2.087	91,71	8,24	0,05
1.000.000	91,14	8,83	0,03

Dado que o modelo proposto neste trabalho para solução do Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica considerando múltiplos cenários de geração (Modelo (3.2-3.13)) é um modelo de otimização multi-objetivo, no próximo capítulo será realizada uma breve revisão da otimização multiobjetivo, destacando alguns dos principais conceitos e algoritmos de solução.

Capítulo 4

Otimização Multiobjetivo

Muitos problemas do mundo real envolvem a otimização simultânea de diversos objetivos. Geralmente, não há uma solução única mas um conjunto de alternativas de soluções. Problemas que têm mais de um objetivo a serem alcançados são classificados como problemas de otimização multiobjetivo (PMOs). Um PMO pode ser definido, portanto, como o problema de encontrar um vetor de variáveis de decisão que satisfaça um conjunto de restrições de igualdade e de desigualdade e que otimize um vetor cujos elementos são as funções objetivo do problema (Coello; Lamont; Van Veldhuizen, 2007). Um problema multiobjetivo composto de k funções objetivo, n variáveis de decisão, m restrições de desigualdade e p restrições de igualdade pode ser formulado matematicamente como segue.

$$\min[f_1(x), f_2(x), \dots, f_k(x)] \quad (4.1)$$

$$\text{sujeito a} \quad (4.2)$$

$$g_i(x) \leq 0 \quad i = 1, 2, \dots, m \quad (4.3)$$

$$h_i(x) = 0 \quad i = 1, 2, \dots, p \quad (4.4)$$

onde $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ é o vetor de variáveis de decisão.

4.1 Conceitos e Definições Básicas

Para ser sucinto, nos ateremos somente aos termos e conceitos mais críticos da otimização multiobjetivo. Para uma discussão mais extensiva de tópicos históricos e teóricos, sugerimos a leitura dos textos (Marler; Arora, 2004; Stadler, 1986).

4.1.1 Dominância

A maioria dos algoritmos multiobjetivos utilizam o conceito de dominância. Nestes algoritmos, duas soluções são comparadas com o objetivo de verificar qual domina a outra. Para definição de dominância no âmbito da otimização multiobjetivo, sejam i e j duas soluções para um determinado problema multiobjetivo com k funções objetivo. Convencionamos neste texto os seguintes operadores: $\prec, \preceq, \succ, \succeq$. Abaixo são listadas as relações entre duas soluções i e j considerando estes operadores.

- $i \succ j$ denota que a solução i é pior do que a solução j para um determinado objetivo.
- $i \prec j$ denota que a solução i é melhor do que a solução j para um determinado objetivo.

- $i \succeq j$ denota que a solução i é pior ou igual à solução j para um determinado objetivo. Em outras palavras, i não é melhor do que j .
- $i \preceq j$ denota que a solução i é melhor ou igual à solução j para um determinado objetivo. Ou seja, i não é pior do que j .

Definição 4.1. Uma solução $x^{(1)}$ domina outra solução $x^{(2)}$ se ambas as seguintes condições forem satisfeitas para 1 e 2 (Deb, 2001):

1. A solução $x^{(1)}$ não é pior do que a solução $x^{(2)}$ em qualquer objetivo. Ou seja, $f_j(x^{(1)}) \preceq f_j(x^{(2)})$ para todo $j = 1, 2, \dots, k$.
2. A solução $x^{(1)}$ é melhor do que a solução $x^{(2)}$ em pelo menos um objetivo. Ou seja, $f_j(x^{(1)}) \prec f_j(x^{(2)})$ para pelo menos um $j \in \{1, 2, \dots, k\}$.

Se alguma das condições acima forem violadas, então a solução $x^{(1)}$ não domina a solução $x^{(2)}$. Caso contrário, $x^{(1)} \preceq x^{(2)}$, então pode-se afirmar que:

- $x^{(2)}$ é dominada por $x^{(1)}$;
- $x^{(1)}$ é não-dominada por $x^{(2)}$ ou $x^{(1)}$ não é inferior a $x^{(2)}$.

Como a relação de dominância permite determinar se uma solução é melhor do que uma determinada outra, a maioria dos métodos multiobjetivos usa este conceito como guia de busca por soluções não-dominadas.

4.1.2 Otimalidade de Pareto

Geralmente os algoritmos para solução de problemas multiobjetivos são algoritmos de busca em vizinhança. Desta forma, sempre que se tiver em mãos um conjunto de soluções factíveis, deve-se fazer comparação entre cada uma delas no intuito de se determinar quais soluções são dominantes e quais soluções são dominadas. À medida que o processo de busca avança, espera-se encontrar e manter o conjunto de soluções que não foram dominadas por nenhuma outra solução durante o processo. Este conjunto de todas as soluções não-dominadas encontradas chama-se *conjunto não-dominado*. O conjunto de soluções não-dominadas pode ser formalmente definido como segue.

Definição 4.2. Dado um conjunto de soluções P e seja P' o conjunto de todas as soluções em P não-dominadas por qualquer outra solução em P . O conjunto P' é chamado de *conjunto não-dominado*.

Se S é o conjunto de soluções factíveis para o problema multiobjetivo e $P = S$, então P' é chamado de *conjunto Pareto-ótimo* ou *fronteira Pareto-ótima*. Assim como existe um ótimo local e um ótimo global na programação mono-objetivo, na programação multiobjetivo existe uma fronteira Pareto-ótima local e uma fronteira Pareto-ótima global.

Definição 4.3. O conjunto de soluções não-dominadas de toda a região factível é chamado de *conjunto Pareto-ótimo global*.

Uma vez que nenhuma solução do conjunto Pareto-ótimo global é dominada por qualquer solução da região factível, então o conjunto Pareto-ótimo global, geralmente chamado apenas de conjunto Pareto-ótimo, é a solução do problema multiobjetivo.

Um conjunto Pareto-ótimo local pode ser definido como segue (Deb, 1999, Miettinen, 1999):

Definição 4.4. *Se para cada membro x pertencente ao conjunto $\underline{P}$ não existe em sua vizinhança nenhuma solução y ($\|x - y\| \leq \epsilon$, onde ϵ é um número pequeno e positivo) que domine qualquer membro de $\underline{P}$, então as soluções pertencentes ao conjunto $\underline{P}$ formam um conjunto Pareto-ótimo local.*

Dado um problema de otimização multiobjetivo com funções objetivo f_1 e f_2 , a serem minimizadas, a Figura 4.1 representa graficamente a definição de conjunto Pareto-ótimo local e conjunto Pareto-ótimo global.

Figura 4.1: Fronteira Pareto-ótima

4.1.3 Ponto Ideal

Como já foi mencionando, a programação multiobjetivo lida com um conjunto de k funções objetivo, geralmente conflitantes, a serem otimizadas. Como os objetivos são geralmente conflitantes, espera-se que a solução ótima de cada uma das funções objetivo seja diferente das demais. O ponto $x^* = [x_1^*, x_2^*, \dots, x_n^*]^T \in \mathbf{x}$ tal que $x_j^* = \min\{f_j(x) : x \in \mathbf{x}, j = 1, 2, \dots, k\}$ é chamado ponto ideal. Em outras palavras, considerando um problema de minimização, o ponto ideal é aquele que minimiza todas as funções objetivo.

Note que se existe um ponto ideal, então o problema em questão pode ser resolvido através da programação mono-objetivo. Ou seja, precisa-se apenas encontrar a solução ótima de uma das funções objetivo e este consistiria a solução ótima de todas as demais.

Como o ponto ideal geralmente não existe na programação multiobjetivo, e como se considera que todas as funções objetivo são igualmente importantes, Deb (2001) conjectura que a programação multiobjetivo tem duas metas:

1. Encontrar um conjunto de soluções o mais próximo possível da fronteira Pareto-ótima.
2. Encontrar um conjunto de soluções com a maior diversidade possível.

A primeira meta é comum para qualquer problema de otimização. Como a solução de problemas de otimização é baseada em modelos e ferramentas matemáticas e computacionais e visa auxiliar no processo de tomada de decisão, espera-se que estas apresentem ao tomador de decisão uma resposta o mais próximo possível da solução ótima. A segunda meta é mais específica à otimização multiobjetivo. A diversidade entre as soluções garante que todas funções objetivos e o balanceamento entre quaisquer duas funções objetivos foi tratado de forma imparcial. A Figura 4.2 ilustra soluções bem distribuídas ao longo da fronteira Pareto-ótima, Figura 4.2(a), e soluções má distribuídas ao longo da fronteira Pareto-ótima, Figura 4.2(b).

Figura 4.2: Ilustração da distribuição das soluções na fronteira Pareto-ótima.

4.1.4 Níveis de Não-Dominância

Alguns métodos de busca multiobjetivo procuram definir somente a fronteira Pareto-ótima da população. Estes métodos dividem, portanto, as soluções encontradas ao longo do processo de busca em soluções dominadas e soluções não-dominadas. Outros métodos usam como estratégia de busca a definição de níveis de não-dominância. Os níveis de não-dominância são resultantes da ordenação do conjunto de soluções encontradas de acordo com sua não-dominância em relação às outras soluções do conjunto. Desta forma, seja P um conjunto de soluções encontradas durante o processo de busca. Define-se o nível 1 de P ($F_1 \subset P$) como o conjunto formado pelas soluções não-dominadas pertencentes a P . Uma vez definido o nível 1, as soluções que o formam são removidas temporariamente de P . Após a remoção das soluções pertencentes ao nível 1, defini-se o nível 2 de P ($F_2 \subset P$) como o conjunto formado pelas soluções não-dominadas pertencentes a P . Em outras palavras, o nível 2 é formado pelas soluções que são dominadas apenas pelas soluções do nível 1. As soluções pertencentes ao nível 2 são temporariamente removidas de P para que seja definido o nível 3. Desta forma, segue-se iterativamente até que todas as soluções pertencentes a P estejam agrupadas em seu respectivo nível de não-dominância.

A Figura 4.3 mostra o resultado de um processo de classificação de soluções em níveis de não-dominância.

Figura 4.3: Classificação das soluções em níveis de não-dominância.

4.2 Algoritmos Para Otimização Multiobjetivo

Muitos métodos têm sido aplicados para solução de problemas multiobjetivos. Uma das primeiras abordagens foi através da otimização mono-objetivo, já que qualquer problema multiobjetivo pode ser convertido a um problema mono-objetivo agregando-se as funções objetivo a uma função escalar $s : X \rightarrow \mathbb{R}$.

Um método de agregação, conhecido também como abordagem de soma ponderada, associa um peso ω_i a cada uma das funções objetivo f_i , conforme a Eq. (4.5).

$$s(x) = \sum_{i=1}^k \omega_i f_i(x) \quad (4.5)$$

Geralmente $\omega_i > 0$ e representa a importância da função objetivo i em relação aos demais objetivos. Estudos teóricos em relação à solução ótima do método de soma ponderada e a solução Pareto-ótima podem ser encontrados nos textos (Chankong; Haimes, 1983, Ehrgott, 2005, Miettinen, 1999).

Existem vários métodos de escalonamento, porém todos exigem do tomador de decisão um conhecimento profundo do domínio do problema e de informações em relação à importância de cada um dos objetivos.

Por outro lado, parece natural a aplicação de algoritmos evolutivos para executar a tarefa de encontrar um conjunto de soluções não-dominadas. Portanto, nesta seção, apresentar-se-ão os principais algoritmos evolutivos para solução de problemas multiobjetivo. Os algoritmos evolutivos são naturalmente adaptados para resolver problemas de otimização multiobjetivo, uma vez que lidam simultaneamente com um conjunto de soluções possíveis. Isto possibilita que sejam encontrados diversos membros do conjunto Pareto-ótimo em uma única execução do algoritmo. Esta natureza intrínseca dos algoritmos evolutivos para solução de Problemas Multiobjetivo (PMOs) pode ser confirmada no número de métodos baseados em algoritmo evolutivo para essa classe de problemas de otimização. O levantamento bibliográfico feito em (Jones; Mirrazavi; Tamiz, 2002) mostrou que a maioria das metaheurísticas aplicadas para resolver PMOs são baseadas em Algoritmos Genéticos (AGs). Porém, outras metaheurísticas como *Simulated Annealing* e busca tabu também têm sido aplicadas para solução de problemas multiobjetivo.

4.3 Algoritmos Genéticos

Os algoritmos evolutivos multiobjetivos são baseados no algoritmo introduzido por John Holland em 1975 (Holland, 1975). Holland (1975) incorporou características da evolução das espécies para desenvolver um método flexível, robusto (independentemente do ponto de partida, os AGs são capazes de encontrar soluções de boa qualidade) e computacionalmente simples de implementar para resolver problemas complexos.

4.3.1 Terminologia

A seguir serão detalhadas as principais terminologias dos AGs.

Cromossomo

O cromossomo é a estrutura de um indivíduo. Cada indivíduo é a representação de uma solução candidata para resolver o problema em foco. O cromossomo é composto por *genes*, que são uma cadeia de símbolos. Existem diversas formas de representar um indivíduo computacionalmente. A mais comum é a representação binária. Nesta representação os *alelos* dos genes somente poderão ter valor 0 ou 1. Porém, conforme sugere Michalewicz (1996), a representação dos indivíduos deve ser a mais natural de acordo com a natureza física do problema em análise.

Avaliação do Indivíduo

Nos AGs a cada indivíduo deve ser atribuído um valor que indica sua adaptação, ou seja, quando se trata de um problema a ser resolvido, o valor de adaptação indica o custo da solução que o indivíduo representa.

A cada iteração dos AGs, também chamada de geração, um conjunto de indivíduos formará uma população. Geralmente a população inicial é formada por indivíduos gerados aleatoriamente. O parâmetro mais importante da população é seu tamanho, que geralmente se mantém constante durante toda a execução.

Operadores Genéticos

Uma vez que os AGs são baseados no princípio da seleção natural, eles devem ser capazes de identificar os indivíduos mais adaptados, já que representam soluções de boa qualidade para o problema em foco, para que estes possam ser combinados para gerar novos descendentes. Este processo é guiado por um método de seleção. Existem diversos métodos de seleção, sendo a seleção por torneio um dos mais usados. Na seleção por torneio, $k \geq 2$ indivíduos são selecionados aleatoriamente a partir da população corrente. Estes indivíduos competem entre si, e será selecionado para reprodução o que tiver a melhor adaptação.

Para gerar uma nova população, dois operadores são importantes: operador de recombinação e o operador de mutação. O operador de recombinação é responsável pela recombinação de características dos pais durante a reprodução, permitindo que as próximas gerações herdem essas características. O operador de mutação consiste em mudar aleatoriamente algum gene de um indivíduo da população. O operador de mutação é usado para garantir diversificação da população e impedir, desta forma, a convergência prematura para um ótimo local.

Elitismo

O objetivo de aplicar elitismo é evitar a perda dos melhores indivíduos encontrados durante a busca. Uma forma simples de implementar o elitismo é selecionar τ indivíduos da população

e copiá-los para a próxima geração sem nenhuma modificação. Os outros $N - \tau$ indivíduos são gerados normalmente considerando os operadores genéticos de seleção, cruzamento e mutação.

Parâmetros

Os parâmetros mais importantes para implementação de um Algoritmo Genético (AG) são o tamanho da população M , o número máximo de gerações (geralmente é o único critério de parada do algoritmo), a probabilidade de cruzamento ρ_c e a probabilidade de mutação ρ_m (Michalewicz, 1996). Para mais detalhes teóricos e práticos dos AGs sugerimos a leitura das referências (Coley, 1997, Michalewicz, 1996).

O Algoritmo 2 apresenta o esboço de um algoritmo genético.

Algoritmo 2: Esboço de um algoritmo genético

- 1 Gerar a população inicial $\mathbb{P}_0$.
 - 2 Atribuir $i = 0$.
 - 3 Avaliar a população $\mathbb{P}_i$.
 - 4 Atribuir o valor de aptidão aos indivíduos de $\mathbb{P}_i$.
 - 5 Aplicar os operadores de seleção, cruzamento e mutação a $\mathbb{P}_i$ e gerar a população $\mathbb{P}_{i+1}$.
 - 6 Atribuir $i = i + 1$. Se o número máximo de gerações não tiver sido alcançado, ou outro critério de parada não tiver sido satisfeito, voltar ao passo 3.
-

Nas próximas seções apresentaremos os principais algoritmos evolutivos para solução de problemas multiobjetivo.

4.3.2 *Vector Evaluated Genetic Algorithm*

Um dos primeiros algoritmos não ponderados a lidar com problemas de otimização multiobjetivo foi o *Vector Evaluated Genetic Algorithm* (VEGA) desenvolvido por Schaffer (1985). O VEGA consiste em um algoritmo genético básico com um mecanismo de seleção modificado. A cada geração, um número de subpopulações é gerado aplicando uma seleção proporcional de acordo com cada função objetivo. Desta forma, para um problema com k objetivos, k subpopulações de tamanho M/k cada são gerados (onde M é o tamanho da população). Realizada a divisão da população, cada subpopulação é preenchida com os melhores indivíduos considerando a função objetivo i .

4.3.3 *Multi-Objective Genetic Algorithm*

Fonseca; Fleming (1993) propuseram uma variação dos algoritmos genéticos chamada de *Multi-Objective Genetic Algorithm* (MOGA). O MOGA utiliza um procedimento de ordenação não-dominado dos indivíduos da população. No MOGA, a todos os indivíduos não-dominados é atribuída ordem 1, enquanto que os demais são ordenados de acordo com sua ordem de não-dominância em relação ao restante da população. Sendo x_i um determinado indivíduo da geração t , que é dominado por $p_i^{(t)}$ indivíduos da geração atual, a Eq. (4.6) é usada para atribuir a ordem de cada indivíduo.

$$ordem(x_i, t) = 1 + p_i^{(t)} \quad (4.6)$$

Segundo (Goldberg; Deb, 1991), a atribuição de aptidão realizada por ordenação, como feito pelo MOGA, é muito susceptiva a uma pressão de seleção muito forte, o que pode levar à convergência prematura. Com intuito de evitar este efeito indesejado, um método de formação de nicho foi aplicado para distribuir a população ao longo da região Pareto-ótima.

4.3.4 *Niched Pareto Genetic Algorithm*

O algoritmo *Niched Pareto Genetic Algorithm* (NPGA) foi proposto em (Horn; Nafpliotis; Goldberg, 1994) com o objetivo de levar em conta o conceito de dominância de Pareto. A ideia básica do algoritmo consiste em selecionar dois indivíduos aleatoriamente e compará-los com um subconjunto da população atual (em torno de 10% da população). O indivíduo não-dominado em relação ao restante da população é o vencedor. Em caso de empate (nenhum é dominado, ou ambos são dominados), o desempate pode ser decidido através de uma medida de distância de cada indivíduo em relação aos outros indivíduos da população.

4.3.5 *Nondominated Sorting Genetic Algorithms*

Proposto por Srinivas; Deb (1994), esta abordagem é baseada na classificação dos indivíduos da população em diversos níveis de não-dominância. O *Nondominated Sorting Genetic Algorithms* (NSGA) varia do algoritmo genético simples apenas na forma em que o operador de seleção trabalha, uma vez que os operadores de recombinação e mutação são aplicados da mesma forma do AG tradicional.

Depois que a seleção de indivíduos da população for realizada, a nova população é ordenada de acordo com a não-dominância dos indivíduos. Primeiramente, os indivíduos não dominados são identificados. A estes indivíduos é atribuído um mesmo valor de aptidão elevado. Para que seja mantida a diversidade da população, faz-se com que as soluções não-dominadas compartilhem os seus valores de aptidão segundo algum critério de mensuração de distância (veja (Goldberg; Richardson, 1987, Miller; Shaw, 1996) para estudo de métodos de compartilhamento de valores de aptidão). Após a execução do compartilhamento e as modificações das aptidões, os indivíduos não-dominados são temporariamente ignorados, e o remanescente da população é submetido ao mesmo processo para se definir o próximo nível de não-dominância. Este processo é repetido até que todos os membros da população tenham um valor de aptidão compartilhada.

Os algoritmos supradescritos não fazem uso de elitismo. Os algoritmos que serão descritos nas próximas seções fazem uso de elitismo, geralmente mantendo armazenada uma população externa com os indivíduos não-dominados encontrados durante a busca.

Coello (2005) resalta algumas questões em relação ao uso de população externa por parte dos algoritmos evolutivos multiobjetivo com elitismo:

- Como o arquivo externo interage com a população principal?
- O que fazer quando o arquivo externo estiver cheio?
- Deve-se impor critérios adicionais para um indivíduo entrar no arquivo externo, ou apenas se usa a dominância de Pareto?

Os algoritmos evolutivos multiobjetivo com elitismo mais representativos serão descritos nas próximas seções.

4.3.6 *Strength Pareto Evolutionary Algorithm*

Zitzler; Thiele (1998a) propuseram um algoritmo evolutivos elitista que chamaram de *Strength Pareto Evolutionary Algorithm* (SPEA). O SPEA introduz elitismo ao manter explicitamente uma população externa $\bar{P}$. Nesta população é armazenado um número fixo de soluções não-dominadas encontradas desde o início da execução do algoritmo. Em cada iteração, deve-se verificar entre todos os indivíduos (da população atual e da população externa) quais são os não-dominados para que estes possam fazer parte da população externa. Abaixo, algumas características importantes do SPEA.

- O SPEA aplica uma técnica de agrupamento para remover elementos do arquivo externo quando este estiver cheio seguindo a meta de preencher a fronteira de Pareto uniformemente.
- A atribuição do valor de aptidão é realizada baseada somente na população externa. Desta forma, o elitismo do SPEA força com que os indivíduos de cada geração compitam com os indivíduos da população externa.
- Todas as soluções da população externa participam da seleção.
- Um método de nicho é aplicado no intuito de preservar a diversidade da população.

4.3.7 *SPEA2:Strength Pareto Evolutionary Algorithm*

Zitzler; Laumanns; Thiele (2001) publicaram um aperfeiçoamento do SPEA e o chamaram de SPEA2. As principais diferenças em relação ao SPEA são: (1) o SPEA2 incorpora uma estratégia de atribuição de aptidão refinada que leva em conta, para cada indivíduo, o número de indivíduos que o dominam e o número de indivíduos que ele domina; (2) no SPEA2, usa-se uma estimativa de densidade de vizinhança que guia a busca de forma mais eficiente e (3) o método de agrupamento é mais eficiente do que o do SPEA, uma vez que preserva as soluções limitantes da fronteira de Pareto. No algoritmo SPEA estas soluções podem ser perdidas durante o processo evolutivo. Mais detalhes sobre o SPEA2 serão apresentados no Capítulo 6.

4.3.8 *Pareto Archived Evolution Strategy*

Algoritmo *Pareto Archived Evolution Strategy* (PAES) foi proposto por Knowles; Corne (1999) e usa a estratégia evolutiva (1 + 1), isto é, um pai gera um único filho. O indivíduo filho é comparado com o indivíduo pai. Se o filho dominar o pai, então o filho é aceito para a próxima geração. Se o pai dominar o filho, então o filho é descartado. No caso de nenhum dominar o outro, a escolha entre filho e pai será realizada comparando ambos com um arquivo em que estão as melhores soluções encontradas até o momento. O indivíduo filho é comparado com cada um dos indivíduos do arquivo externo. O filho será aceito para a próxima geração se dominar algum indivíduo do arquivo externo e todos os dominados serão removidos do arquivo externo. Caso o indivíduo filho não domine nenhum indivíduo do arquivo externo, então filho e pai serão analisados para verificar a distância que estão das soluções não-dominadas. Se o filho estiver em uma região de baixa densidade, então ele é aceito para a próxima geração e uma cópia do filho é adicionada ao arquivo externo. Caso ambos estejam em região de densidade igual, então será

selecionado um aleatoriamente. Nos trabalhos (Corne et al., 2001, Corne; Knowles; Oates, 2000) são encontrados aperfeiçoamentos do PAES.

4.3.9 *Nondominated Sorting Genetic Algorithms-II*

O algoritmo NSGA-II é uma modificação do NSGA proposta em (Deb et al., 2002) que usa (i) um método rápido de ordenação baseado em não-dominância; (ii) uma abordagem elitista e (iii) um método eficiente para determinar a distância de aglomeração. O NSGA-II é um dos poucos algoritmos evolutivos multiobjetivo que não usa um arquivo externo. Em vez disso, seu mecanismo de elitismo consiste em combinar os melhores pais com os melhores filhos obtidos (seleção $(\mu + \lambda)$). Mais detalhes sobre o NSGA-II serão apresentados no Capítulo 6.

4.4 Busca Tabu Para Otimização Multiobjetivo

A busca tabu (Glover, 1989, 1990, Glover; Laguna, 1997) é uma metaheurística que tem sido aplicada, de forma bem sucedida, para solução de problemas combinatórios mono-objetivo. Ela usa exploração adaptativa e memória flexível para guiar a busca no espaço de solução. Ao usar a exploração adaptativa, esta meta-heurística determina a direção de busca baseada em propriedades da solução atual e na história da busca. A memória flexível tem estruturas de curta e longa duração. A memória de curto prazo mantém informação sobre as soluções, ou atributos de soluções visitadas no passado recente. Determinados atributos encontrados em soluções recentemente visitadas são armazenados em uma lista, chamada de lista tabu, no intuito de prevenir ciclos. A memória de longo prazo contém informações sobre a história das soluções e seus atributos encontrados durante a busca.

A busca tabu começa a partir de uma solução inicial S pertencente à região de busca Υ . Uma vizinhança $\mathcal{N}(S)$ deve ser definida para S . Apenas uma parte da vizinhança $V^* \subset \mathcal{N}(S)$ é gerada a partir de S , uma vez que geralmente gerar toda a vizinhança de S é impraticável computacionalmente. Das soluções candidatas V^* , seleciona-se a melhor solução $S^* \in V^*$. A busca tabu impõe que seja mantida em uma memória T os últimos movimentos realizados recentemente no intuito de evitar ciclos. Portanto, considerando um problema de minimização, em cada iteração deve-se mover para a melhor solução $\{S^* : S^* = \min\{V^* \setminus T\}\}$. O Algoritmo 3 delinea os principais passos da busca tabu. Os dois principais parâmetros da busca tabu são o tamanho da lista tabu e o tempo tabu, ou seja, o número de iterações que um elemento permanecerá tabu.

Algoritmo 3: Principais passos da busca tabu

1	Gerar uma solução inicial $S \in \Upsilon$.
2	Iniciar a lista tabu: $T = \emptyset$
3	Para um número máximo de iterações faça
4	Gerar uma vizinhança $V^* \subset \mathcal{N}(S)$.
5	Mover para a melhor solução $\{S^* : S^* = \min\{V^* \setminus T\}\}$.
6	Atualizar a lista tabu $T = T \cup S^*$.
7	Atualizar a solução atual $S = S^*$
8	fim
-

Existem algumas propostas no intuito de aplicar a busca tabu para solução de problemas

multiobjetivos. Baykasoglu (2001) propôs uma abordagem para resolver problemas multiobjetivo usando busca tabu. Em sua abordagem, Baykasoglu (2001) sugere o uso de duas listas. A Primeira, chamada de lista Pareto, armazena as soluções não-dominadas encontradas durante a busca. A segunda lista é chamada de lista candidata e armazena as soluções que são globalmente não-dominadas, mas foram localmente não-dominadas em algum estágio da busca. As soluções da lista candidata são usadas durante a busca para promover diversificação.

Ho et al. (2002) aplica algumas modificações à busca tabu básica para resolver problemas multiobjetivos. Nessa abordagem, são adotados a ordenação Pareto (similar ao esquema de ordenação de Fonseca; Fleming (1993)), um arquivo externo e a ideia de compartilhamento de valor de adaptação. Para uma lista mais extensa de aplicação da busca tabu e outras metaheurísticas para solução de problemas multiobjetivo remetemos o leitor para (Coello; Lamont; Van Veldhuizen, 2007).

Nos estudos comparativos publicados em (Fonseca et al., 2005, Zitzler; Deb; Thiele, 2000) e (Deb et al., 2002) os algoritmos SPEA2 e NSGA-II apresentaram os melhores resultados. Com base nestes estudos, realizamos a implementação dos algoritmos SPEA2 e NSGA-II para solução do problema de Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica com múltiplos cenários de geração. Os detalhes das implementações realizadas serão mostrados no Capítulo 6.

Capítulo 5

Introdução à Computação Paralela

Nas últimas décadas se têm presenciado grandes avanços na tecnologia de microprocessadores. Porém a necessidade de desempenho aumenta em velocidade maior do que a tecnologia de desenvolvimento de microprocessadores pode suportar. Além disso, os limites físicos estão próximos de serem alcançados pelas tecnologias de desenvolvimento de semicondutores. No intuito de acompanhar a demanda por processamento contornando os limites físicos, surgem as arquiteturas de computadores paralelos.

Culler; Singh; Gupta (1999) definem computador paralelo como: *“uma coleção de elementos de processamento que cooperam e se comunicam no intuito de resolver um problema grande rapidamente”*. Hwang; Faye (1984) definem processamento paralelo como uma: *“Forma eficiente do processamento de informações com ênfase na exploração de eventos concorrentes no processo computacional”*.

Portanto, o processamento paralelo implica a divisão de um grande problema em partes menores que devem ser tratadas concomitantemente em elementos de processamento diferentes. Neste ambiente, se faz necessário um meio de comunicação que permita aos elementos de processamento trocarem informações a respeito do que está sendo ou já foi processado.

Neste capítulo serão detalhados os principais conceitos relacionados à computação paralela. Na Seção 5.1 será realizada uma introdução às principais arquiteturas de máquinas paralelas. Na Seção 5.2 serão levantados aspectos relacionadas à paralelização de algoritmos evolutivos multiobjetivos.

5.1 Classificação de Arquiteturas Paralelas

As arquiteturas das máquinas paralelas passaram por uma diversificação considerável durante a década de 1990 (Culler; Singh; Gupta, 1999, Duncan, 1990, Rajkumar, 1999). Por isso, se faz importante formalizar uma taxonomia no intuito de facilitar o estudo dos computadores paralelos.

5.1.1 Taxonomia de Flynn

A taxonomia de Flynn (1966) é o esquema de classificação mais conhecido para computadores paralelos. Segundo este esquema, a categoria de um computador é baseada no paralelismo exibido no fluxo de dados e no fluxo de instruções. Portanto, um computador pode suportar um fluxo simples ou múltiplo de instruções manipulando um fluxo simples ou múltiplo de dados. Daí surgem quatro categorias, conforme se segue.

SISD

A categoria SISD (*single instruction, single data*) refere-se aos computadores que executam um fluxo simples de instrução a um fluxo simples de dados. Desta forma, computadores pertencentes a esta categoria são computadores sequenciais, e nenhum paralelismo é executado. Os primeiros computadores pertenciam a esta categoria. A Figura 5.1 ilustra esta categoria de computadores.

Figura 5.1: Modelo SISD. (Adaptado de Almasi; Gottlieb (1994))

Na Figura 5.1 e nas que seguem:

- **FI**-Fluxo de instruções
- **M**-Memória
- **FD**-Fluxo de dados
- **UC**-Unidade de Controle
- **UP**-Unidade de Processamento

SIMD

A categoria SIMD (*single instruction, multiple data*) refere-se aos computadores em que cada unidade de processamento pode operar em um elemento de dados diferente. Os computadores vetoriais e matriciais se enquadram nesta categoria. Arquiteturas SIMD são utilizadas, geralmente, para manipulação de matrizes e processamento de imagens (Almasi; Gottlieb, 1994). A Figura 5.2 ilustra esta categoria de computadores.

Figura 5.2: Modelo SIMD. (Adaptado de Almasi; Gottlieb (1994))

MISD

A categoria MISD (*multiple instruction, single data*) refere-se aos computadores que executam múltiplos fluxos de instruções diferentes sobre um único fluxo de dados. Não existe computador representante desta categoria.

MIMD

A categoria MIMD (*multiple instruction Stream, multiple data stream*) refere-se aos computadores em que múltiplas unidades de processamento executam diferentes instruções em diferentes conjuntos de dados, de maneira independente. A maioria dos computadores paralelos se enquadra nesta categoria (Almasi; Gottlieb, 1994). A Figura 5.3 ilustra esta categoria de computadores.

Figura 5.3: Modelo MIMD. (Adaptado de Almasi; Gottlieb (1994))

Atualmente existem três tipos de arquiteturas de computadores paralelos, classificadas de acordo com a arquitetura da memória, bastante usadas que podem ser consideradas pertencentes à categoria MIMD. As três arquiteturas são detalhadas abaixo.

- Multiprocessador Simétrico (*Symmetric Multiprocessor* (SMP)) - arquitetura MIMD com arquitetura de memória compartilhada.
- Acesso Não-Uniforme à Memória (*Non-Uniform Memory Access* (NUMA)) - arquitetura MIMD com arquitetura de memória compartilhada.
- Aglomerado de Computadores - arquitetura MIMD com arquitetura de memória distribuída.

5.1.2 Arquiteturas SMP

A arquitetura SMP consiste em um conjunto de computadores semelhantes que compartilham uma mesma memória através de um barramento, ou de alguma outra forma de conexão. Estes sistemas podem aumentar o desempenho de uma aplicação ou do processamento total do sistema. A Figura 5.4 representa uma arquitetura SMP.

Figura 5.4: Arquitetura SMP.

Um sistema com arquitetura SMP é controlado como um todo por um único sistema operacional incumbido de tornar transparente ao usuário a existência de vários processadores. Nesta arquitetura, qualquer processador pode acessar qualquer parte da memória, já que a memória é centralizada. Além disso, o tempo necessário para acessar a memória é igual para

todos os processadores (*Uniform Memory Access* - UMA). Para mais detalhes desta arquitetura sugere-se a leitura de (Culler; Singh; Gupta, 1999).

5.1.3 Acesso Não-Uniforme à Memória

A existência de um barramento limita as arquiteturas SMP a apenas algumas CPUs. Uma alternativa é distribuir a memória primária entre os processadores, criando um sistema em que o acesso à memória local é muito mais rápido do que o acesso à memória não local. Se o conjunto de memórias distribuídas formar um único espaço lógico de endereçamento, o computador paralelo é chamado de multiprocessador distribuído. Este tipo de sistema pode ser chamado também de multiprocessador com acesso não-uniforme à memória (*nonuniform memory access* - NUMA), por que o tempo de acesso à memória lógica pode variar consideravelmente dependendo se o endereço físico pertence ao processador que está tentando acessar o endereço lógico, ou se pertence a outro processador. A Figura 5.5 representa uma arquitetura NUMA. Em (Quinn, 2003) são dados mais detalhes a respeito desta arquitetura.

Figura 5.5: Arquitetura NUMA.

5.1.4 Aglomerado de Computadores

Um aglomerado ou *cluster* de computadores é um computador paralelo formado por computadores interconectados, que podem trabalhar juntos, fazendo com que o programador enxergue o computador paralelo como uma máquina única.

O primeiro aglomerado de computadores foi desenvolvido em 1994 pelos cientistas espaciais da NASA Thomas Sterling e Dom Becker. O computador foi montado inteiramente por hardware fora de uso e softwares gratuitos disponíveis. O sistema era composto por 16 processadores Intel DX4 e conectados por *links* Ethernet de 10 Mbit/sec. Em relação a softwares, o *cluster* rodava o sistema Linux, o compilador GNU, e dava suporte à programação paralela através da biblioteca de troca de mensagem MPI (Merkey, 2004., Quinn, 2003). O computador paralelo de Starling e Becker foi nomeado de *Beowulf*.

5.2 Modelos de Paralelização de Algoritmos Evolutivos Multiobjetivos

Embora as metaheurísticas sejam capazes de encontrar soluções aproximadas de grande qualidade para problemas de otimização combinatória, quando se objetiva fazer busca em regiões muito grandes o esforço computacional exigido pode ser bastante elevado, ou mesmo proibitivo. Com o advento e popularização dos computadores paralelos, vislumbrou-se a possibilidade de aplicá-los para aumentar a região de busca que uma metaheurística pode abordar sem que o tempo de resolução se torne proibitivo.

Quando se investiga a paralelização de uma metaheurística para solução de problemas mono-objetivo geralmente existem dois grandes objetivos em mente: 1) diminuir o tempo computacional necessário para uma metaheurística que já é capaz de encontrar soluções de boa qualidade, porém com tempo elevado; 2) melhorar a qualidade de solução, aumentando a região de busca, diversificação e intensificação, de uma metaheurística que não está obtendo soluções próximas da solução ótima para um determinado problema, ou instância de problema. Portanto, o paralelismo é uma forma possível de não apenas reduzir o tempo de execução de um algoritmo de busca local e metaheurísticas, mas também de melhorar sua eficiência e robustez (Cung et al., 2002).

Existem diversas propostas de classificação de metaheurísticas mono-objetivas paralelas, conforme se pode ver nas referências (Alba, 2005, Crainic; Toulouse, 2002, 2003, Cung et al., 2002). Quando se lida com problemas multiobjetivos, os algoritmos paralelos podem ser projetados e implementados para aumento do seu desempenho de busca. Além disso, a paralelização pode ser usada para melhorar a precisão dos modelos matemáticos, a qualidade das fronteiras Pareto obtidas, a robustez das soluções obtidas e para se poder resolver problemas, ou instâncias, de grande porte.

Alguns modelos diferentes para implementar metaheurísticas paralelas têm sido propostos na literatura. Podemos classificá-los como segue (Miettinen; Ruiz; Wierzbicki, 2008, Talbi, 2009):

- Cooperação paralela autônoma
- Paralelização intra-algoritmo independente do problema
- Paralelização intra-algoritmo dependente do problema

sendo que os dois últimos modelos não alteram o comportamento do algoritmo e portanto são geralmente utilizado para aumentar a velocidade da busca.

5.2.1 Cooperação Paralela Autônoma

Este modelo de algoritmos evolutivos paralelos são baseados em ilhas e utilizados em sistemas com comunicação limitada. No modelo de ilhas, cada processador executa um Algoritmo Evolutivo Multiobjetivo (AEMO) usando uma (sub)população separada. O processador pode colaborar para solução global ao trocar, com certa regularidade, indivíduos migrantes com outros processadores. Este modelo é apropriado também para problemas que exigem uma população muito grande, sendo esta dividida em subpopulações entre os processadores.

Coello; Lamont; Van Veldhuizen (2007) destacam que os algoritmos pertencentes a este modelo podem ser subdivididos em dois grupos: 1) subpopulações cooperativas: neste modelo a população é dividida em subpopulações que cooperam entre si. O número de subpopulações e a forma com que a população é dividida são as principais questões levantadas para este modelo; 2) abordagem com múltiplos inícios: neste modelo, cada processador executa um algoritmo de otimização diferente, não há cooperação entre as populações.

Subpopulações Cooperativas

Estes algoritmos são aplicados visando distribuir a tarefa de encontrar a fronteira Pareto-ótima entre os processadores participantes. Assim, a população de um AEMO é dividida em um número

determinado de subpopulações independentes resultando em diversos AEMOs menores executando simultaneamente. Cada AEMO pode ter operadores e/ou valores de parâmetros diferentes. As referências (Baita et al., 1995, Hiroyasu; Miki; Watanabe, 2000, Muyl; Dumas; Herbert, 2004, Xiao; Armstrong, 2003) são exemplos de trabalhos que utilizam este modelo de programação multiobjetiva paralela. Para ilustrar, a Figura 5.6 mostra quatro ilhas, cada qual com população composta de oito indivíduos. O processo migratório consiste na migração de um indivíduo para as ilhas mais próximas.

Figura 5.6: *Subpopulações cooperativas.*

Na prática, estamos lidando com uma superpopulação de $4 \times 8 = 32$ indivíduos. Veja que o processo migratório entre as ilhas pode ajudá-las tanto na obtenção de uma fronteira Pareto mais próxima da fronteira Pareto-ótima, como para a obtenção de uma boa distribuição dos indivíduos da fronteira encontrada.

Abordagem com Múltiplo Inícios

Este modelo consiste em diversos algoritmos de busca local que executam independentemente em diversos processadores. O principal argumento para a aplicação deste modelo consiste na conjectura de que executar diversos algoritmos estocásticos de otimização com sementes diferentes pode gerar melhores resultados do que executar o mesmo algoritmo uma única vez por um longo período de tempo. As referências (Jozefowicz; Semet; Talbi, 2007, Mezma; Melab; Talbi, 2006) utilizam este modelo, ilustrado pela Figura 5.7.

Figura 5.7: *Abordagem com múltiplos inícios.*

5.2.2 Paralelização Intra-Algoritmo Independente do Problema

Como a maioria das metaheurísticas são métodos iterativos, este modelo visa paralelizar uma iteração do algoritmo. Assim, este modelo se preocupa apenas nos mecanismos de busca que sejam independentes do problema como, por exemplo, a avaliação de uma busca local na vizinhança.

Considerando um algoritmo de população, a sua paralelização aplicando este modelo pode ser realizada de forma que a avaliação da aptidão dos indivíduos seja feita em processadores diferentes. Este modelo é particularmente importante quando a avaliação da aptidão dos indivíduos exigem um esforço elevado, como é o caso do problema abordado nesta tese e outros problemas como a análise computacional da dinâmica dos fluidos, análise de campos elétrico-magnéticos, métodos de elementos finitos (Okabe, 2004, Okabe et al., 2003).

Assumindo que n é o número de indivíduos de uma população, que g é o número máximo de gerações, que k é o número de objetivos, e que t_i é o tempo total para calcular a aptidão de um indivíduo para o objetivo i . O tempo total esperado para convergência de um algoritmo evolutivo de otimização multiobjetiva, denotado por T , pode ser calculado conforme a Eq. (5.1).

$$T = gn \sum_{i=1}^k t_i + \alpha g = g(nt + \alpha) \quad (5.1)$$

onde α é o tempo médio para execução dos operadores genéticos de uma iteração e $t = \sum_{i=1}^k t_i$.

Exemplo 1: Considerando-se a abordagem do problema de planejamento da expansão de redes de transmissão com múltiplos cenários de geração proposto neste trabalho. Supondo-se que o algoritmo de geração de cenários indique, para um determinado sistema de grande porte, a geração de uma amostra com $N_{cenarios} = 6000$ cenários. Para cada cenário supõe-se que seja necessário resolver um PL que demore $0,07s$ ($T_{PL} = 0,07$). Portanto, $t \simeq 6.000 \times 0,07 = 7min$. Se $n = 300$, $\alpha \simeq 0$ e $g = 100$, então

$$T = 100 \times 300 \times 7 \simeq 146 \text{ dias!} \quad (5.2)$$

Para lidar com problemas como este, que exigem um grande esforço computacional, geralmente é usado um cálculo paralelo dos indivíduos. A Figura 5.8 ilustra a ideia básica deste modelo. Este tipo de paralelização é chamada de modelo administrador/trabalhador (Foster, 1995) ou modelo mestre/escravo (Branke et al., 2004, Cantu-Paz, 1998, Cantú-Paz, 1998).

Figura 5.8: Modelo administrador - trabalhador.

Como ilustrado na Figura 5.8, o processador mestre centraliza a população e sempre que precisar calcular o valor de adaptação de um indivíduo submete este indivíduo a um processador escravo ocioso. O processador escravo fica incumbido de calcular o valor de adaptação e enviar o

resultado ao processador mestre.

Assumindo-se agora, considerando o modelo mestre/escravo, que T^s é o tempo total de execução de um algoritmo sem paralelização; que T^c é o tempo total de execução com paralelização e N é o número de nós, processadores, usados. Considerando ainda que $N = n$ (o número de processadores disponíveis é igual ao número de indivíduos da população), que o processador mestre também participa como escravo e que T_{oh} é o tempo gasto para montar as estruturas de comunicação mais o tempo de transferência. As Eqs. (5.3-5.4) podem ser utilizadas para calcular T^s e T^c , respectivamente.

$$T^s = gnt + g\alpha = gNt + g\alpha \quad (5.3)$$

$$T^c = g\alpha + gt + g(N-1)T_{oh} \quad (5.4)$$

Podemos aferir o benefício da paralelização em relação à execução sequencial através da Eq. (5.5).

$$\eta = \frac{T^s}{NT^c} \times 100\% \quad (5.5)$$

De acordo com a Eq. (5.5), quanto mais próximo de 100% estiver η , maior é o ganho com a paralelização.

Utilizando as Eqs. (5.3-5.4) em Eq. (5.5), obtemos:

$$\eta = \frac{Nt + \alpha}{N(\alpha + t + (N-1)T_{oh})} \times 100\% \quad (5.6)$$

Considerando que $\alpha \simeq 0$, a Eq. (5.6) é reduzida à Eq. (5.7).

$$\eta = \frac{1}{1 + \frac{(N-1)T_{oh}}{t}} \times 100\% \quad (5.7)$$

Pela Eq. (5.7), podemos inferir formalmente que quanto maior for o tempo para o cálculo do valor de aptidão e mais veloz for o meio de comunicação, maior será o benefício decorrente da paralelização. Para o exemplo dado no início desta seção, considere que o tempo gasto entre comunicação e preparação dos dados para transferência seja de 0,300s. A Eq. (5.8) calcula o benefício esperado da paralelização.

$$\eta = \frac{1}{1 + \frac{299 \times 0,005}{7}} = 82,41\% \quad (5.8)$$

Conclui-se que, para os dados levados em conta, podemos diminuir o tempo necessário para realizar o planejamento do sistema considerado em 82,41%. Ou seja, em vez de levar 146 dias, levaria 26 dias. A hibridização com o modelo de cooperação paralela autônoma pode diminuir ainda mais o tempo de execução.

5.2.3 Paralelização Intra-Algoritmo Dependente do Problema

Este modelo visa paralelizar operações específicas do problema que está sendo abordado. Este tipo de modelo pode ser implementado realizando a decomposição do problema baseado no

domínio ou baseada na função.

Na abordagem de decomposição baseada no domínio procuramos primeiro decompor os dados associados a um problema. Se possível, dividem-se esses dados em pequenos pedaços de tamanho aproximadamente igual. Em seguida, particiona-se a computação que deverá ser realizada (Foster, 1995). Considerando um algoritmo populacional, a decomposição baseada no domínio pode ser implementada considerando que a população é uma estrutura de dados em que cada uma de suas partes (os indivíduos) serão tratadas por processadores diferentes.

Segundo Foster (1995), a decomposição funcional representa uma maneira diferente e complementar de pensar no problema a ser abordado. Nesta abordagem, o foco inicial é sobre o cálculo que deve ser executada em vez de sobre os dados manipulados pela computação. Considerando um algoritmo multiobjetivo, a decomposição funcional pode ser implementada paralelizando o cálculo das funções objetivo constituintes dos problema.

Considerando o Exemplo 1, uma opção seria realizar o cálculo paralelo do valor de aptidão de cada indivíduo (decomposição funcional). O tempo total de execução esperado pode ser calculado pela Eq. (5.9).

$$T_{exe} = \left(T_{PL} \frac{N_{cenarios}}{N} + T_{oh} \right) gn + g\alpha \quad (5.9)$$

Como nos exemplos anteriores, consideramos $\alpha \simeq 0$. Considerando os mesmos valores de T_{PL} , N , $N_{cenarios}$ e T_{oh} da seção anterior, temos que.

$$T_{exe} = \left(0,07 \times \frac{6000}{300} + 0,300 \right) \times 100 \times 300 = 5,93 \times 10^{-1} \text{ dia.} \quad (5.10)$$

Portanto, considerando este modelo, os 146 dias necessários para resolver o sistema do Exemplo 1 podem ser reduzidos a apenas algumas horas!

Assim, ao se aplicar a abordagem proposta nesta tese a sistemas de grande porte, é muito provável que o melhor modelo de decomposição a ser utilizado seja o modelo de paralelização intra-algoritmo dependente do problema. Caso as instâncias abordadas necessitem de mais gerações e/ou uma população maior do que as consideradas nos exemplos aqui analisados, sugere-se que a melhor estratégia seja a hibridização do modelo desta seção com o modelo de cooperação paralela autônoma, conforme detalhado a seguir.

Uma fórmula para o tempo esperado de resposta de um algoritmo que implemente a hibridização dos modelos de cooperação paralela autônoma com ilhas e o apresentado nesta seção deve abranger duas partes: 1) o tempo esperado de execução do algoritmo evolutivo dentro de cada ilha e 2) tempo de migração entre as ilhas.

O tempo esperado de execução do algoritmo evolutivo dentro das ilhas é semelhante à Eq. (5.9). Conforme a Eq. (5.11).

$$T_{exe1} = \left(T_{PL} \frac{N_{cenarios}}{N_{por_ilha}} + T_{oh} \right) g_i \times n_{por_ilha} + g_i\alpha \quad (5.11)$$

onde N_{por_ilha} é o número de processadores por ilha, dado por N/n_i , sendo n_i o número de ilhas; g_i é o número de gerações executados em cada ilha antes de iniciar o processo de migração e n_{por_ilha} é o número de indivíduos pertencentes à população da ilha, dado por n/n_i .

A segunda parte da equação que modela o tempo esperado do modelo hibridizado é composto

pelo tempo gasto devido à comunicação entre as ilhas, conforme a Eq. (5.12).

$$T_{exe2} = n_{ipm} \times n_i \times T_{oh} \times g_e \quad (5.12)$$

onde g_e é o número de gerações externas e n_{ipm} é o número de indivíduos que devem migrar por iteração.

Uma vez que a parte T_{exe1} deve executar g_e vezes, temos que o tempo esperado da hibridização é dado por $T_{hib} = T_{exe2} + g_e \times T_{exe1}$, ou conforme a Eq (5.13).

$$T_{hib} = g_e \left[n_{ipm} \times n_i \times T_{oh} + \left(T_{PL} \frac{N_{cenarios}}{N_{por_ilha}} + T_{oh} \right) g_i \times n_{por_ilha} + g_i \alpha \right] \quad (5.13)$$

Supondo-se que se deseja quintuplicar o número de gerações e duplicar o tamanho da população. Vamos calcular o tempo esperado de execução da hibridização considerando: $g_e = 25$, $g_i = 20$ ($g_i \times g_e = 500$). Antes de seguir no raciocínio, destaca-se que os parâmetros g_e e g_i têm que ser definidos com cuidado. Veja que se $g_i = 1$ estamos lidando somente com o modelo de cooperação paralela autônoma. Se $g_e = 1$, estamos lidando somente como modelo de paralelização intra-algoritmo dependente do problema. Uma vez que o modelo de paralelização intra-algoritmo dependente do problema visa diminuir o tempo de execução e o modelo de cooperação paralela autônoma visa aumentar a região de busca, deve-se lidar com estes dois parâmetros de acordo com a necessidade da instância em abordagem. No nosso caso, estamos satisfeitos com o tempo dado pelo modelo de paralelização intra-algoritmo dependente do problema. Almeja-se agora aumentar a região de busca, já que hipoteticamente não se está satisfeito com a qualidade da solução. Por isso, atribuímos um valor para g_e maior do que para g_i .

Digamos que queremos dobrar a população distribuindo-a entre as ilhas. Portanto, se anteriormente $n = 300$, agora vamos distribuir os 600 indivíduos equitativamente entre as ilhas. Aqui também será considerando que $\alpha \simeq 0$. A cada iteração consideramos que migra-se de cada ilha um indivíduo para a ilha sucessora mais próxima e um indivíduo para a ilha antecessora mais próxima, assim $n_{ipm} = 2$. Uma vez definidos os parâmetros base, na Figura 5.9 ilustramos o ganho de desempenho em relação ao aumento do número de ilhas que podemos utilizar. Pela referida figura, vemos que o número de ilhas ideal é 77, com o qual conseguimos um tempo computacional de $7min02s$. Se utilizássemos somente o modelo de paralelização intra-algoritmo dependente do problema, o tempo esperado seria de $8min30s$, ou seja, 17,19% mais lento.

Figura 5.9: *Ganho de desempenho em relação ao número de ilhas.*

Um outro parâmetro muito importante é o número de indivíduos que devem migrar. Quanto maior for este número maior é o ganho com diversificação e intensificação da busca. Em outras palavras, maior é a cooperação entre as ilhas. Intuitivamente, pode-se pensar que o aumento deste número pode trazer muito prejuízo ao esforço computacional. A Figura 5.10 ilustra o aumento do custo computacional à medida que n_{ipm} varia entre 1 e 150.

Figura 5.10: *Diminuição do desempenho do modelo hibridizado à medida que aumenta n_{ipm} .*

Através da Figura 5.10 podemos concluir que o aumento do parâmetro n_{ipm} , principalmente quando seguido do aumento do número de ilhas, tem como consequência o aumento do esforço computacional. Já pela Figura 5.11, podemos inferir que à medida que n_{ipm} cresce, devemos diminuir o número de ilhas. Na Figura 5.10, o maior esforço computacional mínimo foi de $T_{hib} = 7min20s$. Portanto, o aumento de apenas 18 segundos pode ser vantajoso, uma vez que com a troca de 150 indivíduos entre as ilhas pode aumentar a qualidade da solução final, já que 150 indivíduos migrando pode representar uma grande cooperação entre as ilhas.

Figura 5.11: Diminuição do número de ilhas em que ocorre o menor esforço esperado à medida que aumenta n_{ipm} .

É fácil perceber que o problema de busca pelo máximo de intensificação/diversificação juntamente com a busca pelo mínimo de perda de desempenho é um problema de otimização multiobjetivo.

Nos próximos capítulos serão apresentados detalhes a respeito da paralelização dos algoritmos evolutivo multiobjetivos implementados nesta tese, bem como detalhes dos resultados obtidos através da execução dos algoritmos paralelos aplicados ao problema objeto deste trabalho.

Capítulo 6

Aplicação de Algoritmos Multiobjetivo ao PERTEE

Neste capítulo, serão apresentados detalhes a respeito das implementações sequenciais e paralelas dos algoritmos NSGA-II e SPEA2 realizadas.

Como mencionado no Capítulo 3, a proposta deste trabalho para lidar com o Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica considerando múltiplos cenários de geração resulta em um modelo com mais de um objetivo. Portanto, foram implementados dois algoritmos multiobjetivo no intuito de compará-lo e verificar se a abordagem do PERTEE com múltiplos cenários de geração aqui proposta é viável do ponto de vista econômico e computacional.

O problema multiobjetivo proposto neste trabalho tem duas funções objetivo: a minimização do investimento e a maximização do atendimento aos múltiplos cenários de geração da amostra de cenários gerada. Conforme menciona Zuluaga (2008), não é possível associar ao congestionamento um custo presente já que este custo somente pode ser calculado no futuro, com o aparecimento de cenários reais de geração para os quais existam limitações de transmissão e seja necessário fazer um redespacho considerando as usinas de geração disponíveis.

Embora o tomador de decisão não tenha o custo presente associado ao congestionamento futuro do Sistema Elétrico de Potência (SEP), a metodologia proposta neste trabalho será capaz de fornecer além do custo de investimento, o nível de segurança em relação ao congestionamento ao antever os cenários mais propensos de ocorrer na operação futura do sistema.

6.1 Detalhamento dos Algoritmos Multiobjetivo Sequenciais

Nesta seção serão detalhados as implementações sequenciais realizadas dos algoritmos NSGA-II e SPEA2.

6.1.1 NSGA-II Aplicado ao PERTEE Com Múltiplos Cenários de Geração

O algoritmo NSGA-II começa com uma população inicial $\mathbb{P}_1$, de tamanho M , da qual uma descendência $\mathbb{Q}_t$ é criada. Ambas as populações são combinadas formando uma população $\mathbb{R}_t$ de tamanho $2M$. Um algoritmo de ordenação por não-dominância é aplicado à população $\mathbb{R}_t$. Uma vez ordenadas, as soluções pertencentes às primeiras fronteiras de Pareto-ótima são atribuídas à nova população $\mathbb{P}_{t+1}$ até que $|\mathbb{P}_{t+1}| = M$. Nem todas as fronteiras caberão por completo em $\mathbb{R}_t$. Portanto, a última fronteira que será usada para preencher $\mathbb{P}_{t+1}$ deverá passar por uma ordenação por distância de aglomeração para que somente os indivíduos presentes em regiões menos densas passem para a próxima geração. A Figura 6.1 ilustra este procedimento de definição dos indivíduos que deverão passar para a próxima geração. O Algoritmo 4 apresenta os principais passos do método NSGA-II. No Algoritmo 4, F_i representa o conjunto formado pelas soluções pertencentes

à fronteira i .

Figura 6.1: Ilustração do algoritmo NSGA-II.

Algoritmo 4: Algoritmo NSGA-II

- 1 Gerar uma população inicial $\mathbb{P}_t$ de tamanho M e uma descendência $\mathbb{Q}_t$, também de tamanho M , dessa população.
 - 2 Atribuir $\mathbb{R}_t = \mathbb{Q}_t \cup \mathbb{P}_t$.
 - 3 Aplicar a ordenação por não-dominância a $\mathbb{R}_t$ identificando as fronteiras:
 $F_i, i = 1, 2, \dots$, etc.
 - 4 Atribuir $\mathbb{P}_{t+1} = \emptyset$ e $i = 1$.
 - 5 **Enquanto** $|\mathbb{P}_{t+1}| + |F_i| < M$ **faça**
 - 6 Atribuir $\mathbb{P}_{t+1} = \mathbb{P}_{t+1} \cup F_i$
 - 7 $i = i + 1$
 - 8 **fim**
 - 9 Realizar uma ordenação de F_i , que é a fronteira que não coube por completo em $\mathbb{P}_{t+1}$, por distância de aglomeração.
 - 10 Inserir $M - |\mathbb{P}_{t+1}|$ elementos de F_i em $\mathbb{P}_{t+1}$.
 - 11 Criar uma população $\mathbb{Q}_{t+1}$ a partir de $\mathbb{P}_{t+1}$ usando os operadores de seleção por torneio de distância de aglomeração, recombinação e mutação. Enquanto um critério de parada não for satisfeito, voltar ao passo 2.
-

Na linha 3 do Algoritmo 4 é definida a que fronteira de Pareto pertence cada um dos elementos da população $\mathbb{R}_t$. Na linha 9 é feita referência a uma ordenação por distância de aglomeração. Esta função ordena a população de tal forma que os elementos localizados em regiões pouco populosas e as soluções limitantes sejam agrupadas para ficarem nas primeiras posições da população. O Algoritmo 5 delinea a função de atribuição de distância de aglomeração para cada elemento de uma população.

Algoritmo 5: Função de atribuição de distância de aglomeração

```

1  Atribuir  $l = |F|$  ▷ Número de elementos pertencentes à fronteira F
2  Para cada indivíduo  $i$  em F faça
3      | Atribuir  $F_{distância}^i = 0$ 
4  fim
5  Para cada objetivo  $k$  faça
6      | Executar  $ordenar(F, k)$  ▷ Ordenar F em ordem crescente de  $k$ 
7      | Atribuir  $F_{distância}^1 = F_{distância}^l = \infty$  ▷ Os indivíduos limitantes
        | da fronteira serão sempre selecionados.
8      | Para  $i = 2, (l - 1)$  faça
9      |     |  $F_{distância}^i = F_{distância}^i + \frac{(F_k^{(i+1)} - F_k^{(i-1)})}{f_k^{max} - f_k^{min}}$ 
10     | fim
11  fim
```

Os parâmetros f_k^{max} e f_k^{min} podem ser os valores de máximo e mínimo, respectivamente, da k -ésima função objetivo da população. O segundo termo do lado direito da atribuição realizada na linha 9 do Algoritmo 5 representa a metade do perímetro do cuboide formado com as soluções vizinhas mais próximas de i postas nos vértices do cuboide (Figura 6.2).

Figura 6.2: Cálculo de distância de aglomeração.

Portanto, as soluções com maior valor para $F_{distância}^i$ serão aquelas que formarem o cuboide com maior perímetro, em outras palavras, as que estiverem em regiões com poucas soluções em sua vizinhança.

Operador de Distância de Aglomeração

Na linha 11 do Algoritmo 4 é mencionado que uma nova geração deve ser criada a partir do operador de distância de aglomeração. Este operador visa determinar quais indivíduos estão mais aptos a alcançar as duas metas da programação multiobjetivo: encontrar soluções próximas a fronteira Pareto-ótima e gerar soluções o mais uniformemente distribuídas ao longo da fronteira Pareto-ótima. Com este propósito, Deb et al. (2002) propuseram um operador de distância de aglomeração. Este operador compara duas soluções considerando dois atributos:

1. A ordem, ou nível, de não-dominância (F_{ordem}^i) na população.

2. A distância de aglomeração local ($F_{distância}^i$) na população.

Desta forma, postas duas soluções i e j para competirem entre si, a solução i vencerá a competição se qualquer uma das condições seguintes for satisfeita:

1. Se i tiver uma ordem de não-dominância em relação a j ($F_{ordem}^i < F_{ordem}^j$).
2. Se i e j tiverem a mesma ordem de não-dominância, mas i tiver uma distância de aglomeração maior do que j ($F_{ordem}^i = F_{ordem}^j \wedge F_{distância}^i > F_{distância}^j$).

Em outras palavras, no momento da seleção dos indivíduos para a nova geração, será dada prioridade primeiramente aos indivíduos não-dominados, e depois aos indivíduos localizados em regiões da fronteira Pareto não muito aglomeradas.

Adequação do Algoritmo NSGA-II Para o PERTEE Com Múltiplos Cenários de Geração

A adequação do algoritmo NSGA-II para solução do PERTEE multiobjetivo consiste nas seguintes modificações realizadas:

Codificação do problema: A codificação do problema é um dos aspectos mais importantes para implementação do um AG. Em nossa abordagem, utilizamos uma codificação decimal para uma proposta de solução. Neste trabalho, a codificação é feita através de um vetor de tamanho igual ao número de ramos candidatos (nl), em que cada elemento representa um ramo candidato do sistema. O valor de cada elemento do vetor deve variar entre 0 e $\bar{n}_{ij}$, ou seja, o número máximo de linhas que podem ser adicionadas. A Figura 6.3 ilustra a proposta de codificação para um sistema fictício de quatro barras.

Figura 6.3: Um sistema fictício de quatro barras.

População inicial: para geração da solução inicial, aplicamos duas heurísticas construtivas modificadas. As heurísticas aplicadas foram a heurística construtiva proposta por Villasana; Garver; Salon (1985) e a heurística construtiva de mínimo esforço proposta por Monticelli et al. (1982).

A heurística de Villasana; Garver; Salon (1985) usa o modelo híbrido linear, descrito na Seção 2.1.1 do Capítulo 2, relaxando as restrições de integralidade. Após a solução do problema de

programação linear (PL) referente ao modelo, um Índice de Sensibilidade (IS) é usado para adicionar linhas ao sistema até que uma solução factível seja encontrada. O IS é definido pela Eq. (6.1).

$$IS = \operatorname{argmax}_{ij} \{n_{ij} \times \bar{f}_{ij} : n_{ij} \neq 0\} \quad (6.1)$$

Neste trabalho, uma modificação foi aplicada ao IS proposto por Villasana; Garver; Salon (1985). A modificação consiste em adicionar não necessariamente o maior valor de $n_{ij} \times \bar{f}_{ij}$ como na proposta de Villasana; Garver; Salon (1985). Na abordagem proposta nesta tese, selecionamos aleatoriamente $k_\alpha \geq 2$ linhas tais que $n_{ij} \neq 0$ e a que tiver o maior valor de IS será escolhida para ser adicionada à configuração atual. O objetivo desta modificação nos algoritmos heurísticos construtivos clássicos é aplicar a ideia de competição no momento de adicionar linhas tornando possível assim gerar diversas soluções factíveis diferentes a partir de uma única configuração inicial. Uma segunda modificação à heurística construtiva de Villasana; Garver; Salon (1985) foi formar um conjunto Λ com $k_\alpha \geq 2$ linhas aleatoriamente selecionadas tais que $n_{ij} \neq 0$ e aplicar o IS definido pela Eq. (6.2).

$$IS = \operatorname{argmax}_{ij} \{n_{ij} \times \bar{f}_{ij} \times \gamma_{ij} : n_{ij} \neq 0 \wedge ij \in \Lambda\} \quad (6.2)$$

A heurística de Monticelli et al. (1982) usa o modelo cc, descrito na Seção 2.1.1 do Capítulo 2, relaxando as restrições de integralidade. O IS sugerido por Monticelli et al. (1982) é definido pela Eq. (6.3).

$$IS = \operatorname{argmax}_{ij} \left\{ -\frac{1}{2}(\theta_i^* - \theta_j^*)^2 \gamma_{ij} \right\} \quad (6.3)$$

onde θ_i^* é o valor da variável θ_i na solução ótima do PL. Esta heurística construtiva também foi alterada para considerar este índice aplicado a apenas $k_\alpha \geq 2$ ramos. Na abordagem proposta nesta tese, foi definido $k_\alpha = 4$ para todas as heurística construtivas.

Desta forma, para gerar uma população inicial, atribuímos a configuração inicial a cada posição da população inicial e depois a submetemos ao Algoritmo 6. Neste trabalho, foi definido $M = 100$.

Seleção: para seleção, escolhem-se $k = 2$ indivíduos aleatoriamente para competirem entre si. O operador de distância de aglomeração é usada para definir qual será o vencedor.

Recombinação: com uma dada probabilidade de recombinação, realiza-se uma recombinação sobre os pais para formar uma nova descendência. Na abordagem proposta nesta tese, a taxa de recombinação ρ_c foi definida em 90%. O método de recombinação utilizamos foi o cruzamento de um ponto, conforme ilustrado na Figura 6.4. Neste trabalho, o ponto p (ponto a partir do qual ocorrerá a recombinação) é selecionado aleatoriamente entre $[0, 25nl]$ e $[0, 75nl]$.

Algoritmo 6: Gera população factível**Parâmetros:** População $\mathbb{P}$, de tamanho M , formada por indivíduos infactíveis.

```

1  Início
2      Atribuir  $i = 0$ 
3      Enquanto  $i \leq M$  faça
4           $i = i + 1$ 
5          Se  $i \leq M$  então
6              Aplicar a heurística usando a
              Eq. (6.1) com competição para
              tornar indivíduos  $P_i$  factível.
7          fim
8           $i = i + 1$ 
9          Se  $i \leq M$  então
10             Aplicar a heurística usando a
             Eq. (6.2) com competição para
             tornar indivíduos  $P_i$  factível.
11         fim
12          $i = i + 1$ 
13         Se  $i \leq M$  então
14             Aplicar a heurística usando a
             Eq. (6.3) com competição para
             tornar indivíduos  $P_i$  factível.
15         fim
16     fim
17 fim

```

	p											
	1											nl
Indivíduo 1	1	3	0	0	0	1	0	1	0	0	2	1
Indivíduo 2	0	0	0	1	0	2	1	0	0	0	1	2
Descendente 1	1	3	0	0	0	1	1	0	0	0	1	2
Descendente 2	0	0	0	1	0	2	0	1	0	0	2	1

Figura 6.4: Ilustração da recombinação de um ponto.

Como não há nenhuma garantia de que os indivíduo gerados pela recombinação sejam factíveis, após este procedimento todos os descendentes são submetidos ao Algoritmo 6.

Mutação: Os indivíduos recém gerados passarão por uma mutação com probabilidade ρ_m . Neste trabalho, definiu-se $\rho_m = 2\%$. Sugere-se que somente 5 alelos (ramos da rede de transmissão)

sejam modificados na mutação. A mutação foi implementada da seguinte forma: seja α_i o número de linhas (alelo) que há no ramo (gene) i . É gerado um número aleatório uniformemente distribuído entre 0 e 1. Se o número gerado for maior ou igual a 0,5, ou se $\alpha_i = 0$, faz-se $signal = 1$. Caso contrário, $signal = -1$. Agora gera-se um número aleatório (na) inteiro que pode receber os valores 1 ou 2. Depois desses passos, a mutação ocorre: $\alpha_i = \max\{0, \alpha_i + signal * na\}$.

Se um indivíduo for selecionado para mutação, uma cópia dele é preservada no conjunto ν . Todos os indivíduos que estiverem no conjunto ν sofrerão mutação. Após a mutação, todos os indivíduos pertencentes ao conjunto ν devem ser submetidos ao Algoritmo 6.

Uma vez concretizado o procedimento de mutação, haverá três conjuntos de indivíduos: a população atual ($\mathbb{P}_t$), os indivíduos resultantes da recombinação (ϑ) e o conjunto dos indivíduos que sofreram mutação (ν). A população descendente $\mathbb{Q}_t$ será formada pelos indivíduos de ϑ e ν . Formada a população descendente, as linhas 2-10 do Algoritmo 4 são executadas. O processo segue iterativamente até o número máximo de gerações ter sido alcançado.

6.1.2 SPEA2 Aplicado ao PERTEE com Múltiplos Cenários de Geração

Uma aproximação da fronteira Pareto-ótima envolve dois objetivos: minimização da distância da fronteira Pareto-ótima, e a maximização da diversidade das soluções geradas. Por isso, o algoritmo SPEA2 dá ênfase a duas fases importantes dos algoritmos multiobjetivos: seleção para reprodução e seleção para criar a nova população. A primeira questão está relacionada com o objetivo de minimizar a distância da fronteira Pareto-ótima e a segunda, com o objetivo de gerar soluções uniformemente distribuídas ao longo da fronteira Pareto-ótima. As duas fases mais críticas do algoritmo SPEA2 são a atribuição do valor de adaptação e a seleção dos indivíduos que formarão a próxima geração. Portanto, nas próximas seções serão detalhadas estas duas etapas do algoritmo SPEA2.

Atribuição do Valor de Adaptação

No algoritmo SPEA2, além da população atual ($\mathbb{P}_t$) é mantido um arquivo externo ($\overline{\mathbb{P}}_t$) formado pelas soluções não-dominadas encontradas durante a execução do algoritmo. A cada indivíduo i pertencente a $\overline{\mathbb{P}}_t$ e a $\mathbb{P}_t$ é atribuído um valor de força $S(i)$, que representa o número de soluções que este indivíduo domina, conforme a Eq. (6.4).

$$S(i) = |\{j : j \in \mathbb{P}_t \cup \overline{\mathbb{P}}_t \wedge i \succ j\}| \quad (6.4)$$

Uma vez determinado o número de soluções que cada um dos indivíduos domina, usa-se a Eq. (6.5) para definir a força de cada um dos dominadores.

$$R(i) = \sum_{j \in \mathbb{P}_t \cup \overline{\mathbb{P}}_t, j \succ i} S(j) \quad (6.5)$$

Em outras palavras, para cada indivíduo i , $R(i)$ representa o quanto esta solução é dominada. Assim, para soluções não-dominadas $R(i) = 0$. Porém, embora $R(i)$ seja um mecanismo de nicho baseado no conceito de dominância, ele falha quando muitos indivíduos não são dominados entre si. Como visto pela Figura 6.5, pode haver muitas soluções que não dominam umas às outras. Em situações como esta, o índice $R(i)$ aplicado sozinho irá falhar em relação ao objetivo de encontrar soluções bem distribuídas ao longo da fronteira Pareto-ótima.

Figura 6.5: Soluções má distribuídas ao longo da fronteira Pareto-ótima.

Por isso, o algoritmo SPEA2 incorpora uma informação a mais sobre a densidade da população. O estimador de densidade de um determinado ponto usado pelo SPEA2 é o inverso da distância do k -ésimo vizinho mais próximo do ponto. Detalhando: para cada indivíduo i , gera-se uma lista de distâncias¹ em relação a todos os outros indivíduos ($\mathbb{P}_t \cup \overline{\mathbb{P}}_t$). Gerada a lista de distâncias, esta deve ser ordenada em ordem crescente, e o k -ésimo elemento da lista, representado como σ_i^k , dá a distância de i . Conforme sugerido por Silverman Bernard (1986), k é definido como sendo igual à raiz quadrada do tamanho da amostra, portanto, $k = \lfloor \sqrt{|\mathbb{P}_t \cup \overline{\mathbb{P}}_t|} \rfloor$. Uma vez definida a distância do indivíduo i , sua estimação de densidade é armazenada na lista de densidades D conforme a Eq. (6.6).

$$D(i) = \frac{1}{\sigma_i^k + 2} \quad (6.6)$$

Soma-se o σ_i^k com dois para garantir que $D(i)$ seja maior do que zero e menor do que um. Finalmente, o valor de adaptação do indivíduo i pode ser atribuído conforme a Eq. (6.7).

$$F(i) = R(i) + D(i) \quad (6.7)$$

Seleção

Para seleção dos indivíduos que farão parte da próxima geração, dois cuidados são levados em conta pelo SPEA2: i) o número de indivíduos no arquivo externo é mantido constante durante toda a execução do algoritmo e ii) o método de truncamento deve evitar que as soluções limitantes sejam removidas.

Pela Eq. (6.7), podemos afirmar que os indivíduos não dominados terão valor de adaptação menor do que um. Isto porque a parte $R(i)$ será zero e foi tomado cuidado para que a parte $D(i)$ da equação seja menor do que um. Desta forma, o primeiro passo a ser realizado quando da seleção dos indivíduos a fazerem parte da nova geração é copiar para o arquivo externo aqueles que tiverem valor de adaptação menor do que um. Este processo pode ser definido formalmente pela Eq. (6.8).

¹Nesta tese foi considerada a distância euclidiana.

$$\bar{\mathbb{P}}_{t+1} = \{i : i \in \mathbb{P}_t \cup \bar{\mathbb{P}}_t \wedge F(i) < 1\} \quad (6.8)$$

Considerando que $\bar{M}$ é o tamanho do arquivo externo, $|\bar{\mathbb{P}}| = \bar{M}$, e que M é o tamanho da população do algoritmo genético, $|\mathbb{P}| = M$, da Eq. (6.8) três situações podem ocorrer, conforme listado abaixo.

1. $|\bar{\mathbb{P}}_{t+1}| = \bar{M}$: o número de soluções não-dominadas é exatamente igual ao tamanho máximo do arquivo externo. Neste caso, nada tem que ser feito e a próxima geração inicia.
2. $|\bar{\mathbb{P}}_{t+1}| < \bar{M}$: o número de soluções não-dominadas não é suficiente para preencher o arquivo externo. Neste caso, faz-se $\bar{\mathbb{R}}_t = \mathbb{P}_t \cup \bar{\mathbb{P}}_t$ e ordena-se a população $\bar{\mathbb{R}}_t$ em ordem crescente do valor de adaptação. Após estes passos, os primeiros M indivíduos de $\bar{\mathbb{R}}_t$ são selecionados para a próxima geração.
3. $|\bar{\mathbb{P}}_{t+1}| > \bar{M}$: o número de soluções não-dominadas é maior do que o tamanho máximo do arquivo externo. Nesta situação, o método de truncamento é acionado no intuito de remover indivíduos de $\bar{\mathbb{P}}_{t+1}$ até que $|\bar{\mathbb{P}}_{t+1}| = \bar{M}$. O método de truncamento ocorre como segue. Primeiro, tenta-se remover o indivíduo com o menor valor de σ_i^k , ou seja, procura-se remover o indivíduo que está mais próximo de seus vizinhos. Havendo mais de um indivíduo com o menor valor de σ_i^k (indivíduos empatados com o menor valor), seleciona-se entre estes indivíduos empatados o que tiver o menor valor para σ_i^l , onde $l = k - 1$. A variável l é iterativamente decrementada em um até que o indivíduo com menor valor para σ_i^l seja único.

O Algoritmo 7 esboça os principais passos do algoritmo SPEA2.

Algoritmo 7: Esboço do algoritmo SPEA2

- 1 **População inicial:** gerar uma população inicial $\mathbb{P}_0$ e um arquivo externo vazio $\bar{\mathbb{P}}_0 = \emptyset$. Atribuir $t = 0$.
 - 2 **Atribuição do valor de adaptação:** calcular o valor de adaptação para cada membro de $\mathbb{P}_t$ e $\bar{\mathbb{P}}_t$ de acordo com a Seção 6.1.2.
 - 3 **Seleção:** copiar todos os elementos de $\mathbb{P}_t$ e $\bar{\mathbb{P}}_t$ para $\bar{\mathbb{P}}_{t+1}$. Se o tamanho de $\bar{\mathbb{P}}_{t+1}$ exceder $\bar{M}$ chame o método de truncamento. Se o tamanho de $\bar{\mathbb{P}}_{t+1}$ for menor do que $\bar{M}$, então preencha $\bar{\mathbb{P}}_{t+1}$ com os indivíduos dominados de $\mathbb{P}_t$ e $\bar{\mathbb{P}}_t$, conforme a Seção 6.1.2.
 - 4 **Condição de parada:** Se t for igual ao número máximo de gerações, então pare.
 - 5 **Seleção para o grupo de reprodução:** realizar torneio binário com reposição com indivíduos em $\bar{\mathbb{P}}_{t+1}$ para criar o grupo de reprodução.
 - 6 **Descendência:** aplicar os operadores de recombinação e mutação aos indivíduos do grupo de reprodução e gerar a geração $\mathbb{P}_{t+1}$. Incrementar $t = t + 1$ e voltar ao passo 2.
-

A adequação do SPEA2 para o PERTEE com múltiplos cenários de geração foi realizada da mesma forma do NSGA-II. Portanto, utilizou-se o mesmo AG para ambos os algoritmos. Com ambos os algoritmos multiobjetivo usando o mesmo AG básico, a diferença ficará somente nas estratégias de busca que cada algoritmo implementa para pesquisar soluções para problemas

multiobjetivo. Portanto, é seguro fazer uma comparação direta entre os dois, como a documentada no Capítulo 7.

No Capítulo 7, apresentaremos detalhes dos testes e comparações realizados com cada um dos algoritmos aqui expostos.

6.2 Paralelização dos Algoritmos NSGA-II e SPEA2

Como já mencionado, recorreremos à computação paralela no intuito de melhorar a qualidade das regiões paretos encontradas pelos algoritmos implementados nesta tese. Nesta seção serão apresentados detalhes da paralelização dos algoritmos SPEA2 e NSGA-II realizadas neste trabalho. Detalhes do cluster *Beowulf* em que foram realizadas todas as execuções e dos resultados obtidos da paralelização podem ser obtidos no Apêndice A.

É importante destacar que, em nossa implementação, antes da migração de indivíduos entre processos, primeiro é realizada uma conversão de tal forma que todos os dados do indivíduo, como cromossomo, valor de aptidão, etc, são armazenados em um vetor do tipo *double precision*. É este vetor que é transferido. Quando o processo destino recebe o vetor representando um indivíduo, o processo faz a conversão oposta, ou seja, os elementos do vetor são convertidos para os dados de um indivíduo. Considere que o indivíduo ilustrado na Figura 6.6 represente um investimento de R\$ 3,45 e satisfaça a 87% dos cenários para um determinado sistema. O processo de conversão, criará o vetor de dupla precisão representado pela Figura 6.7

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

Figura 6.6: *Indivíduo a ser transferido entre processos.*

1,0	2,0	0,0	0,0	2,0	3,0	0,0	0,0	0,0	3,45	0,87
-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------

Figura 6.7: *Vetor representando o indivíduo a ser transferido entre processos.*

6.2.1 Paralelização do Algoritmo NSGA-II

O algoritmo NSGA-II foi paralelizado de acordo com o modelo de cooperação paralela autônoma com subpopulações cooperativas, detalhado na Seção 5.2.1 do Capítulo 5. Os parâmetros são iguais em todas as ilhas aos da implementação sequencial, exceto o tamanho da população em cada nó, que foi definido em $M = 70$. O número identificador do nó na implementação MPI foi utilizado como raiz do gerador de números aleatórios. Desta forma, todos os nós iniciam com uma raiz diferente. No modelo de cooperação paralela autônoma com subpopulações cooperativas, o algoritmo paralelo é composto de duas fases: uma interna, em que os nós executam um número determinado de iterações antes de ocorrerem as migrações; e uma externa, em que é realizada as migrações. Nesta tese, o número de iterações externas foi definido em 50 e o número de iterações internas foi definido em cinco. O número de indivíduos que migram a cada cinco iterações internas foi definido em três. Desta forma, a cada cinco iterações, cada nó se prepara para entrar no processo migratório. Uma vez concluído este processo, uma nova rodada de iterações internas se inicia. O processo segue por 50 iterações.

É importante destacar que nenhuma alteração foi realizada na implementação sequencial

do NSGA-II. Esta é uma das vantagens do modelo de cooperação paralela autônoma com subpopulações cooperativas: pouca ou nenhuma alteração precisa ser realizada caso já exista uma implementação sequencial da metaheurística.

6.2.2 Paralelização do Algoritmo SPEA2

Com o algoritmo *SPEA2 Paralelo* (pSPEA2) decidimos aplicar o modelo paralelização intra-algoritmo independente do problema. Em nossa implementação, os nós executam iterações do SPEA2 considerando uma população externa comum a todos. A cada iteração, todos os nós enviam a um nó mestre a sua população atual. Com a união de todas as populações, se gera a nova população externa que, por sua vez, é enviada a todos os nós para se dar início a uma nova iteração. O Algoritmo 8 esboça os principais passos do algoritmo pSPEA2.

Neste trabalho, foram definidos os seguintes parâmetros: $MaxIt = 300$, $\bar{N} = 30$ e $M = 100$. Os demais parâmetros são os mesmos aplicados na versão sequencial do algoritmo SPEA2.

6.3 Técnicas para Avaliação de Desempenho

A justificativa de paralelização dos algoritmos NSGA-II e SPEA2 pode ser baseada em dois objetivos: 1) diminuir o tempo de resposta dos algoritmos e 2) melhorar a aproximação da fronteira encontrada pelos algoritmos da fronteira Pareto-ótima. Em nosso trabalho, damos maior foco ao objetivo de melhorar a fronteira Pareto encontrada pelas versões sequenciais dos algoritmos implementados.

6.3.1 Diminuição de Tempo de Resposta e Eficiência

Para a paralelização de algoritmos sequenciais, tradicionalmente os pesquisadores têm utilizado duas medidas de avaliação de desempenho: a diminuição do tempo de resposta, mais conhecida na literatura especializada pelo termo inglês *speedup* (doravante será utilizado o termo em inglês por ser amplamente utilizado na literatura especializada); e a eficiência.

Seja um problema P , um algoritmo paralelo A_{np} e uma máquina paralela M com q processadores iguais. Convencionaremos $T_{A_{np},M}$ como o tempo que o algoritmo A_{np} precisa para resolver o problema P executando na máquina M utilizando $np \leq q$ processadores. Supondo que A_s é o melhor algoritmo (o que tem o menor tempo de resposta) conhecido para resolver o problema P em uma máquina sequencial com desempenho igual a um processador da máquina M , define-se o *speedup* $\nu A_{np}, M(np)$ do algoritmo A_{np} rodando na máquina M com np processadores segundo a Eq. (6.9).

$$\nu A_{np}, M(np) = \frac{T_{A_s}}{T_{A_{np},M}} \quad (6.9)$$

Portanto, o *speedup* mede a aceleração observada para o algoritmo paralelo executando em np processadores, em relação ao melhor algoritmo sequencial (Cung et al., 2002).

A eficiência $ef A_{np}, M(np)$ do algoritmo paralelo é calculada através da Eq. 6.10.

$$ef A_{np}, M(np) = \frac{\nu A_{np}, M(np)}{np} \quad (6.10)$$

A eficiência calcula a média da fração de tempo em que cada processador está sendo efetivamente utilizada. O ideal é que em momento algum os processadores estejam ociosos. A

Algoritmo 8: Esboço do algoritmo pSPEA2

```

1
2   ► Em cada nó, faça
3   Início
4       População inicial: gerar uma população inicial  $\mathbb{P}_0$  e um arquivo
        externo vazio  $\overline{\mathbb{P}}_0 = \emptyset$ .
5       Para  $it=1:MaxIt$  faça
6           ► Os passos 6, 7, 8 e 9 são idênticos aos
            do Algoritmo 7.
7           Atribuição do valor de adaptação.
8           Seleção.
9           Seleção para o grupo de reprodução.
10          Descendência.
11          ► Todos os nós enviam ao nó mestre a sua
            população atual.
12          GATHER( $\mathbb{P}_t, M, \mathbb{P}_t^{tmp}, np \times M, \text{Mestre}$ )
13
14          ► No nó mestre, faça
15          Início
16              Atribuição do valor de adaptação: calcular o valor de
                adaptação para cada membro
                de  $\mathbb{P}_t^{tmp}$  e  $\overline{\mathbb{P}}_t$  de acordo com a
                Seção 6.1.2.
17              Seleção: copiar todos os
                elementos de  $\mathbb{P}_t^{tmp}$  e  $\overline{\mathbb{P}}_t$  para
                 $\overline{\mathbb{P}}_{t+1}$ . Se o tamanho de  $\overline{\mathbb{P}}_{t+1}$ 
                exceder  $\overline{N}$ , chame o método de
                truncamento. Se o tamanho de
                 $\overline{\mathbb{P}}_{t+1}$  for menor do que  $\overline{N}$ ,
                então preencha  $\overline{\mathbb{P}}_{t+1}$  com os
                indivíduos dominados de  $\mathbb{P}_t^{tmp}$  e
                 $\overline{\mathbb{P}}_t$ , conforme a seção 6.1.2.
18          fim
19          BCAST( $\overline{\mathbb{P}}_{t+1}, \overline{N}, \text{Mestre}$ )
20      fim
21  fim
  
```

eficiência objetiva calcular o quanto o algoritmo paralelo se aproxima do ideal.

Estas métricas são bastante utilizadas na literatura no intuito de mostrar o quão vantajoso é aumentar o número de processadores para diminuir o tempo de resposta. Em nossa abordagem, o tempo de resposta não é uma variável tão importante, por se tratar de um problema de planejamento, sendo portanto o tempo de resposta dos algoritmos sequenciais aceitável. O nosso maior objetivo em paralelizar os algoritmos é diminuir o valor de investimento para o limitante superior, o que atende a todos os cenários, e melhorar a qualidade das fronteiras Pareto encontradas. Por isso, não faremos uso destas métricas, mas somente das que visam medir a qualidade das fronteiras encontradas.

6.3.2 Métricas Para Avaliar a Aproximação da Fronteira Pareto-Ótima

Nesta tese, foram usadas três métricas para medir a qualidade das fronteiras encontradas pelos algoritmos multiobjetivos implementados: distância geracional (Veldhuizen, 1999), espalhamento (Deb et al., 2002) e hipervolume (Veldhuizen, 1999, Zitzler; Thiele, 1998b).

Distância Geracional

Esta métrica calcula explicitamente a proximidade de um conjunto Q , de tamanho N , com soluções do conjunto P^* Pareto-ótimo conhecido. A distância geracional calcula uma distância média das soluções de Q em relação a P^* de acordo com a Eq. (6.11) (Veldhuizen, 1999).

$$GD = \frac{(\sum_{i=1}^{|Q|} d_i^p)^{1/p}}{|Q|} \quad (6.11)$$

Para $p = 2$, o parâmetro d_i é a distância euclidiana entre o vetor de objetivos da solução $i \in Q$ e o membro de P^* mais próximo de i (Deb, 2001):

$$d_i = \min_{k=1}^{|P^*|} \sqrt{\sum_{m=1}^M (f_m^{(i)} - f_m^{*(k)})^2} \quad (6.12)$$

onde M é o número de objetivos, $f_m^{*(k)}$ é o valor da m -ésima função objetivo do k -ésimo membro de P^* . Como a métrica GD se trata do cálculo de distância de um dado conjunto em relação ao conjunto Pareto-ótimo, então quanto menor for o valor de GD , melhor é a qualidade do conjunto.

Como o conjunto Pareto-ótimo P^* não é conhecido para nenhum dos sistemas testados, seguiremos nesta tese o seguinte critério: o conjunto P^* é definido como sendo o conjunto não-dominado considerando o resultado de todas as execuções realizadas. Portanto, seja Q^* o conjunto resultante da união de todas as soluções encontradas, $Q^* = \{\cup_{i=3}^7 Q_{(nsgaII, np=i)}\} \cup \{Q_{(nsgaII, sequencial)}\} \cup \{\cup_{i=3}^7 Q_{(spea-2, np=i)}\} \cup \{Q_{(spea-2, sequencial)}\}$, P^* é definido como sendo o conjunto de soluções não-dominadas pertencentes a Q^* .

Espalhamento

Esta métrica, proposta por Deb et al. (2002) e definida pela Eq. (6.13), tem como objetivo avaliar a diversidade das soluções não-dominadas.

$$\Delta = \frac{\sum_{m=1}^M d_m^e + \sum_{i=1}^{|Q|} |d_i - \bar{d}|}{\sum_{m=1}^M d_m^e + |Q| \bar{d}} \quad (6.13)$$

Na Eq. (6.13), d_i pode ser qualquer medida de distância entre as soluções vizinhas e $\bar{d}$ é a média dos valores destas distâncias. O parâmetro d_m^e é a distância entre as soluções extremas de P^* e Q correspondente ao m -ésimo objetivo.

Considerando esta métrica, um conjunto Q será ideal se para este conjunto $d_m^e = 0$ e todos os valores de d_i são iguais à média $\bar{d}$. Como o conjunto ideal é o próprio conjunto Pareto-ótimo, então distância das soluções extremas em relação às soluções extremas do conjunto Pareto-ótimo será igual a zero. A segunda condição implica que a distribuição das soluções intermediárias é uniforme.

Hipervolume

Esta métrica propõe medir em um só cálculo tanto a distância em relação ao conjunto Pareto-ótimo, como a diversidade da solução. A métrica hipervolume, proposta em (Zitzler; Thiele, 1999), mede o hipervolume da região do espaço de objetivos que é fracamente dominada por um conjunto aproximado Q (Fonseca et al., 2005, Knowles; Thiele; Zitzler, 2006, Zitzler; Brockhoff; Thiele, 2007, Zitzler; Knowles; Thiele, 2008, Zitzler; Thiele, 1999). Para calcular esta métrica, o espaço de objetivo deve ser limitado ou um ponto limite de referência, que é dominado por todos os pontos, deve ser usado. Matematicamente, para cada solução $i \in Q$, um hipercubo v_i é construído considerando o ponto de referência W e a solução i como os cantos diagonais do hipercubo. O ponto de referência pode ser definido como o vetor dos piores valores das funções objetivo. Após a união de todos os hipercubos for calculada, o hipervolume ($I_H(Q)$) é calculado de acordo com a Eq. (6.14) (Deb, 2001).

$$I_H(Q) = \text{volume}(\cup_{i=1}^{|Q|} v_i) \quad (6.14)$$

A Figura 6.8 mostra o hipervolume, região cinza da figura, formado por um conjunto de soluções não-dominadas, $Q = \{A, B, C, D, E\}$, tomando W como ponto de referência. Na referida figura, f_1 e f_2 devem ser minimizadas.

Figura 6.8: Hipervolume formado pelas soluções não-dominadas.

A Figura 6.9 ilustra os hipercubos formados pelas soluções não-dominadas e o ponto de referência W . Veja que como estamos lidando com apenas dois objetivos, os hipercubos se resumem a retângulos formados (v_1, v_2, v_3, v_4 e v_5), e o hipervolume, se resume à soma das

áreas destes retângulos.

Figura 6.9: *Ilustração dos hipercubos.*

Agora vejamos como a métrica de hipervolume consegue medir a qualidade das soluções tanto em relação à proximidade do conjunto Pareto-ótimo P^* , como em relação à boa distribuição das soluções. Quanto mais próximo o conjunto Q estiver da fronteira Pareto-ótima, mais distantes estarão suas soluções do ponto W . Portanto, o valor de $I_H(Q)$ tenderá a crescer à medida que Q se aproxima de P^* . Se as soluções Q estiverem mal distribuídas, menor será o número de hipercubos, o que fará com que os hipervolumes diminuam.

Para evitar problemas de magnitudes diferentes entre os valores das funções objetivo, costuma-se fazer uma normalização desta métrica dividindo-se $I_H(Q)$ por $I_H(P^*)$ (Deb, 2001, Veldhuizen, 1999):

$$I_{HN} = \frac{I_H(Q)}{I_H(P^*)}. \quad (6.15)$$

Portanto, segundo a Eq. (6.15), quanto mais o conjunto Q se aproximar do conjunto Pareto-ótimo P^* , mais se aproximará de 1,0.

No Capítulo 8 serão fornecidos os resultados computacionais obtidos através da execução dos algoritmos paralelos. As métricas introduzidas nesta seção serão utilizadas para medir a evolução da qualidade das soluções à medida que cresce o número de processadores executando os algoritmos multiobjetivos paralelos apresentados neste capítulo.

Capítulo 7

Testes e Resultados Considerando Algoritmos Sequenciais

Para mostrar os resultados da abordagem multiobjetivo do Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica com múltiplos cenários de geração, três sistemas utilizados na literatura foram considerados: Sistema de Seis Barras com Redespacho da Geração (Garver), o Sistema Teste IEEE de 24 Barras (IEEE24Bar) e o Sistema Sul Brasileiro (SULBr) com redespacho da geração. Detalhes dos sistemas e da amostra de cenários de geração utilizada estão nas Seções 3.3.1, 3.3.2 e 3.3.3, respectivamente, do Capítulo 3. Todas as competições nos algoritmos testados foram binárias, ou seja, com dois participantes. O algoritmo de ordenação *quicksort* (Hoare, 1961, 1962) foi aplicado sempre que houve necessidade de ordenação em todos os algoritmos aqui testados. Para solução dos problemas lineares foi usado o pacote Minos 5.5 (Murtagh; Saunders, 1987). Todos os demais códigos computacionais foram desenvolvidos pelo autor. Os códigos computacionais foram desenvolvidos na linguagem Fortran 90/95 e compilados pelo compilador *GNU Fortran compiler* versão *gcc* 4.4.4 20100630. Todos os testes foram rodados em um computador *intel pentium IV 3.2 GHz*, memória de 1 MB, com sistema operacional *Linux kernel 2.6.32.26*, distribuição *Fedora 12*.

7.1 Testes Com o Algoritmo NSGA-II

Para o algoritmo NSGA-II utilizamos uma população de tamanho $M = 100$.

Uma das desvantagens das metaheurísticas em geral é que não há segurança a respeito do critério de parada. Geralmente se usa um limite de tempo de execução ou um número máximo de iterações. Neste trabalho, deixamos o algoritmo executar por um número máximo de iterações e armazenamos as fronteiras Pareto-ótima a cada cem iterações no intuito de verificar a convergência do algoritmo.

7.1.1 Sistema de (Garver, 1970)

A solução ótima considerando somente o objetivo φ apresenta investimento de \$110,00 e a seguinte topologia.

$$n_{2-6} = 3, n_{3-5} = 1.$$

A configuração que atendeu a todos os cenários $\psi = 1,0$ com o menor custo apresenta investimento de \$220,00 e tem a seguinte topologia.

$$n_{2-3} = 1, n_{2-6} = 3, n_{3-5} = 1, n_{4-6} = 3.$$

Figura 7.1: NSGA-II: Fronteira Pareto-ótima obtida para o sistema Garver.

Para este sistema, a fronteira Pareto-ótima foi encontrada em menos de cem iterações e não foi mais alterada. A Figura 7.1 mostra a fronteira Pareto-ótima encontrada pelo algoritmo NSGA-II para o sistema Garver.

7.1.2 Sistema IEEE 24 Barras

Para este sistema, a configuração que atendeu a todos os cenários $\psi = 1,0$ com o menor custo apresenta investimento de $\$1.057,00 \times 10^4$ e tem a seguinte topologia.

$$n_{1-2} = 1, n_{3-24} = 1, n_{4-9} = 1, n_{5-10} = 2, n_{6-10} = 2, n_{7-8} = 2, n_{8-9} = 1, n_{8-10} = 1, n_{9-12} = 1, n_{10-11} = 2, n_{10-12} = 1, n_{11-13} = 1, n_{12-23} = 1, n_{14-16} = 1, n_{15-21} = 1, n_{15-24} = 1, n_{16-17} = 2, n_{16-19} = 1, n_{17-18} = 1, n_{20-23} = 1, n_{2-8} = 1.$$

A Figura 7.2 mostra as fronteiras Pareto-ótimas armazenadas para o sistema IEEE24Bar.

Figura 7.2: NSGA-II: Fronteira Pareto-ótima obtida para o sistema IEEE24Bar.

Embora a configuração de menor custo com $\psi = 1,0$ tenha sido encontrada em menos de duzentas iterações, pela Figura 7.2 podemos ver que o algoritmo obteve convergência entre

Figura 7.3: *NSGA-II: Fronteira Pareto-ótima obtida para o sistema SULBr.*

quatrocentas a quinhentas iterações.

7.1.3 Sistema Sul Brasileiro

Para este sistema, a configuração que atendeu a todos os cenários $\psi = 1,0$ com o menor custo apresenta investimento de $\$246.983,00 \times 10^3$ e tem a seguinte topologia.

$$n_{14-18} = 1, n_{32-43} = 1, n_{18-19} = 1, n_{20-21} = 1, n_{42-43} = 2, n_{46-6} = 1, n_{16-28} = 1, n_{19-25} = 1, n_{31-32} = 1, n_{28-31} = 1, n_{31-41} = 1, n_{24-25} = 2, n_{40-41} = 1, n_{5-6} = 2.$$

Embora a configuração de menor custo com $\psi = 1,0$ tenha sido encontrada em menos de trezentas iterações, pela Figura 7.3 podemos ver que o algoritmo obteve convergência entre quatrocentas a quinhentas iterações.

7.2 Testes Com o Algoritmo SPEA2

Assim como para o algoritmo NSGA-II, no SPEA2 também usamos como tamanho da população $M = 100$ e o tamanho do grupo de reprodução ($npsize$) também foi definido em 30. O tamanho do arquivo externo foi definido em 30 ($\bar{N} = 30$) para todos os sistemas testados.

Da mesma forma que para o NSGA-II, foi definido em número máximo de gerações para o algoritmo parar, porém a cada cem gerações foi arquivado o arquivo externo no intuito de acompanhar a convergência do algoritmo. Destacamos que como este algoritmo apresentou durante testes iniciais maior dificuldade de convergência, o deixamos executar por 500 iterações.

Sistema de Garver

O algoritmo SPEA2 encontrou a mesma fronteira Pareto-ótima para o sistema Garver encontrada pelo algoritmo NSGA-II e também em menos de cem iterações.

7.2.1 Sistema IEEE 24 Barras

A configuração que atendeu a todos os cenários $\psi = 1,0$ com o menor custo apresenta investimento de $\$1.062,00 \times 10^4$ e tem a seguinte topologia.

$$n_{1-2} = 1, n_{3-24} = 1, n_{4-9} = 1, n_{5-10} = 1, n_{6-10} = 2, n_{7-8} = 2, n_{8-9} = 1, n_{8-10} = 1, n_{9-12} =$$

$$1, n_{10-11} = 2, n_{10-12} = 1, n_{11-13} = 1, n_{12-23} = 1, n_{14-16} = 1, n_{15-21} = 1, n_{15-24} = 1, n_{16-17} = 1, n_{17-18} = 1, n_{20-23} = 1, n_{21-22} = 1, n_{1-8} = 1.$$

A Figura 7.4 mostra as fronteiras Pareto-ótimas armazenadas para o sistema IEEE24Bar.

Figura 7.4: *SPEA2: Fronteira Pareto-ótima obtida para o sistema IEEE24Bar.*

Embora a configuração de menor custo com $\psi = 1,0$ tenha sido encontrada em menos de duzentas iterações, pela Figura 7.4 podemos ver que o algoritmo obteve convergência entre quatrocentas a quinhentas iterações.

O SPEA2 também encontrou o limitante inferior para o sistema IEEE24Bar.

7.2.2 Sistema Sul Brasileiro

A configuração que atendeu a todos os cenários $\psi = 1,0$ com o menor custo apresenta investimento de $\$235.681,00 \times 10^3$ e tem a seguinte topologia.

$$n_{46-16} = 1, n_{18-19} = 1, n_{20-21} = 1, n_{42-43} = 2, n_{46-6} = 1, n_{19-25} = 1, n_{31-32} = 1, n_{28-31} = 1, n_{28-30} = 1, n_{27-29} = 2, n_{28-41} = 1, n_{24-25} = 2, n_{29-30} = 1, n_{40-41} = 1, n_{5-6} = 2.$$

Embora a configuração de menor custo com $\psi = 1,0$ tenha sido encontrada em menos de trezentas iterações, pela Figura 7.5 podemos ver que o algoritmo obteve convergência entre quatrocentas a quinhentas iterações.

Figura 7.5: *SPEA2: Fronteira Pareto-ótima obtida para o sistema SulBr.*

O SPEA2 também encontrou o limitante inferior para o sistema SULBr.

7.3 Análise dos Resultados

Nesta seção mostraremos, baseados nos testes realizados, a robustez da metodologia aqui proposta. Mostraremos os cenários a que as soluções encontradas para o limitante inferior (menor valor para o objetivo φ) e superior (menor custo para $\psi = 1,0$) atendem considerando amostras com o número de cenários testados e com um milhão de cenários.

7.3.1 Sistema de Garver

Detalhes sobre este sistema foram apresentados na Seção 3.3.1. A Tabela 7.1 mostra a porcentagem de cenários não atendidos pela configuração \$110,00 considerando amostras com $N = 573$ e $N = 1.000.000$ cenários. A coluna *MxDNS* mostra a porcentagem da maior demanda não atendida da amostra e a coluna *MdDNS* mostra a porcentagem da média das demandas não atendidas da amostra, ambas em relação à demanda de pico do sistema.

Tabela 7.1: *Cenários não atendidos pela configuração \$110,00 com $N = 573$ e $N = 1.000.000$.*

N	Folga %			MxDNS%	MdDNS%
	Baixa	Média	Alta		
573	81,48	37,41	0,00	29,78	19,55
1.000.000	79,44	44,32	0,00	31,54	27,19

A Tabela 7.2 mostra a porcentagem de cenários não atendidos pela configuração \$220,00 considerando amostras com $N = 573$ e $N = 1.000.000$ cenários.

Tabela 7.2: *Cenários não atendidos pela configuração \$220,00 com $N = 573$ e $N = 1.000.000$.*

N	Folga %			MxDNS%	MdDNS%
	Baixa	Média	Alta		
573	0,00	0,00	0,00	0,00	0,00
1.000.000	0,004	0,00	0,00	1,86	1,75

7.3.2 Sistema IEEE de 24 barras

Detalhes sobre este sistema foram apresentados na Seção 3.3.2. A Tabela 7.3 mostra a porcentagem de cenários não atendidos pela configuração \$ 152,00 $\times 10^4$ considerando amostras com $N = 1.689$ e $N = 1.000.000$ cenários.

Tabela 7.3: *Cenários não atendidos pela configuração \$ 152,00 $\times 10^4$ com $N = 1.689$ e $N = 1.000.000$.*

N	Folga %			MxDNS%	MdDNS%
	Baixa	Média	Alta		
1.689	100,00	100,00	0,00	12,14	9,76
1.000.000	99,98	99,19	66,67	14,73	10,68

A Tabela 7.4 mostra a porcentagem de cenários não atendidos pela configuração \$1.062,00 $\times 10^4$, encontrada pelo algoritmo SPEA2, considerando amostras com $N = 1.689$ e $N = 1.000.000$ cenários.

Tabela 7.4: *SPEA2: Cenários não atendidos pela configuração \$1.062,00 $\times 10^4$ com $N = 1.689$ e $N = 1.000.000$.*

N	Folga %			MxDNS%	MdDNS%
	Baixa	Média	Alta		
1.689	0,00	0,00	0,00	0,00	0,00
1.000.000	0,02	0,00	0,00	0,87	0,51

A Tabela 7.5 mostra a porcentagem de cenários não atendidos pela configuração \$1.057,00 $\times 10^4$, encontrada pelo algoritmo NSGA-II, considerando amostras com $N = 1.689$ e $N = 1.000.000$ cenários.

Tabela 7.5: *NSGA-II: Cenários não atendidos pela configuração \$1.057,00 $\times 10^4$ com $N = 1.689$ e $N = 1.000.000$.*

N	Folga %			MxDNS%	MdDNS%
	Baixa	Média	Alta		
1.689	0,00	0,00	0,00	0,00	0,00
1.000.000	0,04	0,00	0,00	0,71	0,43

7.3.3 Sistema Sul Brasileiro

Detalhes sobre este sistema foram apresentados na Seção 3.3.3. A Tabela 7.6 mostra a porcentagem de cenários não atendidos pela configuração U\$ 70.289,00 $\times 10^3$ considerando amostras com $N = 2.087$ e $N = 1.000.000$ cenários.

Tabela 7.6: *Cenários não atendidos pela configuração U\$ 70.289,00 $\times 10^3$ com $N = 2.087$ e $N = 1.000.000$.*

N	Folga %			MxDNS%	MdDNS%
	Baixa	Média	Alta		
2.087	100,00	90,91	0,00	25,84	16,42
1.000.000	100,00	99,99	0,00	30,98	23,07

A Tabela 7.7 mostra a porcentagem de cenários não atendidos pela configuração U\$235.681,00 $\times 10^3$, encontrada pelo algoritmo SPEA2, considerando amostras com $N = 2.087$ e $N = 1.000.000$ cenários.

Tabela 7.7: *SPEA2: Cenários não atendidos pela configuração $U\$235.681,00 \times 10^3$ com $N = 2.087$ e $N = 1.000.000$.*

N	Folga %			MxDNS%	MdDNS%
	Baixa	Média	Alta		
2.087	0,00	0,00	0,00	0,00	0,00
1.000.000	0,26	0,00	0,00	1,54	1,02

A configuração com custo de $U\$246.983,00 \times 10^3$, encontrada pelo algoritmo NSGA-II, satisfaz a todos os cenários tanto para $N = 2.087$ como para $N = 1.000.000$.

7.3.4 Análise Comparativa Entre os Algoritmos Testados

Nesta seção compararemos o desempenho dos algoritmos NSGA-II e SPEA2. As Figuras 7.6 e 7.7 apresentam as melhores fronteiras Pareto-ótimas encontradas pelos dois algoritmos para os sistemas IEEE24Bar e SULBr, respectivamente. Para os sistema Sistema de Seis Barras com Redespacho da Geração (Garver), ambos os algoritmos encontraram a mesma fronteira.

Figura 7.6: *Comparação entre NSGA-II e SPEA2 para o sistema IEEE24Bar.*

Figura 7.7: Comparação entre NSGA-II e SPEA2 para o sistema SULBr.

Vê-se pelas Figuras 7.6 e 7.7 que as fronteiras encontradas são semelhantes. As Tabelas 7.8 e 7.9 apresentam os dados referentes ao desempenho dos dois algoritmos quando aplicados aos sistemas testados. A coluna intervalo de convergência (FC) é referente ao intervalo de número de gerações em que o algoritmo alcançou estabilidade. A coluna NPL é o número de problemas de programação linear (PPLs) resolvidos até a última geração do intervalo de convergência. Por exemplo, na primeira linha da Tabela 7.8, 305.031 é o número de PPLs resolvidos até a geração de número 100. Da mesma forma, a coluna Tempo (s) refere-se ao tempo em segundos até a última geração do intervalo de convergência.

Tabela 7.8: Dados a respeito do desempenho do algoritmo SPEA2

Sistema	FC	NPL	Tempo (s)
Garver	001 – 100	305.031	390,3
IEEE24Bar	200 – 300	21.753.172	10.457,31
SULBr	200 – 300	37.810.843	39.489,55

Tabela 7.9: Dados a respeito do desempenho do algoritmo NSGA-II

Sistema	FC	NPL	Tempo (s)
Garver	001 – 100	303.510	386,3
IEEE24Bar	400 – 500	36.151.929	17.461,30
SULBr	400 – 500	62.834.369	64.105,36

Os dois algoritmos aqui testados encontram soluções próximas, como se pode constatar pelas Figuras 7.6 e 7.7, e pelas Tabelas 7.8 e 7.9. Pode-se verificar, através das referidas tabelas, que o algoritmo SPEA2 apresenta convergência mais rápida.

Uma explicação para o melhor desempenho do algoritmo SPEA2 frente ao algoritmo NSGA-II pode estar no uso do arquivo externo do algoritmo SPEA2. Como o arquivo externo contém somente soluções não-dominadas (principalmente depois das primeiras gerações) e as recombinações e mutações ocorrem sobre este arquivo, acreditamos que o cruzamento e mutação geram

indivíduos pouco inactivos. Assim, poucas iterações da heurística construtiva serão necessárias para tornar o indivíduo activo. Já o algoritmo NSGA-II não faz uso de um arquivo externo, e indivíduos dominados e pertencentes a qualquer fronteira Pareto-ótima podem ser selecionados para cruzamento e mutação, mesmo com menor probabilidade. Desta forma, há possibilidade de um indivíduo da primeira fronteira fazer cruzamento com indivíduo da última fronteira, gerando filhos muito inactivos, os quais necessitarão de muitas iterações da heurística construtiva para se tornarem activos.

Como se pode constatar nas Tabelas 7.8 e 7.9 uma das maiores desvantagens da metodologia proposta neste trabalho refere-se ao custo computacional elevado. Embora diversas refatorações (Fowler, 2000) tenham sido aplicadas aos algoritmos implementados para garantir a otimalidade dos códigos, o sistema SULBr, mesmo sendo um sistema de médio porte, precisou de algumas horas para convergir. Além disto, acreditamos que o limitante superior pode ser diminuído tanto para este sistema como para o sistema IEEE24Bar permitindo-se a execução dos algoritmos por mais iterações ou com populações maiores. Portanto, propomos a aplicação da computação paralela no intuito de sanar estas suspeitas.

As principais arquiteturas de computadores paralelos e teorias a respeito de aumento esperado de ganho de desempenho foram apresentadas no Capítulo 5. No Capítulo 6 foram introduzidas as principais teorias para medição de desempenho de algoritmos paralelos. No próximo capítulo serão apresentados os resultados obtidos da paralelização dos algoritmos multiobjetivos implementados nesta tese.

Capítulo 8

Testes e Resultados Considerando os Algoritmos Paralelos

Neste capítulo serão apresentados e analisados os resultados computacionais obtidos da aplicação dos algoritmos multiobjetivos paralelos para solução do Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica com múltiplos cenários de geração. Conforme mencionado no Capítulo 7, em que foram apresentados os resultados dos testes computacionais com os algoritmos sequenciais, neste capítulo se objetiva realizar uma análise a respeito do quanto a paralelização dos algoritmos NSGA-II e SPEA2 pode melhorar a qualidade das fronteiras encontradas por estes algoritmos.

8.1 Resultados Computacionais Considerando o Algoritmo NSGA-II

Nesta seção apresentaremos os resultados obtidos da execução do algoritmo NSGA-II Paralelo (pNSGA-II) considerando os sistemas SULBr e IEEE24Bar. Os testes foram realizados considerando de três a sete processadores para ambos os sistemas.

8.1.1 Sistema IEEE 24 Barras

A Figura 8.1 apresenta as fronteiras encontradas pelo algoritmo pNSGA-II aplicado ao sistema IEEE24Bar.

Figura 8.1: *pNSGA-II: Fronteiras do sistema IEEE24Bar.*

Pela Figura 8.1 percebe-se que pouca alteração há nas fronteiras à medida que se aumenta o

número de processadores. A maior diferença está no valor do limitante superior. À medida que se aumenta o número de processadores, há acentuada redução neste valor, como pode ser observado na Tabela 8.1. A Figura 8.2 ilustra a convergência do limitante superior em relação ao aumento do número de processadores.

Tabela 8.1: *pNSGA-II: Diminuição do valor do limitante superior à medida que aumenta o número de processadores para o sistema IEEE24Bar.*

np	$\varphi \times 10^4$
1	1.057,00
3	989,00
4	944,00
5	937,00
6	918,00
7	881,00

Figura 8.2: *pNSGA-II: relação entre o aumento do número de processadores e a diminuição do valor do limitante superior para o sistema IEEE24Bar.*

8.1.2 Sistema Sul Brasileiro

A Figura 8.3 apresenta as fronteiras encontradas pelo algoritmo pNSGA-II aplicado ao sistema SULBr.

Figura 8.3: *pNSGA-II: Fronteiras do sistema SULBr.*

Pela Figura 8.3 percebe-se que pouca alteração há nas fronteiras à medida que se aumenta o número de processadores. As melhores fronteiras foram encontradas com $np = 6$ e $np = 7$, porém a maior diferença está no valor do limitante superior. À medida que se aumenta o número de processadores, há uma ligeira redução neste valor, como pode ser observado na Tabela 8.2. A Figura 8.4 ilustra a convergência do limitante superior em relação ao aumento do número de processadores.

Tabela 8.2: *pNSGA-II: Diminuição do valor do limitante superior à medida que aumenta o número de processadores para o sistema SULBr.*

np	$\varphi \times 10^3$
1	246.983
3	238.006
4	238.006
5	230.274
6	226.901
7	225.158

Figura 8.4: *pNSGA-II*: relação entre o aumento do número de processadores e a diminuição do valor do limitante superior para o sistema SULBr.

8.2 Resultados Computacionais Considerando o Algoritmo SPEA2

Nesta seção apresentaremos os resultados obtidos da execução do algoritmo *SPEA2 Paralelo* (pSPEA2) considerando os sistemas SULBr e IEEE24Bar. Para este algoritmo também foram realizados testes considerando de três a sete processadores para ambos os sistemas.

8.2.1 Sistema IEEE 24 Barras

A Figura 8.5 apresenta as fronteiras encontradas pelo algoritmo pSPEA2 aplicado ao sistema IEEE24Bar.

Figura 8.5: *pSPEA2*: Fronteiras do sistema IEEE24Bar.

Como se pode ver através da Figura 8.5, à medida que se aumenta o número de processadores, melhor é a qualidade da fronteira encontrada pelo algoritmo pSPEA2. Na Tabela 8.3, mostramos o valor do limitante superior em relação entre o número de processadores e o valor do limitante superior encontrado pelo algoritmo pSPEA2 para o sistema IEEE24Bar. A Figura 8.6 foi gerada a partir da Tabela 8.3.

Tabela 8.3: *pSPEA2*: Diminuição do valor do limitante superior à medida que aumenta o número de processadores para o sistema IEEE24Bar.

np	$\varphi \times 10^4$
1	1.062,00
3	1.138,00
4	1.082,00
5	1.037,00
6	1.037,00
7	991,00

Figura 8.6: *pSPEA2*: relação entre o aumento do número de processadores e a diminuição do valor do limitante superior para o sistema IEEE24Bar.

8.2.2 Sistema Sul Brasileiro

A Figura 8.7 apresenta as fronteiras encontradas pelo algoritmo *pSPEA2* aplicado ao sistema SULBr.

Figura 8.7: *pSPEA2*: Fronteiras do sistema SULBr.

Percebe-se, através da Figura 8.5, que à medida que se aumenta o número de processadores há uma ligeira melhora na qualidade da fronteira encontrada pelo algoritmo pSPEA2. Na Tabela 8.4, mostramos o valor do limitante superior em relação entre o número de processadores e o valor do limitante superior encontrado pelo algoritmo pSPEA2 para o sistema SULBr.

Tabela 8.4: *pSPEA2: Diminuição do valor do limitante superior à medida que aumenta o número de processadores para o sistema SULBr.*

np	$\varphi \times 10^4$
1	249.953,00
3	370.574,00
4	370.574,00
5	305.197,00
6	249.953,00
7	249.314,00

Note que houve pouca melhora do limitante superior encontrado pelo pSPEA2 em relação ao limitante superior encontrado pelo SPEA2 sequencial. A Figura 8.8 foi gerada a partir da Tabela 8.4.

Figura 8.8: *pSPEA2: relação entre o aumento do número de processadores e a diminuição do valor do limitante superior para o sistema SULBr.*

8.3 Análise dos Resultados

Nesta seção faremos análise e comparações das soluções e desempenho dos algoritmos implementados nesta tese. Como já mencionamos neste trabalho, consideramos o conjunto Pareto-ótimo P^* aquele formado pelas soluções não-dominadas encontradas por todas as execuções realizadas neste trabalho. A Figura 8.9 mostra o conjunto P^* para o sistema IEEE24Bar e a Figura 8.10 mostra o conjunto P^* para o sistema SULBr.

Figura 8.9: *Soluções não-dominadas encontradas para o sistema IEEE24Bar.*

Figura 8.10: *Soluções não-dominadas encontradas para o sistema SULBr.*

Foram aplicadas as métricas introduzidas na Seção 6.3.2 do Capítulo 6 para medir a qualidade das fronteiras, em relação à fronteira P^* , à medida que se aumenta o número de processadores para execução dos algoritmos paralelos implementados nesta tese. Na Tabela 8.5 são mostrados os valores das métricas para o sistema IEEE24Bar. Na referida tabela, a primeira coluna apresenta o número de processadores; a segunda coluna mostra os valores para a métrica Distância Geracional (GD); a terceira coluna mostra os valores para a métrica Espalhamento (Δ) e a quarta coluna mostra os valores para a métrica Hipervolume normalizado (I_{HN}). As primeiras colunas mostram valores calculados para as fronteiras encontradas pelo algoritmo NSGA-II. As demais colunas são equivalentes às primeiras, mostrando os valores calculados para as fronteiras encontradas pelo algoritmo SPEA2. As Figuras 8.11, 8.12 e 8.13 apresentam gráficos ilustrando a relação entre os valores das métricas distância geracional, espalhamento e hipervolume, respectivamente. Através destes gráficos podemos constatar que as fronteiras encontradas pelo algoritmo NSGA-II são melhores do que as encontradas pelo algoritmo SPEA2 para o sistema IEEE24Bar. Além disso,

verifica-se que à medida que se aumenta o número de processadores, maior é qualidade das respostas.

Tabela 8.5: Métricas para o sistema IEEE24Bar.

NSGA-II				SPEA2			
np	GD	Δ	I_{HN}	np	GD	Δ	I_{HN}
1	3388,77	2,47	0,778	1	8308,36	4,26	0,802
3	1334,77	1,93	0,886	3	10915,59	2,16	0,661
4	907,50	2,02	0,925	4	7633,92	2,47	0,748
5	838,50	1,89	0,934	5	9688,27	2,64	0,831
6	494,06	2,06	0,965	6	5224,41	2,37	0,846
7	343,73	0,99	0,997	7	4729,11	2,02	0,870

Figura 8.11: Convergência da métrica distância geracional em relação ao número de processadores para o sistema IEEE24Bar.

Figura 8.12: Convergência da métrica espalhamento em relação ao número de processadores para o sistema IEEE24Bar.

Figura 8.13: Convergência da métrica hipervolume em relação ao número de processadores para o sistema IEEE24Bar.

As Figuras 8.14, 8.15 e 8.16 apresentam gráficos ilustrando a relação entre os valores das métricas distância geracional, espalhamento e hipervolume, respectivamente. Através destes gráficos podemos constatar que as fronteiras encontradas pelo algoritmo NSGA-II são melhores do que as encontradas pelo algoritmo SPEA2 para o sistema SULBr. Além disso, verifica-se que à medida que se aumenta o número de processadores, maior é a qualidade das respostas.

Tabela 8.6: Métricas para o sistema SULBr.

NSGA-II				SPEA2			
np	GD	Δ	I_{HN}	np	GD	Δ	I_{HN}
1	757,84	2,79	0,931	1	511,98	3,13	0,945
3	260,38	3,04	0,989	3	579,70	2,60	0,885
4	260,38	3,04	0,989	4	4902,22	2,12	0,600
5	313,95	2,94	0,959	5	4996,47	2,99	0,759
6	103,12	1,69	0,995	6	1233,48	3,75	0,892
7	92,35	2,47	0,985	7	1233,48	3,75	0,892

Figura 8.14: Convergência da métrica distância geracional em relação ao número de processadores para o sistema SULBr.

Figura 8.15: Convergência da métrica espalhamento em relação ao número de processadores para o sistema SULBr.

Figura 8.16: Convergência da métrica hipervolume em relação ao número de processadores para o sistema SULBr.

8.4 Conclusões Parciais

Pode-se afirmar portanto que, para os sistemas, modelos e parâmetros utilizados, é proveitoso aumentar o número de processadores no intuito de melhorar a qualidade da fronteira encontrada pelo algoritmo. Porém, a proposta de paralelização do algoritmo SPEA2 não foi muito bem sucedida, principalmente quando comparado com a proposta de paralelização do algoritmo NSGA-II. Alterar os parâmetros do algoritmo pSPEA2 pode ajudar a melhorar qualidade da fronteira encontrada por este algoritmo. Um estudo da alteração dos parâmetros do pSPEA2 para solução do Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica com múltiplos cenários de geração pode ser realizado como trabalho futuro.

Capítulo 9

Conclusões e Trabalhos Futuros

Propõe-se, neste trabalho, uma nova forma de abordar o problema de Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica com múltiplos cenários de geração. A abordagem proposta sugere um modelo multiobjetivo e um algoritmo para gerar uma amostragem dos cenários de geração possíveis de serem encontrados na futura operação do Sistema Elétrico de Potência. Após o desenvolvimento do modelo para abordar o Planejamento da Expansão a Longo Prazo de Redes de Transmissão de Energia Elétrica com múltiplos cenários de geração, foram implementados dois algoritmos multiobjetivos amplamente conhecidos na literatura: *Nondominated Sorting Genetic Algorithms-II* (NSGA-II) e *SPEA2: Strength Pareto Evolutionary Algorithm* (SPEA2). O nosso objetivo foi analisar e comparar o desempenho deste algoritmo frente ao problema objeto desta tese.

Para averiguar a qualidade dos algoritmos implementados e, principalmente, da metodologia para lidar com o PERTEE considerando múltiplos cenários de geração, realizamos diversos testes com três sistemas conhecidos na literatura do problema. Pelos resultados obtidos e após comparação feita com outros resultados encontrados na literatura, verificou-se que a metodologia proposta neste trabalho tem potencial de contribuição para esta área de pesquisa.

Uma outra contribuição deste trabalho foi mostrar que os planos ótimos de expansão que consideram somente a minimização dos custos de investimento falham quase sempre quando aplicados para cenários de geração diferentes do cenário para o qual o plano foi originalmente proposto. Isto significa que embora o investimento para a expansão do sistema tenha sido o menor, custos maiores poderão ser necessários quando da operação do sistema para mitigar o congestionamento da rede de transmissão devido às mudanças constantes nos padrões de fluxo de potência na rede.

Em relação à qualidade dos algoritmos sequenciais, através dos testes realizados, foi possível constatar que os algoritmos NSGA-II e SPEA2 apresentam desempenhos semelhantes quando aplicados para resolver o modelo proposto neste trabalho. Quando comparado o tempo computacional de ambos os algoritmos em relação aos sistemas testados, verificamos que o algoritmo SPEA2 teve melhor desempenho do que o NSGA-II para todos os sistemas testados. Porém, testes com mais sistemas serão necessários no futuro para se verificar se esta tendência se mantém.

Em uma primeira análise, concluímos que devido à complexidade do problema abordado nesta tese e, principalmente, à proposta de construir planos de expansão que atenda a múltiplos cenários de geração, as respostas dadas pelas versões sequenciais dos algoritmos implementados poderiam ser significativamente melhoradas aplicando-se as teorias da computação paralela. Sugerimos e implementamos nesta tese uma proposta de paralelização para cada um dos algoritmos, a

que foram dados os nomes de NSGA-II Paralelo (pNSGA-II) e o *SPEA2 Paralelo* (pSPEA2). Através de uma análise comparativa dos resultados obtidos das execuções realizadas, podemos concluir que ambas as versões propostas de paralelização dos algoritmos NSGA-II e SPEA2 foram capazes de encontrar resultados melhores do que as suas respectivas versões sequenciais. No entanto, a proposta de paralelização do algoritmo pNSGA-II obteve melhores resultados do que a proposta de paralelização do algoritmo pSPEA2. Para analisar as respostas obtidas e compará-las, foram aplicadas métricas amplamente utilizadas na literatura especializada em algoritmos multiobjetivos.

Esta pesquisa pode ser continuada em diversos trabalhos futuros. A seguir, listamos algumas sugestões de trabalhos futuros que acreditamos serem importantes.

- O estudo do comportamento dos índices de confiabilidade considerando os planos de expansão resultantes que foram construídos considerando múltiplos cenários de geração. E o estudo a respeito do atendimento ao critério de segurança $n - 1$ para os mesmos planos.
- O estudo de outros modelos de paralelização dos algoritmos aqui estudados no intuito de diminuir o tempo computacional necessário para resolver o modelo multiobjetivo para o PERTEE com múltiplos cenários de geração e melhorar a qualidade das soluções.
- Análises de desempenho das propostas de paralelização dos algoritmos multiobjetivos considerando o PERTEE com outros objetivos, como por exemplo o critério de segurança $n - 1$. Ou mesmo outros problemas conhecidos na literatura da pesquisa operacional.

Apêndice A

Cluster Utilizado

O cluster *Beowulf* utilizado neste trabalho é composto de 7 computadores pessoais *intel pentium IV*, 3.2 GHz, memória de 1 MB. Todos os computadores com sistema operacional *Linux kernel 2.6.32.26 – 175.fc12.PAE*, distribuição *Fedora 12*. A tecnologia de interconexão dos nós utilizada foi a *100Base-TX Ethernet* com *channel bondig* (Berntsen et al., 1985).

A interface de passagem de mensagens (Message Passing Interface Standard (MPI)) foi utilizada para implementar as trocas de mensagens entre os nós. Na Seção A.1 faremos uma introdução a esta interface.

A.1 Interface de Passagem de Mensagens

O modelo MPI trata-se de uma biblioteca padrão baseada no consenso obtido no *MPI Forum*(MPI-2, 1997, MPI, 1995). O objetivo principal da MPI é definir uma interface que seja amplamente utilizada no desenvolvimento de programas paralelos baseados em passagem de mensagem. O desenvolvedor não deve se preocupar com detalhes de mais baixo nível, tais como tipo de máquina, estrutura da rede, protocolos de baixo nível, entre outros (Foster et al., 1998). Apresentaremos neste trabalho uma descrição parcial e resumida da interface MPI. Para um estudo mais aprofundado, sugerimos as referências (Barney, 2009, Foster, 1995, Gropp; Lusk; Skjellum, 1999, Gropp; Lusk; Thakur, 1999, MPI-2, 1997, MPI, 1995).

O modelo MPI apresenta, no intuito de flexibilizar o desenvolvimento de aplicações paralelas, um conjunto extenso de funções. Porém, estas funções podem ser resumidas a apenas seis dessas funções, denominadas de conjunto básico de funções, listadas abaixo.

- ***MPI_Init***: Deve ser chamada antes de qualquer rotina MPI. Tem a função de iniciar as estruturas necessários para as comunicações.
- ***MPI_Finalize***: Deve ser chamada antes do término do programa paralelo.
- ***MPI_Comm_rank***: Utilizada para encontrar o número identificador do nó (*rank*).
- ***MPI_Comm_size***: Utilizada para se obter o número de processos em um grupo de processos.
- ***MPI_Send***: Envia uma mensagem para um determinado nó.
- ***MPI_Recv***: Recebe uma mensagem de um determinado nó.

Antes de apresentar mais detalhes das rotinas principais do MPI, cabe destacar que todas as funções do MPI são baseadas em quatro conceitos: *processos*, *comunicadores*, *mensagens* e *tipos de dados*.

A.1.1 Processos

Um processo, também denominado como tarefa, é uma entidade em execução sequencial em um determinado processador. No padrão MPI todo processo tem a possibilidade de se comunicar com outros processos do programa MPI.

A.1.2 Comunicadores

Os diferentes processos do programa MPI são interligados através de objetos chamados *comunicadores* ou grupo de processos. Para que seja possível a comunicação entre dois processos, é necessário que estes estejam associados a um mesmo comunicador. A biblioteca MPI fornece um comunicador padrão denominado *MPI_COMM_WORLD*. Porém, é possível que o programador defina seus próprios comunicadores.

A cada processo em um comunicador é associado identificador (*rank*) único, que é um número inteiro. Os valores de identificadores pode variar entre 0 a $np - 1$, onde np é o número de processos pertencentes ao comunicador.

A interface MPI fornece a seguinte rotina para encontrar o identificador de um processo no seu comunicador.

```
1 int MPI_Comm_rank(comm, &rank) //Código em C
2 MPLCOMMRANK(comm, rank, ierr) !Código Fortran
```

A rotina *MPI_Comm_rank* retorna um valor inteiro caso ocorra algum erro durante sua execução. Em Fortran, o valor inteiro é atribuído à variável *ierr*.

Para se obter o número de processos pertencentes a um comunicador, a interface MPI fornece a seguinte rotina.

```
1 int MPI_Comm_size(comm, &count) //Código em C
2 MPLCOMM_SIZE(comm, count, ierr) !Código Fortran
```

A.1.3 Mensagens

Para que haja cooperação entre os processos, é necessários que estes se comuniquem. Toda comunicação entre processos na interface MPI é realizada através de troca de mensagens.

Como já mencionado, os processos se comunicam por mensagens e através de um comunicador. Como podem estar circulando diversas mensagens simultaneamente pelo sistema, até mesmo pelo mesmo comunicador, além dos dados a serem transferidos, há necessidade de informações para identificar o processo remetente, o processo destinatário, um identificador da mensagem (*tag*) e o comunicador a ser utilizado. O padrão MPI garante que mensagens destinadas a um determinado processador com o mesmo valor de *tag* serão recebidas na ordem em que foram enviadas.

A.1.4 Tipos de Dados

Por questões de portabilidade, para que haja a troca de mensagens, a interface MPI exige do programador que seja determinado o tipo de dado da mensagem. A Tabela A.1 lista os tipos de dados predefinidos pela interface.

Tabela A.1: *Tipos MPI predefinidos e seus respectivos correspondentes em C e Fortran.*

Tipos em C		Tipos em Fortran	
MPL_CHAR	signed char	MPL_CHARACTER	character(1)
MPL_SHORT	signed short int	MPL_INTEGER	integer
MPL_INT	signed int		
MPL_LONG	signed long int		
MPL_UNSIGNED_SHORT	unsigned char		
MPL_UNSIGNED	unsigned int		
MPL_UNSIGNED_LONG	unsigned long int	MPL_REAL	real
MPL_FLOAT	float		
MPL_DOUBLE	double		
MPL_LONG_DOUBLE	long double		
		MPL_DOUBLE_PRECISION	double precision
		MPL_COMPLEX	complex
		MPL_DOUBLE_COMPLEX	double complex
		MPL_LOGICAL	logical
MPL_BYTE	dígito de 8 bits	MPL_BYTE	dígito de 8 bits

As comunicações possíveis entre processos podem ser divididas em dois grupos: comunicações ponto-a-ponto e comunicações coletivas.

A.1.5 Comunicações Ponto-a-ponto

As comunicações ponto-a-ponto permitem a troca de mensagens entre dois, a apenas dois, processos MPI. Um processo executa a operação de envio, e o outro a operação de recebimento da mensagem.

As rotinas que podem ser utilizadas para a comunicação ponto-a-ponto, são as rotinas `MPI_send` e `MPI_recv`. Abaixo são apresentadas as interfaces destas duas rotinas.

A interface da rotina `MPI_send` é definida abaixo para as linguagens C, em cima, e Fortran.

```

1  int MPI_send(&buf, count, datatype, dest, tag, comm)
2  MPLSEND(buf, count, datatype, dest, tag, comm, ierr)

```

No código acima, *buf* é um vetor com *count* elementos do tipo *datatype*, identificado por *tag*, que será enviado ao processo *dest*.

A interface da rotina `MPI_recv` é definida abaixo para as linguagens C, em cima, e Fortran.

```

1  int MPI_recv(&buf, count, datatype, source, tag, comm, &status)
2  MPLRECV(buf, count, datatype, source, tag, comm, status, ierr)

```

No código acima, *buf* é um vetor com *count* elementos do tipo *datatype*, identificado por *tag*, que será recebido do processo *dest*. O *status* retorna informações sobre a comunicação efetuada, como o *rank* do processo de origem, o *tag* utilizado na comunicação e o número de dados efetivamente enviados.

Para acessar a informações *status*, é necessário primeiro realizar uma chamada à rotina abaixo.

```

1  int MPI_Get_count( &stat, MPLCHAR, &count)
2  MPLGET_COUNT(stat, MPL_CHARACTER, count, ierr)

```

Barney (2009) apresenta um exemplo de uso das rotinas `MPI_send` e `MPI_recv`. No exemplo, um processo envia uma mensagem ao processo 0 e espera deste uma mensagem de resposta. Abaixo, transcrevemos o código Fortran do exemplo, para sua versão em C, consulte a referência.

```

1  program ping
2  include 'mpif.h' !Biblioteca que implementa as rotinas MPI.

```

```

3  integer np, rank, dest, source, count, tag, ierr
4  integer stat(MPLSTATUS_SIZE)
5  character inmsg, outmsg
6  outmsg = 'x'
7  tag = 1
8
9  call MPLINIT(ierr)
10 call MPLCOMMRANK(MPLCOMMLWORLD, rank, ierr)
11 call MPLCOMMSIZE(MPLCOMMLWORLD, np, ierr)
12
13 if (rank .eq. 0) then
14     dest = 1
15     source = 1
16     call MPLSEND(outmsg, 1, MPLCHARACTER, dest, tag, MPLCOMMLWORLD, ierr)
17     call MPLRECV(inmsg, 1, MPLCHARACTER, source, tag, MPLCOMMLWORLD, stat, ierr)
18
19 else if (rank .eq. 1) then
20     dest = 0
21     source = 0
22     call MPLRECV(inmsg, 1, MPLCHARACTER, source, tag, MPLCOMMLWORLD, stat, ierr)
23     call MPLSEND(outmsg, 1, MPLCHARACTER, dest, tag, MPLCOMMLWORLD, ierr)
24 endif
25 call MPLGETCOUNT(stat, MPLCHARACTER, count, ierr)
26 print *, 'O processo ', rank, ': recebeu ', count, ' caracteres do processo ',
27         stat(MPLSOURCE), ' com a tag ', stat(MPLTAG)
28 call MPLFINALIZE(ierr)
29 end

```

Código A.1: Exemplo em Fortran de uso das rotinas *MPLsend* e *MPLrecv*.

A.1.6 Comunicações Coletivas

Frequentemente, há necessidade que um processo se comunique com um grupo de processos. O padrão MPI estabelece que o grupo envolvido na comunicação é especificado através de um comunicador. Assim, todos os processos pertencentes ao comunicador deverão fazer parte da comunicação coletiva. As operações coletivas pode ser divididas em três classes:

- **Sincronização:** utilizada para forçar um processo a esperar até que os outros membros do grupo alcancem o ponto de sincronização.
- **Comunicação global de dados:** utilizada para envio e recebimento de dados dos membros do grupo.
- **Computação coletiva (redução):** um membro do grupo coleta dados dos outros membros para executar uma determinada operação (máximo, mínimo, adição, multiplicação, etc).

As operações coletivas disponíveis em MPI são as que seguem: barreira, *broadcast*, coleta, distribuição, redistribuição, redução e prefixo. O padrão MPI dita que todas os processos de um

grupo devem realizar a chamada da rotina de comunicação coletiva. Abaixo, apresentaremos as rotinas de comunicação coletiva da interface MPI.

- **Barreira:** cria uma barreira para sincronização do grupo. Um processo deve ficar bloqueado quando chega à barreira até que todos os outros processos do comunicador também chegaram à barreira. Abaixo apresentamos a interface MPI para a rotina de barreira.

```
1      int MPI_Barrier(comm)
2      MPLBARRIER(comm, ierr)
```

- **Broadcast:** um *broadcast* é uma operação em que determinada informação é transmitida de um processo fonte, ou raiz, a todos os processos do grupo. Abaixo apresentamos a interface MPI para a rotina de *broadcast*.

```
1      int MPI_Bcast(&buff, count, datatype, raiz, comm)
2      MPLBCAST(buff, count, datatype, raiz, comm, ierr)
```

Como mencionado na Seção 6.2.2, para a proposta de paralelização do algoritmo SPEA2 utilizou-se desta rotina para enviar a população externa do processo raiz a todos os processos do sistema.

- **Coleta:** uma operação de coleta consiste no recolhimento de dados de todos os processos do grupo e armazenamento do resultado em um vetor no processo raiz da coleta. O processo de *rank i* terá seu dado armazenado na posição *i* do vetor. A rotina MPI para realizar coleta é a `MPI_Gather`. Abaixo apresentamos a interface MPI para a rotina de coleta.

```
1      int MPI_Gather(&sendbuf, sendcnt, sendtype, &recvbuf, recvcnt,
2                  recvtype, raiz, comm)
3      MPLGATHER(sendbuf, sendcnt, sendtype, &recvbuf, recvcnt, recvtype,
4                  raiz, comm, ierr)
```

onde: *sendbuf* é o vetor a ser enviado; *sendcnt* é o tamanho do vetor *sendbuf*; *sendtype* é o tipo de dado do vetor a ser enviado e deve ser igual a *recvtype*; *recvbuf* é o vetor onde todos os dados serão armazenados no processo raiz; *recvcnt* é o tamanho do vetor *sendbuf* multiplicado pelo número de processos pertencentes ao comunicador; *raiz* é o processo raiz onde serão armazenados os dados da coleta. A Figura A.1 ilustra a operação de coleta.

Figura A.1: Ilustração da operação de coleta.

Como mencionado na no Seção 6.2.2, na proposta de paralelização do algoritmo SPEA2 utilizou-se desta rotina de coleta a cada iteração para que o processo mestre receba a população de todos os processos do programa MPI para definição do novo arquivo externo.

- **Distribuição:** a operação de distribuição tem função inversa à rotina de coleta. Portanto, o processo raiz possui um vetor de elementos e distribui elementos deste vetor para cada um dos processos do grupo. O processo de *rank i* deverá receber o elemento *i* do vetor. A rotina MPI para realizar distribuição é a `MPLScatter`. Abaixo apresentamos a interface MPI para a rotina de distribuição.

```

1      int MPI_Scatter(&sendbuf, sendcnt, sendtype,&recvbuf,recvnt,
                    recvtype,raiz,comm)
2      MPLSCATTER(sendbuf, sendcnt, sendtype,&recvbuf,recvnt,recvtype
                    ,raiz,comm, ierr)

```

onde: *sendbuf* é o vetor a ser enviado pelo processo raiz; *sendcnt* é o tamanho do vetor *sendbuf*; *sendtype* é o tipo de dado do vetor a ser enviado e deve ser igual a *recvtype*; *recvbuf* é o vetor para armazenar os dados enviados do processo raiz; *recvcnt* é o tamanho do vetor *sendbuf* dividido pelo número de processos pertencentes ao comunicador; *raiz* é o processo raiz onde serão armazenados os dados da coleta. A Figura A.2 ilustra a operação de distribuição.

Figura A.2: Ilustração da operação de distribuição.

- **Redução:** uma operação de redução visa encontrar um total global, como uma somatória, uma busca de mínimo ou máximo. Há duas variantes da operação de redução: uma em que o total é colocado em um processo raiz e outra em que todos os processos do grupo recebem o resultado da redução.

A rotinas MPI para realizar distribuição são as rotinas `MPLReduce` e `MPLAllreduce`. Abaixo apresentamos a interface MPI para a rotina de redução `MPLReduce`.

```

1      int MPI_Reduce(&sendbuf, &recvbuf,count,datatype,op,raiz,comm)
2      MPLREDUCE(sendbuf, recvbuf,count,datatype,op,raiz,comm, ierr)

```

A interface MPI para a rotina de redução `MPLAllreduce` pode se definida como segue.

```

1      int MPI_Allreduce(&sendbuf, &recvbuf,count,datatype,op,raiz,comm
                        )
2      MPLALLREDUCE(sendbuf, recvbuf,count,datatype,op,raiz,comm, ierr
                    )

```

Nas rotinas acima, *sendbuf* representa os valores locais que serão submetidos para o cálculo de redução global; *recvbuf* é onde o resultado do cálculo de redução será armazenado. Na rotina `MPI_Reduce` somente o processo raiz receberá o resultado do cálculo de redução. Na `MPI_Allreduce`, todos os processos do grupo receberão o resultado do cálculo de redução. O parâmetro *op* fornece a operação de redução desejada. Os operadores de redução pré-definidos pelo padrão MPI são listados na Tabela A.2.

Tabela A.2: Operadores de redução pré-definidos pelo padrão MPI

Operador	Descrição
<code>MPI_MAX</code>	máximo
<code>MPI_MIN</code>	mínimo
<code>MPI_SUM</code>	soma
<code>MPI_PROD</code>	produto
<code>MPI_LAND</code>	e lógico
<code>MPI_BAND</code>	e bit-a-bit
<code>MPI_LOR</code>	ou lógico
<code>MPI_BOR</code>	ou bit-a-bit
<code>MPI_LXOR</code>	ou exclusivo lógico
<code>MPI_BXOR</code>	ou exclusivo bit-a-bit
<code>MPI_MAXLOC</code>	máximo e localização
<code>MPI_MINLOC</code>	mínimo e localização.

Na Tabela A.2, `MPI_MAXLOC` e `MPI_MINLOC` diferem de `MPI_MAX` e `MPI_MIN`, respectivamente, por apresentarem o resultado do cálculo de redução e o *rank* do processo em que o máximo ou mínimo foi encontrado.

A implementação MPI utilizada neste trabalho foi a MPICH-2, disponível no sítio <http://www.mcs.anl.gov/research/projects/mpich2/>.

Apêndice B

Técnica Usada Para Simulação das Variáveis Probabilísticas Contínuas

Como mencionado ao longo deste trabalho, utilizamos funções de distribuição cumulativa de probabilidade no intuito de representar a probabilidade da capacidade de um gerador para um determinado cenário de geração estar distribuída entre zero e a capacidade real do gerador. Portanto, para cada cenário, podemos chamar de G_i^t a variável aleatória contínua que representa o gerador i no cenário t . Desta forma, se faz necessário o uso de técnicas para simular a geração de variáveis aleatórias.

Nas próximas seções apresentaremos dois métodos para simular variáveis aleatórias contínuas.

B.1 Método da Transformada Inversa

Considerando uma variável contínua aleatória com uma função de distribuição F . Seja U uma variável uniforme no intervalo $[0, 1]$. Para qualquer função de distribuição contínua F , uma variável aleatória X definida por Eq. (B.1)

$$X = F^{-1}(U) \quad (\text{B.1})$$

tem distribuição F . Onde F^{-1} é o valor x tal que $F(x) = u$.

Exemplo 2.1. (Ross, 2009): Suponha que queremos gerar uma variável aleatória X com a seguinte função de distribuição.

$$F(x) = x^n, \quad 0 < x < 1 \quad (\text{B.2})$$

Se fizermos $x = F^{-1}(u)$, então

$$u = F(x) = x^n \quad (\text{B.3})$$

ou, equivalentemente, $x = u^{1/n}$.

Portanto, podemos gerar uma variável aleatória X gerando um número aleatório U e atribuir $X = U^{1/n}$.

B.2 Método da Aceitação-Rejeição

Este é um método muito prático para gerar número aleatórios. Digamos que desejamos gerar uma variável aleatória com função de distribuição $g(x)$ no intervalo $x_1 \leq x \leq x_2$. Seja x_{max} o valor de máximo da função $g(x)$. O método da aceitação-rejeição segue os seguintes passos.

1. Gere um número x_u uniformemente distribuído no intervalo $[x_1, x_2]$.

2. Gere um número x_r uniformemente distribuído no intervalo $[0, 1]$.
3. Se $x_r > g(x_u)/x_{max}$, então rejeite x_u e volte ao passo 1. Caso contrário, aceite x_u

B.3 Variável Com Distribuição *Gamma*

Como já mencionado, a variável aleatória G_i^t tem uma função de distribuição de probabilidade *beta*. Porém, uma função de distribuição *gamma* é usada como base para gerar esta distribuição. A função de distribuição de probabilidade *gamma* apresenta as seguintes propriedades (Saucier, 2000).

Função de densidade:

$$f(x) = \begin{cases} \frac{1}{\Gamma(c)} b^{-c} (x-a)^{c-1} e^{-(x-a)/b}, & \text{se } x > a \\ 0, & \text{caso contrário} \end{cases} \quad (\text{B.4})$$

onde $\Gamma(l)$ é a função *gamma*, definida como segue.

$$\Gamma(l) \begin{cases} \equiv \int_0^\infty t^{l-1} e^{-t} dt & \text{se } l \text{ for contínuo} \\ = (l-1)! & \text{se } l \text{ for inteiro} \end{cases} \quad (\text{B.5})$$

Função de distribuição de probabilidade: se c for um inteiro positivo, pode ser definida como segue.

$$F(x) = \frac{x^{a-1}}{b^a \Gamma(a)} e^{-\frac{x}{b}} \quad (\text{B.6})$$

onde $0 \leq x < \infty$, $a > 0$ e $b > 0$.

Para gerar variáveis aleatórias com distribuição *gamma* dois passos devem ser seguidos:

1. Gerar n números uniformemente distribuídos no intervalo $[0, 1]$ e independentes formando a sequência $U_i, i = 1, \dots, n$, onde $n = b$.
2. Calcular X pela seguinte equação:

$$X = b \sum_{i=1}^n (-\ln U_i) = -b \ln \prod_{i=1}^n U_i \quad (\text{B.7})$$

Abaixo listamos o código Fortran 90/95 implementado para gerar variáveis com distribuição de probabilidade *gamma*.

```

1 real(8) function MRandomVariable_gamma(a,b) result(rgamma)
2   real(8), intent(in) :: a,b
3   real(8) :: prod,rgamma
4   integer :: j
5   call MRandomVariable_assert((a .gt. 0).and.(b .gt.0), "MRandomVariable_gamaAB :
      a > 0 and b > 0" )
6   prod = 1.0;
7   do j=1, ceiling(a)
8     prod = prod * MRandomVariable_uniform(dble(0),dble(1));
9   enddo
10  rgamma = -b * log(prod);

```



```

11 end function
12 !Gera números aleatórios distribuídos uniformemente no intervalo [xMin, xMax].
13 real(8) function MRandomVariable_uniform(xMin, xMax) result(runiform)
14   real(8), intent(in) :: xMin, xMax
15   real(8) :: U, runiform
16   call random_number(U)
17   runiform = xMin + (xMax - xMin)*U
18 end function

```

Código B.1: Implementação Fortran 90/95 do gerador de números aleatórios com distribuição de probabilidade *gamma*.

B.4 Variável Com Distribuição Beta

A função de distribuição de probabilidade *beta* apresenta as seguintes propriedades (Saucier, 2000).

Função de densidade:

$$f(x) = \begin{cases} \frac{x^{v-1}(1-x)^{w-1}}{\beta(v,w)} & \text{se } 0 \leq x \leq 1 \\ 0 & \text{caso contrário} \end{cases} \quad (\text{B.8})$$

onde $\beta(v, w)$ é uma função *beta*, definida por $\beta(v, w) \equiv \int_0^1 t^{v-1}(1-t)^{w-1} dt$.

Função de distribuição de probabilidade: se c for um inteiro positivo, pode ser definida como segue.

$$F(x) = \begin{cases} \beta_x(v, w) \beta(v, w) & \text{se } 0 \leq x \leq 1 \\ 0 & \text{caso contrário} \end{cases} \quad (\text{B.9})$$

onde $\beta_x(v, w)$ é uma função *beta* incompleta, definida por $\beta_x(v, w) \equiv \int_0^x t^{v-1}(1-t)^{w-1} dt$.

O algoritmo para gerar números aleatórios que seguem a função de densidade *beta* no intervalo $[x_{min}, x_{max}]$ tem dois passos:

1. Gere duas variáveis com função de distribuição de probabilidade *gamma* da seguinte forma:
 $Y_1 \sim \text{gamma}(1, v)$ e $Y_2 \sim \text{gamma}(1, w)$.
2. Retorne X conforme a Eq. (B.10).

$$X = \begin{cases} x_{min} + (x_{max} - x_{min})Y_1/(Y_1 + Y_2) & \text{se } v \geq w \\ x_{max} - (x_{max} - x_{min})Y_2/(Y_1 + Y_2) & \text{se } v < w \end{cases} \quad (\text{B.10})$$

Abaixo listamos o código Fortran 90/95 implementado para gerar variáveis com distribuição de probabilidade *beta*.

```

1 recursive function MRandomVariable_beta( v, w, xMin, xMax) result(rbeta)
2   real(8), parameter :: p01 = dble(1.0)
3   real(8), intent(in) :: v, w, xMin, xMax
4   real(8) :: rbeta, y1, y2
5   if ( v < w ) then
6     rbeta = xMax - ( xMax - xMin ) * MRandomVariable_beta( w, v, xMin, xMax );
7     return
8   else

```

```

9      y1 = MRandomVariable_gamma(v, p01)
10     y2 = MRandomVariable_gamma(w, p01)
11     rbeta = xMin + ( xMax - xMin ) * y1 / ( y1 + y2 );
12   endif
13 end function

```

Código B.2: Implementação Fortran 90/95 do gerador de números aleatórios com distribuição de probabilidade *beta*.

A Figura B.1(a) mostra a função de distribuição de probabilidade *beta* para os parâmetros $v = 6$ e $w = 2$. A Figura B.1(b) mostra um histograma de 10000 números aleatórios gerados com distribuição *beta* considerando os parâmetros $v = 6$ e $w = 2$ através do código Fortran 90/95 implementado pelo autor.

(a) Função de distribuição *beta*

(b) Histograma dos números gerados pela execução do código Fortran 90/95

Figura B.1: Geração de variáveis aleatórias com função de distribuição de probabilidade *beta*.

A Figura B.2 é uma sobreposição das figuras B.1(a) e B.1(b). Através da qual, podemos confirmar que a aproximação do algoritmo implementado é bastante aceitável.

Figura B.2: *Sobreposição das figuras B.1(a) e B.1(b).*

Apêndice C

Dados dos Sistemas Elétricos Testados

C.1 Sistema de Garver 6 Barras

Os dados deste sistema foram extraídos de Garver (1970). Os dados das barras do sistema Sistema de Seis Barras com Redespacho da Geração (Garver) encontram-se na Tabela C.1 onde são apresentadas a geração máxima (g) e demanda (d) em *MW* de cada barra.

Tabela C.1: *Garver - Dados de barras*

Barra	g	d
1	150,00	80,00
2	0,00	240,00
3	360,00	40,00
4	0,00	160,00
5	0,00	240,00
6	600,00	0,00

Os dados das linhas (existentes e candidatas) do sistema IEEE24Bar encontram-se na Tabela C.2, onde n_{ij}^0 é o número de linhas existentes na configuração inicial, x_{ij} é a reatância do ramo $i - j$, $\bar{f}_{ij}$ é o fluxo máximo de potência entre $i - j$ em *MW* e c_{ij} é o custo de instalação do equipamento de transmissão e está em (10^3 US\$).

Tabela C.2: *Garver - Dados das linhas*

De	Para	n_{ij}^0	x_{ij}	$\bar{f}_{ij}$	c_{ij}
1	2	1	0,4000	120,00	40,00
1	3	0	0,3800	120,00	38,00
1	4	1	0,6000	100,00	60,00
1	5	1	0,2000	120,00	20,00
1	6	0	0,6800	90,00	68,00
2	3	1	0,2000	120,00	20,00
2	4	1	0,4000	120,00	40,00
2	5	0	0,3100	120,00	31,00
2	6	0	0,3000	120,00	30,00
3	4	0	0,5900	102,00	59,00
3	5	1	0,2000	120,00	20,00
3	6	0	0,4800	120,00	48,00
4	5	0	0,6300	95,00	63,00
4	6	0	0,3000	120,00	30,00
5	6	0	0,6100	98,00	61,00

C.2 Dados do Sistema Sul Brasileiro

Os dados das barras encontram-se na Tabela C.3 onde são apresentadas a geração máxima (g) e demanda (d) em MW de cada barra do sistema Sistema Sul Brasileiro (SULBr).

Tabela C.3: *SULBr - Dados de barras*

Barra	g	d
1	0,00	0,00
2	0,00	443,10
3	0,00	0,00
4	0,00	300,70
5	0,00	238,00
6	0,00	0,00
7	0,00	0,00
8	0,00	72,20
9	0,00	0,00
10	0,00	0,00
11	0,00	0,00
12	0,00	511,90
13	0,00	185,80
14	1257,00	0,00
15	0,00	0,00
16	2000,00	0,00
17	1050,00	0,00
18	0,00	0,00
19	1670,00	0,00
20	0,00	1091,20
21	0,00	0,00
22	0,00	81,90
23	0,00	458,10
24	0,00	478,20
25	0,00	0,00
26	0,00	231,90
27	220,00	0,00
28	800,00	0,00
29	0,00	0,00
30	0,00	0,00
31	700,00	0,00
32	500,00	0,00
33	0,00	229,10
34	748,00	0,00
35	0,00	216,00
36	0,00	90,10
37	300,00	0,00
38	0,00	216,00
39	600,00	0,00
40	0,00	262,10
41	0,00	0,00
42	0,00	1607,90
43	0,00	0,00
44	0,00	79,10
45	0,00	86,70
46	700,00	0,00

Os dados das linhas (existentes e candidatas) do sistema SULBr encontram-se na Tabela C.4, onde n_{ij}^0 é o número de linhas existentes na configuração inicial, x_{ij} é a reatância do ramo $i - j$, $\bar{f}_{ij}$ é o fluxo máximo de potência entre $i - j$ em MW e c_{ij} é o custo de instalação do equipamento de transmissão e está em (10^3 US\$).

Tabela C.4: *SULBr - Dados das linhas*

De	Para	n_{ij}^0	x_{ij}	$\bar{f}_{ij}$	c_{ij}
1	7	1	0,0616	270,00	4349,00
1	2	2	0,1065	270,00	7076,00
4	9	1	0,0924	270,00	6217,00
5	9	1	0,1173	270,00	7732,00
5	8	1	0,1132	270,00	7480,00
7	8	1	0,1023	270,00	6823,00
4	5	2	0,0566	270,00	4046,00
2	5	2	0,0324	270,00	2581,00
8	13	1	0,1348	240,00	8793,00
9	14	2	0,1756	220,00	11267,00
12	14	2	0,0740	270,00	5106,00
14	18	2	0,1514	240,00	9803,00
13	18	1	0,1805	220,00	11570,00
13	20	1	0,1073	270,00	7126,00
18	20	1	0,1997	200,00	12732,00
19	21	1	0,0278	1500,00	32632,00
16	17	1	0,0078	2000,00	10505,00
17	19	1	0,0061	2000,00	8715,00
14	26	1	0,1614	220,00	10409,00
14	22	1	0,0840	270,00	5712,00
22	26	1	0,0790	270,00	5409,00
20	23	2	0,0932	270,00	6268,00
23	24	2	0,0774	270,00	5308,00
26	27	2	0,0832	270,00	5662,00
24	34	1	0,1647	220,00	10611,00
24	33	1	0,1448	240,00	9399,00
33	34	1	0,1265	270,00	8288,00
27	36	1	0,0915	270,00	6167,00
27	38	2	0,2080	200,00	13237,00
36	37	1	0,1057	270,00	7025,00
34	35	2	0,0491	270,00	3591,00
35	38	1	0,1980	200,00	12631,00
37	39	1	0,0283	270,00	2329,00
37	40	1	0,1281	270,00	8389,00
37	42	1	0,2105	200,00	13388,00
39	42	3	0,2030	200,00	12934,00
40	42	1	0,0932	270,00	6268,00
38	42	3	0,0907	270,00	6116,00
32	43	1	0,0309	1400,00	35957,00
42	44	1	0,1206	270,00	7934,00
44	45	1	0,1864	200,00	11924,00
19	32	1	0,0195	1800,00	23423,00
46	19	1	0,0222	1800,00	26365,00

46	16	1	0,0203	1800,00	24319,00
18	19	1	0,0125	600,00	8178,00
20	21	1	0,0125	600,00	8178,00
42	43	1	0,0125	600,00	8178,00
2	4	0	0,0882	270,00	5965,00
14	15	0	0,0374	270,00	2884,00
46	10	0	0,0081	2000,00	10889,00
4	11	0	0,2246	240,00	14247,00
5	11	0	0,0915	270,00	6167,00
46	6	0	0,0128	2000,00	16005,00
46	3	0	0,0203	1800,00	24319,00
16	28	0	0,0222	1800,00	26365,00
16	32	0	0,0311	1400,00	36213,00
17	32	0	0,0232	1700,00	27516,00
19	25	0	0,0325	1400,00	37748,00
21	25	0	0,0174	2000,00	21121,00
25	32	0	0,0319	1400,00	37109,00
31	32	0	0,0046	2000,00	7052,00
28	31	0	0,0053	2000,00	7819,00
28	30	0	0,0058	2000,00	8331,00
27	29	0	0,0998	270,00	6672,00
26	29	0	0,0541	270,00	3894,00
28	41	0	0,0339	1300,00	39283,00
28	43	0	0,0406	1200,00	46701,00
31	41	0	0,0278	1500,00	32632,00
32	41	0	0,0309	1400,00	35957,00
41	43	0	0,0139	2000,00	17284,00
40	45	0	0,2205	180,00	13994,00
15	16	0	0,0125	600,00	8178,00
46	11	0	0,0125	600,00	8178,00
24	25	0	0,0125	600,00	8178,00
29	30	0	0,0125	600,00	8178,00
40	41	0	0,0125	600,00	8178,00
2	3	0	0,0125	600,00	8178,00
5	6	0	0,0125	600,00	8178,00
9	10	0	0,0125	600,00	8178,00

C.3 Dados do Sistema IEEE 24 Barras

Os dados das barras encontram-se na Tabela C.5 onde são apresentadas a geração máxima (g) e demanda (d) em *MW* de cada barra do sistema Sistema Teste IEEE de 24 Barras (IEEE24Bar).

Tabela C.5: *IEEE24Bar - Dados de barras*

Barra	g	d
1	576,00	324,00
2	576,00	291,00
3	0,00	540,00
4	0,00	222,00
5	0,00	213,00
6	0,00	408,00
7	900,00	375,00

8	0,00	513,00
9	0,00	525,00
10	0,00	585,00
11	0,00	0,00
12	0,00	0,00
13	1773,00	795,00
14	0,00	582,00
15	645,00	951,00
16	465,00	300,00
17	0,00	0,00
18	1200,00	999,00
19	0,00	543,00
20	0,00	384,00
21	1200,00	0,00
22	900,00	0,00
23	1980,00	0,00
24	0,00	0,00

Os dados das linhas (existentes e candidatas) do sistema IEEE24Bar encontram-se na Tabela C.6, onde n_{ij}^0 é o número de linhas existentes na configuração inicial, x_{ij} é a reatância do ramo $i - j$, $\bar{f}_{ij}$ é o fluxo máximo de potência entre $i - j$ em MW e c_{ij} é o custo de instalação do equipamento de transmissão e está em (10^3 US\$).

Tabela C.6: *IEEE24Bar - Dados das linhas*

De	Para	n_{ij}^0	x_{ij}	$\bar{f}_{ij}$	c_{ij}
1	2	1	0,0139	175,00	3000,00
1	3	1	0,2112	175,00	55000,00
1	5	1	0,0845	175,00	22000,00
2	4	1	0,1267	175,00	33000,00
2	6	1	0,1920	175,00	50000,00
3	9	1	0,1190	175,00	31000,00
3	24	1	0,0839	400,00	50000,00
4	9	1	0,1037	175,00	27000,00
5	10	1	0,0883	175,00	23000,00
6	10	1	0,0605	175,00	16000,00
7	8	1	0,0614	175,00	16000,00
8	9	1	0,1651	175,00	43000,00
8	10	1	0,1651	175,00	43000,00
9	11	1	0,0839	400,00	50000,00
9	12	1	0,0839	400,00	50000,00
10	11	1	0,0839	400,00	50000,00
10	12	1	0,0839	400,00	50000,00
11	13	1	0,0476	500,00	66000,00
11	14	1	0,0418	500,00	58000,00
12	13	1	0,0476	500,00	66000,00
12	23	1	0,0966	500,00	134000,00
13	23	1	0,0865	500,00	120000,00
14	16	1	0,0389	500,00	54000,00
15	16	1	0,0173	500,00	24000,00
15	21	2	0,0490	500,00	68000,00
15	24	1	0,0519	500,00	72000,00

16	17	1	0,0259	500,00	36000,00
16	19	1	0,0231	500,00	32000,00
17	18	1	0,0144	500,00	20000,00
17	22	1	0,1053	500,00	146000,00
18	21	2	0,0259	500,00	36000,00
19	20	2	0,0396	500,00	55000,00
20	23	2	0,0216	500,00	30000,00
21	22	1	0,0678	500,00	94000,00
1	8	0	0,1344	500,00	35000,00
2	8	0	0,1267	500,00	33000,00
6	7	0	0,1920	500,00	50000,00
13	14	0	0,0447	500,00	62000,00
14	23	0	0,0620	500,00	86000,00
16	23	0	0,0822	500,00	114000,00
19	23	0	0,0606	500,00	84000,00

Referências Bibliográficas

- Alba, E. **Parallel Metaheuristics: A new class of algorithms**. NJ, USA: John Wiley & Sons, 2005.
- Albuquerque, L. L. d.; Almeida, A. T. d.; Cavalcante, C. A. V. A.-n. Aplicabilidade da programação matemática multiobjetivo no planejamento da expansão de longo prazo da geração no Brasil. **Pesquisa Operacional**, [S.l.], v.29, p.153 – 177, 04 2009.
- Almasi, G. S.; Gottlieb, A. **Highly parallel computing**. Redwood City, CA, USA: Benjamin-Cummings Publishing Co., Inc., 1994.
- Asada, E. N. et al. A branch-and-bound algorithm for the multi-stage transmission expansion planning. In: **Anais...** [S.l.: s.n.], 2005. p.171–176. Proc. IEEE Power Engineering Society General Meeting.
- Baita, F. et al. Genetic algorithm with redundancies for the vehicle scheduling problem. In: Biethahn, J.; Nissen, V. (Ed.). **Evolutionary Algorithms in Management Applications**. [S.l.: s.n.], 1995. p.341–353.
- Baldick, R.; Kahn, E. Transmission planning issues in a competitive economic environment. **IEEE Transactions on Power Systems**, [S.l.], v.8, n.4, p.1497–1503, 1993.
- Barney, B. **Message passing interface (mpi)**. Disponível em: <https://computing.llnl.gov/tutorials/mpi/>, acesso em 20 de janeiro 01 de 2012.
- Barros, J. R. P. **Metodologia de Planejamento da Expansão da Transmissão Baseada em Custos Marginais de Confiabilidade**. 2004. Tese (Doutorado em Ciência da Computação) — Universidade Federal de Itajubá (UNIFEI).
- Baykasoglu, A. Goal programming using multiple objective tabu search. **Journal of the Operational Research Society**, [S.l.], v.52, n.12, p.1359–1369, 2001.
- Benders, J. F. Partitioning procedure for solving mixed variables programming problems. **Numerische Mathematik**, [S.l.], p.238–262, 1962.
- Berntsen, J. A. et al. MAC Layer Interconnection of IEEE 802 Local Area Networks. **Computer Networks and ISDN Systems**, [S.l.], v.10, n.5, p.259–273, Dec. 1985.
- Bjorndal, M.; Jörnsten, K. Benefits from coordinating congestion management–The Nordic power market. **Energy Policy**, [S.l.], v.35, n.3, p.1978 – 1991, 2007.
- Branke, J. et al. Parallelizing Multi-Objective Evolutionary Algorithms: Cone Separation. In: CONGRESS ON EVOLUTIONARY COMPUTATION (CEC'2004), 2004., Portland, Oregon, USA. **Anais...** IEEE Service Center, 2004. v.2, p.1952–1957.
- Bresesti, P. et al. The benefits of transmission expansions in the competitive electricity markets. **Energy**, [S.l.], v.34, n.3, p.274–280, 2009.
- Buygi, M. O. et al. Market-based transmission expansion planning. **IEEE Transactions on Power Systems**, [S.l.], v.19, n.4, p.2060–2067, 2004.

- Buygi, M. O. et al. **Market based transmission expansion planning: fuzzy risk assessment**. Electric Utility Deregulation, Restructuring and Power Technologies, 2004.(DRPT 2004). Proceedings of the 2004 IEEE International Conference on.
- Cantu-Paz, E. Designing Efficient Master-Slave Parallel Genetic Algorithms. In: GENETIC PROGRAMMING 1998: PROCEEDINGS OF THE THIRD ANNUAL CONFERENCE, University of Wisconsin, Madison, Wisconsin, USA. **Anais...** Morgan Kaufmann, 1998. p.455.
- Cantú-Paz, E. A Survey of Parallel Genetic Algorithms. **Calculateurs Paralleles Reseaux et Systems Repartis**, [S.l.], v.10, n.2, p.141–171, 1998.
- Ceciliano, J. L.; Nieva, R. **Transmission network planning using evolutionary programming**. Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on.
- Cedeno, E. B.; Arora, S. Performance comparison of Transmission Network Expansion Planning under deterministic and uncertain conditions. **International Journal of Electrical Power and Energy Systems**, [S.l.], v.33, n.7, p.1288 – 1295, 2011.
- Chankong, V.; Haimes, Y. **Multiobjective decision making: theory and methodology**. [S.l.]: North-Holland New York, 1983.
- Choi, J.; El-Keib, A. A.; Tran, T. A fuzzy branch and bound-based transmission system expansion planning for the highest satisfaction level of the decision maker. **IEEE Transactions on Power Systems**, [S.l.], v.20, n.1, p.476–484, 2005.
- Choi, J. et al. A method for transmission system expansion planning considering probabilistic reliability criteria. **IEEE Transactions on Power Systems**, [S.l.], v.20, n.3, p.1606–1615, 2005.
- Choi, J.; Mount, T.; Thomas, R. Transmission System Expansion Plans in View Point of Deterministic, Probabilistic and Security Reliability Criteria. In: MORGAN KAUFMANN. **Anais...** [S.l.: s.n.], 2007. p.380. Second International Conference on Innovative Computing, Information and Control, 2007.
- Chung, T. S. et al. Multi-objective transmission network planning by a hybrid GA approach with fuzzy decision analysis. **International Journal of Electrical Power and Energy Systems**, [S.l.], v.25, n.3, p.187–192, 2003.
- Coello, C. A. C. Recent trends in evolutionary multiobjective optimization. **Evolutionary Multiobjective Optimization**, [S.l.], p.7–32, 2005.
- Coello, C. A. C.; Lamont, G. B.; Van Veldhuizen, D. A. **Evolutionary algorithms for solving multi-objective problems**. [S.l.]: Springer-Verlag New York Inc, 2007.
- Coley, D. A. **An Introduction to Genetic Algorithms for Scientists and Engineers**. [S.l.]: World Scientific Publishing Company, 1997.
- Corne, D. et al. **PESA-II: region-based selection in evolutionary multiobjective optimization**. Proceedings of the Genetic and Evolutionary Computation Conference.
- Corne, D.; Knowles, J.; Oates, M. **The Pareto envelope-based selection algorithm for multiobjective optimization**. Parallel Problem Solving from Nature PPSN VI.
- Crainic, T. G.; Toulouse, M. Introduction to the Special Issue on Parallel Meta-Heuristics. **Journal of Heuristics**, [S.l.], v.8, p.247–249, 2002.

- Crainic, T. G.; Toulouse, M. Parallel Strategies for Meta-Heuristics. In: Glover, F.; Kochenberger, G. (Ed.). **Handbook of Metaheuristics**. [S.l.]: Springer New York, 2003. p.475–513. (International Series in Operations Research & Management Science, v.57).
- Culler, D. E.; Singh, J. P.; Gupta, A. **Parallel computer architecture - a hardware / software approach**. [S.l.]: Morgan Kaufmann, 1999.
- Cung, V. D. et al. Strategies for the parallel implementation of metaheuristics. **Essays and surveys in metaheuristics**, [S.l.], p.263–308, 2002.
- Da Silva, E. L. et al. Transmission network expansion planning under a tabu search approach. **IEEE Transactions on Power Systems**, [S.l.], v.16, n.1, p.62–68, 2001.
- David, A. K.; Wen, F. **Transmission planning and investment under competitive electricity market environment**. Power Engineering Society Summer Meeting, 2001. IEEE.
- Deb, K. Multi-objective genetic algorithms: problem difficulties and construction of test problems. **Evolutionary Computation Journal**, [S.l.], v.7, n.3, p.205–230, 1999.
- Deb, K. **Multi-objective optimization using evolutionary algorithms**. [S.l.]: Wiley, 2001.
- Deb, K. et al. A fast and elitist multiobjective genetic algorithm: nsga-ii. **IEEE transactions on evolutionary computation**, [S.l.], v.6, n.2, p.182–197, 2002.
- Duncan, R. A survey of parallel computer architectures. **Computer**, [S.l.], v.23, n.2, 1990.
- Ehrenmann, A.; Smeers, Y. Capacity Expansion in Non-Regulated Electricity Markets. In: ANNUAL HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, 39., Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2006. (HICSS '06).
- Ehrgott, M. **Multicriteria optimization**. [S.l.]: Springer Verlag, 2005.
- Escobar, A. H.; Gallego, R. A.; Romero, R. Multistage and coordinated planning of the expansion of transmission systems. **IEEE Transactions on Power Systems**, [S.l.], v.19, n.2, p.735–744, May 2004.
- Escobar, A. H.; Romero, R. A.; Gallego, R. A. Transmission network expansion planning considering multiple generation scenarios. In: IEEE COMPUTER SOCIETY. **Anais...** [S.l.: s.n.], 2008. p.1–6. Transmission and Distribution Conference and Exposition: Latin America, 2008 IEEE/PES.
- Fan, H.; Cheng, H.; Yao, L. A Bi-Level Programming Model for Multistage Transmission Network Expansion Planning in Competitive Electricity Market. In: **Anais...** [S.l.: s.n.], 2009. p.1 –6. Power and Energy Engineering Conference, 2009. APPEEC 2009. Asia-Pacific.
- Fang, R.; Hill, D. J. A new strategy for transmission expansion in competitive electricity markets. **IEEE Power Engineering Review**, [S.l.], v.22, n.11, p.60–60, 2002.
- Fang, R. S.; David, A. K. Transmission congestion management in an electricity market. **Power Systems, IEEE Transactions on**, [S.l.], v.14, n.3, p.877–883, 1999.
- Flynn, M. J. Very high-speed computing systems. **Proceedings of the IEEE**, [S.l.], v.54, n.12, 1966.
- Fonseca, C. M. et al. **A tutorial on the performance assessment of stochastic multiobjective optimizers**. Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005).

- Fonseca, C. M.; Fleming, P. J. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: OF THE FIFTH INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS. **Proceedings...** [S.l.: s.n.], 1993. v.423, p.416–423.
- Foster, I. **Designing and building parallel programs - concepts and tools for parallel software engineering**. [S.l.]: Addison-Wesley, 1995.
- Foster, I. et al. A Wide-Area Implementation of the Message Passing Interface. **Parallel Computing**, [S.l.], v.24, n.12, 1998.
- Fowler, M. **Refactoring**. [S.l.]: Addison-Wesley, 2000.
- Gallego, R. A.; Monticelli, A.; Romero, R. Transmission system expansion planning by an extended genetic algorithm. **IEEE Proceedings-Generation, Transmission and Distribution**, [S.l.], v.145, p.329, 1998.
- Gallego, R. A.; Romero, R.; Monticelli, A. J. Tabu search algorithm for network synthesis. **IEEE Transactions on Power Systems**, [S.l.], v.15, n.2, p.490–495, 2000.
- Garcia, B. L.; Mahey, P.; LeBlanc, L. J. Iterative improvement methods for a multiperiod network design problem. **European Journal of Operational Research**, [S.l.], v.110, n.1, p.150–165, 1998.
- Garver, L. L. Transmission Network Estimation Using Linear Programming. **IEEE Transactions on Power Apparatus and Systems**, [S.l.], n.7, p.1688–1697, Sept. 1970.
- Glover, F. Tabu Search-Part I. **ORSA Journal on Computing**, [S.l.], v.1, n.3, p.190–206, 1989.
- Glover, F. Tabu search-part II. **INFORMS Journal on Computing**, [S.l.], v.2, n.1, 1990.
- Glover, F.; Laguna, M. **Tabu Search**. [S.l.]: Kluwer Academic Publishers, 1997.
- Goldberg, D. E.; Deb, K. A comparative analysis of selection schemes used in genetic algorithms. **Foundations of genetic algorithms**, [S.l.], v.1, p.69–93, 1991.
- Goldberg, D. E.; Richardson, J. Genetic algorithms with sharing for multimodal function optimization. In: Hillsdale, NJ. **Anais...** Lawrence Erlbaum, 1987. p.41–49. Genetic Algorithms and Their Applications: Proceeding of the Second International Conference on Genetic Algorithms.
- Gropp, W.; Lusk, E.; Skjellum, A. **Using MPI: portable parallel programming with the message-passing interface**. [S.l.]: MIT Press, 1999.
- Gropp, W.; Lusk, E.; Thakur, R. **Using MPI-2: advanced features of the message-passing interface**. [S.l.]: MIT Press, 1999.
- Güler, T.; Gross, G. The economic evaluation of system security criterion selection in the market environment. In: LAWRENCE ERLBAUM. **Anais...** [S.l.: s.n.], 2010. p.1 –8. Power and Energy Society General Meeting, 2010 IEEE.
- Haffner, S. et al. Branch and bound algorithm for transmission system expansion planning using a transportation model. **IEEE Proceedings-Generation, Transmission and Distribution**, [S.l.], v.147, n.3, p.149–156, 2000.
- Haffner, S. L. **O Planejamento da expansão dos sistemas eletricos no contexto de um ambiente competitivo**. 2000. Tese (Doutorado em Ciência da Computação) — UNICAMP SP, Brazil.

- Harvey, S. M.; Hogan, W. W.; Pope, S. L. Transmission capacity reservations implemented through a spot market with transmission congestion contracts. **The Electricity Journal**, [S.l.], v.9, n.9, p.42 – 55, 1996.
- Hiroyasu, T.; Miki, M.; Watanabe, S. The new model of parallel genetic algorithm in multi-objective optimization problems-divided range multi-objective genetic algorithm. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, JULY 2000. **Anais...** [S.l.: s.n.], 2000. v.1, p.333–340.
- Ho, S. et al. A Tabu method to find the Pareto solutions of multiobjective optimal design problems in electromagnetics. **Magnetics, IEEE Transactions on**, [S.l.], v.38, n.2, p.1013–1016, 2002.
- Hoare, C. A. R. Algorithm 64: quicksort. **Commun. ACM**, New York, NY, USA, v.4, p.321–322, July 1961.
- Hoare, C. Quicksort. **The Computer Journal**, [S.l.], v.5, n.1, p.10, 1962.
- Holland, J. H. **Adaptation in natural and artificial systems**. [S.l.]: Ann Arbor, 1975.
- Horn, J.; Nafpliotis, N.; Goldberg, D. E. A niched Pareto genetic algorithm for multiobjective optimization. In: EVOLUTIONARY COMPUTATION, 1994. IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE., PROCEEDINGS OF THE FIRST IEEE CONFERENCE ON. **Anais...** [S.l.: s.n.], 1994. p.82–87.
- Hwang, K.; Faye, A. **Computer architecture and parallel processing**. New York, NY, USA: McGraw-Hill, 1984.
- Jin, Y. et al. New discrete method for particle swarm optimization and its application in transmission network expansion planning. **Electric Power Systems Research**, [S.l.], v.77, n.3-4, p.227–233, 2007.
- Jones, D.; Mirrazavi, S.; Tamiz, M. Multi-objective meta-heuristics: an overview of the current state-of-the-art. **European journal of operational research**, [S.l.], v.137, n.1, p.1–9, 2002.
- Jozefowiez, N.; Semet, F.; Talbi, E. Target aiming Pareto search and its application to the vehicle routing problem with route balancing. **Journal of Heuristics**, [S.l.], v.13, n.5, p.455–469, 2007.
- Knight, U. G. W. The logical design of electrical networks using linear programming methods. **Proceedings of the IEE-Part A: Power Engineering**, [S.l.], v.107, n.33, p.306–314, 1960.
- Knowles, J.; Corne, D. **The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation**. Proceedings of the 1999 Congress on Evolutionary Computation.
- Knowles, J.; Thiele, L.; Zitzler, E. A tutorial on the performance assessment of stochastic multiobjective optimizers. **Tik report**, [S.l.], v.214, 2006.
- Kumar, A.; Srivastava, S. C.; Singh, S. N. Congestion management in competitive power market: a bibliographical survey. **Electric Power Systems Research**, [S.l.], v.76, n.1-3, p.153 – 164, 2005.
- Land, A. H.; Doig, A. G. An automatic method of solving discrete programming problems. **Econometrica: Journal of the Econometric Society**, [S.l.], p.497–520, 1960.
- Latorre, G. et al. Classification of publications and models on transmission expansion planning. **IEEE Transactions on Power Systems**, [S.l.], v.18, n.2, p.938–946, 2003.

- 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**, [S.l.], v.PAS-93, n.5, 1974.
- Leme, A. A. A reforma do setor elétrico no Brasil, Argentina e México: contrastes e perspectivas em debate. **Rev. Sociol. Polit**, [S.l.], v.17, p.33, 2009.
- Levi, V. A. A new mixed-integer methodology for optimal transmission expansion planning. **Electric power systems research**, [S.l.], v.32, n.3, p.227–238, 1995.
- Lu, M. et al. A novel approach to evaluate congestion for composite power system planning in a competitive electricity market. In: **Anais...** [S.l.: s.n.], 2006. p.8 pp. Power Engineering Society General Meeting, 2006. IEEE.
- Léautier, T.-O.; Thelen, V. Optimal expansion of the power transmission grid: why not? **Journal of Regulatory Economics**, [S.l.], v.36, p.127–153, 2009.
- Marler, R. T.; Arora, J. S. Survey of multi-objective optimization methods for engineering. **Structural and Multidisciplinary Optimization**, [S.l.], v.26, p.369–395, 2004.
- Merkey, P. **Beowulf History**. Disponível em: <http://www.beowulf.org/overview/history.html>, acessado em 02 de setembro de 2011.
- Mezmaz, M.; Melab, N.; Talbi, E. Using the multi-start and island models for parallel multi-objective optimization on the computational grid. In: IEEE INTERNATIONAL CONFERENCE ON E-SCIENCE'06 AND GRID COMPUTING. **Anais...** [S.l.: s.n.], 2006.
- Michalewicz, Z. **Genetic algorithms+ data structures=Evolution Programs**. 3.ed. [S.l.]: Springer, 1996.
- Miettinen, K. **Nonlinear Multiobjective Optimization**. [S.l.]: Boston:Kluwer, 1999.
- Miettinen, K.; Ruiz, F.; Wierzbicki, A. P. Multiobjective Optimization Interactive and Evolutionary Approaches. **Theoretical Computer Science**, [S.l.], v.5252, p.27–57, 2008.
- Miller, B. L.; Shaw, M. J. Genetic algorithms with dynamic niche sharing for multimodal function optimization. In: **Anais...** [S.l.: s.n.], 1996. p.786 –791. Proceedings of IEEE International Conference on Evolutionary Computation.
- Monticelli, A. **Fluxo de carga em redes de energia elétrica**. [S.l.]: Editora Edgard Blücher Ltda, 1983.
- Monticelli, A. et al. Interactive Transmission Network Planning Using a Least-Effort Criterion. **IEEE Transactions on Power Apparatus and Systems**, [S.l.], v.PAS-101, p.3919–3925, 1982.
- MPI-2, M. F. **Extensions to the message-passing interface**. [S.l.: s.n.], 1997.
- MPI, M. F. A message-passing interface standard. **International Journal of Supercomputer Applications**, [S.l.], 1995.
- Murray, B. **Electricity markets: investment, performance and analysis**. [S.l.]: Wiley, 1998.
- Murtagh, B. A.; Saunders, M. A. **MINOS 5.4 User's Guide (preliminary)**. [S.l.]: Department of Operations Research Stanford University, Technical report, Revised Jan, 1987.
- Muyl, F.; Dumas, L.; Herbert, V. Hybrid method for aerodynamic shape optimization in automotive industry. **Computers & Fluids**, [S.l.], v.33, n.5-6, p.849–858, 2004.

- Neufville, R. Real options: dealing with uncertainty in systems planning and design. **Integrated Assessment**, [S.l.], v.4, n.1, p.26–34, 2003.
- Okabe, T. **Evolutionary Multi-Objective Optimization**: on the distribution of offspring in parameter and fitness space. [S.l.]: Shaker Verlag, Aachen, 2004.
- Okabe, T. et al. Comparative studies on micro heat exchanger optimisation. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION. **Anais...** [S.l.: s.n.], 2003. v.1, p.647–654.
- Pereira, M. V. F. et al. A decomposition approach to automated generation/transmission expansion planning. **IEEE Transactions on Power Apparatus and Systems**, [S.l.], p.3074–3083, 1985.
- Pereira, M. V. F.; Pinto, L. M. Application Of Sensitivity Analysis Of Load Supplying Capability To Interactive Transmission Expansion Planning. **IEEE Transactions on Power Apparatus and Systems**, [S.l.], v.PAS-104, p.381–389, 1985.
- Quinn, M. J. **Parallel Programming in C with MPI and OpenMP**. [S.l.]: Mcgraw-Hill Higher Education, 2003.
- Rajkumar, B. (Ed.). **High Performance Cluster Computing**: architectures and systems. [S.l.]: Prentice Hall, 1999.
- Romero, R. et al. Constructive heuristic in branch-and-bound structure applied to transmission network expansion planning. **IET Generation, Transmission & Distribution**, [S.l.], v.1, p.318–323, 2007.
- Romero, R. et al. Evaluation of hybrid models for static and multistage transmission system planning. **Sba: Controle & Automação Sociedade Brasileira de Automatica**, [S.l.], v.18, p.106–114, 2007.
- Romero, R.; Gallego, R. A.; Monticelli, A. Transmission System Expansion Planning by Simulated Annealing. **IEEE Transactions on Power Systems**, [S.l.], v.11, n.1, p.364–369, fevereiro 1996.
- Romero, R.; Monticelli, A. A zero-one implicit enumeration method for optimizing investments in transmission expansion planning. **IEEE Transactions on Power Systems**, [S.l.], v.9, n.3, p.1385–1391, Aug 1994.
- Romero, R.; Monticelli, A. A Hierarchical Decomposition Approach for Transmission Network Expansion Planning. **IEEE Transactions on Power Systems**, [S.l.], v.9, n.3, p.1385–1391, Aug 1994.
- Ross, S. M. **Introduction to Probability Models**. 10.ed. [S.l.]: Academic Press, 2009.
- Saucier, R. **Computer generation of statistical distributions**. [S.l.]: Army Research Lab Aberdeen Proving Ground MD., 2000.
- Schaffer, J. **Multiple objective optimization with vector evaluated genetic algorithms**. Proceedings of the 1st International Conference on Genetic Algorithms.
- Shariati, H. et al. Transmission expansion planning considering security cost under market environment. In: **Anais...** [S.l.: s.n.], 2008. p.1430 –1435. Electric Utility Deregulation and Restructuring and Power Technologies, 2008. DRPT 2008. Third International Conference on.
- Shrestha, G. B.; Fonseka, P. A. J. Congestion-driven transmission expansion in competitive power markets. **Power Systems, IEEE Transactions on**, [S.l.], v.19, n.3, p.1658 – 1665, 2004.

- Silva Junior, I. J. **Planejamento da Expansão o de Sistemas de Transmissão Considerando Segurança e Planos de Programação da Geração**. 2005. Tese (Doutorado em Ciência da Computação) — UNICAMP SP, Brazil.
- Silva Junior, I. J. et al. Transmission Network Expansion Planning Considering Uncertainty in Demand. **Power Systems, IEEE Transactions on**, [S.l.], v.21, n.4, p.1565 –1573, 2006.
- Silverman Bernard, W. **Density Estimation for Statistics and Data Analysis**. [S.l.]: London: Chapman Hall, 1986.
- Sousa, A. S.; Asada, E. N. Combined heuristic with fuzzy system to transmission system expansion planning. **Electric Power Systems Research**, [S.l.], v.81, n.1, p.123 – 128, 2011.
- Srinivas, N.; Deb, K. Multiobjective optimization using nondominated sorting in genetic algorithms. **Evolutionary computation**, [S.l.], v.2, n.3, p.221–248, 1994.
- Stadler, W. **Initiators of multicriteria optimization**. Jahn, J., W. Krabs (Eds.): **recent advances and historical development of vector optimization**. Proc. of an Int. Conf. on Vector Optimization, Darmstadt.
- Sum-Im, T. et al. Differential evolution algorithm for static and multistage transmission expansion planning. **IET Generation, Transmission and Distribution**, [S.l.], v.3, n.4, p.365–384, 2009.
- Talbi, E.-G. **Metaheuristics: from design to implementation**. [S.l.]: John Wiley & Sons Inc, 2009.
- Tor, O. B.; Guven, A. N.; Shahidehpour, M. Congestion-Driven Transmission Planning Considering the Impact of Generator Expansion. **Power Systems, IEEE Transactions on**, [S.l.], v.23, n.2, p.781 –789, May 2008.
- Torre, S. d. l.; Conejo, A. J.; Contreras, J. Transmission Expansion Planning in Electricity Markets. **Power Systems, IEEE Transactions on**, [S.l.], v.23, n.1, p.238 –248, 2008.
- Veldhuizen, D. V. **Multiobjective evolutionary algorithms: classifications, analyses, and new innovations**. 1999. Tese (Doutorado em Ciência da Computação) — Dayton OH: Air Force Institute of Technology.
- Villasana, R.; Garver, L. L.; Salon, S. J. Transmission Network Planning Using Linear Programming. **IEEE Transactions on Power Apparatus and Systems**, [S.l.], v.PAS-104, p.349–356, 1985.
- Xiao, N.; Armstrong, M. A specialized island model and its application in multiobjective optimization. In: GENETIC AND EVOLUTIONARY COMPUTATION–GECCO 2003. **Anais...** [S.l.: s.n.], 2003. p.213–213.
- Zitzler, E.; Brockhoff, D.; Thiele, L. The hypervolume indicator revisited: on the design of pareto-compliant indicators via weighted integration. In: EVOLUTIONARY MULTI-CRITERION OPTIMIZATION. **Anais...** [S.l.: s.n.], 2007. p.862–876.
- Zitzler, E.; Deb, K.; Thiele, L. Comparison of multiobjective evolutionary algorithms: empirical results. **Evolutionary computation**, [S.l.], v.8, n.2, p.173–195, 2000.
- Zitzler, E.; Knowles, J.; Thiele, L. Quality Assessment of Pareto Set Approximations. In: Branke, J. et al. (Ed.). **Multiobjective Optimization: interactive and evolutionary approaches**. [S.l.]: Springer, 2008. p.373–404.

- Zitzler, E.; Laumanns, M.; Thiele, L. **SPEA2: improving the strength pareto evolutionary algorithm.** . [S.l.]: Swiss Federal Institute of Technology, TIK-Report, 2001.
- Zitzler, E.; Thiele, L. **An evolutionary algorithm for multiobjective optimization: the strength pareto approach.** [S.l.]: Swiss Federal Institute of Technology, TIK-Report, 1998.
- Zitzler, E.; Thiele, L. Multiobjective optimization using evolutionary algorithms - A comparative case study. In: PARALLEL PROBLEM SOLVING FROM NATURE-PPSN V. **Anais...** [S.l.: s.n.], 1998. p.292–301.
- Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. **Evolutionary Computation, IEEE Transactions on**, [S.l.], v.3, n.4, p.257–271, 1999.
- Zuluaga, A. H. E. **Análise Crítica de Aspectos de Modelagem Matemática no Planejamento da Expansão a Longo Prazo de Sistemas de Transmissão.** 2008. Tese (Doutorado em Ciência da Computação) — Universidade Estadual Paulista - Ilha Solteira.
- Zuluaga, A. H. E.; Romero, R. A.; Gallego, R. A. **Transmission network expansion planning considering multiple generation scenarios.** Transmission and Distribution Conference and Exposition: Latin America, 2008 IEEE/PES.