

UNIVERSIDADE ESTADUAL PAULISTA

CAMPUS DE ILHA SOLTEIRA - FACULDADE DE ENGENHARIA

Departamento de Engenharia Elétrica

Caixa Postal 31

15385.000 - Ilha Solteira - SP

Tel (fax): 55-18-7622125

Planejamento a Longo Prazo da Expansão de Sistemas de Transmissão de Energia Elétrica

Texto que sistematiza o estado da arte em planejamento de sistemas de transmissão assim como nossa contribuição nesta importante linha de pesquisa após do título de Doutor, submetido à Universidade Estadual Paulista - Faculdade de Engenharia de Ilha Solteira, para a obtenção do Título de Livre-Docente.

Rubén Augusto Romero Lázaro

Ilha Solteira, novembro de 1999

1210001173



UNIVERSIDADE ESTADUAL PAULISTA

CAMPUS DE ILHA SOLTEIRA - FACULDADE DE ENGENHARIA

Departamento de Engenharia Elétrica

Caixa Postal 31

15385.000 - Ilha Solteira - SP

Tel (fax): 55-18-7622125

000003640

Planejamento a Longo Prazo da Expansão de Sistemas de Transmissão de Energia Elétrica

Texto que sistematiza o estado da arte em planejamento de sistemas de transmissão assim como nossa contribuição nesta importante linha de pesquisa após do título de Doutor, submetido à Universidade Estadual Paulista - Faculdade de Engenharia de Ilha Solteira, para a obtenção do Título de Livre-Docente.

Rubén Augusto Romero Lázaro

NRD-006101-Proc. 040/01

UNESP - "CAMPUS DE ILHA SOLTEIRA" Ilha Solteira, novembro de 1999
SERVIÇO TÉCN. DE BIBLIOTECA E DOCUMENTAÇÃO

DATA DE CHEGADA	DATA DE TOMBO 31/03/01
REGISTRADO POR Lelia	TOMBO 1173
AQUISIÇÃO Doacas R\$ 19,00 Aut. 5	CLASSIFICAÇÃO RD 431.p

1210001173



3040403-Transm. ener. eletr. dist. energ. elétrica

Sumário

1	Introdução	1
2	Modelagem do Problema de Planejamento de Sistemas de Transmissão	3
2.1	Modelo de Transportes	5
2.1.1	Modelo Matemático	5
2.1.2	Exemplo 2.1: Sistema de 3 Barras	6
2.1.3	Exemplo 2.2: Caso Especial - Sistema de 3 Barras Completamente Ilhado	9
2.2	Modelo Híbrido	10
2.2.1	Modelo Matemático	11
2.2.2	Exemplo 2.3: Equivalência entre as duas formulações matemáticas	13
2.3	Modelo DC	16
2.3.1	Modelo Matemático	16
2.3.2	Exemplo 2.4: Equivalência entre as duas formulações matemáticas	17
2.4	Modificações na Modelagem Básica	20
3	Algoritmos Heurísticos Usados no Planejamento de Sistemas de Transmissão	23
3.1	Heurísticas Construtivas Usando o Modelo de Transportes	25
3.1.1	O Algoritmo de Garver [2]	25
3.2	Heurísticas Construtivas Usando o Modelo DC	27
3.2.1	O Algoritmo de Mínimo Esforço	28
3.2.2	O Algoritmo de Mínimo Corte de Carga	31
3.3	Heurísticas Construtivas que Usam Modelos Mistos	33



3.3.1	O Algoritmo de Villasana-Garver	33
3.3.2	O Algoritmo de Rede Marginal de Levi-Calovic	36
4	Algoritmos de Planejamento Ótimo Usando o Modelo de Transportes	42
4.1	Planejamento Ótimo Usando um Algoritmo de Branch and Bound Especializado	43
4.1.1	Algoritmo B & B Especializado	47
4.2	Planejamento Ótimo Hierarquizado Usando a Decomposição de Benders e Relaxação	47
4.2.1	A Decomposição de Benders	49
4.2.2	Algoritmo de Planejamento Hierarquizado	51
5	Algoritmos de Decomposição Matemática Usando o Modelo DC	54
5.1	A Decomposição Matemática de Benders	56
5.1.1	A Decomposição de Benders Generalizada	57
5.2	Planejamento Ótimo Hierarquizado Usando a Decomposição de Benders e Relaxação	61
5.2.1	Algoritmo de Planejamento Hierarquizado	63
6	Planejamento de Sistemas de Transmissão Usando “Simulated Annealing”	65
6.1	Introdução	65
6.2	Simulated Annealing: Teoria	65
6.2.1	O Algoritmo de Metrópolis	65
6.2.2	Algoritmo de “Simulated Annealing” (SA)	66
6.2.3	Analogia Física e Apresentação da Metaheurística	67
6.2.4	Programa de Esfriamento	67
6.3	SA Aplicado ao Problema de Planejamento	69
6.3.1	A Representação do Problema	70
6.3.2	Mecanismo de Transição	71
6.3.3	O Programa de Esfriamento	74



6.3.4	Algoritmo de “Simulated Annealing” Implementado	77
6.4	SA Paralelo Aplicado ao Problema de Planejamento	79
6.4.1	Algoritmo da Divisão:	81
7	Planejamento de Sistemas de Transmissão Usando o Algoritmo Genético	82
7.1	Introdução	82
7.2	O Algoritmo Genético: Teoria	83
7.2.1	População Inicial	84
7.2.2	Mecanismo de Codificação do Problema	85
7.2.3	Parâmetros de Controle	85
7.2.4	Avaliação do Nível de “Fitness”	86
7.2.5	Seleção e Operadores Genéticos do Ciclo Geracional	86
7.3	O Algoritmo Genético Básico Aplicado ao Problema de Planejamento	88
7.3.1	A Representação e Codificação do Problema	89
7.3.2	O Mecanismo Geracional: Seleção - “Crossover” - Mutação	92
7.3.3	Geração da População Inicial	93
7.3.4	Parâmetros de Controle e Critérios de Parada	95
7.4	Implementações Especializadas no Algoritmo Genético	95
7.4.1	Taxa de Mutação Variável	96
7.4.2	Seleção Limitada	97
7.4.3	Implementação dos “Building Blocks” Através dos Caminhos Novos	97
7.4.4	Formas Especializadas de Geração da População Inicial	100
7.4.5	Parâmetro α Variável	100
7.5	Algoritmo Implementado	101
8	Planejamento de Sistemas de Transmissão Usando Busca Tabu	102
8.1	Busca Tabu: Teoria	102
8.1.1	Memória de Curto Prazo e seus Principais Elementos	106



8.1.2	Memória de Longo Prazo e suas Estratégias Associadas	108
8.1.3	Oscilação Estratégica	110
8.1.4	Path Relinking	110
8.2	Busca Tabu Aplicado ao Problema de Planejamento	111
8.2.1	Algoritmo TS Básico	111
8.2.2	Implementações Especializadas no Algoritmo de Busca Tabu	114
9	Principais Contribuições em Pesquisas do Problema de Planejamento de Sis- temas de Transmissão	118
9.1	Contribuições na Modelagem Matemática	118
9.2	Contribuições no Desenvolvimento de Algoritmos de Solução	119
9.2.1	Algoritmos Heurísticos	119
9.2.2	Algoritmos de Otimização Clássica	120
9.2.3	Algoritmos de Otimização Combinatorial	121
9.2.4	Contribuições no Desenvolvimento de Algoritmos Correlatos	122
9.3	Conclusões e Novas Linhas de Pesquisa	123
A	Dados dos Sistemas Testados	128



Capítulo 1

Introdução

O problema de planejamento de sistemas de transmissão a longo prazo é um problema clássico de sistemas de energia elétrica e cuja modelagem matemática ideal corresponde a um problema de programação não linear inteira mista (PNLIM) e que, além disso, apresenta o fenômeno da explosão combinatorial que mereceu a atenção de muitos pesquisadores que usaram diversas técnicas para resolver um problema complexo desse tipo. As principais dificuldades na resolução deste problema estão relacionadas com a natureza combinatorial do processo de planejamento que normalmente leva a um número explosivo de alternativas, inclusive no caso de sistemas de médio porte. Além disso, o problema de planejamento apresenta uma estrutura multimodal com um número muito elevado de ótimos locais o que leva a maioria dos métodos aproximados a parar numa solução ótima local às vezes de pobre qualidade. Para contornar estes problemas foram apresentadas muitas alternativas na literatura especializada.

Nesta pesquisa se apresenta uma revisão aproximada do *estado da arte* do problema de planejamento estático da expansão dos sistemas de energia elétrica e uma análise resumida de nossa contribuição nesta linha de pesquisa realizado nos últimos dez anos. Neste tipo de planejamento, dada uma configuração inicial e os dados de geração e demanda do horizonte de planejamento (além de outros dados como limites de operação, custos e restrições de investimento), tenta-se determinar o plano de expansão com um custo mínimo, isto é, deve-se determinar *onde* e *que* tipos de novos equipamentos devem ser instalados. Naturalmente este é um subproblema de um caso mais geral, chamado planejamento dinâmico da expansão onde, adicionalmente, deseja-se saber *quando* instalar estes novos equipamentos. Estes resultados preliminares de planejamento posteriormente serão analisados e melhorados testando seu desempenho através de ferramentas de análise mais exatas como fluxo de carga AC, curto circuito, estabilidade transitória, etc.

Como todo problema de otimização matemática, o problema de planejamento de sistemas de transmissão pode ser separado em duas partes claramente definidas: modelagem matemática e a técnica de solução escolhida para resolver esse modelo matemático. Assim, em todas as pesquisas apresentadas e discutidas através dos próximos capítulos sempre aparece evidente a modelagem matemática e a técnica de solução adotada pelos algoritmos desenvolvidos por diferentes pesquisadores nas últimas décadas.

Na literatura especializada existem três modelos ou formulações matemáticas para realizar



a modelagem do problema de planejamento de sistemas de transmissão: modelo DC, modelo de transportes e modelo híbrido. Tradicionalmente, o modelo DC é considerado como sendo o modelo ideal para representar o problema de planejamento sendo os outros modelos versões relaxadas ou simplificadas do modelo DC. Para resolver estes modelos foram usadas várias técnicas que podem ser agrupadas em três grupos: *métodos heurísticos ou aproximados*, *métodos de otimização clássica* e *métodos de otimização combinatorial*. A modelagem matemática do problema de planejamento de sistemas de transmissão é apresentada no capítulo 2.

Os *métodos aproximados* apresentam a vantagem de fornecer soluções de boa qualidade com esforços computacionais pequenos; mas os mesmos raramente encontram a solução ótima de sistemas reais e não fornecem informação da qualidade da solução obtida, isto é, o quão perto da solução ótima se encontra a solução obtida. Entretanto, estes métodos ainda são os mais usados pelas concessionárias de energia elétrica nos trabalhos de planejamento. No capítulo 3 são apresentados e amplamente discutidos cinco algoritmos que usam métodos aproximados ou heurísticos. Esses algoritmos são os mais usados pelas empresas elétricas no planejamento de sistemas reais.

Os *métodos de otimização clássica*, geralmente usando técnicas de decomposição matemática, apresentam a característica de que encontram a solução ótima do problema de planejamento e são muito eficientes em sistemas de pequeno e médio porte, mas para sistemas de grande porte ainda apresentam problemas de elevado esforço computacional e de convergência. No capítulo 4 são apresentados vários algoritmos que encontram a solução ótima do problema de planejamento quando é usado o modelo de transportes e no capítulo 5 são apresentados vários algoritmos que usam a decomposição de Benders para resolver o problema de planejamento quando é usado o modelo DC.

Os *métodos de otimização combinatorial* como “Simulated Annealing”, Algoritmos Genéticos, Busca Tabu, etc., foram aplicados com muito sucesso para resolver vários problemas do campo da pesquisa operacional e também em alguns problemas de engenharia elétrica. Estes métodos apresentam a característica geral de que convergem em soluções ótimas ou quase ótimas mas com um esforço computacional elevado. Os algoritmos combinatoriais são apresentados nos capítulos 6, 7 e 8.

No capítulo 9 são apresentadas, em forma resumida, nossas contribuições nesta área de pesquisa tanto na modelagem matemática assim como no desenvolvimento de algoritmos de solução eficientes para o problema de planejamento de sistemas de transmissão. Também são apresentadas as conclusões mais importantes e nossas linhas de pesquisa que estão sendo desenvolvidas no momento. No apêndice são mostrados os dados de sistemas típicos usados em testes.

Capítulo 2

Modelagem do Problema de Planejamento de Sistemas de Transmissão

A resolução de todo problema de otimização em engenharia compreende a implementação de dois processos consecutivos: a modelagem matemática e a técnica de solução escolhida para resolver esse modelo matemático. No problema de planejamento de sistemas de transmissão, um problema específico de otimização de sistemas de energia elétrica, também existem estas duas etapas: modelagem e técnica de solução. A modelagem matemática do problema de planejamento de sistemas de transmissão a longo prazo é apresentada neste capítulo e as várias técnicas de solução são apresentadas nos capítulos seguintes.

A modelagem matemática consiste na representação de um problema da vida real através de um modelo matemático que relaciona um conjunto de variáveis de decisão através de um conjunto de relações matemáticas que podem assumir formas e tipos variados. A modelagem matemática pode ser uma representação exata ou simplificada do problema da vida real. Em geral, quanto mais exata a modelagem matemática que representa um problema da vida real, mais complexa é a resolução desse modelo matemático. Assim, deve existir um compromisso entre a modelagem matemática adotada e a técnica de solução escolhida: a modelagem matemática deve representar de maneira adequada o problema da vida real e, além disso, permitir sua resolução através das técnicas disponíveis e com esforços computacionais aceitáveis. Nesse contexto, o conceito de modelagem adequada varia com o tempo porque modelos muito complexos podem se tornar adequados no futuro com o aparecimento de novas técnicas de solução e/ou a fabricação de computadores muito mais velozes do que os disponíveis atualmente. É também evidente que a técnica de solução escolhida encontra uma solução para o modelo matemático e não necessariamente para o problema da vida real.

Em planejamento de sistemas de transmissão, o problema da vida real consiste de um sistema elétrico com uma topologia corrente no qual se deseja encontrar o plano de expansão ótima (construção de novos circuitos) para um horizonte de planejamento definido, isto é, onde e que tipos de circuitos devem ser construídos para que o sistema opere adequadamente num horizonte de planejamento para um crescimento especificado da demanda. Nesse contexto, a modelagem matemática ideal para indicar a operação adequada seria a representação do



problema através das relações matemáticas de fluxo de carga AC. Entretanto, em planejamento de sistemas de transmissão não é usada a modelagem matemática de fluxo de carga AC por vários motivos, dos quais os mais importantes são os seguintes: 1) em geral a topologia inicial do sistema elétrico usado em planejamento é um sistema não conexo, isto é, o sistema apresenta um conjunto de barras isoladas ou ilhadas da parte principal do sistema e, pelo menos no contexto atual, não é possível resolver sistemas deste tipo usando a modelagem matemática de fluxo de carga AC e as técnicas de solução conhecidas para resolver esse tipo de problemas; e 2) o problema de planejamento de sistemas de transmissão resolve somente o fornecimento de potência ativa no sistema elétrico; o problema de fornecimento de reativos (planejamento de reativos) é resolvido posteriormente. Neste último caso, mesmo que o sistema elétrico seja conexo, a convergência do modelo AC seria difícil, ou às vezes impossível, pois o problema frequentemente não teria solução. Além disto, não existem técnicas de solução que resolvam simultaneamente os problemas da expansão dos sistemas de transmissão (construção de linhas e/ou transformadores ou planejamento de sistemas de transmissão) e da alocação de reativos no sistema elétrico.

Levando em conta as observações anteriores, atualmente se considera a modelagem matemática conhecida como fluxo de carga DC como sendo a modelagem ideal para ser usada no problema de planejamento de sistemas de transmissão e essa modelagem matemática conhecida como fluxo de carga DC será chamada simplesmente de *modelo DC*. Os principais motivos para esse consenso são os seguintes: 1) testes experimentais exaustivos mostraram que os resultados obtidos usando o modelo DC apresentam resultados muito próximos aos resultados obtidos usando o fluxo de carga AC em relação à distribuição dos fluxos de potência ativa; e 2) existem várias técnicas de solução (algoritmos) que resolvem de maneira adequada os problemas de planejamento que usam o modelo DC. Entretanto, nas três décadas de pesquisas realizadas em planejamento de sistemas de transmissão foram propostos vários modelos matemáticos para representar o problema de planejamento. Três desses modelos ainda são usados nos trabalhos de planejamento de sistemas de transmissão: modelo de transportes, modelo híbrido e modelo DC. Também ainda são muito usados variantes desses modelos. Neste capítulo são apresentados e discutidos esses três modelos matemáticos e suas principais variantes.

Usando os três modelos matemáticos mencionados anteriormente, o problema resultante é um problema de otimização matemática que envolve relações algébricas lineares e/ou não lineares e com variáveis de decisão inteiras e contínuas. Assim, os problemas resultantes correspondem a um campo da pesquisa operacional conhecida como programação não linear inteira mista e, em princípio, para resolver esses problemas podem ser usadas as técnicas de solução de problemas de programação linear inteira mista (PNLIM).

O modelo DC usa as duas leis de Kirchhoff para realizar a modelagem matemática do problema de planejamento. Os modelos de transportes e híbrido são versões relaxadas (aproximadas) do modelo DC considerado como sendo o modelo ideal. Os motivos e as consequências de usar modelos relaxados são discutidos nos próximos capítulos.



2.1 Modelo de Transportes

O modelo de transportes foi a primeira proposta sistemática de modelagem matemática usado com muito sucesso no problema de planejamento de sistemas de transmissão. O modelo foi proposto por Garver em [2] e representou o início de uma sistemática de pesquisa nos problemas de planejamento de sistemas de transmissão sugerindo o uso de modelos diferentes para os problemas de operação e de planejamento.

Garver sugere que devido aos grandes problemas de usar o modelo de fluxo de carga AC usado para operação, deve-se usar modelos mais relaxados que permitam encontrar topologias ou configurações atrativas do crescimento do sistema elétrico mesmo que estas propostas sejam aproximadas. Assim, sugere a utilização de um modelo matemático que deve satisfazer somente a primeira lei de Kirchhoff, isto é, a modelagem matemática proposta não leva em conta a segunda lei de Kirchhoff. Este modelo matemático é conhecido como modelo de transportes. Obviamente, esta modelagem matemática é uma representação menos adequada do problema real que, por exemplo, o modelo DC e, portanto, a solução encontrada pelo modelo de transportes pode ser menos adequada para o problema real.

No modelo de transportes se deseja encontrar uma configuração que produz o menor investimento no plano de expansão do sistema elétrico e adequadas condições de operação desse sistema elétrico. Adequadas condições de operação significa que o sistema deve satisfazer a primeira lei de Kirchhoff e que os circuitos e as usinas de geração operem dentro de seus limites especificados. A primeira lei de Kirchhoff simplesmente especifica que a somatória dos fluxos de potência que entram numa barra do sistema deve ser igual à somatória do fluxo de potência que saem dessa barra do sistema.

2.1.1 Modelo Matemático

Usando o modelo de transportes, a modelagem matemática do problema de planejamento de sistemas de transmissão assume a seguinte forma:

$$\begin{aligned} \min v &= \sum_{(i,j)} c_{ij} n_{ij} \\ \text{s.a.} \quad & S f + g = d \\ & |f_{ij}| \leq (n_{ij} + n_{ij}^o) \bar{f}_{ij} \\ & 0 \leq g \leq \bar{g} \\ & 0 \leq n_{ij} \leq \bar{n}_{ij} \\ & n_{ij} \text{ inteiro} \\ & f_{ij} \text{ irrestrito} \end{aligned} \tag{2.1}$$

em que v é o investimento devido às adições de circuitos no sistema, c_{ij} é o custo de um circuito no caminho $i - j$, n_{ij} é o número de circuitos adicionados no caminho $i - j$, S é a matriz de

incidência nó-ramo do sistema elétrico, f é o vetor de fluxos cujos elementos f_{ij} representam o fluxo total no caminho $i - j$, g é o vetor de geração cujos elementos g_i representam o nível de geração na barra de geração i , d é o vetor de demanda cujos elementos d_i representam a demanda na barra de carga i , n_{ij}^o representa o número de circuitos na configuração base no caminho $i - j$, $\bar{f}_{ij}$ é o fluxo máximo permitido para um circuito no caminho $i - j$, $\bar{g}$ é o vetor de máxima capacidade de geração nas barras de geração e $\bar{n}_{ij}$ é o vetor do número máximo de adições permitida no caminho $i - j$. As variáveis de decisão e a estrutura matemática do modelo de transportes apresentado originalmente por Garver é diferente de (2.1) mas as duas formulações são conceitualmente equivalentes.

No modelo de transportes, o conjunto de restrições $S f + g = d$ representa as equações correspondentes à primeira lei de Kirchhoff, uma equação para cada barra do sistema; as restrições $|f_{ij}| \leq (n_{ij} + n_{ij}^o) \bar{f}_{ij}$ representam as restrições de capacidade de transmissão dos circuitos (linhas e/ou transformadores) e o valor absoluto é necessário pois os fluxos de potência podem fluir em ambos sentidos. As outras restrições são triviais e representam apenas restrições de limite de geração e de circuitos adicionados em cada caminho candidato $i - j$. Finalmente, as variáveis f_{ij} são irrestritas em valor e as variáveis n_{ij} devem ser inteiras representando a maior fonte de complexidade no problema.

Do ponto de vista de pesquisa operacional o sistema (2.1), o modelo de transportes, é um problema de programação linear inteiro misto (PLIM). A resolução do sistema (2.1), isto é, encontrar a solução ótima desse problema não é simples, especialmente para sistemas elétricos de grande porte. Entretanto, se fossem permitidas adições fracionárias de circuitos (linhas de transmissão e/ou transformadores), isto é, se é permitido que os n_{ij} assumam valores contínuos, então o sistema (2.1) se transforma num simples problema de programação linear (PL) mesmo para o caso de sistemas de grande porte. Assim, é evidente que a restrição n_{ij} inteiro produz a maior complexidade no sistema (2.1). Estas características serão aproveitadas para desenvolver vários tipos de algoritmos para resolver o problema de planejamento de sistemas de transmissão quando é usado o modelo de transportes.

A grande vantagem do modelo de transportes é que praticamente não existe diferença entre resolver problemas de sistemas conexos e altamente ilhados. Esta característica decorre diretamente do fato de que a solução do modelo de transportes não depende das reatâncias dos circuitos. Deve-se observar que na formulação matemática, o sistema (2.1), não aparecem as reatâncias (ou susceptâncias) dos circuitos. A desvantagem principal é que a solução apresentada pelo modelo de transportes pode ser distante da solução correspondente ao modelo DC, considerado como sendo modelo ideal. Em outras palavras, a solução ótima do modelo de transportes às vezes pode ficar muito afastado da solução ótima do modelo DC pois a solução do modelo de transportes não necessariamente satisfaz a segunda lei de Kirchhoff.

2.1.2 Exemplo 2.1: Sistema de 3 Barras

A figura 2.1 é usada para apresentar o modelo de transportes de um sistema de 3 barras e 3 linhas de transmissão. Os dados deste sistema, que também são mostradas na figura 2.1, são os seguintes:

Custo das linhas: $c_{12} = 3$, $c_{13} = 2$ e $c_{23} = 2$ US\$

Susceptâncias: $\gamma_{12} = \frac{1}{3}$, $\gamma_{13} = \frac{1}{2}$ e $\gamma_{23} = \frac{1}{2}$.

Geração máxima e carga: $\bar{g} = 80$, $d_2 = 60$ e $d_3 = 20$ MW.

Fluxo máximo por linha: $\bar{f}_{12} = 35$, $\bar{f}_{13} = 40$ e $\bar{f}_{23} = 40$ MW.

Número máximo de adições permitida por caminho: $\bar{n}_{12} = \bar{n}_{13} = \bar{n}_{23} = 2$.

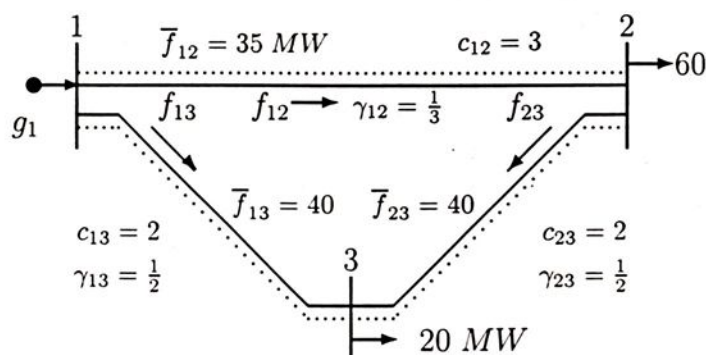


Figura 2.1: Sistema inicial de 3 barras.

Em todos os exemplos, deve-se escolher um sentido para os fluxos f_{ij} . Embora esta escolha seja arbitrária, em todos os exemplos apresentados os sentidos dos fluxos em cada caminho é escolhido no sentido da barra de numeração menor para a barra de numeração maior como é mostrado na figura 2.1. Obviamente, a resolução do problema deve mostrar o sentido correto dos fluxos para cada exemplo. As equações correspondentes à primeira lei de Kirchhoff aplicado para cada barra, considerando positivo o fluxo que entra na barra, produz as seguintes equações:

$$\text{Barra 1:} \quad -f_{12} - f_{13} + g_1 = 0$$

$$\text{Barra 2:} \quad f_{12} - f_{23} - 60 = 0$$

$$\text{Barra 3:} \quad f_{13} + f_{23} - 20 = 0$$

As inequações correspondentes aos limites de capacidade de transmissão produzem as seguintes relações:

$$\text{Linha 1-2:} \quad |f_{12}| \leq 35(1 + n_{12})$$

$$\text{Linha 1-3:} \quad |f_{13}| \leq 40(1 + n_{13})$$

$$\text{Linha 2-3:} \quad |f_{23}| \leq 40(1 + n_{23})$$

A restrição sobre capacidade de geração produz a seguinte restrição:

$$\text{Barra de geração 1:} \quad 0 \leq g_1 \leq 80$$

As restrições sobre o número máximo de adições permitidas em cada caminho candidato a adição produzem as seguintes restrições:

$$\text{Caminho 1-2:} \quad 0 \leq n_{12} \leq 2$$

Caminho 1-3: $0 \leq n_{13} \leq 2$

Caminho 2-3: $0 \leq n_{23} \leq 2$

A função objetivo assume a seguinte forma: $\min v = 3n_{12} + 2n_{13} + 2n_{23}$

Assim, para o exemplo, a modelagem matemática assume a seguinte forma:

$$\min v = 3n_{12} + 2n_{13} + 2n_{23} \quad (2.2)$$

s.a.

$$\begin{aligned} -f_{12} - f_{13} + g_1 &= 0 \\ f_{12} - f_{23} &= 60 \\ f_{13} + f_{23} &= 20 \\ |f_{12}| &\leq 35(1 + n_{12}) \\ |f_{13}| &\leq 40(1 + n_{13}) \\ |f_{23}| &\leq 40(1 + n_{23}) \\ 0 &\leq g_1 \leq 80 \\ n_{12} &\in \{0, 1, 2\} \\ n_{13} &\in \{0, 1, 2\} \\ n_{23} &\in \{0, 1, 2\} \\ f_{12}, f_{13}, \text{ e } f_{23} &\text{ irrestritos} \end{aligned}$$

As incógnitas no problema são: $v, n_{12}, n_{13}, n_{23}, f_{12}, f_{13}, f_{23}$ e g_1 .

As equações correspondentes à primeira lei de Kirchhoff podem ser representadas na seguinte forma matricial:

$$\begin{array}{|c|c|c|} \hline -1 & -1 & \\ \hline 1 & & -1 \\ \hline & 1 & 1 \\ \hline \end{array} \begin{bmatrix} f_{12} \\ f_{13} \\ f_{23} \end{bmatrix} + \begin{array}{|c|} \hline g_1 \\ \hline \\ \hline \end{array} = \begin{array}{|c|} \hline \\ \hline 60 \\ \hline 20 \\ \hline \end{array}$$

No sistema anterior, pode-se verificar a matriz S conhecida como matriz de incidência nó-ramo do grafo do sistema elétrico:

	1	2	3	
1	-1	-1		n_b
2	1		-1	
3		1	1	
	n_l			

Na matriz S existe uma coluna para cada ramo candidato e uma linha para cada barra do sistema, assim a dimensão de S é $n_b \times n_l$. Os elementos de S podem ser obtidos facilmente usando a seguinte regra: cada coluna de S (coluna de um ramo) possui unicamente dois elementos diferentes de zero, vale 1 se o fluxo no ramo analisado está chegando na barra e -1 se o fluxo está saindo da barra.

Levando em conta ainda as seguintes notações vetoriais:

$$c = \begin{bmatrix} 3 & 2 & 2 \end{bmatrix} \quad n = \begin{bmatrix} n_{12} \\ n_{13} \\ n_{23} \end{bmatrix} \quad n^o = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad f = \begin{bmatrix} f_{12} \\ f_{13} \\ f_{23} \end{bmatrix} \quad g = \begin{bmatrix} g_1 \\ 0 \\ 0 \end{bmatrix} \quad d = \begin{bmatrix} 0 \\ 60 \\ 20 \end{bmatrix}$$

o sistema (2.2) assume uma forma equivalente ao sistema (2.1).

2.1.3 Exemplo 2.2: Caso Especial - Sistema de 3 Barras Completamente Ilhado

Na figura 2.2, apresenta-se o sistema de 03 barras completamente ilhado no qual não existem linhas na configuração base.

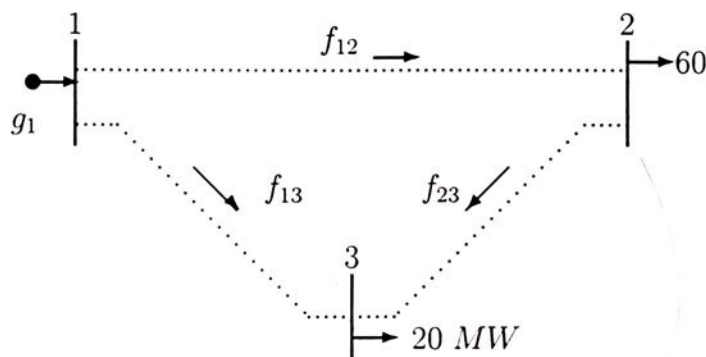


Figura 2.2: Sistema de 3 barras ilhado.

Neste caso se pretende realizar o planejamento de um sistema que não existe, isto é, não existem linhas na configuração base e, portanto, o sistema está completamente ilhado. Entretanto, na modelagem matemática apenas variam as restrições correspondentes aos limites de

fluxos nas linhas pois agora $n_{12}^o = n_{13}^o = n_{23}^o = 0$. Assim, a modelagem matemática assume a seguinte forma:

$$\begin{aligned}
 \min v &= 3n_{12} + 2n_{13} + 2n_{23} & (2.3) \\
 \text{s.a.} & \\
 & -f_{12} - f_{13} + g_1 = 0 \\
 & f_{12} - f_{23} = 60 \\
 & f_{13} + f_{23} = 20 \\
 & |f_{12}| \leq 35n_{12} \\
 & |f_{13}| \leq 40n_{13} \\
 & |f_{23}| \leq 40n_{23} \\
 & 0 \leq g_1 \leq 80 \\
 & n_{12} \in \{0, 1, 2\} \\
 & n_{13} \in \{0, 1, 2\} \\
 & n_{23} \in \{0, 1, 2\} \\
 & f_{12}, f_{13}, \text{ e } f_{23} \text{ irrestritos}
 \end{aligned}$$

As incógnitas no problema são: v , n_{12} , n_{13} , n_{23} , f_{12} , f_{13} , f_{23} e g_1 .

2.2 Modelo Híbrido

O modelo híbrido foi proposto originalmente em [7] num contexto e numa estrutura diferente ao apresentado aqui. Na formulação mais pura, a modelagem matemática do modelo híbrido especifica o seguinte: a parcela do sistema elétrico correspondente aos caminhos onde já existem circuitos na configuração base devem satisfazer as duas leis de Kirchhoff e a outra parcela correspondente aos caminhos novos em que não existem circuitos na configuração base devem satisfazer unicamente a primeira lei de Kirchhoff. Portanto, o modelo híbrido é uma mistura entre o modelo de transportes e o modelo DC. Obviamente, uma vez definida a modelagem matemática desta maneira, a solução ótima encontrada também deve satisfazer as duas leis de Kirchhoff na parte do sistema onde existiam circuitos na configuração base e somente a primeira lei de Kirchhoff na parte do sistema onde não existiam circuitos na configuração base. Em outras palavras, no modelo híbrido, deve-se satisfazer a primeira lei de Kirchhoff em todas as barras do sistema e a segunda lei de Kirchhoff somente naqueles laços que já existem na configuração base. Assim, por exemplo, se no processo de planejamento for adicionado um circuito num caminho novo, então os laços que eventualmente podem aparecer como consequência da adição deste circuito não estão obrigados a satisfazer a segunda lei de Kirchhoff. Aqui reside a diferença principal entre esta formulação do modelo híbrido e a proposta apresentada em [7] onde a modelagem híbrida é usada simplesmente como uma forma de auxílio para o indicador de sensibilidade do algoritmo heurístico proposto. Entretanto, em [7] apareceu a primeira proposta de modelagem híbrida para ser usada no problema de planejamento de sistemas de transmissão.

A idéia de usar o modelo híbrido no problema de planejamento de sistemas de transmissão é

para contornar alguns problemas que apresentavam os modelos de transportes e DC. O modelo de transportes apresenta uma excelente flexibilidade para trabalhar com redes não conexas mas as soluções encontradas podem ficar muito afastadas da solução ótima do modelo DC. Por outro lado, o modelo DC pode apresentar sérios problemas para trabalhar com redes não conexas. Assim, o modelo híbrido permite encontrar soluções mais próximas da solução ótima do modelo DC e com a vantagem de trabalhar eficientemente na parcela correspondente às partes não conexas do sistema. Entretanto, encontrar a solução ótima do problema de planejamento de sistemas de transmissão usando o modelo híbrido é de uma complexidade quase equivalente do modelo DC. Este fato explica, em parte, a falta de pesquisas publicadas usando o modelo híbrido de maneira independente. Assim, a versão do modelo híbrido que está sendo apresentada e as diferentes variantes que aparecem na literatura especializada são usadas apenas para auxiliar no processo de resolução do modelo DC em algoritmos de planejamento de sistemas de transmissão [7, 14].

2.2.1 Modelo Matemático

A formulação matemática do modelo híbrido assume a seguinte forma:

$$\begin{aligned}
 \min v &= \sum_{(i,j)} c_{ij} n_{ij} & (2.4) \\
 \text{s.a.} & \\
 & S f + g = d \\
 & f_{ij} - (\gamma_{ij}^o + x_{ij})(\theta_i - \theta_j) = 0 \quad \forall (i,j) \in \Omega_1 \\
 & |f_{ij}| \leq (n_{ij} + n_{ij}^o) \bar{f}_{ij} \\
 & 0 \leq g \leq \bar{g} \\
 & 0 \leq n_{ij} \leq \bar{n}_{ij} \\
 & n_{ij} \text{ inteiro} \\
 & x_{ij} \text{ discreto} \\
 & f_{ij} \text{ irrestrito} \\
 & \theta_j \text{ irrestrito } \forall j \in \Omega_3
 \end{aligned}$$

em que γ_{ij}^o é a susceptância equivalente dos circuitos existentes no caminho $i-j$ na configuração base, x_{ij} é a susceptância equivalente dos circuitos adicionados no caminho $i-j$ no processo de otimização, Ω_1 representa o conjunto de linhas existentes na configuração base, Ω_2 representa o conjunto de linhas correspondentes aos novos caminhos ($\Omega = \Omega_1 \cup \Omega_2$), Ω_3 representa o conjunto de barras que fazem parte da configuração base, isto é, as barras que não estão ilhadas, e θ_j é a magnitude do ângulo da tensão das barras existentes na configuração base, isto é, para as barras que pertencem a Ω_3 .

No sistema (2.4), o conjunto de equações $S f + g = d$ representam as equações da primeira lei de Kirchhoff, uma equação por barra, para todas as barras do sistema e o conjunto de equações $f_{ij} - (\gamma_{ij}^o + x_{ij})(\theta_i - \theta_j) = 0$ representam as equações correspondentes à segunda lei de Kirchhoff com uma equação para cada caminho que apresenta pelo menos um circuito

na configuração base. Este último conjunto de equações representa a diferença entre os três modelos matemáticos que estão sendo apresentados. No modelo de transportes, o conjunto de equações $f_{ij} - (\gamma_{ij}^o + x_{ij})(\theta_i - \theta_j) = 0$ simplesmente não aparecem, já no modelo híbrido aparecem somente uma parcela dessas equações constituídas pelos caminhos em que existem circuitos na configuração base e, finalmente, no modelo DC aparecem todas as equações desse tipo, uma por cada caminho existente e/ou novos caminhos candidatos a adição de circuitos.

A presença de todas as equações correspondentes à segunda lei de Kirchhoff no modelo DC transforma este modelo num problema mais restrito que os outros modelos. Assim, do ponto de vista de otimização matemática, pode-se dizer que o modelo híbrido é um problema relaxado em relação ao modelo DC, isto é, um conjunto de restrições presentes no modelo DC foram eliminados no modelo híbrido. De igual maneira, o modelo de transportes é um problema relaxado em relação aos modelos híbrido e DC pois no modelo de transportes foram eliminadas todas as restrições correspondentes à segunda lei de Kirchhoff. Uma consequência natural desse processo de relaxação é que uma solução ótima para o modelo de transportes não necessariamente é uma solução ótima para os modelos híbrido e DC e, frequentemente, uma solução ótima do modelo de transportes pode ser infactível em relação aos outros modelos. A mesma observação é verdadeira com a solução ótima do modelo híbrido em relação ao modelo DC. É um assunto aberto a análise teórica e experimental sobre quão boa pode ser as soluções encontradas pelos modelos relaxados. A vantagem evidente dos modelos relaxados é que podem ser mais facilmente resolvidas. Uma observação importante é verificar que as equações da segunda lei de Kirchhoff, $f_{ij} - (\gamma_{ij}^o + x_{ij})(\theta_i - \theta_j) = 0$, são equações ou restrições não lineares transformando o problema em não linear e produzindo um nível de complexidade maior em relação ao modelo de transportes.

Após algumas manipulações algébricas, pode-se chegar a uma formulação equivalente que assume a seguinte forma:

$$\begin{aligned} \min v &= \sum_{(i,j)} c_{ij} n_{ij} & (2.5) \\ \text{s.a.} & \\ & S_2 f_2 + B_1 \theta_1 + g = d \\ & |\theta_i - \theta_j| \leq \bar{\phi}_{ij} \quad \forall (i,j) \in \Omega_1 \\ & |f_{ij}| \leq \bar{f}_{ij} n_{ij} \quad \forall (i,j) \in \Omega_2 \\ & 0 \leq g \leq \bar{g} \\ & 0 \leq n_{ij} \leq \bar{n}_{ij} \\ & n_{ij} \text{ inteiro} \\ & x_{ij} \text{ discreto} \\ & f_{ij} \text{ irrestrito } \forall (i,j) \in \Omega_2 \\ & \theta_j \text{ irrestrito } \forall j \in \Omega_3 \end{aligned}$$

em que S_2 é a matriz de incidência nó-ramo da parcela dos circuitos que não tem linhas na configuração base, (linhas que pertencem a Ω_2), f_2 é o vetor de fluxos das linhas que pertencem a Ω_2 , B_1 é a matriz de susceptâncias da parcela dos circuitos que apresentam linhas adicionadas na configuração base (linhas que pertencem a Ω_1) e θ_1 é o vetor de ângulos $\theta_j \in \Omega_3$, isto é, de

barras conectadas à configuração base. Assim, S_2 é de dimensão $nb \times nc$ em que nb é o número de barras e nc o número de novos caminhos e as linhas de S_2 correspondentes às barras para as quais não chega um caminho novo tem todos seus elementos iguais a zero; f_2 é um vetor de dimensão nc ; B_1 é de dimensão $nb \times nb$ em que as filas e colunas de B_1 correspondentes a barras ilhadas tem todos seus elementos iguais a zero e θ_1 é um vetor de dimensão nb em que as posições correspondentes às barras ilhadas tem valores iguais a zero.

2.2.2 Exemplo 2.3: Equivalência entre as duas formulações matemáticas

Para mostrar que ambas formulações matemáticas são equivalentes usamos o sistema de 3 barras mostrado na figura 2.3 com as incógnitas do problema. Os dados das linhas e de geração e carga são as mesmas da figura 2.1.

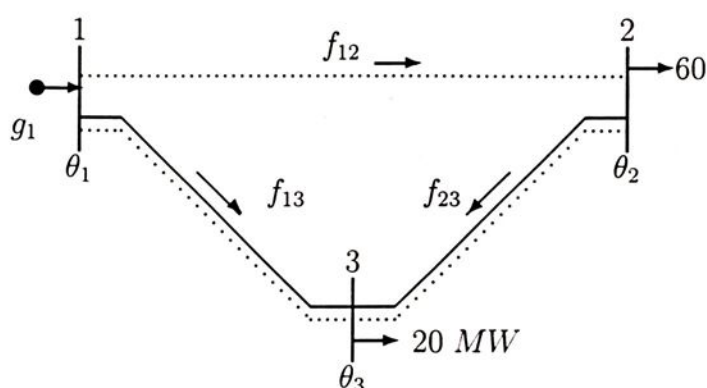


Figura 2.4: Formulação do Modelo Híbrido.

Usando a primeira formulação, as principais relações matemáticas assumem a seguinte forma:

1. Equações da primeira lei de Kirchhoff:

Barra 1: $-f_{12} - f_{13} + g_1 = 0$

Barra 2: $f_{12} - f_{23} = 60$

Barra 3: $f_{13} + f_{23} = 20$

2. Equações da segunda lei de Kirchhoff:

Linha 1-3: $f_{13} = (\frac{1}{2} + x_{13})(\theta_1 - \theta_3)$

Linha 2-3: $f_{23} = (\frac{1}{2} + x_{23})(\theta_2 - \theta_3)$

3. Inequações correspondentes aos limites de fluxos:

Linha 1-2: $|f_{12}| \leq 35n_{12}$

Linha 1-3: $|f_{13}| \leq 40(1 + n_{13})$

Linha 2-3: $|f_{23}| \leq 40(1 + n_{23})$

As outras restrições são triviais e assim, para o exemplo e usando a primeira formulação, a modelagem matemática do modelo híbrido assume a seguinte forma:

$$\begin{aligned}
 \min v &= 3n_{12} + 2n_{13} + 2n_{23} & (2.6) \\
 \text{s.a.} & \\
 & -f_{12} - f_{13} + g_1 = 0 \\
 & f_{12} - f_{23} = 60 \\
 & f_{13} + f_{23} = 20 \\
 & f_{13} = \left(\frac{1}{2} + x_{13}\right)(\theta_1 - \theta_3) \\
 & f_{23} = \left(\frac{1}{2} + x_{23}\right)(\theta_2 - \theta_3) \\
 & |f_{12}| \leq 35n_{12} \\
 & |f_{13}| \leq 40(1 + n_{13}) \\
 & |f_{23}| \leq 40(1 + n_{23}) \\
 & 0 \leq g_1 \leq 80 \\
 & n_{12}, n_{13} \text{ e } n_{23} \in \{0, 1, 2\} \\
 & x_{12} \in \left\{0, \frac{1}{3}, \frac{2}{3}\right\} \\
 & x_{13} \text{ e } x_{23} \in \left\{0, \frac{1}{2}, 1\right\} \\
 & n_{12} = 3x_{12} \\
 & n_{13} = 2x_{13} \\
 & n_{23} = 2x_{23} \\
 & f_{12}, f_{13}, \text{ e } f_{23} \text{ irrestritos} \\
 & \theta_1, \theta_2, \text{ e } \theta_3 \text{ irrestritos}
 \end{aligned}$$

As incógnitas no problema são: v , n_{12} , n_{13} , n_{23} , x_{12} , x_{13} , x_{23} , f_{12} , f_{13} , f_{23} , θ_1 , θ_2 , θ_3 , e g_1 . Obviamente, da formulação matemática se pode eliminar as incógnitas x_{ij} (ou n_{ij}) usando a relação $x_{ij} = \gamma_{ij}n_{ij}$.

Em (2.6), pode-se substituir f_{13} e f_{23} das equações da segunda lei de Kirchhoff nas equações da primeira lei de Kirchhoff e nas restrições de limites de fluxos para obter as seguintes relações:

1. Substituindo nas equações da primeira lei de Kirchhoff:

$$\begin{aligned}
 -f_{12} - \left(\frac{1}{2} + x_{13}\right)(\theta_1 - \theta_3) + g_1 &= 0 \\
 +f_{12} - \left(\frac{1}{2} + x_{23}\right)(\theta_2 - \theta_3) &= 60 \\
 \left(\frac{1}{2} + x_{13}\right)(\theta_1 - \theta_3) + \left(\frac{1}{2} + x_{23}\right)(\theta_2 - \theta_3) &= 20
 \end{aligned}$$

ou na forma matricial:

-1
1
0

 $\uparrow$
 S_2

f_{12}

 $+$

$-(\frac{1}{2} + x_{13})$	0	$(\frac{1}{2} + x_{13})$	θ_1
0	$-(\frac{1}{2} + x_{23})$	$(\frac{1}{2} + x_{23})$	θ_2
$(\frac{1}{2} + x_{13})$	$(\frac{1}{2} + x_{23})$	$-(1 + x_{13} + x_{23})$	θ_3

 $\uparrow$
 B_1
 $+$

g_1
0
0

 $=$

0
60
20

2. Substituindo nas restrições de limite de fluxo:

$$|(\frac{1}{2} + x_{13})(\theta_1 - \theta_3)| = (\frac{1}{2} + x_{13})|(\theta_1 - \theta_3)| \leq 80(\frac{1}{2} + x_{13}) \implies |(\theta_1 - \theta_3)| \leq 80$$

$$|(\frac{1}{2} + x_{23})(\theta_2 - \theta_3)| = (\frac{1}{2} + x_{23})|(\theta_2 - \theta_3)| \leq 80(\frac{1}{2} + x_{23}) \implies |(\theta_2 - \theta_3)| \leq 80$$

Substituindo as relações anteriores no sistema (2.6), pode-se obter o sistema equivalente seguinte:

$$\min v = 3n_{12} + 2n_{13} + 2n_{23} \quad (2.7)$$

s.a.

$$-f_{12} - (\frac{1}{2} + x_{13})\theta_1 + (\frac{1}{2} + x_{13})\theta_3 + g_1 = 0$$

$$f_{12} - (\frac{1}{2} + x_{23})\theta_2 + (\frac{1}{2} + x_{23})\theta_3 = 60$$

$$(\frac{1}{2} + x_{13})\theta_1 + (\frac{1}{2} + x_{23})\theta_2 - (1 + x_{13} + x_{23})\theta_3 = 20$$

$$|f_{12}| \leq 35n_{12}$$

$$|\theta_1 - \theta_3| \leq 80$$

$$|\theta_2 - \theta_3| \leq 80$$

$$0 \leq g_1 \leq 80$$

$$n_{12}, n_{13} \text{ e } n_{23} \in \{0, 1, 2\}$$

$$x_{12} \in \{0, \frac{1}{3}, \frac{2}{3}\}$$

$$x_{13} \text{ e } x_{23} \in \{0, \frac{1}{2}, 1\}$$

$$n_{12} = 3x_{12}$$

$$n_{13} = 2x_{13}$$

$$n_{23} = 2x_{23}$$

$$f_{12} \text{ irrestrito}$$

$$\theta_1, \theta_2 \text{ e } \theta_3 \text{ irrestritos}$$

que também pode ser obtido diretamente da segunda formulação do modelo híbrido verificando a equivalência de ambas formulações.

2.3 Modelo DC

O modelo DC é uma generalização do modelo de fluxo de carga DC que está amplamente desenvolvido em [28]. Este modelo é considerado como sendo ideal para realizar os trabalhos de planejamento de sistemas de transmissão. No modelo DC o sistema elétrico completo deve satisfazer as duas leis de Kirchhoff, isto é, todas as barras do sistema devem satisfazer a primeira lei de Kirchhoff e todos os laços existentes devem satisfazer a segunda lei de Kirchhoff. Testes experimentais exaustivos mostraram que as topologias encontradas usando o modelo DC apresentam bom desempenho quando avaliadas com os modelos tradicionais de operação de sistemas de energia elétrica como o fluxo de carga AC.

2.3.1 Modelo Matemático

A formulação matemática do modelo DC assume a seguinte forma:

$$\begin{aligned}
 \min v &= \sum_{(i,j)} c_{ij} n_{ij} & (2.8) \\
 \text{s.a.} & \\
 & S f + g = d \\
 & f_{ij} - (\gamma_{ij}^o + x_{ij})(\theta_i - \theta_j) = 0 \\
 & |f_{ij}| \leq (x_{ij} + \gamma_{ij}^o) \bar{\phi}_{ij} \\
 & 0 \leq g \leq \bar{g} \\
 & 0 \leq n_{ij} \leq \bar{n}_{ij} \\
 & n_{ij} \text{ inteiro} \\
 & x_{ij} \text{ discreto} \\
 & f_{ij} \text{ irrestrito} \\
 & \theta_j \text{ irrestrito}
 \end{aligned}$$

em que $\Omega \gamma_{ij}^o$ representa a susceptância equivalente dos circuitos existentes no caminho $i - j$ na configuração base, γ_{ij} representa a susceptância de um circuito no caminho $i - j$, x_{ij} representa a susceptância equivalente dos circuitos adicionados no caminho $i - j$ e as seguintes relações são válidas:

$$n_{ij} = \frac{x_{ij}}{\gamma_{ij}} \quad \bar{\phi}_{ij} = \frac{\bar{f}_{ij}}{\gamma_{ij}}$$

e, obviamente, usando as relações anteriores podem ser eliminados as incógnitas x_{ij} ou n_{ij} da formulação do modelo DC.

Após algumas manipulações algébricas, pode-se chegar a uma formulação equivalente que assume a seguinte forma:



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

s.a.

$$B\theta + g = d$$

$$|\theta_i - \theta_j| \leq \bar{\phi}_{ij} \quad \forall (i, j) \in \Omega_1$$

$$x_{ij} |\theta_i - \theta_j| \leq x_{ij} \bar{\phi}_{ij} \quad \forall (i, j) \in \Omega_2$$

$$0 \leq g \leq \bar{g}$$

$$0 \leq n_{ij} \leq \bar{n}_{ij}$$

$$n_{ij} \text{ inteiro}$$

$$x_{ij} \text{ discreto}$$

$$\theta_j \text{ irrestrito}$$

em que B é a matriz de susceptâncias do sistema elétrico e θ é o vetor de ângulos de tensão de barras do sistema elétrico.

2.3.2 Exemplo 2.4: Equivalência entre as duas formulações matemáticas

Para mostrar que ambas formulações matemáticas do modelo DC são equivalentes usamos o sistema de 3 barras do exemplo 2.1.

Usando a primeira formulação, as principais relações matemáticas assumem a seguinte forma:

1. Equações da primeira lei de Kirchhoff:

$$\text{Barra 1:} \quad -f_{12} - f_{13} + g_1 = 0$$

$$\text{Barra 2:} \quad f_{12} - f_{23} = 60$$

$$\text{Barra 3:} \quad f_{13} + f_{23} = 20$$

2. Equações da segunda lei de Kirchhoff:

$$\text{Linha 1-2:} \quad f_{12} = \left(\frac{1}{3} + x_{12}\right)(\theta_1 - \theta_2)$$

$$\text{Linha 1-3:} \quad f_{13} = \left(\frac{1}{2} + x_{13}\right)(\theta_1 - \theta_3)$$

$$\text{Linha 2-3:} \quad f_{23} = \left(\frac{1}{2} + x_{23}\right)(\theta_2 - \theta_3)$$

3. Inequações correspondentes aos limites de fluxos:

$$\text{Linha 1-2:} \quad |f_{12}| \leq 105\left(\frac{1}{3} + x_{12}\right)$$

$$\text{Linha 1-3:} \quad |f_{13}| \leq 80\left(\frac{1}{2} + x_{13}\right)$$

$$\text{Linha 2-3:} \quad |f_{23}| \leq 80\left(\frac{1}{2} + x_{23}\right)$$

pois:

$$\bar{\phi}_{12} = \frac{\bar{f}_{12}}{\gamma_{12}} = \frac{35}{1/3} = 105 \quad \bar{\phi}_{13} = \frac{\bar{f}_{13}}{\gamma_{13}} = \frac{40}{1/2} = 80 \quad \bar{\phi}_{23} = \frac{\bar{f}_{23}}{\gamma_{23}} = \frac{40}{1/2} = 80$$

As outras restrições são triviais e assim, para o exemplo e usando a primeira formulação, a modelagem matemática do modelo DC assume a seguinte forma:

$$\min v = 3n_{12} + 2n_{13} + 2n_{23} \quad (2.10)$$

s.a.

$$-f_{12} - f_{13} + g_1 = 0$$

$$f_{12} - f_{23} = 60$$

$$f_{13} + f_{23} = 20$$

$$f_{13} = \left(\frac{1}{2} + x_{13}\right)(\theta_1 - \theta_3)$$

$$f_{23} = \left(\frac{1}{2} + x_{23}\right)(\theta_2 - \theta_3)$$

$$|f_{12}| \leq 105\left(\frac{1}{3} + x_{12}\right)$$

$$|f_{13}| \leq 80\left(\frac{1}{2} + x_{13}\right)$$

$$|f_{23}| \leq 80\left(\frac{1}{2} + x_{23}\right)$$

$$0 \leq g_1 \leq 80$$

$$n_{12}, n_{13} \text{ e } n_{23} \in \{0, 1, 2\}$$

$$x_{12} \in \left\{0, \frac{1}{3}, \frac{2}{3}\right\}$$

$$x_{13} \text{ e } x_{23} \in \left\{0, \frac{1}{2}, 1\right\}$$

$$n_{12} = 3x_{12}$$

$$n_{13} = 2x_{13}$$

$$n_{23} = 2x_{23}$$

$$f_{12}, f_{13}, \text{ e } f_{23} \text{ irrestritos}$$

$$\theta_1, \theta_2, \text{ e } \theta_3 \text{ irrestritos}$$

As incógnitas no problema são: $v, n_{12}, n_{13}, n_{23}, x_{12}, x_{13}, x_{23}, f_{12}, f_{13}, f_{23}, \theta_1, \theta_2, \theta_3$, e g_1 . Obviamente, da formulação matemática, pode-se eliminar as incógnitas x_{ij} (ou n_{ij}) usando a relação $x_{ij} = \gamma_{ij}n_{ij}$.

A partir do sistema anterior, pode-se chegar à segunda formulação do modelo DC realizando as seguintes transformações:

1. Substituindo as relações da segunda lei de Kirchhoff nas equações da primeira lei de Kirchhoff:

- Primeira equação da primeira lei de Kirchhoff:

$$-\left(\frac{1}{3} + x_{12}\right)(\theta_1 - \theta_2) - \left(\frac{1}{2} + x_{13}\right)(\theta_1 - \theta_3) + g_1 = 0$$

$$\left[-\left(\frac{1}{3} + x_{12}\right) - \left(\frac{1}{2} + x_{13}\right)\right]\theta_1 + \left[\left(\frac{1}{3} + x_{12}\right)\right]\theta_2 + \left[\frac{1}{2} + x_{13}\right]\theta_3 + g_1 = 0$$

- Segunda equação da primeira lei de Kirchhoff:

$$\left(\frac{1}{3} + x_{12}\right)(\theta_1 - \theta_2) - \left(\frac{1}{2} + x_{23}\right)(\theta_2 - \theta_3) = 60$$

$$\left[\left(\frac{1}{3} + x_{12}\right)\right]\theta_1 - \left[\left(\frac{1}{3} + x_{12}\right) + \left(\frac{1}{2} + x_{23}\right)\right]\theta_2 + \left[\frac{1}{2} + x_{23}\right]\theta_3 = 60$$

- Terceira equação da primeira lei de Kirchhoff:

$$\left(\frac{1}{2} + x_{13}\right)(\theta_1 - \theta_3) + \left(\frac{1}{2} + x_{23}\right)(\theta_2 - \theta_3) = 20$$

$$\left[\frac{1}{2} + x_{13}\right]\theta_1 + \left[\left(\frac{1}{2} + x_{23}\right)\right]\theta_2 - \left[\left(\frac{1}{2} + x_{13}\right) + \left(\frac{1}{2} + x_{23}\right)\right]\theta_3 = 20$$

2. Substituindo as relações da segunda lei de Kirchhoff nas inequações de limites de fluxos:

- Primeira inequação de limites de fluxo:

$$\left|\left(\frac{1}{3} + x_{12}\right)(\theta_1 - \theta_2)\right| \leq 105\left(\frac{1}{3} + x_{12}\right) \implies |(\theta_1 - \theta_2)| \leq 105$$

- Segunda inequação de limites de fluxo:

$$\left|\left(\frac{1}{2} + x_{13}\right)(\theta_1 - \theta_3)\right| \leq 80\left(\frac{1}{2} + x_{13}\right) \implies |(\theta_1 - \theta_3)| \leq 80$$

- Terceira inequação de limites de fluxo:

$$\left|\left(\frac{1}{2} + x_{23}\right)(\theta_2 - \theta_3)\right| \leq 80\left(\frac{1}{2} + x_{23}\right) \implies |(\theta_2 - \theta_3)| \leq 80$$

Com as simplificações realizadas anteriormente, o sistema (2.10) se transforma no seguinte sistema:

$$\min v = 3n_{12} + 2n_{13} + 2n_{23} \quad (2.11)$$

s.a.

$$-\left(\frac{5}{6} + x_{12} + x_{13}\right)\theta_1 + \left(\frac{1}{2} + x_{12}\right)\theta_2 + \left(\frac{1}{2} + x_{13}\right)\theta_3 + g_1 = 0$$

$$\left(\frac{1}{3} + x_{12}\right)\theta_1 - \left(\frac{5}{6} + x_{12} + x_{23}\right)\theta_2 + \left(\frac{1}{2} + x_{23}\right)\theta_3 = 60$$

$$\left(\frac{1}{2} + x_{13}\right)\theta_1 + \left(\frac{1}{2} + x_{23}\right)\theta_2 - (1 + x_{13} + x_{23})\theta_3 = 20$$

$$|\theta_1 - \theta_2| \leq 105$$

$$|\theta_1 - \theta_3| \leq 80$$

$$|\theta_2 - \theta_3| \leq 80$$

$$0 \leq g_1 \leq 80$$

$$n_{12}, n_{13} \text{ e } n_{23} \in \{0, 1, 2\}$$

$$x_{12} \in \left\{0, \frac{1}{3}, \frac{2}{3}\right\}$$

$$x_{13} \text{ e } x_{23} \in \left\{0, \frac{1}{2}, 1\right\}$$

$$n_{12} = 3x_{12}$$

$$n_{13} = 2x_{13}$$

$$n_{23} = 2x_{23}$$

$$\theta_1, \theta_2, \text{ e } \theta_3 \text{ irrestritos}$$

que também pode ser obtida diretamente da segunda formulação do modelo DC.

2.4 Modificações na Modelagem Básica

Na utilização de algumas técnicas de resolução às vezes é mais adequado realizar algumas modificações na modelagem básica dos três modelos usados no problema de planejamento de sistemas de transmissão. A modificação mais usada é a utilização de geradores artificiais em todas as barras de carga. O objetivo fundamental da mudança é tornar o problema modificado mais fácil de resolver usando determinados tipos de algoritmos. Do ponto de vista matemático os geradores artificiais constituem apenas novas variáveis (incógnitas) no problema. Portanto, o problema original fica com um número de variáveis acrescido mas pode ser mais fácil de resolver. Logicamente, para que a solução de ambos problemas (original e modificado) sejam equivalentes, na solução final todas as variáveis correspondentes aos geradores artificiais devem ser iguais a zero.

Usando as novas variáveis r_i correspondentes aos geradores artificiais os três modelos usados em planejamento de sistemas de transmissão assumem a seguinte forma:

Modelo de Transportes:

$$\begin{aligned}
 \min v &= \sum_{(i,j)} c_{ij} n_{ij} + \alpha \sum_{k \in \Gamma} r_k & (2.12) \\
 \text{s.a.} & \\
 & S f + g + r = d \\
 & |f_{ij}| \leq (n_{ij} + n_{ij}^o) \bar{f}_{ij} \\
 & 0 \leq r \leq d \\
 & 0 \leq g \leq \bar{g} \\
 & 0 \leq n_{ij} \leq \bar{n}_{ij} \\
 & n_{ij} \text{ inteiro} \\
 & f_{ij} \text{ irrestrito}
 \end{aligned}$$

Modelo Híbrido:

$$\begin{aligned}
 \min v &= \sum_{(i,j)} c_{ij} n_{ij} + \alpha \sum_{k \in \Gamma} r_k & (2.13) \\
 \text{s.a.} & \\
 & S f + g + r = d \\
 & f_{ij} - (\gamma_{ij}^o + x_{ij})(\theta_i - \theta_j) = 0 \quad \forall (i,j) \in \Omega_1 \\
 & |f_{ij}| \leq (n_{ij} + n_{ij}^o) \bar{f}_{ij} \\
 & 0 \leq r \leq d \\
 & 0 \leq g \leq \bar{g} \\
 & 0 \leq n_{ij} \leq \bar{n}_{ij} \\
 & n_{ij} \text{ inteiro} \\
 & x_{ij} \text{ discreto} \\
 & f_{ij} \text{ irrestrito} \\
 & \theta_j \text{ irrestrito } \forall j \in \Omega_3
 \end{aligned}$$

Modelo DC:

$$\begin{aligned}
 \min v &= \sum_{(i,j)} c_{ij} n_{ij} + \alpha \sum_{k \in \Gamma} r_k & (2.14) \\
 \text{s.a.} & \\
 & S f + g + r = d \\
 & f_{ij} - (\gamma_{ij}^o + x_{ij})(\theta_i - \theta_j) = 0 \\
 & |f_{ij}| \leq (x_{ij} + \gamma_{ij}^o) \bar{\phi}_{ij} \\
 & 0 \leq r \leq d \\
 & 0 \leq g \leq \bar{g} \\
 & 0 \leq n_{ij} \leq \bar{n}_{ij} \\
 & n_{ij} \text{ inteiro} \\
 & x_{ij} \text{ discreto} \\
 & f_{ij} \text{ irrestrito} \\
 & \theta_j \text{ irrestrito } \forall j \in \Omega_3
 \end{aligned}$$

em que Γ representa o conjunto de todas as barras k em que existe demanda, r_k representa a geração artificial (ou fictícia) na barra k e α é um parâmetro de penalidade suficientemente grande para tornar pouco atrativa alternativas de investimento com valores de r_k diferentes de zero.

Matematicamente, a inclusão das variáveis artificiais permite obter um sistema de maior tamanho com um incremento das variáveis mas frequentemente mais fácil de resolver usando determinados tipos de algoritmos. É interessante observar que o modelo modificado sempre possui uma solução factível trivial, não necessariamente ótima, fazendo cada $r_k = d_k$ e todas as outras variáveis iguais a zero. Esta característica será adequadamente aproveitada no processo de resolução implementados por alguns algoritmos nos capítulos seguintes.

Capítulo 3

Algoritmos Heurísticos Usados no Planejamento de Sistemas de Transmissão

Como já foi analisado no capítulo de modelagem, o problema de planejamento de sistemas de transmissão pode ser representado usando três modelos matemáticos. Usando o modelo de transportes é obtido um problema de programação linear inteiro misto (PLIM) e usando os modelos híbrido e DC é obtido um problema de programação não linear inteiro misto (PNLIM). Ainda hoje é difícil resolver problemas destes tipos, especialmente para sistemas de grande porte. Portanto, nas pesquisas iniciais sobre o problema de planejamento de sistemas de transmissão foram usados os chamados algoritmos heurísticos construtivos para resolver problemas de planejamento de sistemas reais. Este capítulo está integralmente orientado a analisar vários algoritmos heurísticos construtivos desenvolvidos nas últimas três décadas.

Um algoritmo heurístico construtivo é um procedimento passo-a-passo em que de maneira sistemática se encontra uma boa proposta de expansão do sistema elétrico, isto é, a partir da configuração base, em cada passo é adicionado um ou vários circuitos até que o conjunto de adições realizadas permita uma operação adequada do sistema elétrico. Portanto, em cada passo a configuração do sistema é modificado pela adição de um ou vários circuitos e esta configuração obtida é denominada configuração corrente. O circuito escolhido em cada passo para ser adicionada à chamada configuração corrente é um circuito que corresponde ao caminho mais atrativo identificado pelo chamado critério de sensibilidade, indicador de sensibilidade ou índice de desempenho. Assim, a diferença fundamental entre os vários algoritmos heurísticos construtivos reside no indicador de sensibilidade escolhido e, obviamente, do modelo escolhido (transportes, híbrido, DC).

Um indicador de sensibilidade é basicamente um parâmetro que de alguma maneira está relacionado com a variação da função objetivo devido a alguma variação dos parâmetros do sistema considerando como sistema a configuração corrente. Assim, o indicador de sensibilidade possui as seguintes características:

- Identifica os caminhos mais atrativos para realizar adição de circuitos.



- É um indicador de caráter local, isto é, identifica a melhor estratégia **para a configuração corrente** em contraposição de um indicador de caráter global que identificaria a melhor estratégia para identificar a melhor configuração do sistema.
- Como os indicadores locais nem sempre coincidem com os indicadores globais então os algoritmos heurísticos construtivos, frequentemente, não tem capacidade de encontrar as configurações ótimas globais de sistemas reais.

Resumindo, do ponto de vista teórico um algoritmo heurístico construtivo nem sempre encontra a configuração ótima da expansão de um sistema elétrico. Na prática estes algoritmos heurísticos encontram as configurações ótimas de sistemas pequenos e apenas configurações boas para sistemas elétricos de médio e grande porte. Entretanto, estes algoritmos são muito importantes pelos seguintes motivos:

- Na primeira fase de pesquisas (décadas de 60 e 70) era a única maneira que existia para resolver problemas de planejamento de sistemas elétricos de grande porte.
- A maioria destes algoritmos são robustos e simples de entender, programar e usar.
- Os esforços computacionais destes algoritmos são muito pequenos.
- Muitas características e propriedades destes algoritmos podem ser usadas no desenvolvimento de algoritmos mais complexos como os algoritmos combinatoriais (simulated annealing, algoritmo genético, busca tabu, GRASP, etc.).
- Ainda hoje, esses algoritmos são os mais usados pelas empresas elétricas.

Na análise das diferentes pesquisas publicadas em planejamento de sistemas de transmissão, deve-se realizar uma clara diferenciação entre dois aspectos significativamente diferentes: **a modelagem escolhida** e **o método de resolução**. Assim, a modelagem escolhida pode ser algum dos três apresentados: transportes, híbrido ou DC. Entretanto existem muitos métodos ou técnicas para resolver o problema uma vez escolhido o modelo mais adequado. Estas técnicas podem ser divididas em três grandes grupos: algoritmos heurísticos construtivos, algoritmos de otimização clássica e algoritmos combinatoriais. Entretanto, existem pesquisas publicadas onde é usado mais de um modelo e a técnica de resolução pode ser um algoritmo que usa uma mistura de várias técnicas de resolução mas sempre é possível identificar o modelo e a técnica principal dos modelos e técnicas auxiliares ou complementares. Assim, ao iniciar a apresentação de cada pesquisa, vai-se discutir brevemente os aspectos antes comentados. Este capítulo apresenta os principais algoritmos heurísticos usados no planejamento de sistemas de transmissão.

Os modelos matemáticos apresentados neste capítulo estão ligeiramente modificados em relação à forma original apresentada pelos autores desses modelos e algoritmos, mas conceitualmente ambos modelos, o original e o apresentado neste capítulo, são conceitualmente equivalentes. Portanto, a diferença fundamental corresponde às vezes à técnica de resolução e ao esforço computacional. Neste capítulo os modelos matemáticos são apresentados de modo que em todos eles seja possível usar técnicas de programação linear padrão para resolver seus respectivos modelos matemáticos.



3.1 Heurísticas Construtivas Usando o Modelo de Transportes

Garver [2] desenvolveu o primeiro algoritmo de grande difusão usado no planejamento de sistemas de transmissão. O trabalho de Garver foi pioneiro em vários aspectos e entre os mais importantes podem ser citados os seguintes:

- Sugeriu uma forma sistemática de realizar o planejamento de sistemas de transmissão introduzindo técnicas diferentes aos usados na análise de operação de sistemas elétricos. Assim, sugeriu usar o agora conhecido como modelo de transportes como a forma mais adequada de modelagem matemática para realizar trabalhos de planejamento em contraposição ao uso de modelos mais exatos como o fluxo de carga AC mas que não podiam ser aplicáveis em trabalhos de planejamento. Modelos como o fluxo de carga AC não podem ser usados porque a topologia de sistemas elétricos usados em planejamento são sistemas não conexos com barras e/ou regiões ilhadas e porque ainda não foi resolvido o problema de reativos o que levaria a sérios problemas de convergência mesmo em sistemas não ilhados.
- Inaugurou a fase dos algoritmos heurísticos construtivos que foram muito usados nas décadas seguintes. Estes algoritmos consistem, basicamente, em ir adicionando um circuito em cada passo usando um indicador de sensibilidade até que sejam satisfeitas todas as condições de operação.

Logicamente, o algoritmo desenvolvido por Garver foi uma tentativa em encontrar uma boa solução de um problema complexo e não necessariamente a solução ótima. Encontrar a solução ótima da modelagem proposta por Garver, o modelo de transportes, implicaria em resolver um problema de programação linear inteiro misto (PLIM). A solução ótima de um problema desse tipo pode ser encontrado, por exemplo, usando um algoritmo de Branch and Bound mas esta técnica exige o uso de conhecimentos profundos de programação linear, programação inteira, boa experiência em programação com linguagem de alto nível (FORTRAN, C, etc) e computadores de alta velocidade para resolver problemas de sistemas reais e nem todas essas ferramentas estavam disponíveis na década de 60 quando foi desenvolvido o trabalho de Garver.

3.1.1 O Algoritmo de Garver [2]

Garver sugere usar o modelo de transportes como o modelo matemático mais adequado para resolver problemas de planejamento. O modelo de transportes, amplamente discutido no capítulo 2, assume a seguinte forma:



$$\begin{aligned}
 \min v &= \sum_{(i,j)} c_{ij} n_{ij} & (3.1) \\
 \text{s.a.} & \\
 & S f + g = d \\
 & |f_{ij}| \leq (n_{ij} + n_{ij}^o) \bar{f}_{ij} \\
 & 0 \leq g \leq \bar{g} \\
 & 0 \leq n_{ij} \leq \bar{n}_{ij} \\
 & n_{ij} \text{ inteiro} \\
 & f_{ij} \text{ irrestrito}
 \end{aligned}$$

No modelo de transportes é levada em conta somente a primeira lei de Kirchhoff e os limites de fluxos nos circuitos, isto é, não é levada em conta a segunda lei de Kirchhoff representando um modelo relaxado em relação aos modelos híbrido e DC. Portanto, a solução encontrada para o modelo de transportes não necessariamente é uma solução factível para os modelos híbrido e DC. Além disso, é necessário mencionar que a forma matemática do modelo de transportes apresentado por Garver em [2] é significativamente diferente ao sistema (3.1) apresentado aqui embora sejam conceitualmente equivalentes.

O problema (3.1) é um PLIM mas se aceitamos como solução valores de n_{ij} contínuos então (3.1) se transforma num simples problema de programação linear (PL). Em outras palavras, se relaxamos a integralidade das variáveis inteiras n_{ij} aceitando que assumam valores fracionários então o problema (3.1), muito complexo, transforma-se num problema simples. O problema resultante de (3.1) após relaxar a integralidade das variáveis inteiras leva a um PL conhecido como problema de programação linear correspondente. Deve-se observar que no sistema (3.1) as restrições do tipo $|f_{ij}| \dots$ com valor absoluto podem ser substituídas por duas restrições lineares. Logicamente, uma solução com linhas de transmissão fracionária é inaceitável como proposta global de solução mas pode ser uma idéia excepcional **como estratégia** na tentativa de encontrar uma boa solução com variáveis de investimento inteiras. Assim, a idéia excepcional de Garver consiste em usar o PL correspondente como uma estratégia para encontrar uma boa solução do problema original.

A proposta de Garver consiste em resolver o sistema (3.1) relaxado e encontrar a solução ótima contínua para a configuração corrente n_{ij}^o . Assim, conhecidas as incógnitas n_{ij} , encontradas usando um algoritmo de PL, pode-se encontrar os fluxos de potência em todos os circuitos antigos (n_{ij}^o) e novos (n_{ij}). Portanto, aquele caminho novo ij identificado pelo n_{ij} que leva o maior fluxo de potência representa o caminho mais atrativo de acordo com a proposta de Garver. A proposta de Garver então consiste em adicionar uma linha na configuração corrente naquele caminho mais atrativo e atualizar a configuração corrente de acordo com a adição escolhida. Um processo repetido dessa estratégia, adicionando em cada passo um circuito no caminho mais atrativo, constitui o algoritmo de Garver. O processo termina quando a solução do PL correspondente à configuração corrente apresenta uma solução com todos os $n_{ij} = 0$ o que significa que não é necessário realizar mais adições e o conjunto de adições realizadas representa a proposta de solução do algoritmo de Garver.

O algoritmo de Garver pode ser resumido nos seguintes passos:

1. Assumir a configuração base n_{ij}^o como configuração corrente.
2. Resolver o PL correspondente do sistema (3.1) para a configuração corrente. Se todos os $n_{ij} = 0$ então pare pois foi encontrada uma boa configuração factível. Em caso contrário ir ao passo 3.
3. Calcular os fluxos através de todos os novos circuitos adicionados pelo PL, ($n_{ij} \neq 0$), usando a relação $f_{ij}^v = n_{ij} \bar{f}_{ij}$. Identificar o caminho novo ij com o maior valor de f_{ij}^v e atualizar a configuração corrente adicionando um circuito naquele caminho ij . Voltar ao passo 2.

O algoritmo apresentado anteriormente não tem a mesma estrutura nem as simbologias usadas em [2] mas são conceitualmente equivalentes. Em [2] são usados conceitos como trajetórias sobrecarregadas (overload path) e a determinação de fluxos lineares (linear flow estimation) que são conceitos decorrentes da formulação matemática e da estratégia de programação linear adotada.

O maior esforço computacional do algoritmo corresponde ao passo 2 que consiste em resolver um problema de PL. Entretanto, o algoritmo apresentado pode ainda ser modificado para obter versões alternativas ligeiramente diferentes do algoritmo de Garver. Assim, podem ser realizadas as seguintes modificações:

1. Após a resolução do primeiro PL, pode-se incorporar na configuração base, simultaneamente, a parte inteira de todos os circuitos que apresentam valores de $n_{ij} \geq 1$.
2. Em cada passo, pode-se escolher para adição de um circuito, aquele caminho com maior valor de n_{ij} em lugar de escolher o caminho com maior valor de f_{ij}^v .
3. Em cada passo, pode-se manter o critério de adição daquele caminho com maior valor de f_{ij}^v mas em lugar de adicionar um simples circuito naquele caminho, pode-se adicionar um número de circuitos igual à parte inteira de n_{ij} .

Todas as modificações propostas podem ser incorporadas separadamente ou em conjunto e levam a algoritmos ligeiramente diferentes do algoritmo de Garver. Todos esses algoritmos alternativos são mais rápidos, pois necessitam de menos chamadas ao algoritmo de PL e, geralmente, mas nem sempre, as soluções obtidas são de menor qualidade. Do ponto de vista de otimização matemática o algoritmo de Garver é um algoritmo heurístico construtivo que na prática encontra configurações de boa qualidade mas do ponto de vista teórico não existe garantia de encontrar a configuração ótima global. Na prática isto significa que o algoritmo encontra com facilidade as configurações ótimas de sistemas pequenos mas em sistemas de grande porte essas configurações podem ficar bastante afastados da configuração ótima.

3.2 Heurísticas Construtivas Usando o Modelo DC

Após o trabalho pioneiro de Garver apareceram novas pesquisas sugerindo a utilização do modelo DC e algoritmos heurísticos construtivos para encontrar uma solução desse modelo. É



evidente que o modelo DC é uma representação mais adequada do problema de planejamento de sistemas de transmissão. Entretanto, a dificuldade de resolver problemas usando o modelo DC é maior quando comparado com os problemas formulados usando o modelo de transportes. Em compensação, as configurações encontradas são melhores como alternativa de planejamento. Deve-se lembrar que, em planejamento de sistemas de transmissão, estão sendo usados modelos matemáticos relaxados (transportes, híbrido, DC) para encontrar configurações que depois devem ser usadas e testadas usando o modelo de operação de sistemas de energia elétrica, isto é, o modelo de fluxo de carga AC. Portanto, é evidente que as configurações obtidas usando os modelos de planejamento de sistemas de transmissão frequentemente devem precisar de um ajuste ou reforço adicional nas fases seguintes de estudo (planejamento de reativos, planejamento de curto prazo, etc).

Levando em conta as observações antes mencionadas, existe um consenso entre os pesquisadores do planejamento de sistemas de transmissão de que o modelo considerado ideal na fase de planejamento é o modelo DC. Consequentemente, a maioria das pesquisas publicadas em planejamento de sistemas de transmissão utilizam o modelo DC e, nos últimos anos, os outros modelos (transportes e híbrido) foram utilizados como modelos auxiliares para ajudar a encontrar excelentes configurações do modelo DC.

Dois algoritmos heurísticos construtivos que usam o modelo DC e são muito usados em sistemas brasileiros são o algoritmo de mínimo esforço e o algoritmo de mínimo corte de carga. Ambos algoritmos são relativamente parecidos e a diferença fundamental está no tipo e na determinação do indicador de sensibilidade usado. Esses algoritmos são apresentados em detalhe a seguir.

3.2.1 O Algoritmo de Mínimo Esforço

O algoritmo de mínimo esforço [3] é um algoritmo heurístico construtivo que usa o modelo DC. Assim como no algoritmo de Garver, o algoritmo de mínimo esforço adiciona ao sistema um circuito em cada passo. Entretanto, a modelagem matemática é ligeiramente diferente do modelo DC apresentado no capítulo 2.

Como o algoritmo de mínimo esforço é um processo de passo a passo, em cada passo é resolvido um problema de fluxo de carga DC. Assim, em cada passo existe disponível a chamada configuração corrente constituída por todos os circuitos existentes na configuração base e os circuitos adicionados nos passos anteriores. Assim, em cada passo do algoritmo, deve-se resolver o seguinte problema:

$$B\theta = P \quad (3.2)$$

em que B é a matriz de susceptâncias do sistema constituído pelos circuitos da configuração corrente e P é o vetor de injeções líquidas de potência ativa. Uma vez conhecidos os valores de θ , pode-se encontrar facilmente os fluxos de potência através dos circuitos do sistema usando a relação:



$$f_{ij} = (\theta_i - \theta_j) \gamma_{ij}^{eq} \quad (3.3)$$

em que γ_{ij}^{eq} é a susceptância equivalente dos circuitos existentes no caminho $i - j$. Se existem circuitos sobrecarregados então, deve-se adicionar mais circuitos atrativos ao sistema até eliminar todas as sobrecargas do sistema elétrico.

Se a configuração corrente apresenta corte de carga então o próximo circuito que deve ser adicionado ao sistema deve ser aquele que apresenta o melhor desempenho determinado por um indicador de sensibilidade que apresenta a seguinte forma:

$$SI_{ij}^{me} = \Delta Z_{ij} = -\frac{1}{2}(\theta_i - \theta_j)^2 \gamma_{ij} \quad (3.4)$$

em que os θ_j são obtidos da solução de (3.4) e γ_{ij} é a susceptância de um circuito no caminho $i - j$. Assim, o circuito mais atrativo é representado por aquele caminho que apresenta o maior valor absoluto de SI_{ij}^{me} . Uma variante da metodologia de mínimo esforço consiste em modificar a relação (3.4) para incorporar o efeito dos custos dos circuitos. Assim, também faz parte do algoritmo de mínimo esforço a utilização do seguinte indicador de sensibilidade:

$$SI_{ij}^{mec} = \frac{\Delta Z_{ij}}{c_{ij}} = -\frac{1}{2}(\theta_i - \theta_j)^2 \gamma_{ij} / c_{ij} \quad (3.5)$$

em que c_{ij} é o custo de adição de uma linha no caminho $i - j$. Portanto, no algoritmo de mínimo esforço pode ser usado qualquer um dos indicadores de sensibilidade anteriormente especificados. Obviamente, o uso de cada um dos indicadores de sensibilidade deve produzir propostas de planejamento diferentes.

A justificativa teórica para usar o indicador de sensibilidade (3.4) é que a solução ótima do seguinte problema:

$$\begin{aligned} \min Z &= \frac{1}{2} \sum \frac{f_{ij}^2}{\gamma_{ij}} \\ \text{s.a.} & \\ S f &= P \end{aligned} \quad (3.6)$$

é equivalente à solução obtida com $B\theta = P$ com $f_{ij} = (\theta_i - \theta_j) \gamma_{ij}^{eq}$, isto é, pela aplicação das duas leis de Kirchhoff. Fazendo um paralelo com circuitos resistivos, pode-se afirmar que as duas leis de Kirchhoff num circuito resistivo produzem uma distribuição de fluxos de potência que minimiza as perdas no circuito resistivo. No sistema de potência, pode-se dizer que as duas leis de Kirchhoff produzem uma melhor distribuição dos fluxos no sistema elétrico. Derivando a função objetivo de (3.6) e após algumas manipulações matemáticas, pode-se encontrar a relação (3.4).

Para terminar de apresentar o algoritmo de mínimo esforço ainda é necessário discutir um aspecto complicante. A configuração corrente pode ser um sistema não conexo, isto é, podem

existir barras isoladas da parte conexa do sistema elétrico. Este fato produz dois problemas: 1) para um sistema não conexo o sistema (3.4) normalmente não tem solução, e 2) é necessário ter valores de todos os θ_j para encontrar os valores dos indicadores de sensibilidade. Este problema foi contornado colocando uma rede fictícia superposta com a rede existente para tornar o sistema elétrico conexo. O conceito de rede fictícia foi usado posteriormente por outros pesquisadores, em alguns casos com outros nomes, para contornar o problema de rede elétrica no conexo. Pode-se usar vários tamanhos de rede fictícia. Assim, pode-se colocar uma rede fictícia constituída por circuitos fictícios através de todos os caminhos do sistema elétrico, colocando circuitos fictícios somente na parcela do sistema elétrico correspondente aos caminhos novos ou ainda colocando circuitos fictícios somente num número mínimo de caminhos novos suficientes para tornar o sistema elétrico conexo. Para o algoritmo de mínimo esforço, pode-se colocar um circuito fictício com um valor reduzido, por exemplo, com um valor de $n_{ij} = 0,001$, isto é, colocando uma milésima parte de um circuito.

Pode-se usar uma formulação ligeiramente diferente da formulação original apresentada em [3]. Assim, em vez de resolver o sistema (3.4), resolve-se o seguinte PL:

$$\begin{aligned} \min w &= \sum r_i & (3.7) \\ \text{s.a.} & \\ & B\theta + g + r = d \\ & 0 \leq g \leq \bar{g} \\ & 0 \leq r \leq d \\ & \theta_j \text{ irrestrito} \end{aligned}$$

A solução de (3.7) deve ser exatamente igual à solução de (3.4). Na verdade (3.7) é um PL com um único ponto factível e na solução final sempre acontece que $w = 0$, isto é, está sendo resolvido um sistema de equações algébricas, que apresenta uma única solução, usando a fase I de um PL. Os motivos para usar esta formulação são os seguintes: 1) todos os algoritmos apresentados estão sendo reformulados para que sejam resolvidos usando algoritmos de PL facilitando a apresentação acadêmica, e 2) esta formulação alternativa permite trabalhar, sem nenhuma modificação, com uma variante do problema de planejamento de sistemas de transmissão quando é possível realizar um redespacho da geração. Neste último caso, não é possível usar a formulação (3.4) porque não são conhecidos com antecedência os valores de g e portanto de P .

É importante ainda observar que a única diferença entre as duas formulações discutidas anteriormente é no esforço computacional. É evidente que a formulação (3.7) deve precisar de um maior esforço computacional. Outro aspecto importante é que na formulação (3.4), deve-se fixar um dos θ_j no valor zero como referência angular enquanto que na formulação (3.7) esta precaução não é necessária pois o algoritmo de PL deixa automaticamente um dos θ_j fora da base com valor igual a zero.

Nos passos intermediários do algoritmo de Garver sempre existem adições de circuitos que produzem investimentos diferente de zero e o algoritmo termina justamente quando o algoritmo de PL não produz investimento porque os circuitos adicionados já permitem operar adequadamente o sistema elétrico. Por outro lado, nos passos intermediários do algoritmo de mínimo

esforço sempre existem circuitos sobrecarregados porque os θ_j estão irrestritos e o algoritmo termina quando não existem circuitos sobrecarregados porque os circuitos adicionados permitem que o sistema opere adequadamente.

O algoritmo de mínimo esforço pode ser resumido nos seguintes passos:

Fase I:

1. Assumir a configuração base n_{ij}^o (ou equivalentemente x_{ij}^o) como configuração corrente.
2. Resolver o PL correspondente ao sistema (3.7) para a configuração corrente. Se as sobrecargas foram eliminados então passar para a fase II pois foi encontrada uma configuração factível. Caso contrário, calcular os indicadores de sensibilidade de mínimo esforço usando a relação (3.4) ou a relação (3.5) e ordenar a lista de circuitos candidatos a adição iniciando pelo circuito mais atrativo de acordo com o indicador de sensibilidade, isto é, o circuito mais atrativo é aquele que possui o maior valor absoluto de indicador de sensibilidade encontrado usando (3.4) ou (3.5).
3. Adicionar ao sistema um circuito correspondente ao caminho mais atrativo determinado no passo anterior (o primeiro elemento da lista) para obter a nova configuração corrente.

Fase II:

1. É possível que existam alguns circuitos irrelevantes adicionados durante a fase I devido a outras adições mais importantes realizadas posteriormente. Ordenar os circuitos adicionados em ordem decrescente de seus custos e eliminar aqueles que, uma vez simulada sua saída, não produzem cortes de carga no sistema.

3.2.2 O Algoritmo de Mínimo Corte de Carga

O algoritmo de mínimo corte de carga foi originalmente apresentado em [4]. Este algoritmo também usa o modelo DC modificado usando os geradores artificiais para contornar os problemas de operação do sistema elétrico.

No algoritmo de mínimo esforço os problemas de operação do sistema elétrico são contornados permitindo que os circuitos sejam sobrecarregados. No algoritmo de mínimo corte de carga os circuitos devem permanecer dentro dos limites de operação e, portanto, os problemas de operação são contornados pelos geradores artificiais o que significa que são realizados cortes de carga no sistema. Assim, em cada passo, o algoritmo de mínimo corte de carga minimiza o corte de carga do sistema e o processo termina quando foram adicionados circuitos suficientes e não existem cortes de carga no sistema elétrico. Neste caso também, em cada passo, existe a chamada configuração corrente que é o conjunto de circuitos existentes na configuração base e os adicionados nos passos anteriores.



Em cada passo do algoritmo de mínimo corte de carga se resolve o seguinte PL:

$$\begin{aligned} \min w &= \sum r_i \\ \text{s.a.} \quad & B\theta + g + r = d \\ & |\theta_i - \theta_j| \leq \bar{\phi}_{ij} \\ & 0 \leq g \leq \bar{g} \\ & 0 \leq r \leq d \\ & \theta_j \text{ irrestrito} \end{aligned} \quad (3.8)$$

O indicador de sensibilidade do método de mínimo corte de carga é obtido de (3.8) após algumas manipulações matemáticas e assume a seguinte forma:

$$SI_{ij}^{mc} = \frac{\partial Z}{\partial \gamma_{ij}} = -(\theta_i - \theta_j)(\pi_i - \pi_j) \quad (3.9)$$

em que SI_{ij}^{mc} é o indicador de sensibilidade da função objetivo (3.8) com respeito à susceptância do circuito no caminho $i-j$, os π_j são os multiplicadores de Lagrange das restrições $B\theta + g + r = d$ e os θ_j são os ângulos de tensão das barras do sistema obtidas ao resolver (3.8).

No método de mínimo corte de carga acontece os mesmos problemas que aparecem no método de mínimo esforço em relação aos sistemas não conexos. Portanto, também deve ser usada uma rede fictícia para contornar os problemas de rede não conexa. Assim, é definida uma rede fictícia constituída por circuitos fictícios com valores de n_{ij} muito pequenos, mas com uma capacidade de transmissão muito maior que um circuito normal, isto é, o circuito fictício deve ter uma abertura angular maior que a abertura angular máxima permitida a um circuito normal. Por exemplo, nos exemplos apresentados são usados circuitos fictícios com valores de $n_{ij} = 0,001$ e com aberturas angulares máximas de $|\theta_i - \theta_j| \leq 10\bar{\phi}_{ij}$. A primeira característica do circuito fictício, n_{ij} pequeno, é apenas para dispor de um sistema conexo, evitar problemas numéricos e produzir uma modificação mínima no sistema. Entretanto, a segunda característica é importante em relação à segunda lei de Kirchhoff, isto é, a abertura angular de um circuito fictício deve ser suficientemente grande para que as diferenças angulares nos laços artificiais gerados pelos circuitos fictícios não modifiquem as aberturas angulares nos circuitos normais.

Uma variante do indicador de sensibilidade de mínimo corte de carga pode ser obtida incluindo o efeito dos custos dos circuitos. Assim, uma relação alternativa para usar como indicador de sensibilidade no método de mínimo corte de carga é seguinte:

$$SI_{ij}^{mcc} = SI_{ij}^{mc} / c_{ij} \quad (3.10)$$

em que c_{ij} é o custo de adição de um circuito no caminho $i-j$.

O algoritmo de mínimo corte de carga pode ser resumido nos seguintes passos:

Fase I:

1. Assumir a configuração base n_{ij}^o (ou, equivalentemente, x_{ij}^o) como configuração corrente.
2. Resolver o PL correspondente do sistema (3.8) para a configuração corrente. Se os cortes de carga foram eliminados então passar para a fase II pois foi encontrada uma configuração factível. Em caso contrário, calcular os indicadores de sensibilidade de mínimo corte de carga usando a relação (3.9) ou (3.10) e ordenar a lista de circuitos candidatos a adição iniciando pelo circuito mais atrativo de acordo com o indicador de sensibilidade, isto é, o circuito mais atrativo é aquele que possui o maior valor de indicador de sensibilidade encontrado usando (3.9) ou (3.10) .
3. Adicionar ao sistema um circuito correspondente ao caminho mais atrativo determinado no passo anterior (o primeiro elemento da lista) para obter a nova configuração corrente.

Fase II:

1. É possível que existam alguns circuitos irrelevantes adicionados durante a fase I devido a outras adições mais importantes realizadas posteriormente. Ordenar os circuitos adicionados em ordem decrescente de seus custos e eliminar aqueles que, uma vez simulada sua saída, não produzem cortes de carga no sistema.

3.3 Heurísticas Construtivas que Usam Modelos Mistos

Existem várias pesquisas que usam uma modelagem matemática mista [5, 7], isto é, a modelagem matemática que é resolvida em cada passo do algoritmo heurístico construtivo apresenta características significativamente diferentes das apresentadas no capítulo 2. Entretanto, a solução encontrada satisfaz as exigências do modelo DC, portanto, pode-se afirmar que a modelagem usada nesses métodos corresponde ao modelo DC com a utilização de algumas modificações que substituem os indicadores de sensibilidade dos outros algoritmos heurísticos. Assim, apresenta-se os algoritmos heurísticos construtivos desenvolvidos em [5] e [7].

3.3.1 O Algoritmo de Villasana-Garver

Na proposta de Villasana-Garver [7] é usada uma modelagem matemática que representa duas redes elétricas superpostas, uma rede elétrica que corresponde aos circuitos existentes na chamada configuração corrente e uma rede artificial ou fictícia existente em todos os caminhos em que podem ser adicionados circuitos. A idéia fundamental do algoritmo consiste em que o sistema elétrico deve tentar resolver o problema de operação usando somente os circuitos existentes na configuração corrente e usar os circuitos artificiais somente no caso em que os circuitos existentes sejam insuficientes para resolver o problema de operação.



Na modelagem apresentada em [7] os circuitos da rede existente devem satisfazer as duas leis de Kirchhoff e os limites de fluxos nos circuitos enquanto que os circuitos da rede artificial devem satisfazer somente a primeira lei de Kirchhoff. Em outras palavras, em cada barra do sistema elétrico, deve-se satisfazer a primeira lei de Kirchhoff mas a segunda lei de Kirchhoff deve ser satisfeita somente nos laços formados exclusivamente por circuitos existentes na configuração corrente. Esta formulação também pode ser chamada de híbrida porque a rede formada pelos circuitos existentes apresentam a estrutura do modelo DC e a rede artificial apresenta a estrutura do modelo de transportes. Entretanto, os dois modelos não estão separados porque ambas redes superpostas devem resolver, em conjunto, os problemas de operação.

Em cada passo do algoritmo apresentado em [7], deve-se resolver o seguinte problema:

$$\begin{aligned} \min v_1 &= \sum c_{ij} n_{ij} & (3.11) \\ \text{s.a.} & \\ & S f + B_1 \theta_1 + g = 0 \\ & |\theta_i - \theta_j| \leq \bar{\phi}_{ij} \quad \forall (i, j) \in \Omega_1 \\ & |f_{ij}| \leq \bar{f}_{ij} n_{ij} \quad \forall (i, j) \in \Omega_2 \\ & 0 \leq g_1 \leq \bar{g} \\ & n_{ij} \geq 0 \\ & f_{ij} \text{ irrestrito} \\ & \theta_j, \text{ irrestrito} \end{aligned}$$

em que S é a matriz de incidência nó-ramo da rede artificial, isto é, S é a matriz de incidência nó-ramo do sistema completo, $f = \{f_{ij}\}$ é o fluxo de potência através do circuito artificial no caminho $i - j$, B_1 é a matriz de susceptâncias dos circuitos existentes na configuração corrente, θ_1 é o vetor de ângulos de tensão das barras que estão ligados ao sistema elétrico através de circuitos normais na configuração corrente e n_{ij} é o número de circuitos artificiais que devem ser adicionados para que o sistema “opere adequadamente”.

A formulação (3.11) está apresentada de uma forma diferente ao apresentado em [7] mas é conceitualmente equivalente. Deve-se observar que (3.11) é um problema de programação linear (PL). É evidente a diferença entre (3.11) e a formulação do modelo híbrido apresentado no capítulo 2. O modelo híbrido apresentado no capítulo 2 é um problema de programação não linear inteiro misto cuja solução ótima deve satisfazer a primeira lei de Kirchhoff em todas as barras do sistema e a segunda lei de Kirchhoff em todos los laços formados exclusivamente pelos circuitos existentes na configuração base. Por outro lado, a formulação (3.11) é simplesmente a forma matemática de um passo do algoritmo heurístico apresentado em [7]. É necessário verificar as diferenças entre ambos modelos; assim, por exemplo, em (3.11) n_{ij} é o número de circuitos artificiais enquanto que no modelo híbrido n_{ij} é o número de circuitos que devem ser adicionados ao sistema no caminho $i - j$; também em (3.11) $B_1 \theta_1$ é um conjunto de relações algébricas lineares pois a matriz B_1 tem como elementos parâmetros conhecidos enquanto que no modelo híbrido a relação $B_1 \theta_1$ é um conjunto de relações algébricas não lineares relacionando variáveis x_{ij} (ou n_{ij}) e θ_j .

A formulação (3.11), que é um PL, é usado em [7] para verificar se a configuração corrente

está operando em condições adequadas de operação. Se a configuração corrente não pode operar adequadamente então devem aparecer fluxos de potência através dos circuitos fictícios. Nesse caso, em [7] se sugere adicionar um circuito naquele caminho identificado pelo circuito artificial que transporta maior fluxo de potência e depois atualizar a configuração corrente com a adição implementada. O processo iterativo termina quando $v_1 = 0$ na resolução de (3.11) o que indica que os circuitos já adicionados permitem operar o sistema elétrico adequadamente. Portanto, em cada passo do algoritmo apresentado em [7] se resolve o sistema (3.11) e em cada passo esse sistema muda apenas na estrutura da matriz B_1 , incorporando o efeito do último circuito adicionado ao sistema, e também toda vez que é adicionado pela primeira vez um circuito que corresponde a um novo caminho, deve-se adicionar a correspondente restrição $|\theta_i - \theta_j| \leq \bar{\phi}_{ij}$.

A justificativa usada para escolher o circuito que deve ser adicionado ao sistema é empírica e, basicamente, segue os mesmos critérios que justificam o critério de adição usado no algoritmo de Garver para o modelo de transportes. Portanto, não há uma justificativa matemática rigorosa mas a observação experimental de que a rede fictícia identifica os caminhos mais atrativos. Obviamente, os problemas de rede não conexa são contornados pela rede fictícia. No final do processo, quando as adições realizadas operam adequadamente o sistema, a rede fictícia é eliminada e existe disponível a configuração final do problema constituído pelos circuitos existentes na configuração base e as adições de circuitos realizados no processo iterativo. Esta configuração final é uma solução que é factível para o modelo DC mas não necessariamente a solução ótima porque foi obtido usando um algoritmo heurístico construtivo.

O algoritmo de Villasana-Garver pode ser resumido nos seguintes passos:

Fase I:

1. Assumir a configuração base, sem nenhuma adição, como configuração corrente e montar o sistema (3.11) para essa configuração.
2. Resolver o PL correspondente ao sistema (3.11) para a configuração corrente. Se a resolução do PL não apresenta investimento, isto é, se todos os $n_{ij} = 0$ então pare, pois foi encontrada uma boa configuração factível. Caso contrário calcular o investimento corrente que é a soma do investimento das linhas adicionadas mais o investimento contínuo fornecido pelo PL e ir ao passo 3.
3. Adicionar na configuração corrente um circuito naquele caminho fictício que leva o maior fluxo de potência entre todos os caminhos fictícios para obter a nova configuração corrente. Montar o sistema (3.11) para a configuração corrente, atualizar o investimento correspondente aos circuitos adicionados e voltar ao passo 2.

Fase II:

1. É possível que existam alguns circuitos irrelevantes adicionados durante a fase I devido a outras adições mais importantes realizadas posteriormente. Ordenar os circuitos adicionados em ordem decrescente de seus custos e eliminar aqueles que, uma vez simulada sua saída, não produzem cortes de carga no sistema.

3.3.2 O Algoritmo de Rede Marginal de Levi-Calovic

O algoritmo de rede marginal foi apresentado em [5]. Este algoritmo é muito parecido ao algoritmo de Garver-Villasana mas apresenta algumas contribuições adicionais interessantes. Em vez de usar duas redes superpostas como em [7], em [5], realiza-se a montagem e resolve-se dois modelos matemáticos correspondentes a duas redes diferentes em cada passo do algoritmo heurístico construtivo. O primeiro modelo matemático corresponde ao modelo DC para a chamada configuração corrente, isto é, verifica-se se o sistema opera adequadamente para a configuração corrente. Caso o sistema não opere adequadamente para a configuração corrente, procede-se a montagem da chamada rede marginal usando os resultados obtidos ao resolver o primeiro modelo matemático.

O algoritmo apresentado em [5] resolve primeiro o seguinte problema de programação linear (PL):

$$\begin{aligned} \min w &= \sum r_i & (3.12) \\ \text{s.a.} & \\ & B_1 \theta_1 + g + r = d \\ & |\theta_i - \theta_j| \leq \bar{\phi}_{ij} \quad \forall (i, j) \in \Omega_1 \\ & 0 \leq g \leq \bar{g} \\ & 0 \leq r \leq d \\ & \theta_j, \text{ irrestrito} \end{aligned}$$

em que B_1 é a matriz de susceptâncias dos circuitos existentes na configuração corrente e θ_1 é o vetor dos ângulos das barras ligadas ao sistema elétrico através dos circuitos existentes na configuração corrente, isto é, não existem ângulos para as barras ilhadas na configuração corrente.

Se durante o processo iterativo a resolução de (3.12) não apresenta corte de carga, isto é, $w = 0$ então significa que as adições já realizadas permitem que o sistema opere adequadamente e foi encontrada uma solução factível para o problema de planejamento de sistemas de transmissão. Por outro lado, se a resolução de (3.12) apresenta corte de carga, então a configuração corrente ainda não permite uma operação adequada do sistema e, deve-se adicionar o circuito mais atrativo para obter a nova configuração corrente. Esse circuito mais atrativo é encontrado resolvendo o problema da rede marginal.

O modelo da rede marginal é uma modelagem matemática montada a partir da chamada rede marginal que é montada a partir da resolução do sistema (3.12), levando em conta os seguintes aspectos:

- As gerações na rede marginal são formadas pelas gerações residuais obtidas na resolução de (3.12). Por exemplo, se na resolução de (3.12) uma barra de geração está gerando um valor de $g = 400.MW$ e sua capacidade máxima de geração é de $\bar{g} = 450$ então na rede marginal, nesta barra de geração, deve aparecer um gerador com uma capacidade



máxima de geração de $\bar{g}_m = 450 - 400 = 50MW$, isto é, com a capacidade de geração não usada ou residual.

- As demandas ou cargas na rede marginal são formadas pelos cortes de carga apresentados na resolução de (3.12). Assim, por exemplo, se uma barra de carga apresenta um corte de carga de $r_m = 20MW$ então, na rede marginal, a demanda nesta barra é de $d_m = r_m = 20MW$. Assim, na rede marginal as demandas e gerações estão constituídas pela parcela de demanda não fornecida e pela parcela de capacidade de geração não gerada por insuficiência dos circuitos elétricos na configuração existente.
- Os circuitos na rede marginal são obtidas de uma forma um pouco mais complexa definindo vários tipos de circuitos:

1. Circuitos saturados:

Se na resolução do problema (3.12) os circuitos existentes no caminho $i - j$ estão no limite ou saturados então, nesse caminho, o circuito correspondente na rede marginal deve apresentar um custo c_{ij} , isto é, equivalente a um circuito normal do sistema elétrico. Assim, por exemplo, se na resolução de (3.12) o caminho $i - j$ apresenta $f_{ij} = \bar{f}_{ij} = 100$ então, na rede marginal qualquer adição nesse caminho deve ser considerado com seu correspondente custo de c_{ij} . Em outras palavras, neste caso, não existe capacidade residual nesse caminho que seja aproveitável porque os circuitos estão trabalhando no limite.

2. Circuitos não saturados:

Se na resolução do problema (3.12) os circuitos existentes no caminho $i - j$ estão trabalhando abaixo de seus limites ou estão não saturados então, nesse caminho, o circuito correspondente na rede marginal deve apresentar um custo igual a zero pela capacidade residual existente e um custo de c_{ij} acima desta capacidade residual. Assim, por exemplo, se na resolução de (3.12) o caminho $i - j$ tem a capacidade $\bar{f}_{ij} = 100$ e apresenta um fluxo de $f_{ij} = 70$ no sentido $i - j$ então na rede marginal qualquer adição nesse caminho deve apresentar custo zero até atingir um fluxo de $f_{ij} = 100 - 70 = 30$ no sentido $i - j$ e todo fluxo acima desse valor deve ser considerado com seu correspondente valor de c_{ij} . Além disso, no mesmo exemplo, qualquer adição nesse caminho deve apresentar custo zero para fluxo de potência no sentido $j - i$ até atingir o valor de $f_{ij} = 100 + 70 = 170$ e acima desse valor deve ser considerado um custo de c_{ij} para o circuito adicionado relativo ao excesso de fluxo de potência. Em outras palavras, a variável de investimento n_{ij} que determina o número de circuitos adicionados deve ter custo zero até atingir um valor especificado $\bar{n}_{ij}$ e acima desse valor deve ter um custo c_{ij} . Assim, a rede marginal aproveita a capacidade disponível dos circuitos existentes com custo zero, o que é muito lógico, e utiliza os circuitos que implicam investimento somente quando não é possível usar as capacidades residuais existentes.

Em [5] ainda é definido o chamado circuito quase saturado mas este pode ser eliminado por ser pouco relevante.

Com as observações anteriores a rede marginal produz um modelo matemático que assume a seguinte forma:

$$\begin{aligned}
\min v_1 &= \sum c_{ij} n_{ij} & (3.13) \\
\text{s.a.} & \\
& S f + g_m = d_m \\
& |f_{ij}| \leq \bar{f}_{ij} n_{ij} \\
& 0 \leq g_m \leq \bar{g}_m \\
& n_{ij} \geq 0 \\
& f_{ij}, \text{ irrestrito}
\end{aligned}$$

em que g_m e d_m é o vetor de demanda e geração residual ou marginal e o custo c_{ij} é linear por partes e assume a forma mostrada na figura 3.1.

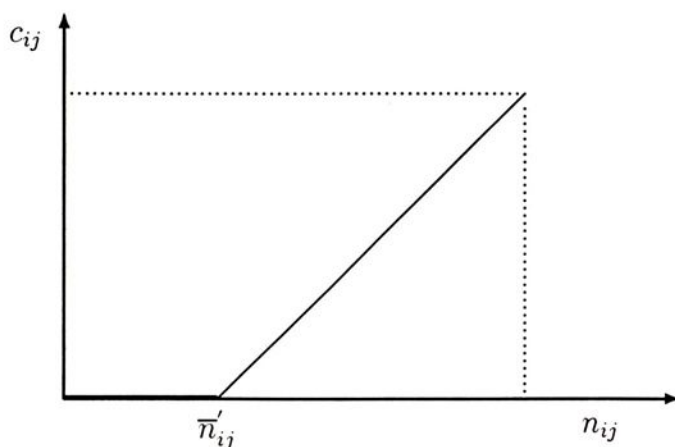


Figura 3.1: Função de custo na rede marginal.

Na figura 3.1 o valor $\bar{n}'_{ij}$ é a margem de circuito disponível sem custo na rede marginal. Assim, $\bar{n}'_{ij}$ pode ser obtido através da seguinte relação:

$$\bar{n}'_{ij} = 1 - \frac{f'_{ij}}{\bar{f}_{ij}} \quad (3.14)$$

em que f'_{ij} é o fluxo de potência através dos circuitos no caminho $i - j$ na resolução de (3.12) e $\bar{f}_{ij}$ é o fluxo máximo de potência através dos circuitos no caminho $i - j$. Assim, os valores de $\bar{n}'_{ij}$ são obtidos a partir dos resultados da solução de (3.12).

O modelo de rede marginal apresentado em (3.13) é um pouco diferente ao apresentado em [7]. A principal diferença está na modelagem do circuito não saturado. Assim, por exemplo, se no caminho $i - j$ existe um circuito com capacidade de 100 MW e na resolução de (3.12) existe um fluxo de 70 MW no sentido $i - j$, então no modelo de rede marginal apresentado em [5] não existe custo para fluxo de potência de até $100 - 70 = 30$ MW no sentido $i - j$ e de até $100 + 70 = 170$ no sentido de $j - i$. Por outro lado, no modelo apresentado em (3.13) não existe

custo (custo igual a zero) para fluxos de potência de até 30 MW em ambos sentidos. Assim o modelo (3.13) aparentemente é mais restrito. Entretanto, posteriormente será discutido que a caracterização de circuito não saturado é a principal limitação do modelo de rede marginal apresentado em [5]. Também, em [5] o modelo de rede marginal é resolvido usando um algoritmo de fluxo em redes. Entretanto, podem ser usados algoritmos de programação linear para resolver ambos problemas do algoritmo de rede marginal.

Como o problema (3.13) apresenta uma função objetivo linear por partes então esse problema pode ser resolvido por várias técnicas de programação linear. A maneira mais simples consiste em realizar uma simples transformação de variáveis. Assim, a variável n_{ij} é substituída por duas variáveis:

$$n_{ij} = n'_{ij} + n''_{ij} \quad \text{com} \quad 0 \leq n'_{ij} \leq \bar{n}'_{ij}; \quad 0 \leq n''_{ij} \quad (3.15)$$

A transformação anterior é realizada para cada circuito que não está saturado ou não está no limite porque, para os circuitos saturados, $\bar{n}'_{ij} = 0$ e a transformação não é necessária. Com as transformações anteriores o sistema (3.13) se transforma no seguinte problema de programação linear (PL):

$$\begin{aligned} \min v_1 &= \sum c_{ij} n''_{ij} & (3.16) \\ \text{s.a.} & \\ & S f + g_m = d_m \\ & |f_{ij}| \leq \bar{f}_{ij}(n'_{ij} + n''_{ij}) \\ & 0 \leq n'_{ij} \leq \bar{n}'_{ij} \\ & 0 \leq g_m \leq \bar{g}_m \\ & 0 \leq n''_{ij} \\ & f_{ij}, \text{ irrestrito} \end{aligned}$$

No modelo de rede marginal existe ainda um assunto que deve ser analisado. Em (3.13) foram considerados como parte da rede marginal todos os circuitos candidatos a adição. Entretanto, na resolução do sistema (3.12) não participam os circuitos correspondentes aos novos caminhos, onde ainda não foram realizadas adições. Uma maneira simples de contornar este problema é considerar todos os circuitos correspondentes aos novos caminhos como circuitos saturados e essa estratégia está implicitamente incorporada nas formulações (3.13) e (3.16). Em [5] são incorporadas na rede marginal somente uma parcela dos circuitos correspondentes aos novos caminhos onde ainda não foram realizadas adições. A idéia fundamental dessa implementação é reduzir o tamanho do problema. Em [5] são apresentadas três estratégias para identificar os circuitos correspondentes aos novos caminhos que devem ser incorporados na rede marginal sendo que duas delas são as seguintes:

- Se $\sigma_{ij} = (\theta_i - \theta_j)(\pi_i - \pi_j) < 0$ e $(\theta_i - \theta_j) > 0$.
- Se $(\theta_i - \theta_j) > 0$.

Esta idéia de redução de circuitos, correspondentes a novos caminhos na rede marginal, pode ser facilmente incorporada em (3.16) simplesmente eliminando todos os circuitos de novos caminhos que não satisfazem o critério escolhido.

Em relação ao algoritmo apresentado em [5] é importante realizar duas observações fundamentais:

- Em [5] se reconhece implicitamente que o sistema (3.12) deve ser conexo. Assim, em [5] primeiro se resolve este problema usando circuitos atrativos para tornar o sistema elétrico conexo. Na verdade, esta implementação não é necessária e o sistema (3.12) apresenta solução para um sistema não conexo. A necessidade de rede conexa seria apenas para implementar a escolha de circuitos de novos caminhos que devem ser incorporados na rede marginal.
- Assim como em [7], em testes realizados em sistemas reais podem aparecer propostas de solução em aparente conflito. Assim, por exemplo, na resolução de (3.12), para um determinado caminho, pode aparecer um fluxo de potência no sentido $i \rightarrow j$ e nesse mesmo caminho pode aparecer um fluxo no caminho $j \rightarrow i$ na resolução do modelo de rede marginal usando (3.16). A modelagem matemática permite e incentiva esta possibilidade ao permitir fluxos com custo zero para os circuitos não saturados para os fluxos em ambos sentidos. A consequência deste conflito leva a propostas de adições inadequadas. Como em cada passo do algoritmo resolve-se os sistemas (3.12) e (3.16) para realizar um simples reforço, com a adição de um único circuito, aparentemente não é adequado considerar como boa alternativa um caminho no qual os fluxos aparecem em conflito na resolução de ambos sistemas. Portanto, pelo menos não se deveria incentivar esta possibilidade. Este problema aparentemente não é fácil de contornar na metodologia apresentada em [7] mas em [5] existem várias formas simples de eliminar este problema.

O algoritmo de rede marginal pode ser resumido nos seguintes passos:

Fase I:

1. Assumir a configuração base, sem nenhuma adição, como configuração corrente e montar o sistema (3.12) para essa configuração, isto é, usar o modelo DC para a configuração corrente.
2. Solução do modelo DC para a configuração corrente:
Resolver o PL correspondente do sistema (3.12) para a configuração corrente. Se a resolução do PL não apresentar corte de carga então foi terminada a fase I, pois foi encontrada uma boa configuração factível e, deve-se passar para a fase II. Em caso contrário ir ao passo 3.
3. Montagem e resolução da rede marginal:
Usando os resultados do passo anterior identificar as demandas e gerações residuais assim como a capacidade de transmissão residual dos circuitos da rede existente. Montar o modelo de rede marginal do sistema (3.16) e resolver o PL resultante.



4. Reforço da configuração corrente:

Adicionar na configuração corrente um circuito naquele caminho da rede marginal que leva o maior fluxo de potência para obter a nova configuração corrente. Atualizar o investimento correspondente aos circuitos adicionados e voltar ao passo 2.

Fase II:

1. É possível que existam alguns circuitos irrelevantes adicionados durante a fase I devido a outras adições mais importantes realizadas posteriormente. Ordenar os circuitos adicionados em ordem decrescente de seus custos e eliminar aqueles que, uma vez simulada sua saída, não produzem cortes de carga no sistema.



Capítulo 4

Algoritmos de Planejamento Ótimo Usando o Modelo de Transportes

Neste capítulo se apresenta várias técnicas para encontrar a solução ótima do problema de planejamento de sistemas de transmissão quando é usado o modelo de transportes. Quando é usado o modelo de transportes o problema de otimização resultante é um PLIM, amplamente discutido em capítulos anteriores, que assume a seguinte forma:

$$\begin{aligned} \min v &= \sum_{(i,j)} c_{ij} n_{ij} & (4.1) \\ \text{s.a.} & \\ & S f + g = d \\ & |f_{ij}| \leq (n_{ij} + n_{ij}^o) \bar{f}_{ij} \\ & 0 \leq g \leq \bar{g} \\ & 0 \leq n_{ij} \leq \bar{n}_{ij} \\ & n_{ij} \text{ inteiro} \\ & f_{ij} \text{ irrestrito} \end{aligned}$$

No estágio atual do desenvolvimento das técnicas de otimização matemática é possível resolver um problema de programação linear inteiro misto (PLIM) usando várias técnicas. A maior limitante está representado pelo esforço computacional e pela sofisticação necessária na implementação computacional para sistemas de grande porte. Neste capítulo são apresentados vários algoritmos que permitem encontrar a solução ótima para o modelo de transportes.

Uma metodologia adequada para encontrar a configuração ótima do problema de planejamento de sistemas de transmissão quando é usado o modelo de transportes consiste em usar um algoritmo de Branch and Bound. Em princípio, qualquer problema de programação linear inteiro misto (PLIM) pode ser resolvido usando um algoritmo de Branch and Bound (B & B). Como o problema resultante da utilização do modelo de transportes é um PLIM então é possível usar um algoritmo B & B. Entretanto, deve-se desenvolver um eficiente algoritmo de B & B e uma adequada implementação computacional para que o esforço computacional não seja proibitivo na resolução de problemas de grande porte.



Outra alternativa para resolver o problema de planejamento de sistemas de transmissão quando é usado o modelo de transportes consiste em usar técnicas de decomposição matemática que permite decompor o problema complexo de planejamento de sistemas de transmissão em dois subproblemas: um subproblema de investimento também chamado de mestre e um subproblema de operação também chamado escravo. A técnica de decomposição matemática usada é a decomposição de Benders e os dois subproblemas, mestre e escravo, são resolvidos de maneira iterativa até que seja encontrada a solução ótima global. O subproblema de operação é um problema de programação linear e o subproblema de investimento é um problema de programação inteira. Portanto, a complexidade do problema foi transferida para a resolução do subproblema mestre em forma iterativa. O subproblema mestre, um problema de programação inteira, pode ser resolvido usando vários métodos e neste capítulo são apresentados dois métodos: um algoritmo de Branch and Bound e um algoritmo de enumeração implícita de Balas especializado.

Finalmente, outra alternativa para resolver o problema de planejamento de sistemas de transmissão consiste em usar a decomposição de Benders dentro de um contexto hierarquizado usando relaxação. Assim, numa primeira fase é resolvido o PL correspondente do modelo de transportes (relaxando a integralidade das variáveis de investimento) usando a decomposição de Benders e na fase final é resolvido o problema original também usando a decomposição de Benders. A vantagem consiste em usar os cortes de Benders gerados na primeira fase para acelerar o processo de convergência na fase final. A diminuição do esforço computacional é evidente pois na fase inicial se resolve um PL usando a decomposição de Benders sendo os dois subproblemas (mestre e escravo) problemas de PL permitindo diminuir as iterações necessárias para a convergência na fase final onde o subproblema mestre é um problema inteiro (PI). Outra estratégia alternativa consiste em substituir a fase inicial por um conjunto de restrições geradas usando características específicas do problema de planejamento de sistemas de transmissão. Estas restrições, como será mostrado posteriormente, são restrições relaxadas pois não eliminam as soluções factíveis do problema original mas aceleram de maneira significativa o processo de convergência. Todos esses métodos são apresentados separadamente.

4.1 Planejamento Ótimo Usando um Algoritmo de Branch and Bound Especializado

O modelo matemático apresentado em (4.1) corresponde a um problema de programação linear inteiro misto (PLIM) e, portanto, um algoritmo de Branch and Bound convencional tem capacidade para encontrar a solução ótima global de esse tipo de problemas.

Um algoritmo de Branch and Bound é conceitualmente simples. A idéia básica consiste em resolver um problema PLIM resolvendo um conjunto de problemas de programação linear (PL) que são versões relaxadas do problema PLIM. Assim, inicialmente é resolvido o problema original (4.1) após relaxar a integralidade das variáveis de investimento. Este problema de PL, conhecido como problema linear correspondente, e que chamaremos de P_0 é resolvido usando um algoritmo de PL. Se P_0 apresenta uma solução inteira para as variáveis inteiras então essa solução é ótima global para o problema original (4.1). Entretanto, normalmente a solução de P_0 apresenta uma solução com algumas variáveis inteiras com valor corrente contínua. Neste caso, a estratégia é separar ou dividir o problema P_0 em dois subproblemas P_1 e P_2 escolhendo



uma variável inteira com valor corrente contínua para realizar a separação. Assim, se a variável inteira n_{ij} possui um valor corrente contínuo de n_{ij}^* então os problemas sucessores são obtidos da seguinte forma:

- Subproblema P_1 :

é o problema original P_o acrescido da restrição: $n_{ij} \leq [n_{ij}^*]$ em que $[n_{ij}^*]$ é o maior inteiro contido em n_{ij}^* .

- Subproblema P_2 :

é o problema original P_o acrescido da restrição: $n_{ij} \geq [n_{ij}^*] + 1$.

O problema de programação linear P_o original foi dividido em dois subproblemas de programação linear menores P_1 e P_2 . Em outras palavras, já que não foi possível encontrar a solução ótima do problema original com variáveis inteiras resolvendo P_o então o problema é separado em dois problemas menores para tentar encontrar a solução ótima nos problemas menores usando técnicas de PL. Os subproblemas gerados são novamente resolvidos e podem gerar dois novos subproblemas usando uma variável inteira com valor corrente contínua para realizar a separação. Entretanto, às vezes a solução de um subproblema permite descobrir que já não é necessária sua separação em dois novos subproblemas e que, pelo contrário, o subproblema deve ser sondado ou eliminado para futuras avaliações. Um subproblema pode ser sondado quando sua solução apresenta alguma das seguintes características:

1. A solução é inteira, isto é, os valores correntes de todas as variáveis inteiras possuem valores inteiros. Nesse caso, o subproblema pode ter outras soluções inteiras mas todas elas devem ser de menor qualidade que a solução inteira encontrada e, portanto, não é necessário procurar melhores soluções na região factível desse subproblema e deve ser sondado.
2. A solução corrente não é inteira mas a função objetivo do subproblema é de menor qualidade que a função objetivo de uma solução inteira já encontrada resolvendo outros subproblemas. Nesse caso, o subproblema corrente pode ter soluções inteiras na sua região factível mas elas devem ser de pior qualidade que a solução inteira já encontrada e, portanto, o subproblema deve ser sondado.
3. A solução do subproblema é infactível. Nesse caso, os problemas sucessores desse subproblema serão mais infactíveis que o subproblema corrente. Portanto, o subproblema deve ser sondado porque não existem pontos factíveis no subproblema.

As características apresentadas anteriormente é conhecida como testes de sondagem. É importante observar que um problema sucessor como P_1 é um problema mais restrito que seu problema predecessor porque P_1 possui uma restrição adicional. Assim, a região factível de P_1 é menor ou igual que a região factível de P_o . Portanto, como o problema é de minimização, a função objetivo de P_1 deve ser maior ou igual à função objetivo de P_o (mais restrito). As observações anteriores justificam os testes de sondagem. Assim, o algoritmo de Branch and Bound consiste fundamentalmente de uma estratégia de separação do problema em problemas menores até que esses subproblemas sejam sondados. Os subproblemas gerados podem ser apresentados

num gráfico chamado de árvore de branch and bound em que os nós representam os subproblemas e as arestas representam a variável inteira usada para separação. O algoritmo termina quando todos os subproblemas gerados foram sondados. A solução ótima global é a melhor solução inteira encontrada que é chamada de incumbente durante o processo de resolução.

No algoritmo de B & B na verdade está sendo implementada uma estratégia de enumeração implícita e/ou explícita de todas as soluções inteiras da região factível do problema original. Este fato garante que a melhor solução encontrada seja ótima global. Assim, o algoritmo de B & B é conceitualmente simples mas a implementação computacional apresenta maior complexidade e podem aparecer problemas de esforço computacional e de memória na resolução de um problema de grande porte.

A eficiência de um algoritmo de B & B depende de vários tipos de decisão que devem ser realizadas durante o processo de resolução. Obviamente, o melhor algoritmo de B & B é aquele que gera um menor número de nós na árvore de B & B, isto é, aquele algoritmo que realiza o menor número de chamadas ao algoritmo de PL, necessário para resolver cada subproblema gerado. As principais decisões que determinam a qualidade de um algoritmo de B & B estão relacionados com os seguintes aspectos:

1. Determinação de uma solução inteira inicial: incumbente inicial:

Um algoritmo de B & B pode iniciar o processo sem incumbente inicial ou pode usar uma estratégia heurística rápida para gerar uma boa incumbente inicial. Dispor de uma boa incumbente é muito importante para que funcione o segundo teste de sondagem. Assim, uma boa incumbente encontrada o mais rapidamente possível melhora o desempenho do segundo teste de sondagem e portanto do esforço computacional do algoritmo de B & B.

2. A variável escolhida para separação:

A solução de um subproblema pode apresentar muitas variáveis inteiras com valores correntes contínuas. Assim, a escolha de cada uma dessas variáveis para separação do subproblema produzem árvores de B & B diferentes. Obviamente, existe uma variável que produz a menor árvore de B & B mas não existem técnicas sistemáticas que permite identificar a melhor variável que deve ser usada para separar um subproblema. Entretanto, existem regras empíricas que permitem identificar as variáveis mais atrativas que podem ser usadas para separação.

3. Escolha do próximo subproblema que deve ser analisado:

Em determinado momento da resolução de um problema usando o algoritmo de B & B existem muitos subproblemas que devem ser analisados. Assim, um aspecto importante é descobrir qual é o próximo subproblema que deve ser analisado. Obviamente, a escolha de cada subproblema produz árvores de B & B diferentes e existe um subproblema que uma vez escolhido produz a menor árvore de B & B. Também não existem técnicas sistemáticas para identificar o subproblema que deve ser analisado primeiro. Entretanto, também existem regras empíricas que permitem identificar os subproblemas mais atrativos. Um critério muito popular é escolher os subproblemas usando a regra LIFO que escolhe para analisar sempre o último subproblema gerado. Esta estratégia permite um melhor desempenho do algoritmo de PL pois o critério permite que frequentemente seja resolvido um subproblema depois de resolver o subproblema predecessor. Assim, na resolução do subproblema pode ser usado um algoritmo de PL do tipo dual simplex com



variáveis canalizadas usando a base ótima do subproblema predecessor encontrando mais rapidamente a solução ótima do subproblema. Esta estratégia permite resolver mais rapidamente muitos subproblemas.

4. Usando as características específicas do problema:

Os algoritmos de B & B mais eficientes são os algoritmos especializados, isto é, os algoritmos que incorporam as características específicas do problema. Para cada tipo de problema podem ser gerados estratégias eficientes de sondagem ou ainda podem ser geradas informações adicionais como novas restrições que ao serem incorporados ao problema melhoram significativamente os testes de sondagem reduzindo significativamente o tamanho da árvore de B & B. Também é possível aproveitar a estrutura do problema para elaborar algoritmos de PL eficientes que são chamados pelo algoritmo de B & B.

Um algoritmo de Branch and Bound Básico assume a seguinte forma:

1. Inicialização:

Escolher a incumbente inicial $Z_{inc} = z^* = \infty$. Resolver o PL correspondente (PLI original com variáveis contínuas). Se a solução do PL é inteiro, pare pois ela é ótima global. Em outro caso armazenar o valor da função objetivo como limitante inferior z_{inf} dos subproblemas sucessores e ir ao passo 2.

2. Branching:

Escolher a primeira variável inteira x_j com valor corrente contínuo x_j^* para realizar a separação. Gerar dois novos subproblemas a partir do subproblema corrente adicionando a restrição $x_j \leq [x_j^*]$ para gerar um subproblema e a restrição $x_j \geq [x_j^*] + 1$ para gerar o outro subproblema onde $[x_j^*]$ é o maior inteiro contido em x_j^* .

3. Bounding:

Resolver o último PL gerado usando um algoritmo de PL e armazenar a solução ótima $z_{pl} = z_{inf}$ como limitante inferior para os prováveis PL's sucessores.

4. Testes de Sondagem:

Após resolver o PL (subproblema), deve-se verificar os testes de sondagem. O problema é sondado (eliminado para futuros desdobramentos) se satisfaz algum dos seguintes testes de sondagem:

- (a) Teste 1: Se $z_{inf} \geq z^*$, em que z^* é o valor da incumbente.
- (b) Teste 2: Se a solução do PL é infactível.
- (c) Teste 3: Se o ótimo do PL é inteiro, isto é, todas as variáveis inteiras assumem valores inteiros. Neste caso, se o ótimo é menor que a incumbente então, deve-se atualizar a incumbente e aplicar o teste 1 para todos os subproblemas ainda não sondados.

5. Se o problema não foi sondado ir ao passo 2 para fazer branching. Em outro caso verificar se existem subproblemas não sondados. Se todos os subproblemas foram sondados, pare, pois foi terminado o processo; em caso contrário voltar ao passo 3 para resolver o último subproblema gerado.

O algoritmo básico, embora seja conceitualmente simples, apresenta complexidade na implementação computacional. Este algoritmo apresenta excelente desempenho para sistemas pequenos mas em sistemas reais apresenta várias limitações relacionadas com o esforço computacional. Assim, em aplicações reais, deve-se introduzir melhorias significativas no algoritmo básico.

4.1.1 Algoritmo B & B Especializado

Existem duas estratégias diferentes para melhorar o desempenho do algoritmo de B & B: (1) introduzindo melhorias de tipo geral próprios dos algoritmos de B & B, e (2) introduzindo informação específica do tipo de problema que está sendo resolvido.

É possível introduzir estratégias de tipo geral no algoritmo de B & B tais como: (1) no passo inicial introduzir um algoritmo heurístico para encontrar uma boa incumbente inicial; (2) escolher outras estratégias de branching como escolher aquela variável que faz parte do ótimo contínuo, ou a variável que apresenta o maior valor fracional ou ainda a variável que apresenta maior custo, e ainda estes critérios podem ser usados de maneira adaptativa; e (3) usando outra estratégia para escolher o próximo subproblema que deve ser analisado. Nesta linha de pesquisa é possível ainda incorporar outras estratégias mais sofisticadas.

Outra estratégia interessante é incorporar conhecimento sobre o tipo de problema para acelerar o processo de convergência. Para o problema de planejamento existem várias características que podem ser aproveitadas. Pode-se mencionar que a consequência prática de implementar características específicas de um tipo de problema é produzir uma redução significativa do esforço computacional melhorando o desempenho dos chamados testes de sondagem [37].

4.2 Planejamento Ótimo Hierarquizado Usando a Decomposição de Benders e Relaxação

Uma estratégia alternativa para encontrar a configuração ótima do problema de planejamento de sistemas de transmissão quando é usado o modelo de transportes é implementar o chamado planejamento hierarquizado usando a decomposição de Benders. Como o problema é um PLIM, pode-se demonstrar que é possível encontrar a solução ótima deste problema usando a decomposição de Benders. Anteriormente foi amplamente analisado que o problema resultante ao usar o modelo de transportes é um problema de programação linear inteiro misto e quando é relaxada a integralidade das variáveis de investimento, o problema resultante é um simples problema de programação linear.

O nome de planejamento hierarquizado aparece devido a que no processo de resolução são



resolvidas nas fases iniciais versões relaxadas do problema original. Neste caso, inicialmente se resolve o problema de planejamento usando o modelo de transportes mas relaxando a integralidade das variáveis de investimento n_{ij} , isto é, usando a decomposição de Benders se resolve um problema de programação linear. Aparentemente, este processo é irrelevante mas a importância em resolver este problema relaxado é que os cortes de Benders gerados nesta fase são restrições válidas e importantes do problema original ou não relaxado (com variáveis inteiras). Portanto todas as restrições, cortes de Benders, geradas ao resolver o problema relaxado podem ser usados para resolver de maneira mais rápida o problema original. Assim, um algoritmo hierarquizado para resolver o problema de planejamento de sistemas de transmissão quando é usado o modelo de transportes consiste de duas fases mostradas a seguir:

1. Fase I: Resolução do problema relaxado:

Resolve-se o problema com as variáveis de investimento relaxadas e usando a decomposição de Benders. Neste caso, os subproblemas de investimento e de operação obtidos usando a decomposição de Benders são problemas de programação linear. Portanto, esses subproblemas podem ser rapidamente resolvidos usando algoritmos especializados de PL. Os cortes de Benders obtidos nesta fase são incorporados no subproblema de investimento na seguinte fase.

2. Fase II: Resolução do problema original:

Resolve-se o problema original com as variáveis de investimento inteiras e usando a decomposição de Benders. Neste caso, o subproblema de operação é um problema de programação linear e o subproblema de investimento é um problema de programação linear inteira (PLI). Os cortes de Benders gerados na fase anterior são incorporados ao subproblema de investimento. O subproblema de operação pode ser rapidamente resolvido por eficientes algoritmos de PL. Entretanto, a resolução do subproblema mestre exige de um esforço computacional muito maior. Nesta fase o subproblema de investimento deve ser resolvido usando um algoritmo especializado de PLI. Para o problema de planejamento de sistemas de transmissão foram usados dois algoritmos [15]: um algoritmo de Branch and Bound semelhante ao usado neste capítulo e um algoritmo de enumeração implícita de Balas especializado.

A estratégia de usar um algoritmo hierarquizado foi originalmente apresentada em [14] para resolver o problema de planejamento de sistemas de transmissão quando é usado o modelo DC. Assim, a idéia pode ser usada também para resolver o problema de planejamento quando é usado o modelo de transportes. O algoritmo hierarquizado apresenta vários atrativos quando é usado para resolver o modelo DC como será analisado no próximo capítulo. Entretanto, quando é usado o modelo de transportes o atrativo mais importante é a redução do esforço computacional quando comparado com um algoritmo de fase única convencional. Um algoritmo de fase única que usa a decomposição de Benders, que será analisada posteriormente, é basicamente a Fase II apresentada anteriormente mas sem os cortes de Benders (restrições) obtidas na fase I. Assim, desde o início do processo iterativo de Benders, deve-se resolver o subproblema de investimento usando um algoritmo de PLI incrementando o esforço computacional do processo iterativo.



4.2.1 A Decomposição de Benders

A decomposição de Benders é a técnica de decomposição matemática mais adequada para ser usada no problema de planejamento de sistemas de transmissão. A decomposição de Benders permite decompor o problema complexo de planejamento em dois subproblemas: um subproblema de operação ou escravo e um subproblema de investimento ou mestre. As variáveis de operação como os fluxos e os ângulos das tensões de barras fazem parte do subproblema de operação que é um problema de programação linear que pode ser facilmente resolvido usando algoritmos eficientes de PL. Por outro lado, as variáveis de investimento fazem parte do subproblema de investimento que é um problema de programação linear inteiro (PLI) com uma única variável contínua β que aparece como parte da decomposição. Assim, a decomposição de Benders resolve o problema de planejamento de sistemas de transmissão através de uma solução iterativa dos subproblemas de operação e de investimento. Portanto, é evidente que toda a complexidade na resolução do problema de planejamento é transferida para a resolução do subproblema de investimento em forma iterativa.

Para usar a decomposição de Benders é mais adequada usar a formulação matemática modificada incorporando os geradores artificiais como foi analisado no capítulo 2. Assim, o modelo matemático do problema de planejamento de sistemas de transmissão para o modelo de transportes e usando os chamados geradores artificiais assume a seguinte forma:

$$\begin{aligned} \min v &= \sum_{(i,j)} c_{ij} n_{ij} + \alpha \sum r_i & (4.2) \\ \text{s.a.} & \\ & S f + g + r = d \\ & |f_{ij}| \leq (n_{ij} + n_{ij}^o) \bar{f}_{ij} \\ & 0 \leq g \leq \bar{g} \\ & 0 \leq r \leq d \\ & 0 \leq n_{ij} \leq \bar{n}_{ij} \\ & n_{ij} \text{ inteiro} \\ & f_{ij} \text{ irrestrito} \end{aligned}$$

em que o parâmetro α pode ser usado para compatibilizar as unidades na função objetivo.

Usando a decomposição de Benders, o problema de planejamento de sistemas de transmissão usando o modelo de transportes, apresentado em (4.2), pode ser decomposto nos seguintes subproblemas:

- Subproblema de operação (escravo):

$$\begin{aligned}
 \min w &= \alpha \sum r_i & (4.3) \\
 \text{s.a.} & \\
 & S f + g + r = d \\
 & |f_{ij}| \leq \bar{f}_{ij}(n_{ij}^k + n_{ij}^o) \\
 & 0 \leq g \leq \bar{g} \\
 & 0 \leq r \leq d \\
 & f_{ij} \text{ irrestrito}
 \end{aligned}$$

- Subproblema de investimento (mestre):

$$\begin{aligned}
 \min v &= \sum_{(i,j)} c_{ij} n_{ij} + \beta & (4.4) \\
 \text{s.a.} & \\
 & w^k + \sum \sigma_{ij}^k (n_{ij} - n_{ij}^k) \leq \beta \\
 & \beta \geq 0 \\
 & 0 \leq n_{ij} \leq \bar{n}_{ij} \\
 & n_{ij} \text{ inteiro}
 \end{aligned}$$

mas como o parâmetro $\sigma_{ij}^k = -|\pi_{ij}^k| \bar{f}_{ij}$, o subproblema mestre assume a seguinte forma:

$$\begin{aligned}
 \min v &= \sum_{(i,j)} c_{ij} n_{ij} + \beta & (4.5) \\
 \text{s.a.} & \\
 & \sum |\pi_{ij}^k| \bar{f}_{ij} n_{ij} + \beta \geq w^k + \sum |\pi_{ij}^k| \bar{f}_{ij} n_{ij}^k \\
 & \beta \geq 0 \\
 & 0 \leq n_{ij} \leq \bar{n}_{ij} \\
 & n_{ij} \text{ inteiro}
 \end{aligned}$$

em que a variável contínua β aparece como consequência da decomposição e os π_{ij}^k são os multiplicadores de Lagrange das restrições $|f_{ij}| \leq \bar{f}_{ij}(n_{ij}^k + n_{ij}^o)$ do subproblema escravo obtidos na iteração k , w^k é a solução do subproblema escravo na iteração k (representa o corte de carga do sistema se $\alpha = 1$) e n_{ij}^k é a solução do mestre na iteração k . É importante observar que na resolução do subproblema escravo os n_{ij}^k são conhecidos e as variáveis (incógnitas) do problema são exclusivamente as variáveis de operação. Por outro lado, na resolução do mestre são conhecidos os parâmetros π_{ij}^k , $\bar{f}_{ij}$, w^k e os n_{ij}^k e as variáveis do problema são unicamente as variáveis de investimento n_{ij} e β .

É possível usar a decomposição de Benders em cada nível do algoritmo hierarquizado. No primeiro nível simplesmente se resolve um problema de programação linear usando decomposição de Benders. Também, no segundo nível se usa a decomposição de Benders mas para

resolver um problema de programação linear inteira. Pode-se demonstrar que um algoritmo desse tipo pode encontrar a solução ótima de um problema de planejamento de sistemas de transmissão quando é usado o modelo de transportes. Assim, em cada nível de algoritmo hierarquizado é usado a decomposição de Benders para separar o problema de planejamento em dois subproblemas: um subproblema de investimento e um subproblema de operação como é mostrado na figura 4.1(a). No processo de resolução os subproblemas trocam informação: o **mestre** informa ao escravo acerca das decisões de investimento (incremento na capacidade de transmissão) e o **escravo** informa ao mestre acerca das necessidades de transmissão através dos cortes de Benders. O mestre basicamente controla os aspectos econômicos do processo de planejamento; o escravo representa as equações de fluxo de potência com diferentes níveis de detalhe de acordo com o nível corrente de hierarquização. O escravo, com a alternativa de investimento fornecida pelo mestre, minimiza o uso dos geradores artificiais para satisfazer a demanda sem violações das restrições de operação. Se, num estágio do processo de solução, as decisões de investimento são tais que permite um ponto de operação sem a utilização dos geradores artificiais então foi obtida uma solução do problema nesse nível de hierarquização e o processo entra na seguinte fase. Em outro caso, se não é possível obter um ponto de operação sem recorrer aos geradores artificiais então, deve-se enviar informação acerca das necessidades de investimento ao mestre através dos cortes de Benders como é mostrado na figura 4.1(a). Estes cortes são incorporados ao mestre como restrições adicionais; assim eles jogam um rol crucial no processo de planejamento pois eles informam ao mestre acerca das necessidades adicionais de capacidade de transmissão.

No mestre existem dois tipos de restrições como é mostrado na figura 4.1(b). Existem as restrições nas variáveis de investimento (como o número máximo de circuitos a adicionar em cada ramo) e uma representação equivalente das restrições de operação (um conjunto reduzido de restrições que representam implicitamente as restrições críticas de operação). A representação equivalente das restrições de operação estão constituídos basicamente dos cortes de Benders que se encontram disponíveis em qualquer estágio do processo e também de outros tipos de informação simplificada acerca da operação da rede. Assim, é possível encontrar e incorporar no mestre outros tipos de restrições diferentes dos cortes de Benders que podem acelerar o processo de convergência do algoritmo. Esta parte do problema será analisada com maior detalhe na última seção deste capítulo. Portanto, nesta parte serão considerados exclusivamente os cortes de Benders para implementar o subproblema mestre. Entretanto, um aspecto importante que deve ser levado em conta é que quanto mais o mestre conhece acerca da operação da rede, melhores são as decisões de investimento e, portanto, mais rápido o processo de convergência ao ótimo global.

4.2.2 Algoritmo de Planejamento Hierarquizado

As duas fases do algoritmo hierarquizado são resumidas na seguinte forma:

Fase I: Modelo de Transportes - Adições Contínuas

Resolver o problema usando o modelo de transportes e relaxando a integralidade das variáveis de investimento. Assim, é relaxado o tipo de variável. Os subproblemas resultantes de operação e investimento são resolvidos alternadamente até atingir a convergência. Ambos subproblemas resultantes são problemas de programação linear (PL) e são resolvidos usando



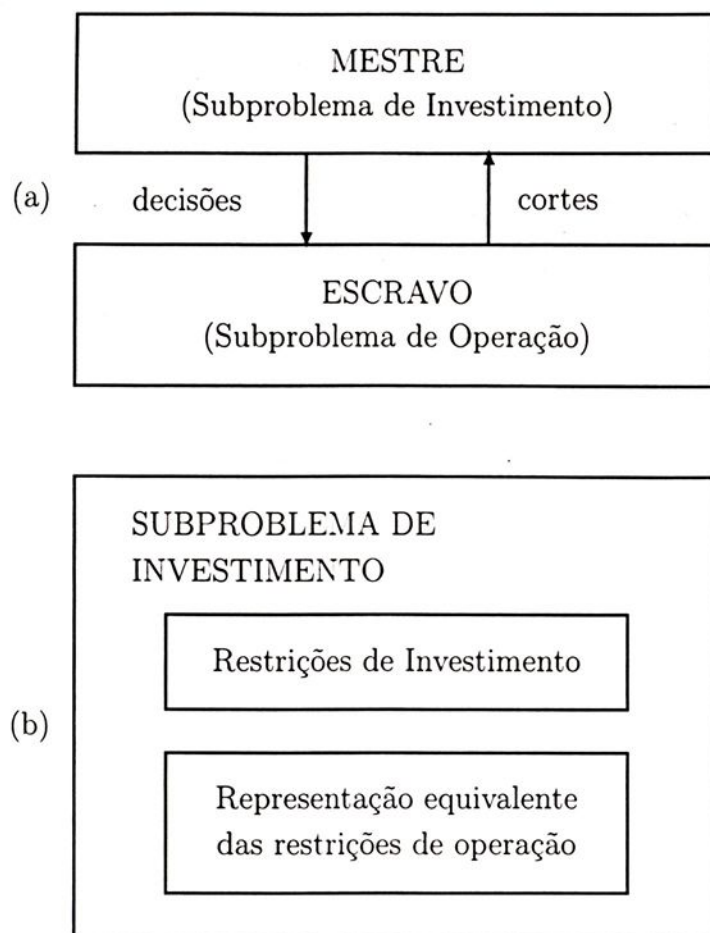


Figura 4.1: A Decomposição de Benders.

técnicas de PL. A solução ótima obtida, assim como os cortes de Benders gerados neste processo, são reutilizadas na seguinte fase.

Fase II: Modelo de Transportes - Adições Inteiras

Nesta fase são reintroduzidas a natureza inteira das variáveis de investimento. Nestas condições o mestre é um problema de programação linear inteira (PLI). O subproblema de operação é um PL e pode ser resolvido usando técnicas de PL mas o subproblema de investimento deve ser resolvido usando técnicas de programação inteira. Portanto, é usada a decomposição de Benders para resolver duas formas diferentes do problema de planejamento quando é usado o modelo de transportes, sendo que a primeira forma com as variáveis de investimento relaxadas é resolvida simplesmente para auxiliar na resolução do problema na fase II ajudando a diminuir o esforço computacional do processo integral.

O algoritmo hierarquizado para o modelo de transportes, apresentado passo a passo, assume a seguinte forma:

Fase I: Modelo de transportes com variáveis contínuas

1. Assumir uma proposta de investimento inicial: $n_{ij}^1 = 0$.

2. Subproblema de operação: Escravo.

Resolver o subproblema de operação (4.3) para a proposta de investimento corrente n_{ij}^k usando um algoritmo de PL. Se o corte de carga $w \leq \epsilon_w$ que é um valor de tolerância previamente especificada então termina a fase I e, deve-se passar à fase II. Em caso contrário ir ao passo 3.

3. Gerar o corte de Benders:

$$\sum |\pi_{ij}^k| \bar{f}_{ij} n_{ij} + \beta \geq w^k + \sum |\pi_{ij}^k| \bar{f}_{ij} n_{ij}^k \quad (4.6)$$

4. Subproblema de investimento: Mestre.

Resolver o subproblema de investimento (4.5) com as restrições de investimento relaxadas, isto é, $n_{ij} \geq 0$ e com os cortes de Benders já gerados. Assim, (4.5) é simplesmente um PL com um conjunto de restrições lineares (os cortes de Benders gerados). Voltar ao passo 2.

Fase II: Modelo de transportes com variáveis inteiras.

1. Subproblema de investimento: Mestre.

Resolver o subproblema de investimento (4.5) com os cortes de Benders gerados nas duas fases. Portanto, (4.5) é um PLI e deve ser resolvido usando um algoritmo de programação linear inteira. Com a proposta de investimento obtida ir ao passo 2.

2. Subproblema de operação: Escravo.

Resolver o subproblema de operação (4.3) para a proposta de investimento corrente n_{ij}^k usando um algoritmo de PL. Se o corte de carga $w \leq \epsilon_w$ que é um valor de tolerância previamente especificada então termina o processo integral pois foi encontrada a configuração ótima. Em caso contrário ir ao passo 3.

3. Gerar o corte de Benders:

$$\sum |\pi_{ij}^k| \bar{f}_{ij} n_{ij} + \beta \geq w^k + \sum |\pi_{ij}^k| \bar{f}_{ij} n_{ij}^k \quad (4.7)$$

e voltar ao passo 1.



Capítulo 5

Algoritmos de Decomposição Matemática Usando o Modelo DC

Neste capítulo se apresenta a técnica de decomposição de Benders para resolver o problema de planejamento de sistemas de transmissão quando é usado o modelo DC. Neste caso o problema de otimização resultante é um PNLIM, amplamente discutido em capítulos anteriores, e cujo modelo modificado assume a seguinte forma:

$$\begin{aligned} \min v &= \sum_{(i,j)} c_{ij} n_{ij} + \alpha \sum_{k \in \Gamma} r_k & (5.1) \\ \text{s.a.} & \\ & S f + g = d \\ & f_{ij} - (\gamma_{ij}^o + x_{ij})(\theta_i - \theta_j) = 0 \\ & |f_{ij}| \leq (x_{ij} + \gamma_{ij}^o) \bar{\phi}_{ij} \\ & 0 \leq r \leq d \\ & 0 \leq g \leq \bar{g} \\ & 0 \leq n_{ij} \leq \bar{n}_{ij} \\ & n_{ij} \text{ inteiro} \\ & x_{ij} \text{ discreto} \\ & f_{ij} \text{ irrestrito} \\ & \theta_j \text{ irrestrito } \forall j \in \Omega_3 \end{aligned}$$

ou a forma equivalente:

$$\min v = \sum_{(i,j)} c_{ij} n_{ij} + \alpha \sum_{k \in \Gamma} r_k \quad (5.2)$$

s.a.

$$B\theta + g = d$$

$$|\theta_i - \theta_j| \leq \bar{\phi}_{ij} \quad \forall (i, j) \in \Omega_1$$

$$x_{ij} |\theta_i - \theta_j| \leq x_{ij} \bar{\phi}_{ij} \quad \forall (i, j) \in \Omega_2$$

$$0 \leq g \leq \bar{g}$$

$$0 \leq n_{ij} \leq \bar{n}_{ij}$$

$$n_{ij} \text{ inteiro}$$

$$x_{ij} \text{ discreto}$$

$$\theta_j \text{ irrestrito}$$

Das várias técnicas de otimização clássicas existentes para otimizar problemas de programação não linear inteiro mistos, a decomposição matemática de Benders é praticamente a única usada para resolver o problema de planejamento de sistemas de transmissão. O motivo principal é a natureza do problema: o problema de planejamento de sistemas de transmissão não apresenta uma estrutura especial mas a decomposição de Benders permite separar o problema complexo em dois subproblemas com características especiais. Os subproblemas são um subproblema de investimento com as variáveis de investimento inteiras e um subproblema de operação com as variáveis de operação. Assim, o subproblema de investimento, também chamado de problema mestre, é um problema de programação linear inteiro e o subproblema de operação, também chamado de problema escravo, é um problema de de programação linear.

No capítulo anterior já foi apresentada a decomposição de Benders para o problema de planejamento de sistemas de transmissão quando é usado o modelo de transportes. Portanto, são válidas todas as observações realizadas no capítulo anterior em relação à utilização da decomposição de Benders ao problema de planejamento de sistemas de transmissão. Entretanto, existe uma grande diferença entre usar a decomposição de Benders para o modelo de transportes e o modelo DC. Quando é usado o modelo de transportes, pode-se demonstrar teoricamente que um algoritmo baseado na decomposição de Benders converge na solução ótima global. Por outro lado, quando é usado o modelo DC não existe garantia de convergência para o ótimo global de um algoritmo baseado na decomposição de Benders. Assim, usando o modelo DC um algoritmo baseado na decomposição de Benders pode convergir em soluções ótimas locais. Esta característica é uma consequência do caráter não convexo do problema de planejamento quando é usado o modelo DC e, nessas condições, a teoria existente permite gerar apenas cortes de Benders aproximados (e lineares) e que eventualmente podem eliminar a solução ótima da região factível definido pelos cortes de Benders já gerados produzindo a convergência para uma solução ótima local. Existem várias maneiras de contornar em parte este problema de convergência para soluções ótimas locais produzindo vários tipos de algoritmos baseados na decomposição de Benders com qualidades de convergência e esforços computacionais diferentes.

Uma proposta inicial pode ser a utilização de um algoritmo puro baseado na decomposição de Benders como apresentado em [11]. Nesta proposta é usado o modelo DC e as únicas restrições do subproblema mestre são os cortes de Benders gerados em cada passo do processo iterativo



além das restrições próprias de investimento, isto é, trata-se da aplicação da decomposição de Benders convencional. Esta proposta apresenta três problemas quando são usados em sistemas reais. O primeiro problema é o elevado número de cortes de Benders que devem ser gerados para atingir a convergência do processo o que eleva o esforço computacional do processo. O segundo problema é que em cada passo do processo deve ser resolvido o subproblema mestre usando um algoritmo de PLI. Um algoritmo de PLI (Branch and Bound ou enumeração implícita de Balas) normalmente precisa de um esforço computacional elevado e como o processo gera muitos cortes de Benders e em cada caso se deve resolver novamente o problema mestre, é evidente que o esforço computacional se torna proibitivo. O terceiro problema é que quando o tamanho do sistema cresce, mais facilmente, o processo converge para soluções ótimas locais.

Várias propostas foram apresentadas para contornar o problema de elevado esforço computacional e a convergência a soluções ótimas locais [10, 11, 14]. As propostas visam fundamentalmente diminuir os cortes de Benders gerados, diminuir o número de chamadas ao algoritmo de PLI e melhorar a convergência do processo. Neste capítulo é apresentado em detalhe a proposta apresentada em [14] chamado de algoritmo de Benders hierarquizado para o planejamento de sistemas de transmissão usando o modelo DC. Esta proposta é uma generalização do algoritmo de Benders hierarquizado para o modelo de transportes apresentado no capítulo anterior.

5.1 A Decomposição Matemática de Benders

Como já foi mencionado a decomposição matemática de Benders permite decompor o problema complexo de planejamento de sistemas de transmissão em dois subproblemas: um subproblema de investimento também chamado de mestre e um subproblema de operação também chamado escravo. Os dois subproblemas, mestre e escravo, são resolvidos de maneira iterativa até que seja encontrada a solução ótima global. O subproblema de operação é um problema de programação linear e o subproblema de investimento é um problema de programação inteira. Portanto, a complexidade do problema foi transferida para a resolução do subproblema mestre em forma iterativa. O subproblema mestre, um problema de programação inteira, pode ser resolvido usando vários métodos tais como um algoritmo de Branch and Bound e um algoritmo de enumeração implícita de Balas especializado.

Em [14] é apresentado o algoritmo de planejamento hierarquizado que resolve o problema de planejamento de sistemas de transmissão usando o modelo DC. Nesse algoritmo de três fases se resolve versões relaxadas do modelo DC (modelo de transportes e modelo híbrido) nas duas primeiras fases e ainda relaxando a integralidade das variáveis de investimento, isto é, os subproblemas nas duas fases iniciais são simplesmente problemas de programação linear (PL).

É possível usar a decomposição de Benders em cada nível do algoritmo hierarquizado. No primeiro nível simplesmente se resolve um problema de programação linear usando decomposição de Benders para o modelo de transportes com variáveis de investimento contínuas. No segundo nível se usa a decomposição de Benders mas para resolver um problema de programação não linear usando a decomposição de Benders, isto é, o modelo híbrido com variáveis de investimento contínuas. Finalmente na terceira fase se resolve o problema PNLIM, isto é, o modelo DC com variáveis de investimento inteiras. Pode-se demonstrar que um algoritmo desse tipo tem maiores chances de encontrar a solução ótima de um problema de planejamento



de sistemas de transmissão quando é usado o modelo DC. Assim, em cada nível do algoritmo hierarquizado é usado a decomposição de Benders para separar o problema de planejamento em dois subproblemas: um subproblema de investimento e um subproblema de operação como é mostrado na figura 5.1(a). No processo de resolução os subproblemas trocam informação: o **mestre** informa ao escravo acerca das decisões de investimento (incremento na capacidade de transmissão) e o **escravo** informa ao mestre acerca das necessidades de transmissão através dos cortes de Benders. O mestre basicamente controla os aspectos econômicos do processo de planejamento; o escravo representa as equações de fluxo de potência com diferentes níveis de detalhe de acordo com o nível corrente de hierarquização. O escravo, com a alternativa de investimento fornecida pelo mestre, minimiza o uso dos geradores artificiais para satisfazer a demanda sem violações das restrições de operação. Se, num estágio do processo de solução, as decisões de investimento são tais que permite um ponto de operação sem a utilização dos geradores artificiais então foi obtida uma solução do problema nesse nível de hierarquização e o processo entra na seguinte fase. Em outro caso, se não é possível obter um ponto de operação sem recorrer aos geradores artificiais então, deve-se enviar informação acerca das necessidades de investimento ao mestre através dos cortes de Benders como é mostrado na figura 5.1(a). Estes cortes são incorporados ao mestre como restrições adicionais; assim eles jogam um rol crucial no processo de planejamento pois eles informam ao mestre acerca das necessidades adicionais de capacidade de transmissão.

No mestre existem dois tipos de restrições como é mostrado na figura 5.1(b). Existem as restrições nas variáveis de investimento (como o número máximo de circuitos a adicionar em cada ramo) e uma representação equivalente das restrições de operação (um conjunto reduzido de restrições que representam implicitamente as restrições críticas de operação). A representação equivalente das restrições de operação estão constituídos basicamente dos cortes de Benders que se encontram disponíveis em qualquer estágio do processo e também de outros tipos de informação simplificada acerca da operação da rede. Assim, é possível encontrar e incorporar no mestre outros tipos de restrições diferentes dos cortes de Benders que podem acelerar o processo de convergência do algoritmo. Portanto, nesta parte serão considerados exclusivamente os cortes de Benders para implementar o subproblema mestre. Entretanto, um aspecto importante que deve ser levado em conta é que quanto mais o mestre conhece acerca da operação da rede, melhores são as decisões de investimento e, portanto, mais rápido o processo de convergência ao ótimo global.

5.1.1 A Decomposição de Benders Generalizada

No algoritmo hierarquizado de Benders são usados os três modelos matemáticos, transportes, híbrido e DC. Portanto, deve-se desenvolver a forma matemática dos subproblemas que aparecem ao usar a decomposição de Benders para cada tipo de modelo matemático. A forma dos subproblemas mestre e escravo assim como a forma do corte de Benders para cada modelo matemático pode ser obtido usando a técnica de decomposição de Benders generalizada que está amplamente desenvolvido em [11, 33].

Para os modelos matemáticos apresentados no capítulo 2, isto é, para os modelos de transportes, híbrido e DC modificados, os subproblemas mestre e escravo assumem a seguinte forma [14, 33]:



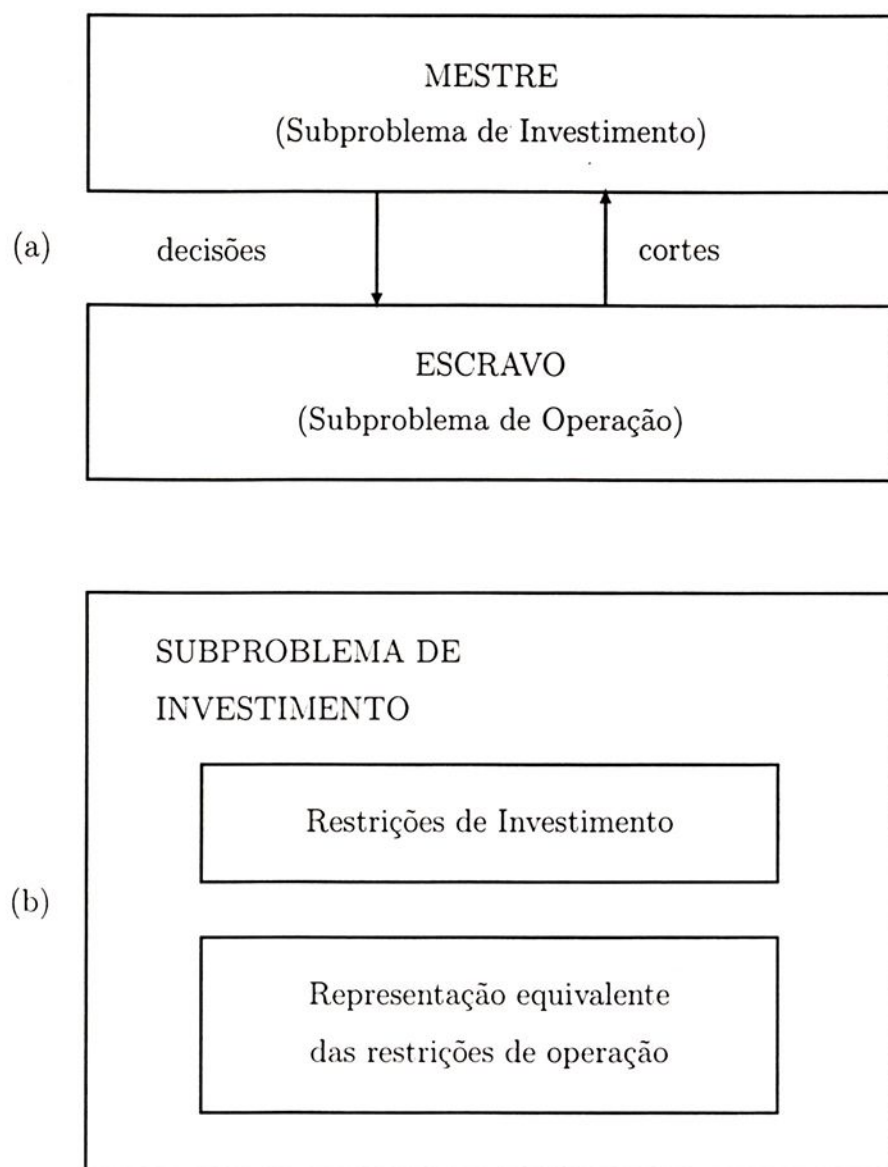


Figura 5.1: A Decomposição de Benders.

1. Modelo de transportes:

- Subproblema de operação (escravo):

$$\begin{aligned}
 \min w &= \alpha \sum r_i & (5.3) \\
 \text{s.a.} & \\
 & S f + g + r = d \\
 & |f_{ij}| \leq \bar{f}_{ij}(n_{ij}^k + n_{ij}^o) \\
 & 0 \leq g \leq \bar{g} \\
 & 0 \leq r \leq d \\
 & f_{ij} \text{ irrestrito}
 \end{aligned}$$

- Subproblema de investimento (mestre):

$$\begin{aligned}
 \min v &= \sum_{(i,j)} c_{ij} n_{ij} + \beta & (5.4) \\
 \text{s.a.} & \\
 & w^k + \sum \sigma_{ij}^k (n_{ij} - n_{ij}^k) \leq \beta \\
 & \beta \geq 0 \\
 & 0 \leq n_{ij} \leq \bar{n}_{ij} \\
 & n_{ij} \text{ inteiro}
 \end{aligned}$$

em que o parâmetro σ_{ij}^k é igual a:

$$\sigma_{ij}^k = -|\pi_{ij}| \bar{f}_{ij} \quad (5.5)$$

Portanto, o subproblema mestre assume a seguinte forma:

$$\begin{aligned}
 \min v &= \sum_{(i,j)} c_{ij} n_{ij} + \beta & (5.6) \\
 \text{s.a.} & \\
 & \sum |\pi_{ij}^k| \bar{f}_{ij} n_{ij} + \beta \geq w^k + \sum |\pi_{ij}^k| \bar{f}_{ij} n_{ij}^k \\
 & \beta \geq 0 \\
 & 0 \leq n_{ij} \leq \bar{n}_{ij} \\
 & n_{ij} \text{ inteiro}
 \end{aligned}$$

2. Modelo DC:

- Subproblema de operação (escravo):

$$\begin{aligned}
 \min w &= \alpha \sum r_i & (5.7) \\
 \text{s.a.} & \\
 & B\theta + g + r = d \\
 & |\theta_i - \theta_j| \leq \bar{\phi}_{ij} \quad \forall (i, j) \in \Omega_1 \\
 & x_{ij}^k |\theta_i - \theta_j| \leq x_{ij}^k \bar{\phi}_{ij} \quad \forall (i, j) \in \Omega_2 \\
 & 0 \leq g \leq \bar{g} \\
 & 0 \leq r \leq d \\
 & \theta_j \text{ irrestrito}
 \end{aligned}$$

- Subproblema de investimento (mestre):

$$\begin{aligned}
 \min v &= \sum_{(i,j)} c_{ij} n_{ij} + \beta & (5.8) \\
 \text{s.a.} & \\
 & w^k + \sum \sigma_{ij}^k (n_{ij} - n_{ij}^k) \leq \beta \\
 & \beta \geq 0 \\
 & 0 \leq n_{ij} \leq \bar{n}_{ij} \\
 & n_{ij} \text{ inteiro}
 \end{aligned}$$

em que o parâmetro σ_{ij}^k é igual a:

$$\sigma_{ij}^k = (\pi_i - \pi_j)(\theta_i - \theta_j)\gamma_{ij} \quad (5.9)$$

3. Modelo Híbrido:

- Subproblema de operação (escravo):

$$\begin{aligned}
 \min w &= \alpha \sum r_i & (5.10) \\
 \text{s.a.} & \\
 & S_2 f_2 + B_1 \theta_1 + g + r = d \\
 & |\theta_i - \theta_j| \leq \bar{\phi}_{ij} \quad \forall (i, j) \in \text{omega}_1 \\
 & |f_{ij}| \leq \bar{f}_{ij} n_{ij} \quad \forall (i, j) \in \Omega_2 \\
 & 0 \leq g \leq \bar{g} \\
 & 0 \leq r \leq d \\
 & \theta_j \text{ irrestrito}
 \end{aligned}$$

- Subproblema de investimento (mestre):

$$\begin{aligned}
 \min v &= \sum_{(i,j)} c_{ij} n_{ij} + \beta & (5.11) \\
 \text{s.a.} & \\
 & w^k + \sum \sigma_{ij}^k (n_{ij} - n_{ij}^k) \leq \beta \\
 & \beta \geq 0 \\
 & 0 \leq n_{ij} \leq \bar{n}_{ij} \\
 & n_{ij} \text{ inteiro}
 \end{aligned}$$

em que o parâmetro σ_{ij}^k é igual a:

$$\sigma_{ij}^k = -|\pi_{ij}| \bar{f}_{ij} \quad (5.12)$$

para os (i, j) correspondentes a novos caminhos. Para os caminhos em que existem circuitos na configuração base é usada a seguinte relação:

$$\sigma_{ij}^k = (\pi_i - \pi_j)(\theta_i - \theta_j) \gamma_{ij} \quad (5.13)$$

em que os π_j e π_{ij} correspondem às mesmas relações dos modelos de transportes e DC.

Assim, em princípio, pode-se usar a decomposição matemática de Benders para resolver o problema de planejamento de sistemas de transmissão usando cada um dos três modelos matemáticos de maneira isolada. Entretanto, no algoritmo de planejamento hierarquizado é encontrada uma solução do modelo DC e os outros dois modelos são usados apenas para ajudar ao algoritmo a encontrar a solução ótima global e diminuir o esforço computacional do processo.

5.2 Planejamento Ótimo Hierarquizado Usando a Decomposição de Benders e Relaxação

Uma estratégia para tentar encontrar a configuração ótima do problema de planejamento de sistemas de transmissão quando é usado o modelo DC é implementar o chamado planejamento hierarquizado usando a decomposição de Benders como foi apresentada anteriormente. Como o problema é um PNLIM, nem sempre é possível encontrar a solução ótima deste problema usando a decomposição de Benders.

O nome de planejamento hierarquizado aparece devido a que no processo de resolução são resolvidas nas fases iniciais versões relaxadas do problema original. Neste caso, inicialmente se resolve o problema de planejamento usando o modelo de transportes mas relaxando a integralidade das variáveis de investimento n_{ij} , isto é, usando a decomposição de Benders se resolve um problema de programação linear. Aparentemente, este processo é irrelevante mas a importância em resolver este problema relaxado é que os cortes de Benders gerados nesta fase são restrições válidas e importantes do problema original ou não relaxado (com variáveis inteiras). Portanto todas as restrições, cortes de Benders, geradas ao resolver o problema relaxado

podem ser usados para resolver de maneira mais rápida o problema original. Na seguinte fase se resolve o problema usando o modelo híbrido também relaxando a integralidade das variáveis de investimento. Nesta fase, o mestre incorpora os cortes de Benders gerados na fase anterior. Finalmente, na última fase se resolve o problema original, isto é, o modelo DC com as variáveis de investimento inteiras. Assim, o algoritmo está constituído das seguintes fases:

1. Fase I: Resolução do modelo de transportes relaxado:

Resolve-se o modelo de transportes com as variáveis de investimento relaxadas e usando a decomposição de Benders. Neste caso, os subproblemas de investimento e de operação obtidos usando a decomposição de Benders são problemas de programação linear. Portanto, esses subproblemas podem ser rapidamente resolvidos usando algoritmos especializados de PL. Os cortes de Benders obtidos nesta fase são incorporados no subproblema de investimento na seguinte fase.

2. Fase II: Resolução do modelo híbrido relaxado:

Resolve-se o modelo híbrido com as variáveis de investimento relaxadas e usando a decomposição de Benders. Neste caso, também os subproblemas de investimento e de operação obtidos usando a decomposição de Benders são problemas de programação linear. Portanto, esses subproblemas podem ser rapidamente resolvidos usando algoritmos especializados de PL. Na resolução do mestre nesta fase são usados os cortes de Benders gerados na fase anterior. Os cortes de Benders obtidos nesta fase assim como os gerados na fase anterior são incorporados no subproblema de investimento na seguinte fase.

3. Fase III: Resolução do problema original:

Resolve-se o problema original com as variáveis de investimento inteiras e usando a decomposição de Benders. Neste caso, o subproblema de operação é um problema de programação linear e o subproblema de investimento é um problema de programação linear inteira (PLI). Os cortes de Benders gerados nas fases anteriores são incorporados ao subproblema de investimento. O subproblema de operação pode ser rapidamente resolvido por eficientes algoritmos de PL. Entretanto, a resolução do subproblema mestre exige de um esforço computacional muito maior. Nesta fase o subproblema de investimento deve ser resolvido usando um algoritmo especializado de PLI. Para o problema de planejamento de sistemas de transmissão foram usados dois algoritmos [15]: um algoritmo de Branch and Bound semelhante ao usado no capítulo anterior e um algoritmo de enumeração implícita de Balas especializado.

A estratégia de usar um algoritmo hierarquizado foi originalmente apresentada em [14] para resolver o problema de planejamento de sistemas de transmissão quando é usado o modelo DC. O algoritmo hierarquizado apresenta vários atrativos quando é usado para resolver o modelo DC. Além dos atrativos mencionados para o modelo de transportes no capítulo anterior, quando é usado para o modelo DC apresenta o atrativo de que ajuda de maneira significativa para que a metodologia de decomposição de Benders encontre mais frequentemente as soluções ótimas globais.



5.2.1 Algoritmo de Planejamento Hierarquizado

O algoritmo hierarquizado para o modelo DC, apresentado passo a passo, assume a seguinte forma:

Fase I: Modelo de transportes com variáveis contínuas

1. Assumir uma proposta de investimento inicial: $n_{ij}^1 = 0$.

2. Subproblema de operação: Escravo.

Resolver o subproblema de operação (5.3) para a proposta de investimento corrente n_{ij}^k usando um algoritmo de PL. Se o corte de carga $w \leq \epsilon_w$ que é um valor de tolerância previamente especificada então termina a fase I e, deve-se passar à fase II. Em caso contrário ir ao passo 3.

3. Gerar o corte de Benders:

$$\sum |\pi_{ij}^k| \bar{f}_{ij} n_{ij} + \beta \geq w^k + \sum |\pi_{ij}^k| \bar{f}_{ij} n_{ij}^k \quad (5.14)$$

4. Subproblema de investimento: Mestre.

Resolver o subproblema de investimento (5.6) com as restrições de investimento relaxadas, isto é, $n_{ij} \geq 0$ e com os cortes de Benders já gerados. Assim, (5.6) é simplesmente um PL com um conjunto de restrições lineares (os cortes de Benders gerados). Voltar ao passo 2.

Fase II: Modelo híbrido com variáveis contínuas.

1. Subproblema de operação: Escravo.

Resolver o subproblema de operação (5.10) para a proposta de investimento corrente n_{ij}^k usando um algoritmo de PL. Se o corte de carga $w \leq \epsilon_w$ que é um valor de tolerância previamente especificada então termina a fase II e, deve-se passar à fase III. Em caso contrário ir ao passo 2.

2. Gerar o corte de Benders:

$$w^k + \sum \sigma_{ij}^k (n_{ij} - n_{ij}^k) \leq \beta$$

usando as relações (5.12) e (5.13).

3. Subproblema de investimento: Mestre.

Resolver o subproblema de investimento (5.11) com as restrições de investimento relaxadas, isto é, $n_{ij} \geq 0$ e com os cortes de Benders já gerados nas duas fases. Assim, (5.11) é simplesmente um PL com um conjunto de restrições lineares (os cortes de Benders gerados). Voltar ao passo 1.



Fase III: Modelo DC com variáveis inteiras.

1. Subproblema de investimento: Mestre.

Resolver o subproblema de investimento (5.8) com os cortes de Benders gerados nas três fases. Portanto, (5.7) é um PLI e deve ser resolvido usando um algoritmo de programação linear inteira. Com a proposta de investimento obtida ir ao passo 2.

2. Subproblema de operação: Escravo.

Resolver o subproblema de operação (5.7) para a proposta de investimento corrente n_{ij}^k usando um algoritmo de PL. Se o corte de carga $w \leq \epsilon_w$ que é um valor de tolerância previamente especificada então termina o processo integral pois foi encontrada a configuração ótima. Em caso contrário ir ao passo 3.

3. Gerar o corte de Benders:

$$w^k + \sum \sigma_{ij}^k (n_{ij} - n_{ij}^k) \leq \beta$$

usando a relações (5.9) e voltar ao passo 1.



Capítulo 6

Planejamento de Sistemas de Transmissão Usando “Simulated Annealing”

6.1 Introdução

Neste capítulo se inicia a apresentação dos chamados algoritmos combinatoriais usados no problema de planejamento de sistemas de transmissão a longo prazo. Assim, neste capítulo é analisado o algoritmo de simulated annealing apresentado em [16, 17, 34]. Em [16] é apresentado um algoritmo de simulated annealing sequencial em [17] é apresentado um algoritmo de simulated annealing paralelo. Testes experimentais mostram que os algoritmos combinatoriais (simulated annealing, o algoritmo genético e busca tabu) encontram soluções ótimas ou quase-ótimas de muitos tipos de problemas complexos do campo da pesquisa operacional. Para o problema de planejamento de sistemas de transmissão, os resultados apresentados em [16, 17, 34] mostram que o algoritmo encontra frequentemente as configurações ótimas de sistemas de pequeno e de medio porte e soluções quase-ótimas de sistemas de grande porte.

Annealing é uma das técnicas usadas pelos físicos na construção de cristais perfeitos. Nessa técnica um material é esquentado até uma temperatura elevada e depois esfriada lentamente mantendo durante o processo o chamado quase equilíbrio termodinâmico. O processo pára quando o material atinge seu estado de energia mínima na qual se transforma num cristal perfeito. Assim, o algoritmo de simulated annealing tenta simular um processo equivalente para encontrar a configuração ótima de um problema complexo.

6.2 Simulated Annealing: Teoria

6.2.1 O Algoritmo de Metrópolis

A metodologia de “simulated annealing” (SA) foi proposta no início da década de 80, como uma nova ferramenta a ser usada na solução de problemas grandes e complexos. SA surgiu do campo da termodinâmica como consequência de uma comparação entre os problemas formulados neste campo com os do campo da pesquisa operacional. SA é uma metodologia



simples e de grande potencialidade para ser aplicada a uma grande variedade de problemas.

A idéia original que deu lugar ao algoritmo SA é o chamado *algoritmo de Metrópolis*, o que por sua vez está baseado no *método de Monte-Carlo*, com o qual se estudam as propriedades de equilíbrio na análise do comportamento microscópico dos corpos.

O algoritmo de Metrópolis gera uma seqüência de estados de um sólido, ou seja: dado um sólido num estado i e com energia E_i , gera-se o estado seguinte j mediante a aplicação de um mecanismo de transição através de um pequeno distúrbio. A energia do próximo estado é E_j ; se a diferença de energia $E_j - E_i$ é menor ou igual a zero, o estado j é aceito. Se a diferença de energia é maior que zero, o estado j é aceito com certa probabilidade, a qual é dada como:

$$e^{\left\{\frac{E_i - E_j}{k_b T}\right\}}$$

em que T denota a temperatura e k_b uma constante física conhecida como *constante de Boltzmann*. A regra de aceitação descrita é chamada *critério de Metrópolis* e o algoritmo como *algoritmo de Metrópolis*.

Se a diminuição da temperatura é feita de maneira paulatina, o sólido poderá alcançar o estado de equilíbrio em cada nível de temperatura. No algoritmo de Metrópolis isto se consegue após gerar um grande número de transições num dado nível de temperatura. Para cada valor de temperatura T , o sólido deve atingir um equilíbrio térmico, caracterizado pela probabilidade de estar num estado i com energia E_i dado pela distribuição de Boltzmann:

$$P_T\{X = i\} = \frac{1}{Z(T)} e^{\left(\frac{-E_i}{k_b T}\right)}$$

em que: X denota uma variável estocástica do estado atual do sólido, $Z(T) = \sum_j e^{\left(\frac{-E_j}{k_b T}\right)}$ é um fator de normalização, conhecido como função partição; k_b é a constante de Boltzmann; e $e^{\left(\frac{-E_i}{k_b T}\right)}$ é conhecido como fator de Boltzmann.

6.2.2 Algoritmo de “Simulated Annealing” (SA)

Na figura 6.1 é mostrada a estrutura básica do algoritmo SA. Neste algoritmo, aplica-se uma ação combinada do mecanismo de geração de alternativas e do critério de aceitação. T_k denota o valor do parâmetro de controle (temperatura) e N_k o número de alternativas geradas na k -ésima iteração do algoritmo. Inicialmente quando T é grande, grandes deteriorações da função objetivo podem ser aceitas; quando T decresce, somente pequenas deteriorações são aceitas e finalmente quando T tende a zero, nenhuma deterioração é aceita. Esta característica faz com que o algoritmo SA seja diferente em relação aos algoritmos de busca local como os algoritmos heurísticos construtivos.

A partir do estado i com custo $f(i)$ gera-se o estado j com custo $f(j)$. O critério de aceitação determina se este novo estado é aceito; para isto calcula-se a seguinte probabilidade:



$$P_T\{aceita_j\} = \begin{cases} 1 & \text{se } \leftrightarrow f(j) \leq f(i) \\ e^{\frac{f(i)-f(j)}{T}} & \text{se } \leftrightarrow f(j) > f(i) \end{cases}$$

6.2.3 Analogia Física e Apresentação da Metaheurística

Fazendo uma analogia entre os parâmetros que participam do processo de annealing e os métodos de otimização se tem:

TERMODINÂMICA

OTIMIZAÇÃO

Configuração

Solução factível

Configuração fundamental

Solução ótima

Energia da configuração

Custo da solução

Temperatura

Parâmetro de ajuste

A temperatura não corresponde a um significado real no campo da otimização; assim deve ser adequadamente calculado e atualizado durante o processo. Em otimização uma transição é aceita se a função objetivo diminui; em caso contrário a transição é aceita de maneira probabilística sendo assim possível uma degradação da função objetivo. Esta última característica permite a SA sair de soluções ótimas locais. Na medida em que evolui o processo, diminui a temperatura e a probabilidade de aceitar uma solução de pior qualidade. Assim, SA parte de uma temperatura elevada, permitindo que o processo passe através de configurações de diversa qualidade e, posteriormente, ao diminuir a temperatura diminui também a possibilidade de aceitar soluções piores.

6.2.4 Programa de Esfriamento

O programa de esfriamento controla o processo SA até atingir a convergência. A qualidade de um algoritmo SA depende significativamente da qualidade do correspondente programa de esfriamento. Um programa de esfriamento é definido pela determinação ou escolha de cada um dos seguintes parâmetros:

- Temperatura inicial.
- Taxa de esfriamento ou de diminuição de temperatura.
- Número de tentativas de transição em cada nível de temperatura.
- Critério de parada.



Algoritmo “Simulated Annealing”;

inicia

inicializar (T_0, N_0) ;

$k := 0$;

configuração inicial S_i

repetir o procedimento

do $L := 1$ **to** N_k

gere $(S_j$ **de** $S_i)$;

se $f(j) \leq f(i)$ **faça** $S_i := S_j$;

caso contrário

if $\exp\left(\frac{f(i)-f(j)}{T_k}\right) > \text{random}[0, 1]$ **faça** $S_i := S_j$;

end do;

$k := k + 1$;

Cálculo do comprimento de (N_k) ;

Cálculo do parâmetro de controle (T_k) ;

Critério de parada

termina;

Fonte: [Aarts & Korst (1989)]

Figura 6.1: Descrição do algoritmo “annealing”

Existes muitas formulações matemáticas que justificam a escolha desses parâmetros de controle do programa de esfriamento que podem ser analisados para uma análise mais rigorosa [20, 21]. Entretanto, todos os argumentos matemáticos estão baseados no suposto de que o processo deve atingir o quase-equilíbrio em cada nível de temperatura o que teoricamente leva a uma exigência de esforço computacional muito elevado para a maioria dos problemas de aplicações reais. Na prática, esses parâmetros são calibrados empiricamente para cada tipo de problema e para esforços computacionais não proibitivos.

6.3 SA Aplicado ao Problema de Planejamento

SA é uma técnica de otimização combinatorial que foi desenvolvida analisando o processo de ‘annealing’ em sólidos. ‘annealing’ é um dos processos de construção de cristais perfeitos. No processo de ‘annealing’ o material sólido é esquentado a temperaturas elevadas onde esse material assume uma estrutura atômica altamente desordenada. Depois o material é resfriado lentamente, mantendo o quase-equilíbrio térmico, até atingir o estado de mínima energia onde o material assume a estrutura de um cristal perfeito. Este processo de ‘annealing’, desenvolvido pelos físicos para a construção de cristais perfeitos, foi usado para desenvolver um algoritmo com capacidade de resolver problemas do campo da pesquisa operacional. A grande vantagem destes métodos combinatoriais é que eles tem capacidade de resolver problemas complexos e de grande porte, sem as restrições existentes nos métodos de otimização clássicos como exigências de convexidade, diferenciabilidade, etc.

SA foi usado com sucesso em muitas áreas da pesquisa operacional e em algumas aplicações em engenharia elétrica. Tem a característica de que apresenta soluções ótimas ou quase-ótimas, mas a um esforço computacional elevado. O *programa de resfriamento* controla esta qualidade da solução e o esforço computacional. Teoricamente, SA converge à solução ótima de maneira assintótica com probabilidade igual a 1.

Um problema de otimização combinatorial que pode ser resolvido usando SA pode ser formulado como um par (G, v) onde G representa um conjunto finito (embora muito grande) de configurações e v é o custo associado a cada configuração. Assim, resolver um problema combinatorial deste tipo consiste em encontrar a configuração de custo mínimo entre todas as configurações possíveis. Assim, SA parte de uma configuração inicial e depois gera um conjunto de configurações candidatas controladas pelo programa de resfriamento. Estas configurações candidatas são aceitas se apresentam um custo menor que a configuração corrente, mas elas também podem ser aceitas se o custo for maior que a configuração corrente de acordo com um critério probabilístico. Esta última característica, aceitar configurações com custo mais elevado, permite ao SA sair de eventuais ótimos locais. Este processo é controlado pelo programa de esfriamento.

A aplicação de um algoritmo de SA para resolver um problema de otimização combinatorial precisa da especificação de três aspectos significativamente distintos e independentes:

1. Uma adequada representação do problema.
2. Um mecanismo de transição.



3. Um programa de esfriamento ("cooling schedule").

Para cada tipo de problema, deve-se definir claramente cada um destes três aspectos e geralmente existem várias maneiras de se definir ou caracterizar cada um destes itens, o que leva a algoritmos de **SA** substancialmente diferentes. A seguir são discutidos, separadamente, cada um destes aspectos do algoritmo de **SA** para o problema de planejamento de sistemas de transmissão.

6.3.1 A Representação do Problema

O problema de planejamento de sistemas de transmissão usando o modelo DC modificado assume a seguinte forma:

$$\begin{aligned} \min \quad & v = \sum_{ij} c_{ij} n_{ij} + \sum_i \alpha_i r_i \\ \text{s.a.} \quad & B(x^k + \gamma^o)\theta + g + r = d \\ & (x_{ij}^k + \gamma_{ij}^o) |\theta_i - \theta_j| \leq (x_{ij}^k + \gamma_{ij}^o) \bar{\phi}_{ij} \\ & 0 \leq g \leq \bar{g} \\ & 0 \leq r \leq d \\ & n_{ij} \text{ inteiro} \end{aligned} \tag{6.1}$$

A representação do problema deve caracterizar corretamente o espaço de configurações possíveis e uma forma adequada de determinar a função de custo. Na formulação (6.1), ambos aspectos estão claramente definidos: a factibilidade de uma configuração é determinada pelas restrições e a função de custo está constituída por dois termos diferentes. A primeira parte corresponde aos custos dos circuitos adicionados e a segunda parte corresponde a uma penalidade devido ao corte de carga. Na verdade, a formulação (6.1) já é uma modificação do problema de planejamento que na sua formulação original somente leva em conta os custos dos circuitos adicionados na função objetivo. Entretanto, nessa formulação original seria muito difícil e pouco eficiente implementar um algoritmo de **SA** pois essa formulação apresenta três grandes limitações:

1. Precisa de uma configuração inicial factível para iniciar o processo, o que não é fácil encontrar em sistemas de grande porte.
2. A partir da configuração corrente, deve-se passar a outra configuração factível, isto é, sem corte de carga, o que também é difícil de encontrar.
3. Esse mecanismo de transição, passar de uma configuração sem corte de carga a outra configuração sem corte de carga, limitaria a capacidade do algoritmo de **SA** de sair de um ótimo local. Assim, representar a função objetivo com penalidade devido aos cortes de carga ou sem ela, representam duas alternativas diferentes de implementar algoritmos de **SA** para o problema de planejamento.

Em [16], usa-se a formulação (6.1) com a função objetivo com o termo correspondente a penalidade devido ao corte de carga. O motivo da escolha é que esta formulação não apresenta os problemas anteriormente mencionados, a mesma apresenta uma grande flexibilidade para passar de uma tentativa de configuração para outra. Nesta formulação, qualquer configuração candidata ou proposta de adição de circuitos, n_{ij}^k , é “factível” para a formulação (6.1), isto é, as configurações que apresentam e não apresentam corte de carga. Assim, o espaço das configurações factíveis foi acrescido para facilitar o processo de transição. Por outro lado, o processo deve priorizar as configurações e regiões factíveis do problema original, isto é, das regiões sem corte de carga. Este objetivo é atingido de duas maneiras complementares:

1. Implicitamente, pois as configurações com corte de carga são fortemente penalizadas produzindo uma função de custo elevada o que levará à rejeição de configurações deste tipo; quanto mais elevado o corte de carga de uma configuração, mais elevada será a chance de ser rejeitada. Entretanto, configurações com pequeno corte de carga serão mais facilmente aceitas.
2. Usando limitantes inferiores e superiores para a parte correspondente aos custos dos circuitos de uma configuração, informação que sempre está disponível para cada sistema e ainda pode ser melhorada usando informação do processo de convergência do próprio problema.

Configuração Típica

Relacionado com a representação do problema é interessante analisar o que significa uma configuração típica no problema de planejamento de sistemas de transmissão de acordo com o tipo de representação que foi adotado. Assim, uma configuração é uma proposta de investimento com valores de n_{ij} inteiros. Para verificar se uma configuração é factível ou infactível, deve-se resolver o PL resultante de (6.1) após fixar os valores de $n = \{n_{ij}\}$ que são conhecidos. Entretanto, para o algoritmo SA todas as configurações são “factíveis” e as infactibilidades são penalizadas na função objetivo através dos cortes de carga tornando as configurações realmente infactíveis em pouco atrativas.

A representação do problema, isto é, a formulação do modelo DC com configurações realizando um processo de transição entre configurações factíveis e infactíveis apresentados para o algoritmo SA permanecem válidas para sua utilização no algoritmo genético e no algoritmo busca tabu que são analisados nos próximos capítulos.

6.3.2 Mecanismo de Transição

O mecanismo de transição é o processo que consiste em transformar a configuração corrente em uma configuração subsequente. Este processo é realizado em três etapas:

1. Geração de uma configuração candidata a partir da configuração corrente, o que é implementado definindo uma *estrutura de vizinhança* para o problema.

2. Determinação da variação da função de custo para a configuração candidata.
3. Decidir se a configuração candidata é aceita, transformando-a em configuração corrente.

A seguir são apresentados detalhes destes aspectos para o problema de planejamento de sistemas de transmissão.

Estrutura de vizinhança

A estrutura de vizinhança é o mecanismo que permite gerar uma configuração candidata n_{ij}^{k+1} a partir da configuração corrente, n_{ij}^k .

Um mecanismo de geração simples, que chamar-se-á *mecanismo básico de transição* pode ser implementado identificando três tipos simples de obtenção de uma configuração candidata a partir da configuração corrente:

1. Adicionar um circuito em um caminho candidato.
2. Trocar circuitos, adicionando um circuito em um caminho candidato e retirando outro circuito em outro caminho candidato.
3. Retirar um circuito em um caminho candidato.

Estes mecanismos simples para gerar as novas alternativas na sequência *adição-troca-retirada* foram implementados da seguinte maneira:

1. Escolhe-se aleatoriamente a linha i como sendo candidata a *adição*. A configuração candidata é obtida adicionando-se uma linha na posição i da configuração G_k . Se esta configuração é aceita pelos critérios de **SA** então foi obtida a configuração G_{k+1} e deve-se repetir o item 1. Caso contrário, tenta-se a troca de linhas.
2. Escolhe-se aleatoriamente a linha j entre aquelas linhas que já estão adicionadas na configuração atual. A *troca* consiste em adicionar uma linha na posição i e retirar uma linha da posição j na configuração G_k , para obter a configuração candidata. Se esta configuração é aceita pelos critérios de **SA** então foi obtida a configuração G_{k+1} e deve-se voltar ao item 1. Caso contrário, tenta-se a retirada de linha no item seguinte.
3. A linha j escolhida no item anterior é considerada candidata a *retirada*. Assim, a configuração candidata é obtida retirando-se uma linha na posição j da configuração G_k . Se esta configuração é aceita pelos critérios de **SA** então foi obtida a configuração G_{k+1} e deve-se voltar ao item 1. Caso contrário, fracassou as três tentativas (*adição-troca-retirada*) e volta-se ao item 1.

No esquema anterior, cada vez que o processo volta ao item 1, o contador de iterações para o nível de temperatura T_k deve ser incrementado em 1, até atingir o limite de N_k ; depois o processo deve passar à outro nível de temperatura. Entretanto, pode-se verificar que para cada



iteração controlada por N_k são realizadas até três tentativas. Assim, no esquema anterior, é evidente que para determinado nível de temperatura a quantidade de tentativas para a obtenção de novas configurações está dentro do intervalo $[N_k, 3 * N_{k+1}]$.

A sequência de operações é *adição-troca-retirada*, isto é, primeiro tenta-se a adição de uma linha, depois a troca e finalmente a retirada de uma linha. No problema de planejamento o processo se inicia geralmente com um corte de carga muito grande o que, intuitivamente, levaria à conclusão de priorizar a adição de linhas; entretanto, uma vez obtida uma configuração sem corte de carga é evidente que a prioridade seria a de retirar linhas. Baseados nos argumentos anteriores, a sequência de *adição-troca-retirada* deve ser invertida em determinados momentos do processo a fim de melhorar o desempenho do algoritmo.

O mecanismo básico de transição deve ser implementado de maneira eficiente com sequência trocada: *adição-troca-retirada* quando a configuração corrente apresenta corte de carga maior que uma tolerância especificada w_o , onde deve ser priorizada a adição de linhas; ou *retirada-troca-adição* quando a configuração corrente apresenta um corte de carga menor que uma tolerância especificada w_o , e portanto deve ser priorizada a retirada de linhas.

A primeira sequência é acionada mais freqüentemente nas fases iniciais do processo onde o corte de carga é elevado e a segunda sequência nas fases finais onde praticamente todas as configurações com corte de carga são rejeitadas. Este mecanismo básico de transição apresenta a vantagem de sua simplicidade e a possibilidade de implementar um eficiente algoritmo de PL para calcular rapidamente o corte de carga da configuração candidata a partir da solução da configuração corrente, como será mostrado mais adiante; entretanto, apresenta limitações para sistemas de grande porte com muitas barras não conexas.

Um mecanismo de geração de configurações mais elaborado deve levar em conta a topologia da rede elétrica, assim o processo de adição e retirada de linhas em novos caminhos deve garantir a conectividade do sistema, evitando que sejam adicionadas linhas ineficientes ou retiradas de linhas em novos caminhos que deixem outras linhas sem possibilidade de transportar energia elétrica. Em sistema grandes e complexos, como o Norte-Nordeste brasileiro, às vezes é necessária a adição de 3 a 6 linhas sucessivas para garantir a conectividade do sistema. Para garantir sempre a conectividade do sistema elétrico foi elaborado um *mecanismo de geração modificado* que mantém sempre uma conectividade adequada no sistema eliminando adições ou retiradas ineficientes. Este mecanismo de geração modificado realiza, às vezes, a adição, troca ou retirada de um conjunto de linhas, de caminhos novos completos, o que leva à caracterização de uma estrutura de vizinhança mais sofisticada. Esta estratégia eficiente de geração de caminhos está adequadamente apresentada em [34].

Avaliação da função objetivo

Dada a configuração corrente, somente a primeira parcela da função objetivo na formulação (6.1) pode ser determinada facilmente. A determinação da segunda parcela implica na resolução do seguinte problema de programação linear (PL):

$$\begin{aligned}
\min \quad & w = \sum_i \alpha_i r_i \\
s.a. \quad & B(x^k + \gamma^o)\theta + g + r = d \\
& (x_{ij}^k + \gamma_{ij}^o) |\theta_i - \theta_j| \leq (x_{ij}^k + \gamma_{ij}^o) \bar{\phi}_{ij} \\
& 0 \leq g \leq \bar{g} \\
& 0 \leq r \leq d
\end{aligned} \tag{6.2}$$

Nesta formulação são alocados, de forma fictícia, geradores artificiais r_i que fornecem a potência necessária nessa barra quando as linhas que chegam à mesma não podem transportar a potência necessária, assim a somatória de todos os r_i , $w = \sum_i \alpha_i r_i$ é conhecido como corte de carga, isto é, a quantidade de carga que o sistema não pode fornecer.

Assim, conhecida a configuração corrente n_{ij}^k , a formulação (6.1) que é um problema PNLIM se transforma em um PL mostrado em (6.2). Portanto, a avaliação da função objetivo para cada configuração candidata implica na resolução de (6.2) usando um algoritmo de PL.

Esta característica do problema de planejamento, avaliar cada configuração candidata através de um PL, eleva consideravelmente o esforço computacional de uma iteração do algoritmo de SA para o problema de planejamento de sistemas de transmissão, quando comparado com outros problemas típicos de pesquisa operacional como o problema do caixeiro viajante onde a variação da função objetivo, da configuração corrente para a configuração candidata, é realizada simplesmente através de 4 somas algébricas. Esta característica do problema de planejamento, de precisar de um PL para analisar cada configuração candidata, também aparece em outros problemas de engenharia elétrica e não depende da forma de representação da função de custo no problema de planejamento.

Critério de aceitação

O critério de aceitação de uma configuração candidata é determinado pelo critério de aceitação de Metrópolis $P(a)$, mostrado a seguir:

$$Pa = \begin{cases} 1 & \text{se } \Delta v < 0 \\ \exp(-\frac{\Delta v}{T}) & \text{se } \Delta v \geq 0 \end{cases} \tag{6.3}$$

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

6.3.3 O Programa de Esfriamento

O programa de esfriamento é a estratégia geral de controle do processo SA desde o início do mesmo até atingir a convergência e está caracterizado pela escolha ou determinação dos quatro seguintes parâmetros:

- Valor inicial do parâmetro de controle T_o , chamado de temperatura inicial.



- Número de tentativas de transição N_k a cada nível de temperatura T_k .
- Taxa de diminuição de temperatura, $T_{k+1} = g(T_k)T_k$, em que $g(T_k)$ é uma função de T_k .
- Temperatura final T_f ou critério de parada.

Uma vez definidos esses parâmetros fica caracterizado um programa de resfriamento. Assim, pode-se implementar muitos programas de resfriamento dependendo do tipo de escolha ou a forma usada para obter cada um desses quatro parâmetros. A eficiência de um algoritmo de SA, isto é, a qualidade da configuração final, assim como o esforço computacional para atingir a convergência, depende da escolha adequada dos parâmetros de controle no programa de resfriamento. A seguir são apresentados e discutidos os critérios de escolha ou determinação desses parâmetros.

Temperatura inicial, T_o

Uma forma de determinar o valor da temperatura inicial T_o é simulando o processo correspondente ao primeiro nível de temperatura, isto é, T_o é construtivamente determinado. Em [21] é proposta a seguinte fórmula construtiva:

$$T_o = \frac{\overline{\Delta V}^+}{\ln\left(\frac{m_2}{m_2 X_o - m_1(1 - X_o)}\right)} \quad (6.4)$$

em que X_o é escolhido e os outros parâmetros são determinados através de um teste experimental de m_o tentativas; assim, m_1 e m_2 correspondem ao número de transições com funções de custo decrescentes e crescentes respectivamente e $\overline{\Delta V}^+$ é o valor médio das diferenças de custo Δv de todas as m_2 transições com aumento na função de custo. Logicamente, $m_o = m_1 + m_2$. No presente trabalho foi escolhido um $X_o = 0,85$ o que significa que na temperatura inicial são aceitas em torno de 85% das configurações candidatas.

Escolha de N_k

A escolha de N_k , número de tentativas a cada nível de temperatura, deve garantir o quase-equilíbrio em cada nível, assim a escolha deste parâmetro está ligada a determinação da taxa de diminuição de temperatura. A maioria dos algoritmos escolhem o valor de N_k em função do tamanho do problema ou do número de variáveis e esse valor pode ser mantido constante ou pode variar com a diminuição da temperatura. Em [16] foi implementada uma maneira crescente para o cálculo do número de tentativas em função do número de variáveis N_o , que é o número de linhas ou caminhos candidatos a adição:

- N_k variável: $\rho \geq 1,0$

$$N_{k+1} = \rho N_k \quad (6.5)$$

Determinação de T_{k+1}

Existem muitos critérios para determinar a taxa de diminuição da temperatura, todos eles baseados no critério básico de manter o quase-equilíbrio do processo. Assim existem critérios simples, como manter uma taxa de diminuição fixa e outros um pouco mais elaborados, que avaliam o desempenho do processo no nível de temperatura T_k para determinar o valor de T_{k+1} . Assim, neste último caso a taxa de diminuição é variável e depende da evolução do processo de otimização. Em [34] são analisadas três formas diferentes de determinar o valor de T_{k+1} :

- *Taxa constante:* $\beta \in [0, 50; 0, 99]$

$$T_{k+1} = \beta T_k \quad (6.6)$$

- *Taxa variável:* $\delta \in [0, 01; 0, 20]$

$$T_{k+1} = \frac{T_k}{\left[1 + \frac{\ln(1+\delta)T_k}{3\sigma(T_k)}\right]} \quad (6.7)$$

em que $\sigma(T_k)$ é o desvio padrão da função de custo das configurações geradas na temperatura T_k .

- *Taxa variável:* $\lambda \leq 1, 0$

$$T_{k+1} = \frac{T_k}{\exp\left(\frac{\lambda T_k}{\sigma(T_k)}\right)} \quad (6.8)$$

Observação: Usando as relações (6.7) e (6.8), obtém-se desempenhos similares a (6.6) em relação a qualidade das configurações encontradas, mas o número de iterações necessário para convergência cresce de maneira proibitiva. A influência dos parâmetros δ e λ assim como de $\sigma(T_k)$ é pequena e produzem taxas de diminuição de temperaturas variáveis, mas elevadas, variando entre 0,94 e 0,99. Portanto, resultados experimentais indicam que é mais prático usar a relação (6.6) com uma adequada calibração dos parâmetros.

Critério de parada: T_f

Existem vários critérios de parada desde os mais simples até os mais elaborados. Os mais simples sugeridos pelos pesquisadores são:

- Fixar no início do processo um número determinado de níveis de temperatura tradicionalmente entre 6 e 50.
- Usar a taxa de melhoria da função objetivo como critério de parada; assim se a função objetivo ou a incumbente não melhora para vários níveis consecutivos de temperatura então o processo atinge a convergência.
- Fixar um número mínimo de aceitações durante o último nível de temperatura; se no último nível não forem aceitas um número maior ao mínimo estabelecido, então o processo atinge a convergência.

Embora existam outros critérios mais elaborados, em [16, 34] foram usados o segundo e terceiro critério simples descritos acima, baseado em testes preliminares que justificam que os critérios de parada não influenciam muito na qualidade da solução encontrada. Assim, nos testes realizados no presente trabalho, o processo pára se a função objetivo não melhora entre 2 a 4 níveis consecutivos de temperatura, ou se durante o último nível de temperatura não foram aceitos entre 1 a 6 novas propostas. Nos dois critérios de parada a seleção do valor depende dos sistemas testados.

6.3.4 Algoritmo de “Simulated Annealing” Implementado

Em [16] foi implementado o seguinte algoritmo que é válido para todos os programas de resfriamento; a figura 6.2 mostra de maneira geral o algoritmo de SA.

1. Determinar a temperatura inicial: T_o .
2. Escolher ou calcular os parâmetros de controle ($\beta, \rho, etc.$) usados nas relações (6.5)-(6.8), assim como também o parâmetro μ , usado para definir o número de iterações na temperatura inicial, obtida através da relação $N_o = \mu \cdot N_t$, onde N_t é o número de variáveis do sistema, isto é, o número de linhas candidatas. Escolha também w_o que define o corte de carga para fazer a seqüência *adição-troca-retirada* de linhas ou vice-versa. Valores típicos para w_o estão na faixa de 0,3 % a 1,0 % da demanda total do sistema e ε_w é o mínimo corte de carga permitido para que a configuração seja considerada factível. Em termos de planejamento de sistemas de transmissão foram usados valores de ε_w da ordem de 0,1 % a 0,3 % da demanda total do sistema.
3. Incrementar o contador de iterações n_{con} . Se $n_{con} \geq N_k$ ir ao passo 7; caso contrário, ir ao passo 4.
4. Se o corte de carga corrente w_o^k for menor que w_o , ir ao passo 5; caso contrário, ir ao passo 6.
 - (a) Obter uma configuração candidata simulando a *retirada* de uma linha i aleatoriamente escolhida; com a configuração candidata, assim obtida, processar um PL e testar sua aceitação usando (6.3). Se for aceita, atualize a configuração corrente e as outras grandezas do processo (w_o^k , função de custo corrente, melhor configuração encontrada (incumbente), etc.) e volte ao passo 3; caso contrário, ir a 5(b).
 - (b) Obter uma configuração candidata simulando a *troca* da linha i com uma linha j aleatoriamente escolhida; com a configuração candidata, assim obtida, processar um PL e testar sua aceitação usando (6.3). Se for aceita, atualize a configuração corrente e as outras grandezas do processo e volte ao passo 3; caso contrário, ir a 5(c).
 - (c) Obter uma configuração candidata simulando a *adição* da linha j aleatoriamente escolhida; com a configuração candidata, assim obtida, processar um PL e testar sua aceitação usando (6.3). Se for aceita, atualize a configuração corrente e as outras grandezas do processo. Em qualquer caso, volte ao passo 3.
5. Idem ao passo 5 com a seqüência trocada:



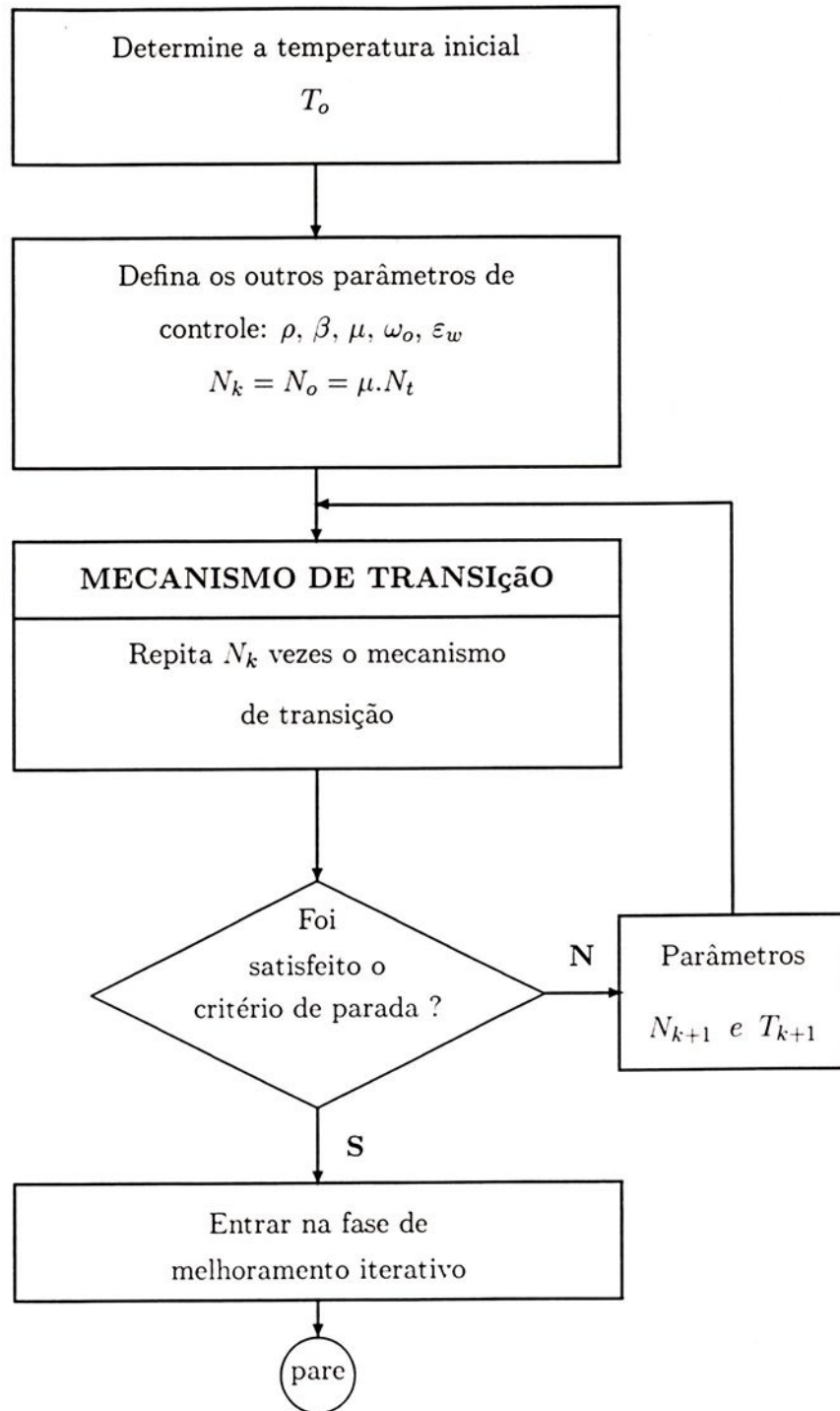


Figura 6.2: Diagrama de Fluxo do Algoritmo de S.A

- (a) Simular a *adição* de uma linha. Se a configuração for aceita, volte ao passo 3; caso contrário, ir a 6(b).
 - (b) Simular a *troca* de linhas. Se a configuração for aceita, volte ao passo 3; caso contrário, ir a 6(c).
 - (c) Simular a *retirada* de uma linha. Em qualquer caso, volte ao passo 3.
6. Testar o critério de parada: se o critério for satisfeito, ir ao passo 9; caso contrário, ir ao passo 8.
 7. Atualizar os parâmetros de controle: N_{k+1} e T_{k+1} e voltar ao passo 3.
 8. O processo de **SA** foi concluído, entrar na fase de melhoramento iterativo (busca local) e parar todo o processo.

O melhoramento iterativo (busca local) implementado é muito rápido, pois nesta fase somente são testadas aquelas configurações candidatas que produzem redução do investimento, portanto, somente são consideradas para avaliação os seguintes casos:

- **Retirada de linha:** Neste caso a tentativa de retirar uma linha que leva a uma diminuição do investimento e cada avaliação implica processar um PL. Mas como geralmente as linhas com adição são pequenas, esta parte leva a um esforço computacional muito pequeno.
- **Troca de linhas:** Neste caso, deve-se simular a retirada de uma linha que tem adição na configuração atual e a adição de cada uma das linhas candidatas. Esta parte também implica pouco esforço computacional pois somente são testados através do PL aquelas trocas que implicam numa diminuição em investimento, pois as outras combinações, isto é, aquelas que incrementam o investimento, são automaticamente rejeitadas.

Na fase de melhoramento iterativo não é avaliada a possibilidade de adição de linhas pois implicaria em um incremento do investimento. São aceitas todas as configurações que diminuem o investimento e apresentam um corte de carga menor que a tolerância ε_w especificada. Nesta fase, se é aceita alguma configuração, então o processo deve ser reinicializado. Também nesta fase, a função objetivo considerada é somente o custo das linhas, isto é, $v' = \sum_{(i,j)} c_{ij} \eta_{ij}$.

6.4 SA Paralelo Aplicado ao Problema de Planejamento

O esforço computacional requerido pelo algoritmo sequencial "Simulated Annealing" depende em grande parte da natureza e tamanho do problema de otimização apresentado; assim, para o caso do planejamento de redes de transmissão de grande porte, este esforço de cálculo pode levar dias até conseguir a convergência. Existem vários fatores que podem melhorar o desempenho do algoritmo em relação ao esforço computacional e da qualidade da solução; os mais importantes são: (a) projeto do algoritmo sequencial, (b) aceleração do "hardware", (c) projeto de algoritmos paralelos.



No algoritmo **SA** o maior consumo de tempo está na etapa de geração de tentativas para a formação de uma cadeia de Markov. Nos algoritmos paralelos se toma vantagem desta característica atribuindo a cada processador uma parcela destas tentativas. As tentativas são particionadas em partes iguais, de acordo ao número de processadores.

Uma forma de acelerar o processo de convergência de um algoritmo de **SA** é implementar uma versão paralela do algoritmo. Implementações paralelas de **SA** foram usadas em muitas aplicações de pesquisa operacional, obtendo-se diferentes tipos de desempenho. As implementações com maiores sucessos foram as implementações paralelas especializadas ("tailored") que são possíveis especialmente em problemas que permitem partição de tarefas como alocação de VLSI e problemas de roteamento ("routing"). Para problemas mais complexos e que não permitem uma partição, somente são possíveis as implementações paralelas de caráter geral.

A parcela do algoritmo de **SA** que consome maior esforço computacional é a geração das N_k configurações candidatas para o nível de temperatura T_k . A avaliação de cada uma das N_k configurações candidatas implica em fazer os seguintes testes:

1. A partir da configuração corrente, selecionar uma configuração candidata usando a estrutura de vizinhança.
2. Calcular a diferença da função objetivo da configuração candidata com a configuração corrente.
3. Decidir se a configuração candidata deve ser aceita ou rejeitada.
4. Substituir a configuração corrente pela configuração candidata que foi aceita.

A implementação eficiente de um algoritmo de **SA** paralelo deve ser realizada na sequência de passos mostrada anteriormente. Além disso, pode-se fazer as seguintes observações:

1. Os três primeiros passos são praticamente independentes e podem ser implementados em paralelo para diferentes configurações candidatas sem alterar a natureza sequencial do algoritmo. Por outro lado, executar o passo 4 implica num processo estritamente sequencial pois não podem ser aceitas novas configurações em paralelo;
2. O número de vezes em que são executados os passos 1-3 praticamente não varia em todo o processo para os diferentes níveis de temperatura: entretanto, a frequência de execução do passo 4 varia muito durante o processo, sendo freqüentemente executada para valores elevados de temperatura e quase não é executada para valores pequenos de temperatura;
3. O esforço computacional necessário para implementar cada um dos quatro passos varia muito de um tipo de problema para outro. Assim, por exemplo, no problema do caixeiro viajante o esforço computacional para executar os passos 2 e 3 são parecidos pois é muito simples calcular a diferença de custos entre a configuração corrente e a configuração candidata, com um esforço computacional da mesma ordem que calcular a função exponencial que define o critério de aceitação do passo 3;



4. No problema de planejamento de sistemas de transmissão a longo prazo, o esforço computacional requerido pelo passo 2 é muito grande quando comparado com os outros passos. Executar o passo 2 implica resolver um problema de programação linear (PL) de grande porte para determinar o valor da função objetivo da configuração candidata. Assim, o tempo gasto nos outros passos são totalmente irrelevantes comparados com o tempo gasto no passo 2 no problema de planejamento;
5. A qualidade da solução final obtida é parecida a obtida por um algoritmo seqüencial, desde que o número de processadores não seja muito grande. Quando o número de processadores aumenta muito, o subconjunto de configurações analisadas por cada processador pode ser muito pequeno, não permitindo que o processo possa atingir o estado de quase-equilíbrio.

Baseadas nas observações anteriores e nas particularidades próprias do problema de planejamento de sistemas de transmissão, em [17, 34] foram implementados algoritmos de SA paralelos de caráter geral conhecidos na literatura especializada de SA: *algoritmo paralelo da divisão* e *algoritmo paralelo de agrupamento* ("clustering"). Em [17, 22, 34], pode-se encontrar uma análise mais rigorosa sobre a implementação de algoritmos SA paralelos. Entretanto, apresenta-se de maneira resumida o algoritmo paralelo da divisão.

6.4.1 Algoritmo da Divisão:

No *algoritmo da divisão* [17, 22], para cada nível de temperatura a tarefa de busca de novas configurações é dividida entre todos os processadores. Se N_k é o número de tentativas de busca de novas configurações realizadas na temperatura T_k e np o número de processadores, então, nessa temperatura, cada processador realiza N_k/np tentativas, sendo que cada processador gera sua própria seqüência de configurações. O algoritmo da divisão permite duas estratégias diferentes. Na *primeira estratégia*, não existe comunicação entre processadores; isto é, cada processador realiza um conjunto de N_k/np tentativas e passa ao seguinte nível de temperatura com sua configuração corrente sem trocar informação (as configurações) com os outros processadores e, no final do processo, das np configurações (uma por processador) é escolhida a melhor configuração como sendo a configuração final. Na *segunda estratégia* existe troca de informação toda vez que existe mudança de temperatura onde é escolhida a melhor configuração das np configurações disponíveis (uma por processador). Assim, esta melhor configuração é usada como configuração inicial por todos os processadores no nível de temperatura seguinte.

O algoritmo da divisão apresenta pequena comunicação e portanto a eficiência pode ser elevada. Quando o número de processadores aumenta, o número de tentativas que cada processador realiza em cada nível de temperatura pode tornar-se pequeno; isto pode levar a um comportamento diferente quando comparado com um algoritmo seqüencial. Assim, no processamento paralelo, pode-se aumentar o número de tentativas N_k a cada nível de temperatura e/ou diminuir a taxa de diminuição de temperatura, incrementando o parâmetro β , a fim de que cada processador possa atingir o quase-equilíbrio do ponto de vista de SA. Em ambas estratégias, pode-se iniciar o processo com uma única configuração inicial ou com configurações diferentes.



Capítulo 7

Planejamento de Sistemas de Transmissão Usando o Algoritmo Genético

7.1 Introdução

Neste capítulo é apresentado um algoritmo genético (genetic algorithms - GA) aplicado ao problema de planejamento de sistemas de transmissão apresentado em [19, 34]. Em [19, 34] é desenvolvido um algoritmo genético baseada nas idéias fundamentais do algoritmo genético (AG) padrão e algumas implementações especializadas que levam em conta as características específicas do problema de planejamento de sistemas de transmissão. O algoritmo desenvolvido encontra facilmente as configurações ótimas de sistemas de pequeno e de medio porte e configurações quase ótimas de sistemas de grande porte [19, 34].

O algoritmo genético contorna adequadamente os problemas de ótimos locais pois trabalha com uma população ou conjunto de configurações, o que permite visitar regiões muito distantes na procura de caminhos promissores. GA simula o princípio de seleção natural que acontece na natureza e que leva à sobrevivência dos indivíduos geneticamente melhores dotados na luta com outros indivíduos por recursos escassos. Assim, o algoritmo genético parte de uma população inicial, realiza um processo de seleção das melhores configurações ("strings") com direito a participar na geração dos novos descendentes ("allsprings"), faz a recombinação ("crossover") entre estas melhores configurações e finalmente um processo de mutação determina os elementos da nova geração.

O algoritmo genético simula o processo de seleção natural para resolver um problema de otimização combinatorial. O mecanismo de busca do algoritmo genético é de natureza probabilística, mas adequadamente direcionada. Os elementos ou configurações da população inicial geralmente são obtidos de forma aleatória. Assim, cada elemento ou gene de uma configuração ("strings") é obtida aleatoriamente. Também, cada "strings" tem uma função objetivo que é usada no processo de seleção para determinar o número de descendentes a que tem direito essa configuração. Portanto, o número de descendentes que de fato é designado a cada configuração é estatisticamente proporcional a qualidade da correspondente função objetivo ("fitness") e finalmente as operações de "crossover" e mutação terminam gerando a nova população. Cada parte deste processo de obtenção de uma nova geração (seleção, "crossover" e mutação) é

controlada de maneira probabilística. Um algoritmo genético é praticamente definido depois de escolher: a população inicial, os parâmetros de controle (tamanho da população, taxa de “crossover” e taxa de mutação) e o critério de parada.

Implementações especializadas geralmente levam em conta as características específicas do problema para implementar algoritmos genéticos especializados. Outras alternativas incorporam características de outros algoritmos como tabu search e simulated annealing.

7.2 O Algoritmo Genético: Teoria

O algoritmo genético é uma metodologia usada para resolver problemas de otimização combinatorial que apresentou um grande sucesso na última década na resolução de problemas de grande porte em áreas muito diversas. O algoritmo genético foi inicialmente formulado por Holland [24] baseado no princípio de seleção natural que acontece na natureza e que fornece maiores chances de sobrevivência aos indivíduos melhores dotados. Matematicamente, pode-se considerar o algoritmo genético como uma técnica de otimização combinatorial que tem uma alta probabilidade de encontrar a solução ótima global de problemas grandes e complexos e que apresentam muitas soluções ótimas locais.

O algoritmo genético está baseado no princípio de seleção natural. Na natureza os indivíduos melhores dotados tem maiores chance de sobrevivência e a capacidade de adaptação a um meio mutante é fundamental na sobrevivência de indivíduos e espécies. Foi baseada nesta lei natural que as espécies sobreviveram e evoluíram na Terra. As características específicas de um indivíduo determina sua capacidade de sobrevivência e, em última instância, essa capacidade específica é determinada pelo conteúdo genético do indivíduo; isto é, pela unidade elementar chamada *gene* na biologia. Assim, nas mudanças do material genético das espécies acontece a evolução das mesmas. A seleção natural leva à sobrevivência dos indivíduos melhores dotados e no processo de reprodução estes indivíduos transmitem aos descendentes os melhores genes; por outro lado, os indivíduos menos dotados morrem no processo de competição por espaço, alimentos, etc. Assim, o princípio de seleção natural permite que somente os indivíduos melhores dotados gerem descendentes. Foi assim que aconteceu a evolução das espécies. A evolução acontece quando os cromossomos de dois indivíduos bem dotados se recombina durante a reprodução. Neste processo acontece uma troca de material genético que pode levar à geração de um indivíduo muito bem dotado ao receber o melhor material genético dos pais. Este processo repetido leva à evolução, isto é, uma seleção com a sobrevivência dos indivíduos melhores dotados e a correspondente reprodução (com troca de material genético) entre os mesmos. Adicionalmente, no processo de evolução, existe um fenômeno chamado *mutação*, que é uma mudança do código genético dos indivíduos como consequência de sua interação com o meio ambiente.

No algoritmo genético o processo de otimização se inicia com a seleção de uma população, isto é, um conjunto de soluções (configurações) candidatas. Cada configuração é qualificada pelo valor da função objetivo que apresenta. Geralmente a codificação de uma configuração é realizada em sistema binário. Todos os elementos da população são classificados pela qualidade de sua correspondente função objetivo. Assim, cada elemento recebe uma determinada probabilidade para gerar os elementos (configurações) na geração seguinte. Os elementos melhores

qualificados (“higher fitness”) neste processo têm, no sentido probabilístico, maior probabilidade de participar na geração dos elementos da nova população. Essa nova geração é obtida com as operações de recombinação (“crossover”) e mutação. Dessa forma, as melhores configurações são recombinadas para gerar os elementos da nova população e, nesse processo, a recombinação leva a uma transferência de material genético entre as melhores configurações. Neste processo, eventualmente devem aparecer elementos da nova geração (filhos) que recebem o melhor material genético das configurações melhor qualificadas, produzindo novas configurações de excelente qualidade. A mutação termina a formação da nova população. Na natureza, a mutação produz a regeneração da perda de material genético. No algoritmo genético, a mutação produz uma modificação esporádica e aleatória de algum gene de um elemento (configuração) da população. Resumindo, pode-se dizer que os **AG** são uma técnica de otimização combinatorial que inicia o processo de otimização com uma população inicial, faz um ordenamento seletivo dos elementos da população avaliando a qualidade da função objetivo e gera uma nova população num processo de três etapas: na primeira é realizada a seleção (onde à cada configuração é designada um número determinado de descendentes), na segunda etapa é realizado o “crossover” entre os elementos melhores qualificados e com direito a gerar descendentes; e na etapa final, é realizada uma mutação aleatória para obter a nova população.

O algoritmo genético em sua forma mais simples é composto pelos seguintes componentes:

- População inicial,
- Mecanismo de codificação do problema,
- Parâmetros de controle,
- Avaliação da qualidade da função objetivo,
- Seleção e operadores genéticos do ciclo geracional.

7.2.1 População Inicial

A população inicial pode ser gerada de várias maneiras. Uma delas é gerar populações iniciais de maneira aleatória; uma outra, poderá ser a geração das populações iniciais usando algoritmos aproximados e cujas respostas se obtém em tempos curtos, com qualidades de respostas relativamente boas.

Fazer evoluir uma população gerada inicialmente de maneira aleatória, até chegar a ter configurações perto da solução ótima, é um bom teste para conhecer como está funcionando a implementação dos **AG**, já que teoricamente as características essenciais e críticas da solução final devem ser consequência do processo evolutivo e não das características dos métodos usados para gerar a população inicial, como é o caso de sistemas de pequeno e médio porte. Já em sistemas de grande porte, o processo necessita de um bom programa inicializador pelas razões que mais adiante serão expostas.



7.2.2 Mecanismo de Codificação do Problema

Em seus trabalhos, Holland representa os cromossomos (configurações) por “strings” binários, isto é, listas de zeros e uns. Este tipo de codificação tem sido empregado em diferentes aplicações; inclusive em muitas aplicações onde não é óbvio seu emprego. Apesar disto, existem uma grande variedade de aplicações onde essa representação não é efetiva; assim, alguns pesquisadores têm começado a explorar outras formas de representação; em geral aquelas que tem relação com aplicações práticas dos AG.

Alguns tipos de problemas são mais facilmente representáveis em codificação binária que outros e, portanto melhor condicionados para serem resolvidos por um algoritmo genético. Existem outros cujas variáveis são binárias, assim não existe nada a modificar ou adequar. Também existem problemas onde as variáveis de decisão são inteiras com limitante superior. Neste caso, deve-se fazer uma representação binária destas variáveis, o que aumenta o tamanho do problema. O problema mais complicado de resolver em termos de representação das variáveis acontece quando as variáveis de decisão são contínuas. Neste caso, deve-se fazer a representação binária de uma variável contínua variando em um intervalo.

No problema de planejamento de sistemas de transmissão as variáveis de decisão são as linhas adicionadas em cada caminho candidato e são portanto variáveis inteiras e, no caso mais geral, cada variável de decisão pode ser representada por 4 dígitos binários que equivale a adicionar mais de 8 linhas em cada caminho candidato. Assim, no problema de planejamento de sistemas de transmissão o problema de representação é resolvido de uma maneira relativamente simples. Entretanto, para sistemas de grande porte como o Norte-Nordeste brasileiro, esta representação levaria a representar cada elemento (configuração) da população por um vetor de 732 elementos binários o que tornaria computacionalmente cansativa a implementação do “crossover” e a preparação dos vetores para o cálculo da função objetivo. Mais adiante são apresentadas idéias alternativas de representação fundamentadas em três idéias básicas: representação binária, representação por bits independentes e representação decimal.

7.2.3 Parâmetros de Controle

As características básicas, assim como a eficiência de um algoritmo genético, são determinadas pela definição dos *parâmetros de controle*, pela forma de escolha da população inicial e pelo critério de parada adotado. Na formulação básica do algoritmo, os parâmetros de controle são: o tamanho da população, a taxa de “crossover” e a taxa de mutação. Portanto, os parâmetros de controle juntamente com o mecanismo de geração da população inicial e o critério de parada definem praticamente a qualidade de um algoritmo genético.

Valores típicos usados na literatura especializada para os parâmetros de controle são [23]:

- Tamanho de população entre 30 e 200.
- Taxa de “crossover” entre 0,5 e 1,0. e
- Taxa de mutação entre 0,001 e 0,050.



7.2.4 Avaliação do Nível de “Fitness”

A função empregada para atribuir os valores “fitness” aos membros da população deverá possuir boa sensibilidade, tal que consiga fazer uma boa discriminação destes últimos. Assim por exemplo, os resultados da avaliação de uma função objetivo pode atribuir um valor de 1,09 a um indivíduo muito bom; por outro lado, pode dar um valor de 0,96 a um indivíduo muito ruim. Se estes valores forem empregados para selecionar os indivíduos na fase de recombinação, seria necessário muito tempo de computação para que os descendentes do primeiro influam mais que os do pior indivíduo.

A normalização se faz trocando a referência dos valores das funções, isto é, subtraindo um valor preestabelecido de cada valor das funções. Desta maneira os indivíduos bons participarão em maior porcentagem na formação da nova geração.

Esta normalização deverá privilegiar as melhores configurações, porém sem descartar o material genético presente na população, já que se a normalização estiver influenciada fortemente pelo melhor indivíduo, a busca ficará em uma região do espaço de configurações perto daquela que influenciou na escolha. Caso contrário, se não existir influência das melhores soluções, pode-se ter uma busca demasiada lenta.

Assim, como no exemplo anterior, para um indivíduo bom se atribui um valor de 1,09 e para um indivíduo ruim um valor de 0,96.

O critério sem normalização é:

	$F(x_1)$	$F(x_2)$	Δ
Valores atribuídos aos indivíduos	1,09	0,96	0,13
Participação na nova população	53%	47%	13%

O critério com normalização se obtém subtraindo 0,80 de cada valor, ou seja:

	$F(x_1)$	$F(x_2)$	Δ
Valores atribuídos aos indivíduos	0,29	0,16	0,13
Participação na nova população	64%	36%	81%

Pode-se observar que o critério com normalização produz uma melhor seletividade, contornando em parte o problema de valores de $F(x)$ muito próximos. Existem ainda outras formas de normalização que poderiam ser implementadas para contornar este problema.

7.2.5 Seleção e Operadores Genéticos do Ciclo Geracional

O mecanismo de *seleção* tenta modelar o mecanismo de sobrevivência dos indivíduos melhores dotados na natureza. Assim, no algoritmo genético, uma vez definida a população,



determina-se a função objetivo ou seu equivalente “fitness” de cada elemento ou configuração da população e aquelas com funções objetivos de melhor qualidade recebem uma maior ponderação para participar na geração dos elementos da nova população.

A mais simples implementação de seleção é o chamado *esquema de seleção proporcional*. Neste esquema cada configuração tem direito a gerar um número de descendentes proporcional à qualidade da função objetivo. Assim, as configurações com maiores valores da função objetivo participarão na geração de muitos descendentes e aquelas de menor valor da função objetivo praticamente não participarão na geração de novas configurações.

O esquema de seleção proporcional é implementado usando o chamado *esquema de seleção da roleta*. No esquema da roleta cada configuração (“string”) é alocada num setor circular da roleta, proporcional ao valor da correspondente função objetivo. Um processo aleatório determina um número de descendentes a que tem direito cada configuração. Entretanto, é possível substituir o esquema da roleta por mecanismos mais eficientes e rápidos que designe a cada configuração um número de descendentes proporcional a porcentagem determinada para cada configuração e fazendo um ajuste no final do processo para determinar um número de descendentes igual ao tamanho da população corrente. Uma vez terminado o processo de seleção e portanto conhecido o número de descendentes a que tem direito cada configuração da população corrente, deve-se implementar o “crossover”.

Na formulação original do algoritmo genético o “crossover” é a operação fundamental e crucial e à mutação é designada uma função secundária. Uma vez definida a seleção, escolhe-se um par de configurações que tem direito a gerar descendentes e este par é submetido a uma recombinação para gerar dois novos descendentes. A escolha das duas configurações para “crossover” pode ser feita de forma aleatória ou de forma sistemática, iniciando a recombinação com aquelas configurações que tem direito a um maior número de descendentes. Existem quatro tipos de “crossover”: simples (“single-point”), com dois pontos (“two-point”), multiponto (“multi-point”) e uniforme.

No “crossover” simples, escolhe-se aleatoriamente um ponto de recombinação; assim, o material genético além desse ponto é trocado. Seja L o tamanho ou número de elementos de uma configuração; então, escolhe-se aleatoriamente um número entre 1 e $L-1$. Seja L_p o número aleatório selecionado, então as duas configurações escolhidas para “crossover” trocam a parte correspondente às primeiras L_p posições. Essas duas novas configurações, assim geradas, são as candidatas a configurações na nova população. No “crossover” de dois pontos, são selecionados de maneira aleatória dois pontos de recombinação, então é trocado o material genético contido entre estes dois pontos. No processo de generalização do “crossover” pode-se implementar o chamado “crossover” multiponto e uniforme. Neste último caso, *cada elemento* (bit) é selecionado de maneira aleatória; começando pelo primeiro, é eleito aleatoriamente o primeiro pai que contribui com o primeiro bit ao primeiro filho. Na sequência, o segundo filho recebe o bit do segundo pai. Este processo continua até que todos os bits sejam atribuídos. Cada tipo de “crossover” apresenta vantagens e desvantagens.

Em relação a operação de “crossover”, os algoritmos genéticos definem a chamada *taxa de “crossover”* (ρ_c). Um algoritmo genético geral não procede a recombinação de todas as configurações escolhidas. A taxa de “crossover” determina probabilisticamente a porcentagem de configurações que serão submetidas efetivamente a recombinação neste algoritmo. Assim uma parcela das configurações da população corrente passa integralmente a constituir a lista

de novas configurações candidatas para a nova população sem “crossover”. Portanto, uma vez escolhida as duas configurações candidatas, gera-se um número aleatório entre 0 e 1. Se o número aleatório gerado for maior que ρ_c , a taxa de “crossover”, então as duas configurações passam diretamente para a lista de configurações candidatas sem “crossover” e, caso contrário, procede-se ao “crossover”. Tipicamente os valores de ρ_c variam entre 0,9 e 1,0.

A operação de *mutação* termina com o processo de obtenção da nova geração da população. O processo de mutação implica modificar de 0 para 1 ou vice-versa um elemento (gene) de uma configuração. Este processo é controlado por um parâmetro denominado *taxa de mutação* (ρ_m). No algoritmo genético simples, a mutação é considerada como um operador secundário comparado com o “crossover” e tem a finalidade de restaurar a perda de material genético que pode acontecer na população. Assim, por exemplo, se na posição k todas as configurações (“strings”) tem valor 0 na representação binária, e na configuração ótima o valor correto nessa posição é 1, então o “crossover” não pode regenerar um 1 nessa posição; entretanto a operação de mutação pode resolver essa dificuldade.

7.3 O Algoritmo Genético Básico Aplicado ao Problema de Planejamento

Muitos problemas de otimização combinatorial que são encontrados na prática são resolvidos usando métodos aproximados devido a grande dificuldade em encontrar sua solução ótima. Os métodos combinatoriais, como é o caso dos **AG**, foram projetados para lidar com este tipo de problema. Matematicamente, pode-se considerar o algoritmo genético como uma técnica de otimização combinatorial que tem uma alta probabilidade de encontrar a solução ótima de problemas grandes, complexos e que apresentam muitas soluções ótimas locais. Este é o caso do planejamento da expansão das redes de transmissão de grande porte, que devido às restrições de integralidade (isto é, as linhas só assumem valores inteiros) se tornam um problema combinatorial, e é de difícil resolução, devido ao grande número de soluções ótimas locais possíveis. Por exemplo, o sistema Norte-Nordeste brasileira, com 89 nós e 183 linhas chega a ter até $(11 + 1)^{183}$ possíveis soluções, isto é, supondo um máximo de 11 linhas candidatas por ramo.

No algoritmo genético aparece um conjunto de componentes que define o tipo e a qualidade do algoritmo. Estes componentes são divididos em quatro partes a fim de facilitar a apresentação, ou seja:

1. A representação e codificação do problema de planejamento.
2. O mecanismo geracional: seleção - “crossover” - mutação.
3. Geração da população inicial.
4. Parâmetros de controle e o critério de parada.



7.3.1 A Representação e Codificação do Problema

Representação

A função objetivo do modelo DC modificado já teve uma transformação inicial, pois além da parcela correspondente ao custo das linhas adicionadas, foi incluída outra parcela correspondente a uma penalidade correspondente ao corte de carga que produz uma determinada configuração. Assim, no algoritmo genético, uma configuração (“string”) da população está constituída unicamente pelas variáveis de investimento n_{ij} e o efeito das outras variáveis contínuas de operação são consideradas implicitamente em conjunto como corte de carga no sistema. Portanto, do ponto de vista do algoritmo genético, as variáveis que formam uma configuração são inteiras, isto é, o número de linhas adicionadas em cada caminho candidato.

O valor da função objetivo determina a qualidade da configuração ou elemento de uma população. Assim, as configurações com valores da função objetivo ou seu equivalente “fitness” de melhor qualidade participam com maior frequência na geração dos elementos da nova população ou geração. Na *formulação tradicional* dos algoritmos genéticos o problema a otimizar é formulado na forma de maximização. A *transformação* de um problema de minimização em um problema equivalente de maximização para usar adequadamente um algoritmo genético não é muito trivial pois ela pode comprometer a seletividade do processo de seleção. No problema de planejamento, a transformação do problema de minimização em um problema de maximização foi realizada da seguinte forma:

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

em que K é uma constante que pode ser fixa ou variável durante o processo de otimização, e sua escolha é feita baseando-se nos valores das funções objetivo das configurações iniciais ou, melhor ainda, pode assumir o valor correspondente àquela função objetivo de maior valor em cada geração.

No caso em que o algoritmo inicie o processo com configurações obtidas por outros métodos, K pode ser fixo e assumir um valor em torno de 20% maior do valor da função objetivo de uma configuração típica usada na geração da população inicial. Quando a população inicial é gerada aleatoriamente K assume um valor variável em cada geração e assume um valor 5% maior que a pior configuração. Todas estas implementações visam manter a seletividade dos AG.

Outra forma alternativa que fornece resultados muito parecidos é fazer a seguinte transformação:

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

No presente trabalho foram implementadas ambas alternativas.



Codificação do Problema

No algoritmo genético tradicional as variáveis do problema são codificadas em *sistema binário* pois desta forma permite uma melhor implementação das operações de “crossover” e mutação. No problema de planejamento, as variáveis que formam uma configuração são inteiras e, portanto, a codificação mais óbvia consistiria em representar a variável inteira em sistema binário. Entretanto, analisando o problema específico de planejamento, pode-se verificar que existem três formas de codificar uma configuração:

- 1. Codificação em sistema binário.
- 2. Codificação em bits independentes.
- 3. Manter a codificação decimal.

A figura 7.1 ajuda a analisar estes 3 tipos de codificação.

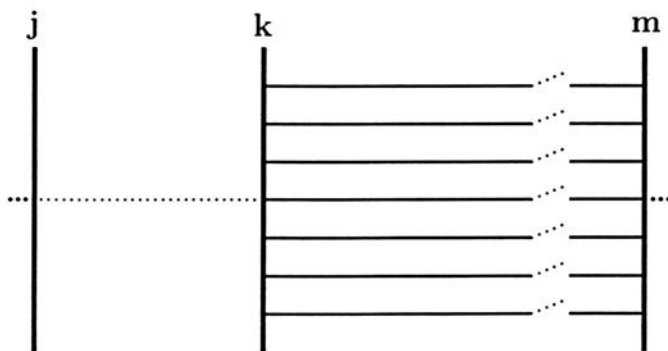
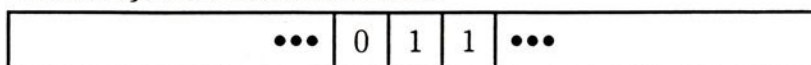


Figura 7.1: O problema de codificação

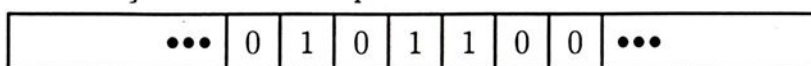
Na figura 7.1, existe a possibilidade de adicionar até 7 linhas entre as barras *k* e *m*. Supondo que na configuração corrente foram adicionadas 3 linhas. A figura 7.2 mostra a codificação da configuração somente para o caminho *k – m* onde foram realizadas 3 adições. As três formas de codificação apresentam vantagens e desvantagens quando são considerados os aspectos mais importantes do processo.

Um aspecto importante é o *espaço necessário* para o armazenamento da configuração. No exemplo, a codificação tradicional em *sistema binário* usa 3 casas binárias, a representação por *bits independentes* usa 7 casas binárias e a *representação decimal* usa uma casa decimal. Um problema de grande porte, como o Norte-Nordeste brasileiro, tipicamente permite adicionar mais de 7 linhas por caminho candidato, existem em torno de 190 caminhos candidatos e uma população típica tem entre 160 e 200 configurações. Assim, entre as três alternativas existe uma diferença enorme em requisitos de memória necessárias para o armazenamento das configurações, mostrando a grande vantagem da codificação decimal. Adicionalmente, as duas primeiras formas de codificação requerem esforço computacional maior, pois cada configuração

Codificação em sistema binário



Codificação em bits independentes



Codificação decimal

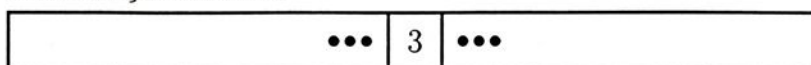


Figura 7.2: Diferentes formas de codificação

deve ser transformada em seu equivalente decimal para usar a subrotina de PL que determina o corte de carga correspondente a cada configuração.

Outro aspecto importante é que a codificação binária apresenta um problema conhecido como *penhasco de Hamming* ("Hamming cliffs") que significa que existe uma grande distância de Hamming para dois inteiros consecutivos. Por exemplo os números inteiros 3 e 4 são representados em sistema binário como 011 e 100. Assim um algoritmo genético deve transformar, usando "crossover" e mutação, todos os seus dígitos para passar de 3 para 4 linhas, as quais formam configurações vizinhas. Por outro lado, a codificação usando bits independentes e a codificação decimal não apresenta este problema.

Outro aspecto mais importante ainda, relacionado com a codificação escolhida, acontece na implementação do "crossover" e da mutação. A figura 7.3 ilustra este aspecto que é crucial na implementação de um algoritmo genético eficiente. A figura 7.3 apresenta, em codificação binária tradicional, duas configurações do caminho candidato $k - m$ da figura 7.1. Assim, mostram-se duas configurações antes do "crossover", depois do "crossover" e depois de fazer a mutação. Antes do "crossover" as configurações indicam a alocação de 3 e 4 linhas respectivamente. Após o "crossover", as novas configurações indicam 7 e 0 linhas adicionadas respectivamente; portanto, *no ponto de recombinação* não acontece uma troca de informação adequada (trocar 3 linhas com 4 linhas no caminho $k - m$). O que realmente acontece no ponto de recombinação é uma mutação violenta (de 3 para 7 linhas e de 4 para 0 linhas). Igualmente, no caso da mutação, houve uma mutação na posição central da segunda configuração mas isto implica passar de 0 a 3 linhas indicando também uma mutação brusca. Este problema não acontece com as outras duas formas de codificação.

Um aspecto final em relação a codificação é que na codificação binária e por bits independentes, as operações de "crossover" e mutação são implementadas, do ponto de vista computacional, de uma maneira trivial, enquanto que na representação decimal, deve-se redefinir a operação de mutação. Também é possível realizar uma codificação binária de forma implícita, isto é, manter uma codificação decimal mas realizar as operações de "crossover" e mutação simulando a codificação binária.

As observações levam a pensar que uma codificação decimal com um aumento da taxa de mutação seria equivalente e até mais eficiente que uma codificação binária em termos de qualidade de soluções encontradas e com um menor esforço computacional. Mantendo uma codificação decimal, o "crossover" seria até mais eficiente pois no ponto de "crossover" realmente

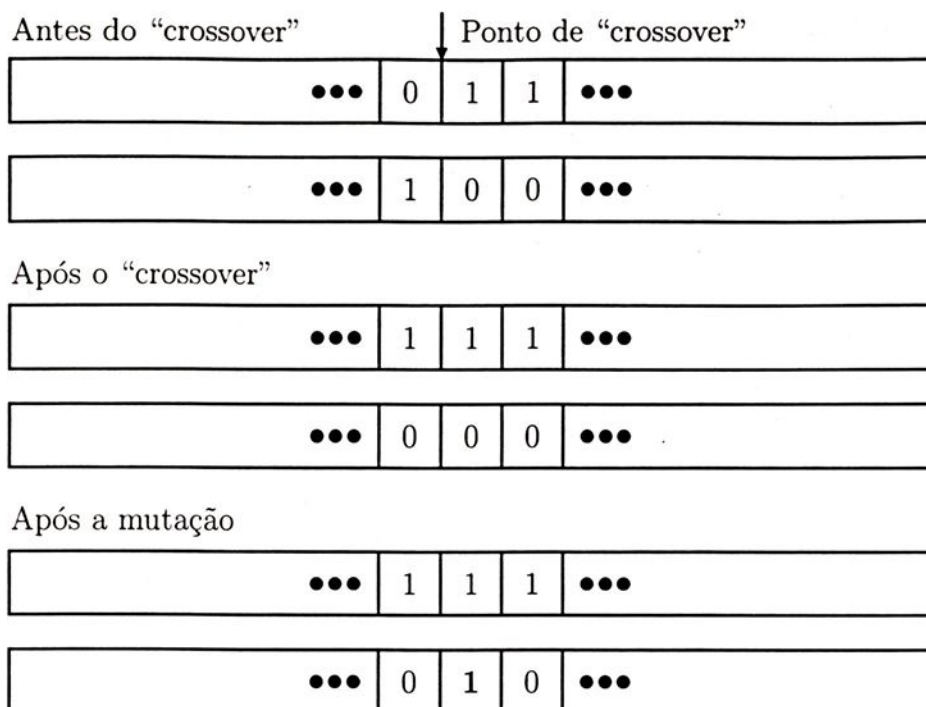


Figura 7.3: "Crossover" e mutação

aconteceria transferência adequada de informação (no exemplo da figura 7.3, 3 linhas passam para a configuração 1 e 4 linhas passam para a configuração 2) e a mutação simplesmente consistiria em aumentar ou diminuir em uma unidade o número de linhas para um caminho candidato. Esta codificação ainda facilita o trabalho computacional na hora de testar as configurações através de um PL. Pode-se ainda incorporar mutações maiores e assim é possível definir mutações simples, duplas, etc. Estas modificações foram incorporadas e testadas no algoritmo desenvolvido juntamente com a codificação binária implícita.

7.3.2 O Mecanismo Geracional: Seleção - "Crossover" - Mutação

A partir da população corrente é determinada uma nova população através de um processo geracional que implica realizar sucessivamente as operações de seleção, "crossover" e mutação.

Nos **AG** o *mecanismo de seleção* é a primeira etapa no processo de obtenção de uma nova população a partir da população corrente. No processo de seleção, determina-se o número de descendentes ("offspring") a que tem direito cada configuração ("string") da população corrente. Nas etapas seguintes são implementados o "crossover" e a mutação para terminar de gerar os elementos (configurações) da nova população.

A figura 7.4 mostra as operações de "crossover" e mutação para duas configurações de uma rede simples e usando a codificação binária. Nesta, observa-se em primeira instância o procedimento de "crossover" entre configurações, para isto seleciona-se o ponto de recombinação que neste caso corresponde ao quarto bit; então, o bloco identificado entre a posição seguinte e o último bit será trocado entre as duas configurações, neste caso a troca de blocos entre

os dois cromossomos pais só inclui a linha 2-3. No passo seguinte observa-se o procedimento de mutação, o qual é determinado pela taxa de mutação; neste caso em cada configuração se efetuará uma mutação. Na primeira configuração foi selecionada a terceira posição que corresponde a linha 1-3, adicionando duas novas linhas; na segunda configuração foi selecionada a segunda posição, que corresponde à linha 1-2, retirando uma linha.

7.3.3 Geração da População Inicial

No caso do problema de planejamento de redes de transmissão gera-se a população inicial de duas maneiras. A primeira corresponde a uma forma aleatória na qual se controla o número de posições diferentes de zero, assim como também só é gerado um número aleatório de linhas para ser armazenadas nessas posições que é igual ou inferior ao máximo número de linhas. Tipicamente o número de posições diferentes de zero na configuração final está na faixa de [15 - 20]% do total de posições; assim, para a geração aleatória, só são ocupadas entre [30 - 40]% das posições. Este tipo de inicialização aleatória será empregada em sistemas pequenos (sistema Garver) e de médio porte (sistema Sul brasileiro). Com estes sistemas se chega, com alguma frequência, à solução ótima, inicializando com configurações geradas de maneira aleatória.

Para sistemas de grande porte, como o Norte-Nordeste brasileiro, mostra-se pouco efetivo, devido ao grande número de regiões no espaço de soluções que contém esta rede. Iniciando o processo com configurações geradas de maneira aleatória, o mesmo pode ficar preso em alguma das regiões, apresentando grande dificuldade para passar às regiões de melhor qualidade. Isto se deve ao fato de que, iniciando o processo e até conseguir chegar às regiões mais atrativas, teria que enfrentar mudanças bruscas entre regiões, o qual torna o problema de difícil resolução.

Para algumas aplicações, especialmente sistemas de grande porte, pode ser conveniente inicializar usando métodos aproximados. Dentro das técnicas aproximadas mais conhecidas para gerar configurações iniciais no estudo do planejamento das redes de transmissão estão os métodos:

- Método de mínimo esforço [3].
- Método de mínimo corte de carga [4].
- Método de Garver para o modelo de transportes [2].

As características destas técnicas é a de gerar configurações com pequeno esforço computacional e com valores de custo de investimento de relativamente boa qualidade com relação as melhores soluções já conhecidas. Todos esses algoritmos inicializadores geram configurações num processo passo a passo como foram amplamente analisados no capítulo 3.

Após efetuar diversas simulações foi selecionada a metodologia de Garver por apresentar configurações de melhor qualidade especialmente na alocação adequada de linhas de grande capacidade de transmissão nas primeiras fases do processo de adição. Além disso, como a metodologia de Garver leva em conta somente a primeira lei de Kirchhoff, as configurações obtidas



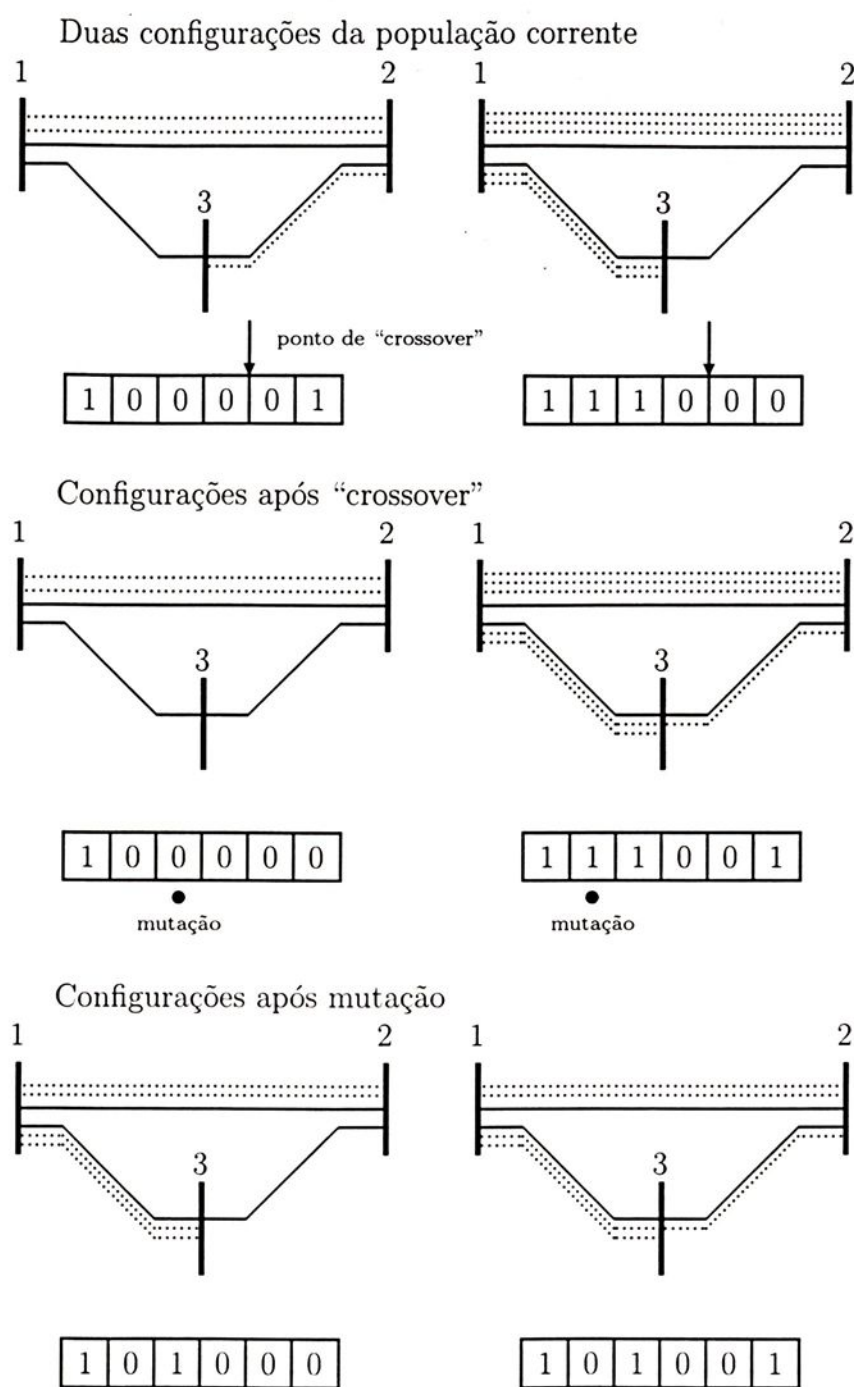


Figura 7.4: "Crossover" e mutação no sistema binário

não adiciona linhas pequenas, permitindo que esta parte da alocação seja adequadamente executada pelo algoritmo genético. Mais adiante é descrito o inicializador de configurações baseado na metodologia de Garver.

7.3.4 Parâmetros de Controle e Critérios de Parada

A escolha ótima dos parâmetros de controle de um algoritmo genético é um tópico que ainda apresenta muitos pontos polêmicos entre os especialistas. Estes parâmetros não podem ser selecionados de maneira independentemente pois os mesmos estão estreitamente relacionados.

O bom desempenho de um algoritmo genético depende em grande parte de uma boa escolha dos parâmetros de controle, taxa de "crossover", taxa de mutação e tamanho da população; além dos anteriores, é também importante uma adequada seleção do tipo de "crossover" (simples, duplo ou múltiplo) e a maneira como será feita a mutação. Com relação ao tamanho da população, esta dependerá do sistema analisado. A literatura recomenda uma população entre [30 - 200].

Em [19, 34] foram estudados sistemas de pequeno, médio e grande porte representando redes elétricas de diferentes tamanhos. Nos três casos se empregou uma taxa de "crossover" de 0,9. Com relação a taxa de mutação, estas são geralmente altas e variáveis durante o processo. Assim por exemplo, a rede pequena foi estudada com taxas de mutação que variam entre [4 - 8]% por cromossomo (configuração), a de médio porte com taxas entre [2 - 4]%, por cromossomo. Já para redes de grande porte, foram estudadas dois tipos de mutação: na primeira, é feita uma mutação como nos casos anteriores, com uma taxa de mutação por cromossomo entre [1 - 5]%; na segunda, foi estudada um tipo de mutação controlado por "simulated annealing".

Com relação ao tamanho da população, este é escolhido de acordo com o tamanho do sistema estudado; ou seja, foram selecionadas faixas baseadas numa faixa mais ampla citada na literatura, nas quais se estudarão os sistemas citados. Para o sistema pequeno, foi escolhido um tamanho da população entre [20 - 30] cromossomos; sistemas médios foram analisados com população entre [50 - 80] cromossomos e sistemas grandes com um tamanho da população entre [160 - 200] cromossomos (configurações).

Como critérios de parada foram empregados dois critérios simples: o primeiro, por número máximo de gerações, e o segundo, se durante as últimas k gerações a solução incumbente não apresenta melhoria.

7.4 Implementações Especializadas no Algoritmo Genético

Além das modificações mencionadas no algoritmo genético básico é possível realizar outras implementações diretamente relacionadas com as características específicas do problema de planejamento de sistemas de transmissão. Essas modificações são:

1. Implementação de uma taxa de mutação variável.



2. Taxa de mutação controlada por “Simulated Annealing” (SA).
3. Seleção limitada.
4. “Building blocks” e rede inicial não conexa.
5. Geração da população inicial usando métodos aproximados.
6. Parâmetro de penalidade α variável.

7.4.1 Taxa de Mutação Variável

A mutação permite encontrar novas alternativas ou regiões promissoras ao alterar o estado em uma posição da configuração. Este aspecto é mais importante ainda nas fases finais do processo quando a maioria das configurações apresentam funções objetivo muito parecidas, onde uma taxa de mutação relativamente elevada pode produzir em muitas configurações modificações significativas dando mais dinamismo ao processo. Em [19, 34] foram usadas taxas de mutação relativamente elevadas e variáveis.

Os algoritmos genéticos básicos usam taxas de mutação pequenas dando um papel secundário a este operador. Entretanto, novas pesquisas realizadas em problemas reais verificam a importância de se usar uma taxa de mutação mais alta e, ainda mais, variável. Por exemplo, para uma configuração dada é gerado um número aleatório entre 0 e 5, assim esse número aleatório define o número de pontos para mutação. Logo, para cada ponto de mutação escolhido também aleatoriamente, implementa-se uma mutação simples ou dupla na representação decimal, isto é, adiciona-se ou retira-se uma ou duas linhas na configuração do sistema. Esta última decisão (adição ou retirada) permite ainda duas variantes:

1. Implementação simples.
2. Avaliação do corte de carga da configuração candidata, com adição de linhas em caso de existir corte de carga e retirada de linhas caso contrário.

Logicamente esta segunda alternativa requer uma chamada à subrotina de PL. Todas as implementações em relação a mutação antes mencionadas, tem dois objetivos claramente diferenciados:

1. Produzir mudanças bruscas na configuração a fim de romper a homogeneidade das configurações da população, o que é um objetivo muito mais importante do previsto.
2. Recuperar a perda de material genético, isto é, fazer pequenas alterações nas configurações a fim de produzir uma configuração que não é possível obter por “crossover”.



7.4.2 Seleção Limitada

No algoritmo genético básico, cada configuração tem direito a gerar um número de descendentes determinado pela qualidade da função objetivo correspondente. Assim, o número de descendentes de uma configuração não está limitado. Entretanto, esta implementação pode causar convergência prematura do processo. Nas primeiras gerações, geralmente as funções objetivo são de pobre qualidade e a presença de uma ou algumas configurações com função objetivo de boa qualidade pode eliminar o efeito das outras configurações, produzindo a convergência prematura num ótimo local. Este fenômeno é muito conhecido e existem várias formas de controle deste problema [24]. Todas estas formas de controle visam limitar o número de descendentes de cada configuração. Assim, uma forma muito simples de controlar este problema é limitar o número máximo de descendentes para cada configuração independentemente da qualidade de sua função objetivo.

7.4.3 Implementação dos “Building Blocks” Através dos Caminhos Novos

A grande capacidade do algoritmo genético em encontrar configurações ótimas ou quase-ótimas é explicada pela sistemática formação dos chamados *blocos construtivos* (“*building blocks*”) nas configurações da população. Para explicar a formação de “building blocks”, precisa-se da definição de alguns conceitos básicos dos AG.

Um “*schema*” representa um subconjunto de todas as possíveis “strings” que têm os mesmos bits em determinadas posições. Por exemplo, o “schema” $**000$ representa um subconjunto de todos os números binários de cinco dígitos que tem três posições fixas, assim este “schema” representa os seguintes quatro “strings”: {00000, 01000, 10000, 11000}. Relacionado com um “schema”, define-se o *valor médio da função objetivo do “schema”* (“average fitness of the schema”) como sendo o valor médio da função objetivo de todos os “strings” ou elementos do “schema”. Adicionalmente, define-se a *distância de definição* (“defining length”) de um “schema” como a distância entre duas posições fixas mais externas do “schema”. Assim, por exemplo, o “schema” $**011*$ tem “defining length” igual a 2 e o “schema” $1*0**1$ tem “defining length” igual a 5.

Define-se então um “*building block*” como aquele “schema” que tem um valor objetivo de excelente qualidade (“high fitness values”) e pequena distância de definição (“defining length”). Portanto, um “building block” é um excelente “schema” no sentido de que possui armazenada uma informação de excelente qualidade pois tem uma função objetivo de alta qualidade e, além disso, apresenta uma “defining length” pequena, o que significa que essa informação de qualidade *está concentrada*. A pequena “defining length” vai diminuir a chance de que um “building block” seja destruído pela operação de “crossover”. Por exemplo, o “schema” $1*0**1$ pode ser mais facilmente destruído pela operação de “crossover” que o “schema” $**011*$ que tem a informação mais “concentrada”.

A hipótese de “*building blocks*” supõe que, os operadores de “crossover” e mutação, fazem a geração, promoção e justaposição de “building blocks” para formar os “strings” ótimos. O “crossover” tende a conservar a informação genética presente nos “building blocks” das confi-



gurações que devem ser cruzadas e a mutação promove a geração de novos “building blocks”. Este mecanismo de formação de “building blocks” é a *essência fundamental* do funcionamento dos AG.

Em [34], além de implementar todas as operações básicas de geração de “building blocks” através do “crossover” e da mutação, que fazem parte de um algoritmo genético básico, foram realizadas algumas implementações adicionais usando a “essência fundamental” da formação e conservação dos “building blocks”. Esta implementação foi realizada para uma parte do sistema elétrico, aquela que corresponde aos novos caminhos de adição e às barras isoladas da configuração base. A figura 7.5, que mostra uma pequena parcela de um sistema elétrico, ilustra este caso particular de barras isoladas.

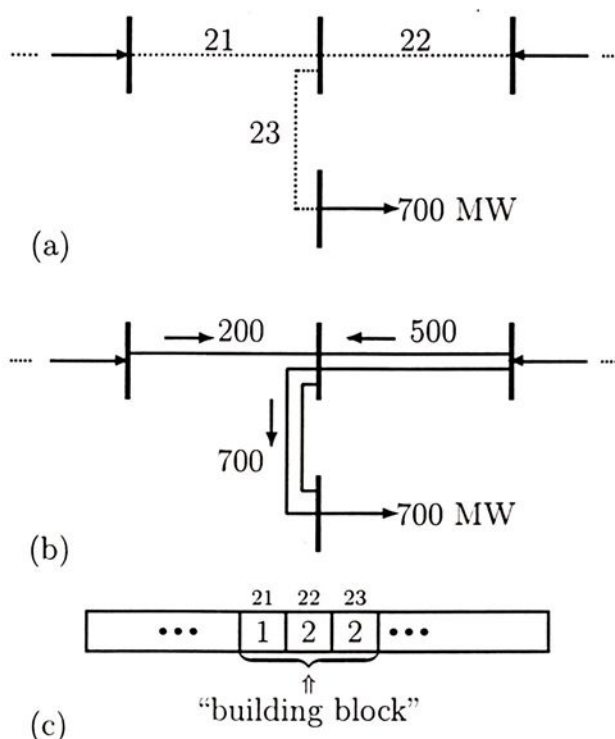


Figura 7.5: “Building blocks” e barras isoladas

Na figura 7.5 (a) aparece uma carga de 700 MW, dois pontos prováveis de fornecimento de potência e três caminhos candidatos. Em (b) aparece a combinação de adições que resolve da maneira mais adequada o fornecimento da carga de 700 MW, e em (c) são apresentadas estas adições em codificação decimal. Nestas condições são importantes as seguintes observações:

1. Para passar da figura 7.5(a) para 7.5(b) na verdade foi gerada uma espécie de “building block”.
2. A combinação de adições nos três caminhos é que constitui uma informação de excelente qualidade e não a informação de adições isoladas.
3. Tanto “crossover” como mutação podem destruir o “building block” da figura 7.5(b).
4. Na figura 7.5(a), na implementação de adições, não faz sentido adicionar linhas ao caminho 23 sem fazer adições simultaneamente ao caminho 21 e/ou 22, pois uma adição isolada em 23 é completamente inútil.

5. Na codificação de uma configuração, normalmente os caminhos 21, 22 e 23 não ocupam casas consecutivas na representação da configuração e no sistema elétrico existem muitos subsistemas como os mostradas na figura 7.5. Assim, com os caminhos 21, 22 e 23 ocupando casas distantes na configuração, os mesmos podem ser facilmente destruídos pelo operador crossover.

Levando em conta as observações anteriores, em [34] foi realizada uma implementação computacional que contorna adequadamente os problemas antes mencionados e aproveita ao máximo a idéia dos blocos construtivos. Assim, foi realizada uma implementação computacional que identifica “building blocks” para a parte do sistema correspondente aos novos caminhos. Um “building block” está constituído por linhas adicionadas em novos caminhos e que são consecutivas ou adjacentes. Desta forma, aparecem “building blocks” em zonas específicas geralmente em torno de grandes barras de carga ou geração. Nesta estratégia, uma vez definida as configurações da população inicial, o primeiro passo é a identificação do número e os elementos dos “building blocks” correspondentes aos novos caminhos. Apontadores adequados identificam os elementos de um “building block”.

A operação de “crossover” não destrói estes tipos especiais de “building blocks” pois todos os elementos dos mesmos entram numa única nova configuração. Esta estratégia também elimina a possibilidade de que o “crossover” forneça configurações com adições inadequadas (linhas que não transmitem potência porque não fecham um caminho). Assim, estes “building blocks” aparecem ou desaparecem com a operação de mutação ou com a eliminação das configurações onde elas estão presentes. É interessante observar que com esta estratégia, estes “building blocks” apresentam “defining length” ideal pois, por definição, estes “building blocks” não são destruídos por “crossover”, mesmo que seus elementos não sejam codificados em casas vizinhas ou consecutivas.

A operação de mutação pode aumentar ou diminuir o número destes “building blocks” presentes nas configurações do sistema, assim como o número de elementos de um “building block” específico. Por exemplo, na figura 7.5(b), uma mutação em 21 e duas mutações sucessivas em 23 levaria à eliminação desse “building block”. Por outro lado, uma mutação em 21 simplesmente levaria a um “building block” com somente 2 elementos. De maneira similar, a mutação pode criar um novo “building block”. Esta implementação dá um papel mais importante à mutação, sendo que deve ser usada uma taxa de mutação maior que as tradicionalmente empregadas.

Mutação pode produzir configurações indesejáveis. Observe na figura 7.5(b) que, se a mutação elimina a adição no caminho 21 e depois elimina as adições no caminho 23, então as adições no caminho 22 são irrelevantes ou indesejáveis. Portanto, na implementação da mutação, deve-se implementar uma estratégia que evite configurações indesejáveis. Desta forma, o algoritmo *contorna um problema*, a adição inadequada de linhas em novos caminhos, e *aproveita* uma característica específica do problema de planejamento formando “building blocks” com linhas vizinhas, aproveitando todas as vantagens que apresentam os “building block” gerados pelo algoritmo genético básico. Embora seja possível generalizar esta idéia de implementar “building blocks” para o sistema elétrico completo, considerando também os caminhos correspondentes a reforços em linhas já existentes, a implementação computacional cresce em complexidade e se torna difícil identificar os blocos independentes. Assim, estes outros caminhos funcionam como “building blocks” de um único elemento e eles formam “building blocks” tradicionais com as operações de “crossover” e mutação convencionais. Testes realizados em sistemas altamente

ilhados mostram uma melhoria significativa dos AG com as implementações mencionadas.

7.4.4 Formas Especializadas de Geração da População Inicial

Existem várias maneiras de determinar a população inicial para o algoritmo genético. Em [34] foi usado um simples, porém eficiente, procedimento de inicialização baseado no algoritmo de Garver [2]. Neste, a rede de transmissão é modelada pelo modelo de transportes (a lei das tensões de Kirchhoff não é representada no modelo, assim o processo fica mais rápido porém a solução é relaxada para o modelo DC). O algoritmo de Garver foi modificado de forma que configurações radicalmente diferentes sejam geradas; com isto se introduz diversidade no material genético que é produzido na etapa inicial do processo de otimização. O algoritmo usado para gerar a população inicial é resumido como segue:

1. Determinar um plano de expansão ótimo usando variáveis de decisão contínuas.
2. Adicionar a parte inteira de todas as linhas/transformadores adicionadas no passo 1 e formar a nova configuração base.
3. Usando a configuração anterior, preparar uma lista de NL linhas/transformadores com investimento fracionário e que nesta configuração transportem os maiores fluxos de potência. (Com cada elemento da lista será obtida uma configuração).
4. Para cada elemento da lista, fazer o que segue:
 - (a) Adicionar um circuito (linha/transformador) que identifica a lista para a configuração base e assim formar a configuração.
 - (b) Usando um algoritmo de PL testar a factibilidade da configuração corrente. Se não existe necessidade de adição de novas linhas/transformadores, então foi encontrada uma configuração que deve ser incorporada à população inicial. Voltar para (a) para analisar outra configuração da lista. Caso contrário, ir a (c).
 - (c) Obter uma nova configuração adicionando à configuração corrente a linha/transformador que transporta maior fluxo de potência, e voltar a (b).

7.4.5 Parâmetro α Variável

O parâmetro α penaliza a função objetivo com corte de carga tornando as configurações propostas pouco atrativas no processo de seleção. O parâmetro α deve assumir um valor relativamente elevado para garantir que o processo pare com uma configuração sem corte de carga. Mas, nas fases iniciais do processo, um α grande produz uma prematura eliminação de configurações com corte de carga e que eventualmente poderiam levar o processo para regiões promissoras. Assim, é desejável que nas fases iniciais do processo as configurações com corte de carga razoáveis tenham chance de sobrevivência por algumas gerações. Isto pode ser obtido com um α variável, iniciando-se com um α menor e incrementando-se seu valor após cada geração.



7.5 Algoritmo Implementado

Em [19] é proposto o seguinte algoritmo genético:

- **Passo 1:** Definir o tamanho da população T_p e determinar a população inicial de forma aleatória, usando configurações conhecidas ou usando o algoritmo de Garver modificado. Definir também k_{max} , o critério de parada que pára o processo se, durante k_{max} gerações, a incumbente não varia, e k_{stop} que é o número máximo de gerações permitidas.
- **Passo 2:** Para cada configuração (“strings”) ou elemento da população, determinar o valor da função objetivo (“fitness”) correspondente, usando o algoritmo de PL. Atualizar, se possível, a incumbente. k_1 conta o número de gerações em que a incumbente permanece sem variar, e k_2 conta o número de gerações.
- **Passo 3:** Se $k_1 > k_{max}$ pare, pois o processo convergiu. Se $k_2 > k_{stop}$ pare, pois foi excedido o número máximo de gerações permitidas. Caso contrário, vá ao passo 4.
- **Passo 4:** Usando o mecanismo de seleção proporcional ou o mecanismo de seleção com limitação de número de descendentes, determinar o número de descendentes (“offsprings”) que corresponde a cada “strings”.
- **Passo 5:** Realizar “crossover” de maneira sistemática entre cada par de configurações que devem participar na formação da nova geração, até gerar um número de descendentes igual ao tamanho da população. A taxa de “crossover” ρ_c controla estatisticamente o número de configurações que serão cruzadas.
- **Passo 6:** Implementar a mutação de maneira sistemática. Cada configuração candidata realiza até n_m mutações (simples ou duplas) onde n_m é proporcional à taxa de mutação ρ_m . Este processo também é controlado estatisticamente pela taxa de mutação ρ_m . Voltar ao passo 2.



Capítulo 8

Planejamento de Sistemas de Transmissão Usando Busca Tabu

Busca Tabu (Tabu Search - TS) é um procedimento metaheurístico utilizado para gerenciar um algoritmo heurístico de busca local evitando que o processo pare em um ótimo local. Assim TS realiza uma exploração através do espaço de configurações contornando adequadamente os ótimos locais.

TS foi inventado por Fred Glover na década de 80 e, na década de 90 está sendo intensamente usado para resolver problemas complexos em áreas muito variadas da pesquisa operacional. Entretanto, a aplicação de TS em problemas de sistemas de energia elétrica ainda se encontra na fase inicial.

Neste Capítulo é desenvolvido um algoritmo de Busca Tabu para ser usado no problema de planejamento de sistemas de transmissão. Assim, inicialmente é apresentada uma análise teórica sobre os fundamentos de TS que na verdade é um resumo de [25], depois são discutidos as características mais importantes das funções de TS que podem ser implementados num algoritmo TS para o problema de planejamento. A continuação são apresentadas as principais características do problema de planejamento que podem ser incorporadas no algoritmo TS.

8.1 Busca Tabu: Teoria

Busca Tabu foi desenvolvido inventando um conjunto de princípios (funções) que, em forma integrada, permitem resolver um problema da maneira mais inteligente. Assim, TS teve sua origem em conceitos que originalmente faziam parte do campo da inteligência artificial. O princípio filosófico de Busca Tabu pode ser explicado com as próprias palavras de seu inventor F. Glover [25, 27] que transcrevemos com pequenas modificações:

*TS está baseado na premissa de que a resolução de um problema pode ser considerado inteligente se esse processo incorpora a **memória adaptativa** e a **exploração sensível**. O uso de **memória adaptativa** contrasta com as técnicas sem memória (como Simulated Annealing e o Algoritmo Genético) e com as técnicas de memória rígida (como as técnicas de inteligência*

artificial e de branch and bound). De igual maneira, a idéia de **exploração sensível** em TS está inspirada na suposição de que uma escolha ruim realizada por uma **estratégia** produz mais informação que uma boa escolha aleatória (numa crítica evidente, por exemplo, a SA que faz escolhas aleatórias). Assim, se a estratégia que guia um algoritmo que usa memória (como TS) faz uma escolha ruim (passa a uma configuração de baixa qualidade) então, pode-se aproveitar essa informação (escolha ruim) para evitar voltar a visitar essa configuração (ruim) e, ainda melhor, para **modificar (melhorar) a própria estratégia** que guia o processo de busca para ter capacidade de encontrar ou escolher configurações de melhor qualidade.

A figura 8.1 mostra os elementos principais (funções) de TS. Portanto, as pesquisas em TS consistem em usar as funções indicadas na figura 8.1 de maneira integrada e eficiente para resolver cada problema específico levando em conta que cada uma das funções podem ser implementadas de várias formas diferentes dependendo das características do problema e do nível de sofisticação da implementação do algoritmo TS. Outra linha de pesquisa muito ativa consiste em inventar outras funções para que sejam incorporadas como parte da estrutura de TS da figura 8.1 e/ou encontrar variantes de implementação das funções mostradas na figura 8.1. Um algoritmo TS para um problema específico pode usar algumas ou todas as funções mostradas na figura 8.1 integradas das formas mas variadas. As características específicas de cada problema e a experiência do pesquisador deve determinar a forma e a qualidade do algoritmo TS implementado. É oportuno lembrar que TS nasceu somente com algumas das funções indicadas na figura 8.1.

Apresentamos brevemente a essência da lógica de trabalho de Busca Tabu. TS resolve problemas genéricos do tipo:

$$\begin{array}{ll} \text{Min} & f(x) \\ \text{s.a.} & x \in X \end{array} \quad (8.1)$$

TS resolve (8.1) iniciando com um processo similar a qualquer algoritmo heurístico de busca local. Na busca local, dada uma configuração x (solução), define-se uma vizinhança de x como sendo o conjunto de todas as configurações $x' \in N(x)$ que podem ser obtidos pela aplicação de um mecanismo de transição a partir de x . A estrutura de vizinhança define as condições para que x' seja vizinho de x . Assim, por exemplo, no problema de planejamento de sistemas de transmissão se pode definir como vizinho de uma configuração x todas aquelas configurações que podem ser obtidas a partir de x pela adição de um circuito, a retirada de um circuito ou troca de dois circuitos (retirada de um circuito e adição de outro circuito) como foi implementado em SA. No **algoritmo de busca local**, a partir da configuração corrente, passa-se para a configuração vizinha que apresenta uma maior diminuição da função objetivo. Um procedimento repetitivo desta estratégia leva ao algoritmo de busca local a parar no momento em que não existe nenhuma configuração vizinha que produz uma diminuição da função objetivo o que significa que foi encontrado um ótimo local.

TS é diferente a um algoritmo de busca local em dois aspectos fundamentais:

1. A partir da configuração corrente, passa-se à melhor configuração vizinha ou à menos pior o que implica que é permitida uma degradação da qualidade da função objetivo.

PRINCIPAIS CARACTERÍSTICAS DE BUSCA TABU

Memória Adaptativa

Seletividade (incluindo esquecimento estratégico)

Abstração e decomposição (usando memória explícita e por atributos)

Tempo:

Recência de eventos

Frequência de eventos

Diferenciação entre curto e longo prazo

Qualidade e impacto:

Atração relativa quanto a escolhas alternativas

Impacto de mudanças de relações em estrutura ou restrições

Contexto

Interdependência regional

Interdependência estrutural

Interdependência sequencial

Exploração Sensível

Imposição estratégica de proibições e induções
(*condições tabu e níveis de aspiração*)

Enfoque concentrado em boas regiões e em boas características das soluções
(*processos de intensificação*)

Caracterização e exploração de novas regiões promissoras
(*processo de diversificação*)

Padrões de busca não monótonos
(*oscilação estratégica*)

Integração e geração de novas soluções
(*path relinking*)

Figura 8.1: Funções de TS

2. O conjunto de vizinhos de x não se caracteriza de maneira estática. Assim, TS define uma nova estrutura de vizinhança, $N^*(x)$ que varia **dinamicamente em estrutura e tamanho** durante todo o processo de otimização. Esta estratégia permite a TS realizar uma busca eficiente e inteligente. A figura 8.2 mostra uma relação entre $N(x)$ e $N^*(x)$. Os elementos de $N^*(x)$ são determinadas de varias formas. A maneira de ilustração, pode-se indicar as seguintes formas:

- Usando uma lista tabu que armazena atributos de configurações consideradas tabu (proibidas). Neste caso $N^*(x) \subset N(x)$ pois alguns vizinhos definidos pela estrutura de vizinhança e cujos atributos fazem parte da lista tabu estão proibidos. Esta estratégia evita retornar a configurações já visitadas e evita o chamado ciclagem.
- Usando estratégias para diminuir a vizinhança ou a lista de configurações candidatas. Geralmente o número de configurações $x' \in N(x)$ pode ser muito grande e avaliar a função objetivo de todas essas configurações para encontrar aquela que apresenta melhor desempenho pode precisar de elevado esforço computacional. TS apresenta pelo menos 4 métodos diferentes para encontrar uma vizinhança de tamanho reduzido tal que $N^*(x) \subset N(x)$. Esta estratégia visa diminuir o esforço computacional necessária em cada transição.
- Usando configurações de elite e path relinking para caracterizar e encontrar novas configurações candidatas. Esta estratégia visa encontrar novas configurações de alta qualidade que dificilmente seriam encontradas a partir da definição de $N^*(x)$. é fácil verificar que neste caso a relação $N(x) \subset N^*(x)$ já não é mais verdadeira.
- Redefinir o conjunto $N(x)$ durante o processo de otimização.

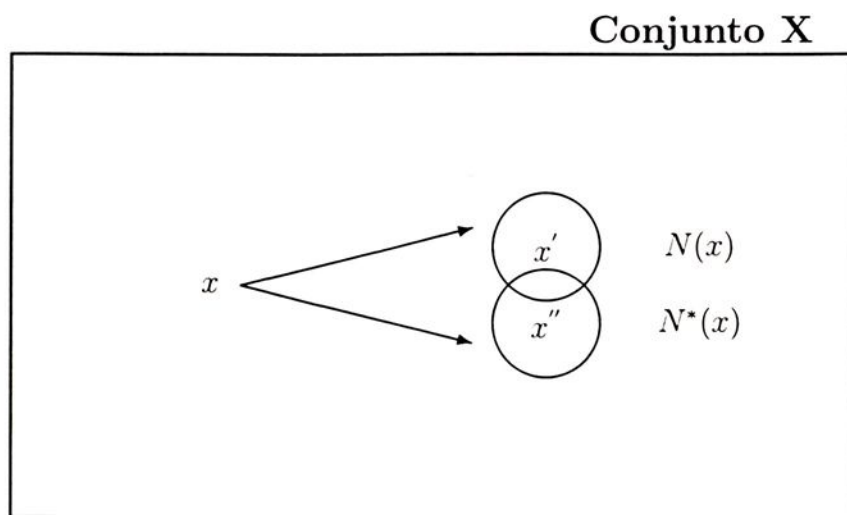


Figura 8.2: Transição em TS

Apresentamos a seguir as quatro principais estratégias de TS: memória de curto prazo, memória de longo prazo, oscilação estratégica e path relinking.

8.1.1 Memória de Curto Prazo e seus Principais Elementos

Um algoritmo TS pode ser implementado simplesmente usando a estratégia de curto prazo e muitas das pesquisas iniciais foram implementadas desta forma. A memória de curto prazo usa basicamente a informação de atributos de configurações que foram modificados no passado recente. Esta informação é conhecida como memória baseada em recência (recency) ou em fatos recentes. A idéia desta estratégia é considerar todos os atributos selecionados no passado recente como sendo proibidos, portanto, todas as configurações candidatas que possuem algum dos atributos proibidos (tabu-ativos) são excluídos na conformação do conjunto $N^*(x) \subseteq N(x)$ de configurações vizinhas para avaliação.

A idéia básica da memória baseada em recência é evitar revisitar configurações já visitadas e aquelas configurações que compartilham os atributos tabu-ativos. Esta última parte pode representar um problema pois podem ser eliminados de $N^*(x)$ configurações muito atrativas. O critério de aspiração contorna em parte esta limitação. No problema de planejamento de sistemas de transmissão os atributos podem ser os circuitos que fazem parte de uma configuração. Assim, por exemplo, um **circuito que é adicionado** na configuração deve permanecer proibido de ser **retirado** nas próximas k configurações geradas, sendo k um número inteiro escolhido. De igual maneira, um circuito que é retirado da configuração corrente deve permanecer proibido de ser adicionado nas próximas k configurações. É evidente que nesta implementação devem existir duas listas de atributos proibidos.

A figura 8.3 mostra uma iteração do algoritmo TS que usa memória de curto prazo sendo que uma iteração significa que, a partir da configuração corrente, deve-se avaliar todas as configurações candidatas $x' \in N^*(x)$ e escolher a melhor como sendo a nova configuração corrente.

Na parte **A** da figura 8.3 é gerada todas as configurações candidatas $x' \in N(x)$. Na parte **B** é acionada a lista tabu. Podem ser criadas uma ou várias listas tabu dependendo do tipo do problema e do nível de sofisticação do algoritmo. A lista tabu pode ser operada de duas formas: 1) uma implementação continua que não elimina automaticamente uma configuração candidata que possui atributos proibidos; assim configurações deste tipo são penalizadas na função objetivo (o seu equivalente) em **C** para que se tornem pouco atrativas, e 2) uma implementação discreta em que toda configuração com atributos proibidos são eliminados automaticamente. Neste último caso, deve-se eliminar o bloco **D** e da saída de **C** voltar diretamente ao bloco **A**. O número k de iterações que um atributo permanece proibido é chamado de status tabu (tabu tenure) e o tamanho de k pode variar em cada lista e durante o processo produzindo algoritmos TS com diferentes tipos de desempenho.

No bloco **C** é implementado o chamado **critério de aspiração** cuja finalidade é fornecer maior flexibilidade ao processo. Neste caso uma configuração que possui atributo(s) proibido(s) pode ter a proibição descartada se essa configuração satisfaz um critério de aspiração. Na verdade aqui se tenta contornar a “rigidez” da lista tabu que armazena atributos proibidos. Uma configuração candidata de excelente qualidade pode ter um atributo proibido porque compartilha esse atributo com uma configuração já visitada no passado recente. Assim, a lista tabu ao proibir revisitar configurações visitadas no passado recente pode também proibir visitar configurações novas de excelente qualidade. Este fato é contornado pelo critério de aspiração que elimina a proibição de uma configuração candidata se ela produz uma função objetivo de



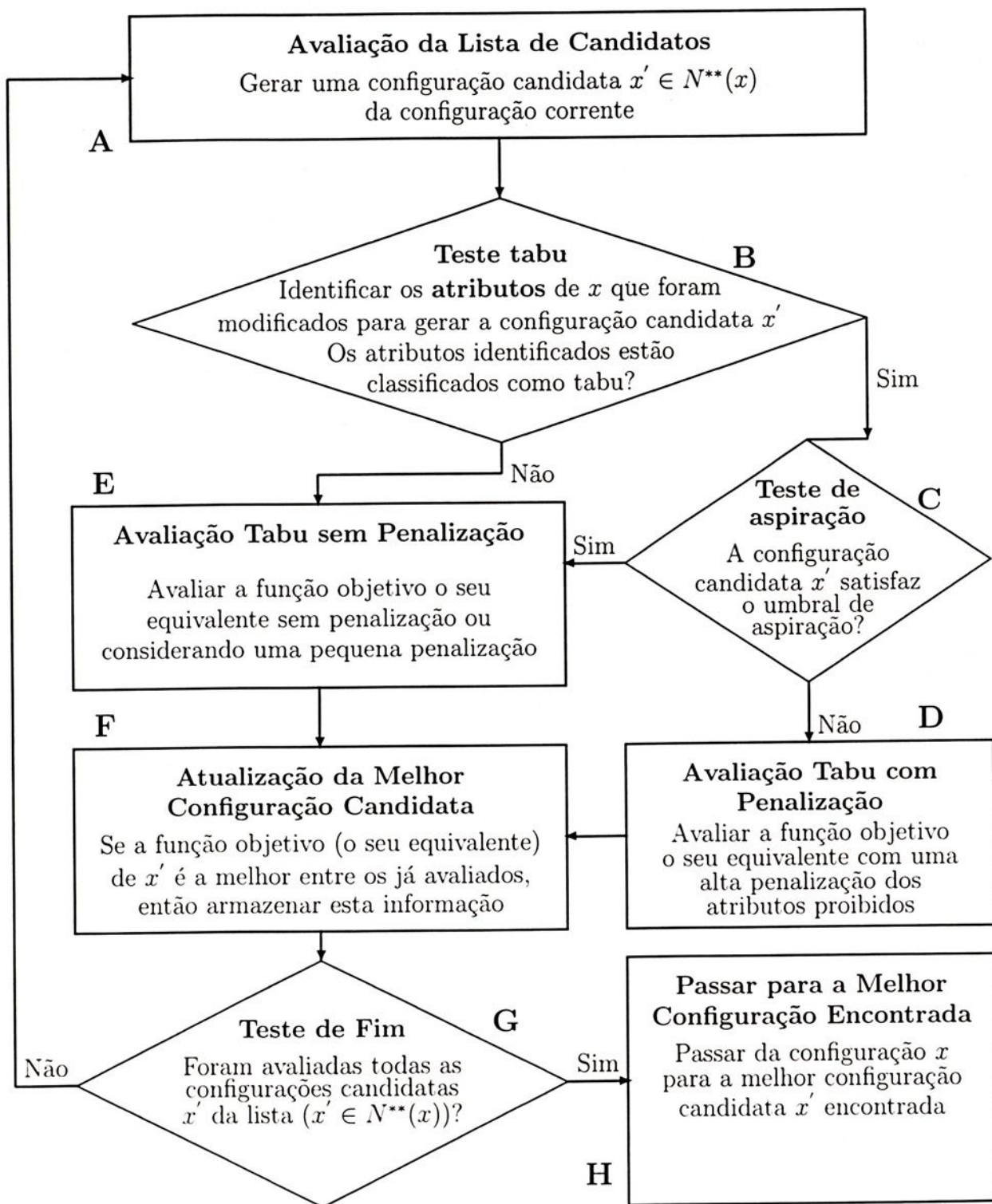


Figura 8.3: Execução de um movimento TS com memória de curto prazo

boa qualidade.

Existem várias formas de definir o critério de aspiração. Obviamente, o critério mais rígido é definir o critério de aspiração da seguinte forma: se uma configuração candidata produz uma função objetivo melhor que a incumbente (melhor configuração já encontrada) então deve-se eliminar a proibição e armazenar essa configuração candidata no bloco **F**.

Na figura 8.3, no bloco **E** é realizada a avaliação da função objetivo (o seu equivalente) para cada configuração candidata. No bloco **F** é armazenada a melhor configuração candidata encontrada, no bloco **G** é verificada se já todas as configurações candidatas foram analisadas e em **H** é realizada a transição em que a configuração corrente é substituída pela melhor configuração candidata, isto é, a configuração corrente é substituída pela melhor configuração candidata encontrada.

Existe um último aspecto importante no algoritmo TS com memória de curto prazo: **a diminuição do tamanho de configurações candidatas que devem ser analisadas**. Para ilustrar este fato sejam: $N(x)$ as configurações candidatas definidas pelo critério ou a estrutura de vizinhança, $N^{**}(x)$ um subconjunto reduzido de $N(x)$ obtido usando algum critério de redução de configurações candidatas e $N^*(x)$ os elementos de $N^{**}(x)$ que não estão proibidos pela lista tabu. Para o problema de planejamento de sistemas de transmissão podemos definir, por exemplo, $N(x)$ como todas as configurações que podem ser obtidas a partir da configuração corrente da seguinte forma: adicionando um circuito, retirando um circuito ou trocando dois circuitos (adicionando um circuito e retirando outro circuito previamente adicionado). Para um sistema real esta definição de vizinho significa que $N(x)$ teria vários milhares de configurações candidatas e como a avaliação de cada configuração candidata precisa da resolução de um problema de PL, é evidente que esta definição de vizinho é impraticável pois precisaria da resolução de vários milhares de PL para realizar um simples passo do algoritmo TS. Uma forma de contornar este problema seria redefinir a vizinhança.

TS contorna o problema anterior de uma forma mais eficiente encontrando um subconjunto reduzido $N^{**}(x)$ de $N(x)$, por exemplo da ordem de algumas dezenas no caso do problema de planejamento. TS possui quatro estratégias para realizar este processo de redução de configurações candidatas: aspiração plus, conformação de uma lista reduzida de configurações candidatas de elite, filtros sucessivos e estratégia de mudança acotada. Entretanto, é possível implementar uma estratégia de redução de configurações candidatas ainda mais eficiente para o problema de planejamento de sistemas de transmissão e cujos detalhes serão apresentadas na próxima seção.

8.1.2 Memória de Longo Prazo e suas Estratégias Associadas

A incorporação de uma estratégia de longo prazo fornece ao algoritmo TS uma sofisticação adicional e um desempenho melhor na maioria das aplicações. Isto significa que além da memória de curto prazo (estratégia de todo algoritmo TS básico) é incorporada uma estratégia adicional chamada de estratégia de longo prazo no algoritmo TS.

Existem três aspectos fundamentais relacionados com a memória de longo prazo: 1) a memória baseada em frequência, 2) a estratégia de intensificação, e 3) a estratégia de diversifi-



cação.

Memória Baseada em Frequência

A **memória baseada em frequência** consiste basicamente em armazenar a informação do número de vezes em que um atributo foi escolhido para gerar ou participar na conformação das configurações durante o processo TS. Assim existem dois tipos de memória baseadas em frequência: as **frequências de transição** que armazenam o número de vezes em que um atributo é retirado ou adicionado para formar novas configurações, e a **frequência de residência ou permanência** que armazena a informação do número de vezes em que um atributo permanece nas novas configurações ou em todas as configurações geradas durante o processo TS. Esta informação baseada em frequência pode ser utilizada para penalizar ou incentivar configurações conformados com determinados atributos.

No problema de planejamento de sistemas de transmissão é possível armazenar a informação do número de vezes em que um circuito é adicionado ou retirado na conformação das configurações no processo TS (seria a frequência de transição). De igual maneira, também é possível armazenar número de vezes em que um circuito permanece adicionado em configurações consecutivas ou o número de configurações em que um circuito participa com adição (seria a frequência de residência).

Estratégia de Intensificação

A **estratégia de intensificação** consiste em mudar os critérios de seleção das novas configurações candidatas, isto é, mudar $N^{**}(x)$ (ou $N^*(x)$) para incentivar a conformação de novas configurações aproveitando a informação acumulada no processo TS. Intensificação pode ser implementada de várias formas diferentes tais como:

- Eliminando vizinhos ou incorporando novos vizinhos x' para a configuração corrente x , tipicamente considerando configurações de elite ou configurações com esses atributos como sendo "vizinhos" de x .
- Modificando a caracterização de configuração vizinha, isto é, redefinindo o conjunto $N(x)$.
- Retornando a regiões atrativas para realizar uma busca mais intensa na vizinhança dessas regiões atrativas. A figura 8.4 mostra uma estratégia para esta alternativa.
- Realizando intensificação por decomposição.

Estratégia de Diversificação

A estratégia de diversificação foi projetada para levar o processo de busca para regiões novas e atrativas. Esta estratégia é implementada mudando a definição de vizinhança ou de configurações candidatas $N(x)$, incorporando "vizinhos" constituídos por atributos que foram

pouco usados. Nesta parte podem ser usadas as estruturas de memória usadas em intensificação ou em memória de curto prazo mas com critérios de duração modificados.

As estratégias de diversificação podem também usar uma forma de **memória de longo prazo baseado em recência** visando incrementar o estado tabu dos atributos das configurações visitadas.

8.1.3 Oscilação Estratégica

Oscilação estratégica fornece um excelente mecanismo para realizar uma interação entre intensificação e diversificação a medio ou longo prazo. Esta técnica opera quando foi atingido um nível crítico em que o método TS normalmente pararia. Assim, antes de parar após atingido o nível crítico, pode-se mudar as regras de escolha de novas configurações candidatas (mudar $N(x)$) para passar fora do limite do nível crítico. Uma vez passado o nível crítico, muda-se novamente as regras de escolha de configurações candidatas tanto no afastamento como no retorno ao nível crítico. Um processo repetitivo desta estratégia produz um movimento oscilatório em torno do nível crítico.

O exemplo mais simples de oscilação estratégica acontece com o problema da mochila. Neste caso as variáveis binárias são modificadas, por exemplo usando uma estratégia TS com memória de curto prazo, até chegar à fronteira da factibilidade. Nesse momento, pode-se entrar na região infactível, mudando a forma de avaliação da função objetivo ou seu equivalente na região infactível. Um processo de afastamento e retorno para a região de fronteira produz uma busca oscilatória que permitiria encontrar configurações factíveis de excelente qualidade.

No problema de planejamento de sistemas de transmissão, pode-se usar a estratégia de oscilação estratégica de uma maneira mais eficiente. Assim, por exemplo, pode-se iniciar o processo a partir de uma configuração infactível e usar uma estratégia TS com memória de curto prazo até encontrar uma configuração factível. Uma vez encontrada uma configuração factível, pode-se realizar um processo de intensificação aceitando simplesmente configurações factíveis que permitam encontrar o ótimo local correspondente a essa região. Uma vez terminada a intensificação, pode-se implementar uma rápida diversificação para entrar novamente na região infactível e reiniciar a estratégia TS de memória de curto prazo para voltar novamente a uma nova região factível e reiniciando a estratégia de busca oscilatória. Em cada parte do processo devem ser implementadas regras específicas para definir as configurações candidatas e evitar voltar a visitar regiões já visitadas. Obviamente, é possível implementar esta estratégia com um maior nível de sofisticação. Assim, o processo integral TS pode ser um processo de oscilação estratégica modificando as regras em cada fase do processo de oscilação estratégica.

8.1.4 Path Relinking

Path relinking permite uma interessante integração entre intensificação e diversificação. Esta estratégia permite encontrar novas configurações usando as trajetórias que “unem” configurações de elite. Nesta estratégia é escolhida uma configuração de elite chamada de configuração base e uma ou várias configurações de elite chamadas de configurações guia. Assim, pode-se obter novas configurações a partir da configuração base e incorporando atributos presentes nas

configurações guia. As configurações, assim obtidas, podem ser usadas para intensificação ou diversificação dependendo da quantidade de atributos incorporadas na configuração base.

A idéia de path relinking é obter configurações constituídos por atributos de alta qualidade pertencentes a configurações de alta qualidade. Assim, na implementação de path relinking para encontrar uma configuração atrativa para continuar o processo, em cada passo é escolhido, de maneira preferencial, o atributo mais atrativo presente nas configurações guia e usando os critérios usuais de TS.

Na implementação de path relinking, uma vez definida a configuração base (inicial) e as configurações guia de elite, deve-se continuar o processo TS na forma usual mas induzindo a escolha de atributos presentes nas configurações de elite. Este efeito é obtido tipicamente dando uma ponderação especial aos atributos presentes nas configurações de elite “forçando” sua escolha em relação aos atributos que não se encontram presentes nas configurações guia. Neste contexto, implementar path relinking significa fazer uma mudança na caracterização da vizinhança $N(x)$ onde as configurações candidatas que possuem atributos presentes nas configurações guia recebem ponderação especial, por exemplo, incentivando sua função objetivo.

Path relinking também pode ser usada para gerar novas configurações de uma maneira mais rápida. Assim, pode-se incorporar na configuração base e numa única operação todos os atributos considerados atrativos e presentes nas configurações guia. Dependendo da quantidade de atributos incorporados, a configuração assim obtida pode ser usada para intensificação ou diversificação. Esta estratégia foi implementada no problema de planejamento de sistemas de transmissão [34]. Outra estratégia é produzir uma configuração para realizar intensificação usando configurações de elite que possuem muitos atributos parecidos ou então produzir uma configuração para diversificação usando configurações de elite que possuem atributos significativamente diferentes.

8.2 Busca Tabu Aplicado ao Problema de Planejamento

8.2.1 Algoritmo TS Básico

Nesta seção, é apresentado um algoritmo de TS aplicado ao problema da expansão de sistemas de transmissão apresentado em [18, 34]. A figura 8.4 mostra o módulo básico de funcionamento do algoritmo TS.

No esquema mostrado na figura 8.4, o processo de otimização é iniciado com uma configuração inicial que pode ser obtida aleatoriamente mas é muito mais eficiente usar o algoritmo de Garver [2] para encontrar uma boa configuração inicial. No módulo I, esta configuração é submetida a um processo de TS com oscilação estratégica que compreende de três partes claramente diferenciadas: (1) processo TS com memória de curto prazo, listas tabu e critério de aspiração. (2) intensificação e (3) pequena diversificação. Esta parte do processo, que corresponde à parte principal do algoritmo TS, é mais amplamente apresentada posteriormente. Finalmente, no módulo II, é realizada uma estratégia de diversificação que pode ser implementada de várias formas e depois o processo pode retornar ao módulo I num processo cíclico até que seja satisfeito um critério de parada. Portanto, a estratégia mostrada na figura 8.4 corres-



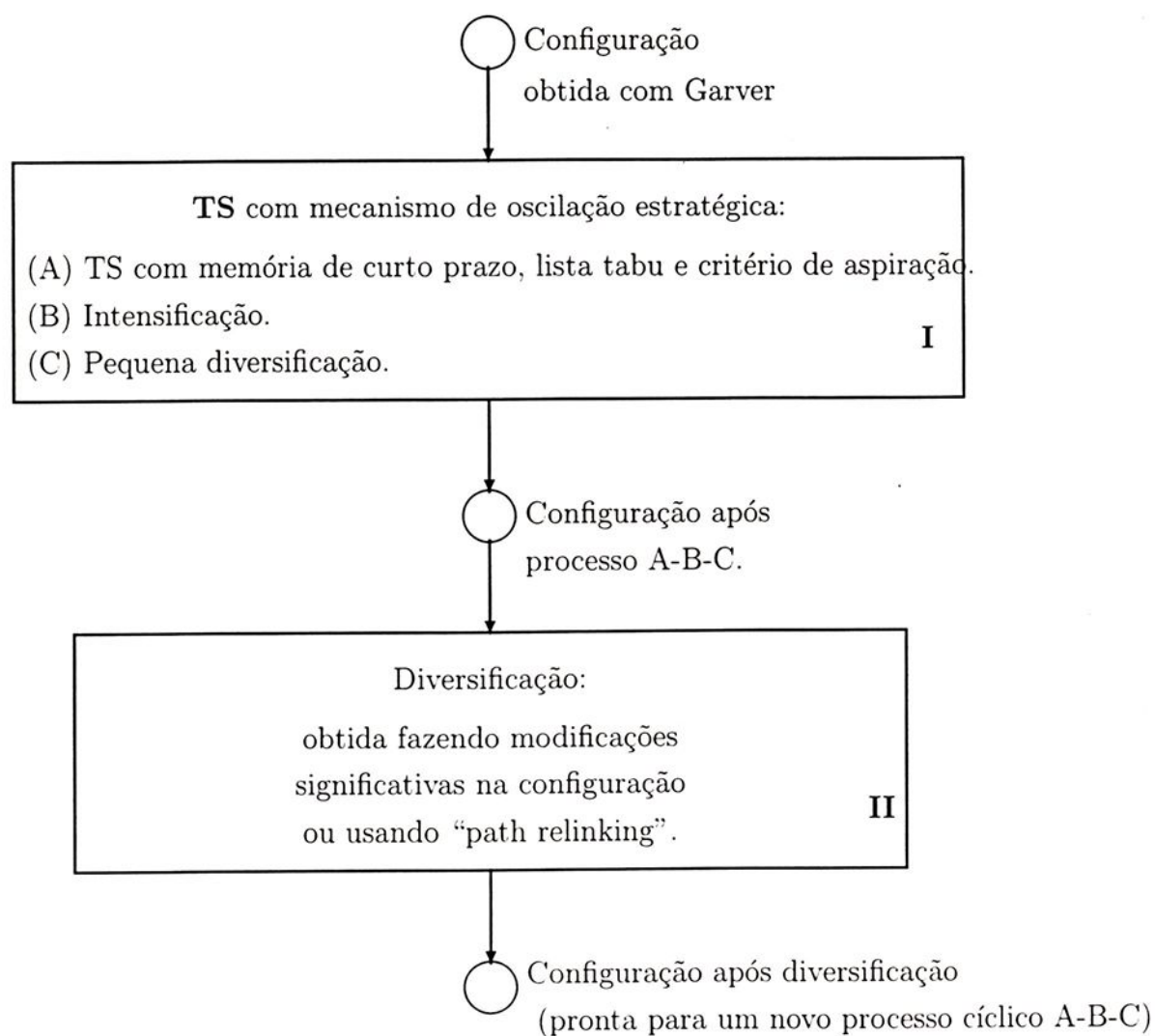


Figura 8.4: Mecanismo de "Tabu Search"

ponde a uma forma de integrar algumas das funções TS mostradas na figura 8.1 para resolver o problema de planejamento de sistemas de transmissão propostas em [18, 34]. Evidentemente, podem ser desenvolvidas outras estratégias TS para resolver um mesmo tipo de problema levando a versões diferentes do algoritmo TS. é importante analisar detalhadamente cada uma das partes da estratégia apresentada na figura 8.4.

Determinação de uma Configuração Inicial

Foi verificado experimentalmente que os algoritmos combinatoriais apresentam melhor desempenho quando os processos são iniciados com boas configurações iniciais em lugar de configurações obtidas aleatoriamente. Esta característica é mais importante ainda no caso do algoritmo TS. Em [18], pode-se observar que boas configurações iniciais diminuem o esforço computacional do processo e frequentemente encontram configurações de melhor qualidade.

Em [18] foi usado o algoritmo de Garver para encontrar boas configurações iniciais mas podem ser usados os outros algoritmos heurísticos construtivos. O algoritmo de Garver é particularmente adequado para ser usado no algoritmo TS apresentado em [18, 34] porque nas configurações de Garver, que satisfazem somente a primeira lei de Kirchhoff, aparecem as adições mais importantes do sistema elétrico e essas configurações são inactíveis para o modelo DC e portanto adequados para iniciar o processo **A**. Assim, embora essas configurações sejam inactíveis estão muito perto da região factível e possuem muitos atributos de alta qualidade incorporados.

Também é possível usar outros algoritmos heurísticos construtivos que usam o modelo DC e nesse caso o algoritmo seria iniciado no processo **B**. Entretanto, em sistemas de grande porte esses algoritmos adicionam muitos circuitos de pequena importância e sua retirada pode precisar de um significativo esforço computacional mas essa alternativa também deveria ser avaliada experimentalmente para verificar seu desempenho.

Processo de Oscilação Estratégica

O processo de oscilação estratégica mostrado no módulo I da figura 8.4. Os três processos chamados **A**, **B** e **C** realizam as seguintes tarefas:

- A parte do processo indicada por **A** corresponde ao processo TS com memória de curto prazo, listas tabu e critério de aspiração e nesta parte do processo as transições são realizadas através de configurações inactíveis e a estratégia consiste em encontrar uma configuração factível de boa qualidade e, portanto, essa parte do processo termina quando é encontrada essa configuração factível. Nesta parte a forma de determinação de vizinhos candidatos segue uma estratégia diferente como é apresentada posteriormente.
- A parte do processo indicada por **B** corresponde a um conjunto de transições realizadas através de configurações factíveis. Assim, o processo **B** corresponde a uma intensificação em torno de uma boa configuração factível, portanto, o critério de vizinhança nesta parte do processo também deve ser diferente e deve permitir somente transições entre configurações factíveis, isto é, não é permitida configurações com corte de carga.

- A parte do processo indicada por **C** corresponde a uma simples transição que deve produzir o retorno à região infactível. Assim, nesta parte do processo o critério de vizinhança também deve ser diferente assim como a atualização das listas tabu. A idéia básica é sair da região factível e iniciar um novo retorno à região factível (novo processo **A**) mas as listas tabu devem proibir o retorno à região factível já visitada. O processo cíclico A-B-C, que corresponde ao mecanismo de oscilação estratégica, deve ser implementado até satisfazer um critério de parada que pode ser um número máximo de chamadas ao algoritmo de PL e/ou uma ausência de melhoria na função objetivo num número determinado de processos cíclicos. Nesse caso, deve-se implementar a diversificação proposta no módulo II.

Processo de Diversificação

Na figura 8.4 o processo TS poderia terminar após o módulo I. Entretanto, processos mais sofisticados do algoritmo TS deve incorporar outras funções TS como a diversificação o que é realizado no módulo II do algoritmo TS apresentado. Diversificação pode ser implementada de duas maneiras diferentes:

- Realizando algumas modificações na configuração através de adições, retiradas e/ou trocas de linhas ou penalizando atributos.
- Fazendo “path relinking”.

A primeira forma de diversificação consiste em realizar pequenas modificações na topologia da configuração através de adições, retiradas ou trocas de circuitos. Outra forma consiste em penalizar por algumas transições os atributos muito acionados durante o processo integral, informação que está presente nas listas de memória baseada em frequência, para priorizar o uso de outros atributos pouco usados levando o processo a regiões não visitadas e potencialmente atrativas.

Uma outra forma de realizar diversificação é usando “path relinking”. Neste caso é realizado “path relinking” entre a configuração candidata à diversificação e uma das configurações de elite. A decisão de realizar uma das duas formas de diversificação pode ser realizada em forma aleatória ou de maneira alternada.

8.2.2 Implementações Especializadas no Algoritmo de Busca Tabu

Em [18, 34] são realizadas algumas implementações especializadas em determinadas partes do algoritmo TS para tornar o algoritmo mais eficiente. A seguir são comentadas brevemente essas implementações especializadas.

Algoritmo TS com Configuração Única de Partida

A estrutura do algoritmo TS proposto na figura 8.4 pode ser incorporado num esquema mais dinâmico como o apresentado na figura 8.5. O algoritmo proposto na figura 8.5 é um pouco diferente ao algoritmo e essa diferença consiste nos seguintes aspectos:

- Ambos algoritmos iniciam o processo com uma configuração inicial de partida. Entretanto, na implementação do módulo I da figura 8.5 que corresponde ao processo TS com memória de curto prazo com lista tabu e critério de aspiração, adicionalmente é gerada uma lista de configurações de elite. As configurações de elite são armazenadas numa lista. Assim, ao entrar no módulo II, a diversificação e o novo processo TS com memória de curto prazo é iniciado a partir de uma configuração de elite antes que iniciar com a chamada configuração corrente.
- No algoritmo na figura 8.5 as configurações de elite são armazenadas numa lista de tamanho k e essas configurações são atualizadas durante todo o processo. Uma configuração encontrada é considerada de elite se é melhor que alguma configuração de elite já armazenada e possui um número mínimo de atributos diferentes. Assim, o processo de diversificação - algoritmo TS com memória de curto prazo é iniciado a partir da melhor configuração de elite. O processo geral termina quando todas as configurações de elite já foram usadas para reiniciar o processo ou quando foi atingida um número máximo de chamadas ao algoritmo de PL.
- No final do processo, o algoritmo da figura 8.5 apresenta um conjunto de K configurações de elite além da melhor configuração encontrada o que é interessante em trabalhos de planejamento. Assim, o novo algoritmo proposto é diferente do algoritmo apresentado na figura 8.4 pela incorporação das chamadas configurações de elite no processo TS.

Um nível adicional de sofisticação do algoritmo TS usado no problema de planejamento de sistemas de transmissão pode ser obtido usando um conjunto de k configurações iniciais para iniciar o processo TS.

O Problema da Estrutura de Vizinhança

Na figura 8.2 foi apresentada as dificuldades computacionais do algoritmo TS para realizar uma transição, isto é, para passar da configuração corrente para a melhor configuração candidata ou vizinha. O caráter agressivo do algoritmo TS exige que seja escolhida a melhor configuração vizinha não proibida o que implica avaliar todas as configurações vizinhas. Anteriormente foram mencionadas várias técnicas de redução de vizinhos candidatos para algoritmos TS de caráter geral. Entretanto, em [18, 34] são propostas técnicas que permitem encontrar um número reduzido de configurações candidatas de excelente qualidade. Para o algoritmo TS proposto anteriormente, as configurações vizinhas candidatas são determinadas de maneira diferente para cada uma das etapas do processo TS com memória de curto prazo, lista tabu e critério de aspiração, isto é, para cada uma das partes do processo **A - B - C** do processo de oscilação estratégica. Assim, em cada uma dessas etapas a lista reduzida de vizinhos candidatos é encontrada da seguinte maneira:



1. Determinação de vizinhos durante a estratégia A:

Nessa parte do processo as configurações apresentam corte de carga e as configurações vizinhas devem ser obtidas de adições/trocas de circuitos a partir da configuração corrente. Como a configuração corrente apresenta corte de carga e a estratégia consiste em encontrar uma configuração sem corte de carga então é encontrado uma lista de configurações vizinhas obtidas da seguinte forma:

- (a) As melhores adições propostas pelos métodos heurísticos construtivos propostos em [2, 3, 4], isto é, algoritmo de Garver que usa o modelo de transportes [2], o método de mínimo esforço que usa o modelo DC [3] e o método de mínimo corte de carga [4] que também usa o modelo DC. Assim são usadas as propostas sugeridas pelos indicadores de sensibilidade desses algoritmos amplamente analisados no capítulo 3. Assim, é necessário mencionar a importância desses algoritmos na geração de configurações iniciais assim como na determinação de configurações vizinhas de excelente qualidade. Em cada caso podem ser usadas a melhor ou as melhores propostas desses algoritmos. Esta informação pode ser obtida ao custo de uma chamada ao algoritmo de PL.
- (b) Circuitos ou conjunto de circuitos vizinhos de barras onde existem grandes perdas de carga ou geração (corte de carga ou geração limitada porque não existem linhas suficientes saindo da barra de geração). Esta informação sempre está disponível sem custo computacional adicional.
- (c) Uma lista pequena de adições e adições/trocas de circuitos aleatoriamente escolhidas.

2. Determinação de vizinhos durante a estratégia B:

Esta parte do processo corresponde a transições entre configurações factíveis, isto é, sem corte de carga. Portanto, como a estratégia consiste em encontrar configurações sem corte de carga na vizinhança da configuração sem corte de carga corrente então a idéia é realizar pequenas alterações na topologia corrente o que equivale a realizar uma intensificação visando encontrar um ótimo local da região. Assim, a lista de configurações vizinhas da configuração corrente neste caso é formada como segue:

- (a) Retirada de circuitos de baixo custo.
- (b) Retirada de alguns circuitos escolhidos aleatoriamente.
- (c) Troca de circuitos de baixo custo.
- (d) Troca de alguns circuitos escolhidos aleatoriamente.

3. Determinação de vizinhos durante a estratégia C:

A parte C da estratégia consiste em remover alguns circuitos da configuração corrente sem corte de carga (tipicamente as mais baratas) excluindo os circuitos adicionados na parte B para evitar ciclagem. Com esta implementação o processo entra novamente na região infactível e, deve-se repetir o processo cíclico A-B-C de oscilação estratégica. Assim, esta estratégia de um único passo escolhe um ou vários circuitos que devem ser retirados da configuração corrente.



Implementação do “Path Relinking”

O “path relinking” é usado para encontrar configurações de excelente qualidade para reiniciar o processo de busca. A forma mais simples é obter uma excelente configuração fazendo “path relinking” entre configurações de elite encontrando uma configuração que possui atributos de duas ou mais configurações de elite.

A forma mais simples de fazer “path relinking” é identificando o número de posições na configuração onde existem adições diferentes nas duas configurações submetidas a “path relinking”. Portanto, assumindo uma das configurações como referência, copia-se uma parcela das posições diferentes que estão presentes na outra configuração e assim gera-se uma nova configuração, que na verdade é a configuração de referência com alguns de seus atributos modificados usando a outra configuração. Obtém-se assim uma nova configuração potencialmente atrativa.



Capítulo 9

Principais Contribuições em Pesquisas do Problema de Planejamento de Sistemas de Transmissão

Neste capítulo são apresentadas nossas principais contribuições em pesquisas relacionadas com o problema de planejamento de sistemas de transmissão assim como as pesquisas que estão sendo desenvolvidas no momento. Também são mencionadas as principais conclusões e possibilidades de pesquisa no futuro.

As contribuições podem ser separadas em três partes para facilitar a apresentação: (1) contribuições na formulação básica, (2) contribuições no desenvolvimento de algoritmos de solução dos modelos matemáticos e, (3) contribuições no desenvolvimento de algoritmos de problemas correlatos.

9.1 Contribuições na Modelagem Matemática

A modelagem matemática do problema de planejamento de sistemas de transmissão apresentaram uma grande evolução desde as primeiras formulações apresentadas por Garver em [2]. Portanto, a evolução da modelagem matemática até assumir a forma matemática com os três modelos apresentados no capítulo 2 contou com a participação de vários pesquisadores sendo difícil identificar o aporte de cada pesquisador. Entretanto, pode-se mencionar que os maiores aportes foram apresentados em [2, 3, 7, 30, 32]. De nossa parte, pode-se mencionar apenas pequenas contribuições na melhor interpretação do modelo híbrido apresentado em [14].



9.2 Contribuições no Desenvolvimento de Algoritmos de Solução

Os algoritmos de solução desenvolvidos para resolver o problema de planejamento de sistemas de transmissão podem ser agrupados em três grandes grupos: (a) algoritmos heurísticos basicamente de tipo construtivos, (b) algoritmos de otimização clássica e, (3) algoritmos de otimização combinatorial. Uma forma de avaliar o desempenho dos diferentes algoritmos consiste em verificar o desempenho de cada um dos algoritmos na solução de sistemas elétricos, isto é, avaliando o desempenho experimental. Deve-se verificar fundamentalmente o esforço computacional e a qualidade da solução encontrada. Para o caso de planejamento, logicamente, o aspecto mais importante é a qualidade da solução encontrada. Outros requisitos desejáveis estão relacionados com aspectos como facilidade de implementação computacional, facilidade de interpretação, robustez, etc. A maneira de ilustração, mostramos o desempenho da maioria dos algoritmos quando são testados em três sistemas elétricos amplamente conhecidos na literatura especializada de planejamento de sistemas de transmissão: (1) um sistema didático de 6 barras e 15 circuitos, (2) um sistema real de medio porte do sistema sul brasileiro reduzido de 46 barras, 79 circuitos e demanda de 6880 MW e, (3) um sistema de grande porte do sistema norte-nordeste brasileiro reduzido de 87 barras, 183 circuitos e demanda de 20325 MW para o plano 2002 e de 29754 MW para o plano 2008. Para os dois primeiros sistemas existem dados disponíveis para planejamento com reprogramação da geração e sem reprogramação da geração. Para o sistema norte-nordeste brasileiro existem dados disponíveis somente para o caso de sem reprogramação da geração.

9.2.1 Algoritmos Heurísticos

Como foi mencionado anteriormente, o algoritmo heurístico encontra soluções de boa qualidade mas com esforços computacionais muito pequenos.

O algoritmo heurístico mais eficiente para o modelo de transportes é o algoritmo de Garver [2]. Entretanto o algoritmo de Garver somente encontra a solução ótima do sistema de 6 barras sem redespacho. Em todos os outros casos encontra apenas soluções subótimas às vezes de qualidade duvidosa. Por exemplo, para o sistema norte-nordeste, plano 2002, o algoritmo de Garver encontra uma solução com um investimento de 1340 milhões de dólares. Entretanto, um algoritmo de Branch and Bound sem decomposição que estamos desenvolvendo e que no momento tem capacidade de encontrar configurações sub-ótimas encontrou uma solução com investimento de 1208 milhões de dólares. Portanto, a diferença de investimento entre a solução encontrada com o algoritmo de Garver e a solução ótima é de mais de 132 milhões de dólares.

Os algoritmos heurísticos para o modelo DC mais conhecidos e usados, pelo menos no Brasil, são os apresentados em [3, 4, 5, 6, 7]. Todos esses algoritmos encontram apenas a solução ótima do sistema de 6 barras sem reprogramação da geração. Portanto, até o final da década de 80 não se conheciam as soluções ótimas dos outros sistemas. Com o aparecimento de soluções ótimas desses sistemas encontrados usando algoritmos mais sofisticados apareceram vários problemas que deveriam ser rediscutidos: (1) os algoritmos heurísticos não são tão eficientes em relação à qualidade das soluções encontradas. (2) deve-se redefinir o papel dos algoritmos heurísticos e, (3) existe necessidade de desenvolver algoritmos mais eficientes.



Em [35] retomamos as pesquisas sobre algoritmos heurísticos construtivos levando em conta as observações anteriores. Assim foi desenvolvido um algoritmo heurístico construtivo integrado que usa os indicadores de sensibilidade dos melhores algoritmos heurísticos construtivos para realizar cada passo do algoritmo e, adicionalmente, são incorporadas outras características específicas do problema de planejamento. O produto final é um algoritmo que encontra soluções de melhor qualidade com um aumento do esforço computacional. Este algoritmo heurístico construtivo integrado encontra a solução ótima do sistema de 6 barras com e sem reprogramação da geração, encontra também a solução ótima do sistema de 46 barras com reprogramação da geração. Nos outros casos encontra apenas soluções sub-ótimas mas de melhor qualidade que os outros algoritmos heurísticos construtivos. No momento estamos melhorando este algoritmo heurístico incorporando os indicadores de sensibilidade de outros algoritmos heurísticos construtivos e outras características específicas do problema de planejamento de sistemas de transmissão.

9.2.2 Algoritmos de Otimização Clássica

Entre os algoritmos de otimização clássica que existem na bibliografia de programação não linear, a decomposição de Benders foi praticamente a única usada para o problema de planejamento de sistemas de transmissão. A principal justificativa é que a decomposição de Benders permite separar “naturalmente” o problema de planejamento em dois subproblemas que são resolvidos de maneira iterativa: um subproblema de investimento com as variáveis de investimento inteiras e um subproblema de operação com as variáveis de operação reais.

As primeiras pesquisas usando a decomposição de Benders para o modelo DC foram apresentadas em [10, 11, 32]. Entretanto, a decomposição de Benders, para o modelo DC, também pode convergir a soluções ótimas locais além de apresentar problemas de instabilidade numérica e problemas de convergência. Assim as pesquisas anteriores encontraram apenas a solução ótima global do sistema de 6 barras sem reprogramação da geração. Para o sistema de 6 barras com reprogramação e para os dois casos do sistema de 46 barras foram encontradas apenas soluções ótimas locais.

Para contornar os problemas mencionados anteriormente foi proposto um algoritmo de decomposição de Benders hierarquizado como já foi analisado no capítulo 5. Este algoritmo encontrou as soluções ótimas para os dois casos do sistema de 6 barras e de 46 barras. Para o sistema de 6 barras com reprogramação da geração encontrou uma solução com investimento de 110 (quando a melhor conhecida era de 130). Para o sistema de 46 barras com reprogramação da geração encontrou uma solução com investimento de 70289000 dólares e de 154420000 para o caso de sem reprogramação da geração. Foi especialmente surpreendente a solução de 70,289 milhões de dólares, muito distante de soluções encontradas com outros algoritmos. Esses resultados justificaram a tese de doutorado apresentado em [33]. Nesse algoritmo o subproblema mestre foi resolvido usando um algoritmo de enumeração implícita de Balas especializado. Do conjunto da pesquisa foram publicados dois artigos publicados em [14, 15] que ainda fazem parte integrante da tese de doutorado. Entretanto, o algoritmo de Benders hierarquizado não teve capacidade de resolver o sistema norte-nordeste brasileiro.

Em relação aos algoritmos baseados na decomposição de Benders está em aberto pesquisas relacionadas com os problemas de convergência e da capacidade de resolver sistemas de grande



porte. Para o modelo DC, este é um tema de pesquisa que está em aberto.

Levando em conta os problemas anteriormente mencionados, retomamos as pesquisas baseadas em otimização clássica. Inicialmente está sendo usado o modelo de transportes que é relativamente mais fácil de resolver e que pode apresentar indicativos eficientes para depois retomar o modelo DC. Assim foi implementado um algoritmo de decomposição de Benders hierarquizado para o modelo de transportes onde o subproblema mestre se resolve usando um algoritmo de Branch and Bound especializado. Os resultados desta pesquisa foram apresentados em [36]. Este algoritmo encontra as soluções ótimas para os dois casos do sistema de 6 e 46 barras. Um aspecto interessante deste algoritmo é que tem capacidade de encontrar soluções ótimas alternativas globais. Este algoritmo ainda não converge para o sistema norte-nordeste brasileiro. Outra possibilidade alternativa consiste em resolver o modelo de transportes sem a decomposição de Benders e resolvendo o problema integral usando um algoritmo de Branch and Bound. Os primeiros resultados desta pesquisa foram apresentados em [37] onde foram encontradas as soluções ótimas do sistema de 6 e 46 barras. Para o sistema norte-nordeste brasileiro este algoritmo encontra soluções sub-ótimas de excelente qualidade. É possível que este algoritmo, após algumas melhorias adicionais, encontre as soluções ótimas de sistemas de grande porte para o modelo de transportes.

9.2.3 Algoritmos de Otimização Combinatorial

Nos últimos anos foram desenvolvidos algoritmos de otimização combinatorial para resolver o problema de planejamento de sistemas de transmissão a longo prazo quando é usado o modelo DC. Os algoritmos combinatoriais tais como simulated annealing, genéticos, tabu search, etc. apresentam características especiais: (1) geralmente encontram soluções ótimas ou quase ótimas de excelente qualidade, (2) apresentam uma grande flexibilidade de implementação para diferentes modelos matemáticos de um tipo de problema e, (3) o esforço computacional é relativamente elevado tornando ainda esse tipo de algoritmos pouco atrativos para o trabalho diário de planejamento nas empresas elétricas.

Após o doutorado nossa primeira experiência com algoritmos combinatoriais foi o desenvolvimento de um algoritmo sequencial de simulated annealing para o problema de planejamento de sistemas de transmissão [16]. Posteriormente foi desenvolvido um algoritmo paralelo de simulated annealing e cujos resultados foram apresentados em [17]. Os resultados encontrados com esses algoritmos para sistemas de grande porte como os do norte-nordeste brasileiro foram surpreendentes. Foram verificados dois fatos importantes: (1) para sistemas de grande porte os algoritmos heurísticos construtivos apresentam soluções de pouca qualidade, (2) entre o intervalo de soluções encontradas pelos algoritmos heurísticos construtivos e as melhores soluções encontradas pelo algoritmo de simulated annealing existem uma quantidade de soluções incrivelmente grande, (3) existem muitas soluções com investimentos muito próximas mas com topologias radicalmente diferentes. Os resultados desta pesquisa levaram à publicação de dois artigos em revista de alto nível [16, 17].

Numa segunda etapa foram implementados algoritmos genéticos para o problema de planejamento de sistemas de transmissão quando é usado o modelo DC. Estes algoritmos encontraram soluções apenas ligeiramente melhores que os encontrados pelo algoritmo de simulated annealing. Entretanto, como representa uma proposta significativamente diferente foi possível



a publicação dos resultados dessa pesquisa através de um artigo em revista de alto nível [19].

Numa seguinte etapa foi desenvolvido um algoritmo tabu search para o problema de planejamento de sistemas de transmissão e, finalmente, foi terminada a parte básica fundamental de implementação de algoritmos combinatoriais com a implementação de um algoritmo híbrido para o problema de planejamento. Assim, em [18] foi apresentado o algoritmo híbrido junto com uma análise comparativa dos três algoritmos combinatoriais (simulated annealing, genéticos, tabu search). Os resultados obtidos com o algoritmo tabu search foram apresentados em [20].

Os algoritmos combinatoriais encontram as soluções ótimas do sistema de 6 barras e 46 barras facilmente. Para o sistema de grande porte, sistema norte-nordeste brasileiro, as melhores soluções encontradas usando os algoritmos combinatoriais foram as seguintes: (1) uma solução com investimento de 1401 milhões de dólares para o plano 2002 e, (2) várias soluções com investimento em torno de 2574 milhões de dólares para o plano 2008. Em relação aos resultados encontrados, deve-se fazer as seguintes observações: (1) essas soluções foram encontradas com o algoritmo híbrido e o algoritmo tabu search, (2) essas soluções são muito superiores aos encontrados usando algoritmos heurísticos construtivos, (3) não se conhecem as soluções ótimas do sistema de grande porte, norte-nordeste brasileiro, para os planos 2002 e 2008 e, (4) existem evidências de que as soluções encontradas estão muito próximas da solução ótima global.

Em resumo, nosso aporte após o doutorado, na implementação de algoritmos combinatoriais para o problema de planejamento de sistemas de transmissão foi grande e significativa. Os resultados dessa pesquisa levaram à publicação de muitos artigos em revistas e anais de congresso sendo que cinco deles correspondem a publicações de alto nível [16, 17, 18, 19, 20]. Logicamente, esse trabalho não foi desenvolvido por uma única pessoa; na verdade foram desenvolvidas por uma equipe e os pesquisadores dessa equipe aparecem como autores nas publicações antes mencionadas.

9.2.4 Contribuições no Desenvolvimento de Algoritmos Correlatos

Em relação ao problema de planejamento de sistemas de transmissão existem dois tipos de pesquisa que podem ser considerados correlatos: (1) pesquisas para o desenvolvimento de técnicas de solução de subproblemas decorrentes do tipo de solução e da modelagem escolhida e, (2) pesquisas de problemas mais complexos.

Praticamente em todos os algoritmos apresentados anteriormente existe a necessidade de resolver subproblemas. Na maioria dos casos esses subproblemas são problemas de programação linear que assumem formas variadas dependendo do modelo escolhido e da técnica de solução escolhida. Às vezes, deve-se resolver problemas de PL milhares ou centenas de milhares de vezes como não caso de sistemas de grande porte e quando são usados os algoritmos combinatoriais. Na maioria de nossas pesquisas foram usadas o software de PL chamado MINOS e considerado como um dos mais eficientes para resolver problemas de PL. Entretanto estamos desenvolvendo algoritmos de PL especializados para resolver problemas de PL relacionados com os subproblemas que aparecem no problema de planejamento de sistemas de transmissão. Os resultados preliminares dessa pesquisa foram apresentados em [38].

Outra linha de pesquisa consiste em desenvolver algoritmos para problemas mais complexos



ou genéricos como o planejamento dinâmico de sistemas de transmissão ou o planejamento de sistemas de transmissão considerando um mercado de livre acesso. Atualmente estamos trabalhando em pesquisas com esse tipo de problema. Entretanto, nossa demora em realizar pesquisas desse tipo se deve a nossa participação muito ativa no problema básico de planejamento onde ainda existem muitos aspectos que devem ser pesquisados. Em [21] apresentamos os primeiros resultados obtidos com o planejamento dinâmico de sistemas de transmissão quando é usado o modelo de transportes.

9.3 Conclusões e Novas Linhas de Pesquisa

Do relatório apresentado, pode-se mencionar as seguintes conclusões:

1. Os algoritmos heurísticos construtivos devem ser reavaliados em relação a sua utilização no planejamento de sistemas de transmissão a longo prazo. Aparentemente, a principal aplicação desse tipo de algoritmos seja como elementos auxiliares de outros algoritmos mais eficientes mas que requerem de esforços computacionais mais elevados.
2. Os algoritmos de otimização clássica encontram soluções ótimas de sistemas de pequeno e de médio porte mas apresentam problemas de convergência em sistemas de grande porte. Portanto, esse aspecto é um tema interessante a ser pesquisado.
3. Os algoritmos combinatoriais são os mais eficientes para o problema de planejamento de sistemas de transmissão em termos da qualidade da solução encontrada. Entretanto, deve-se realizar novas pesquisas visando principalmente diminuir o esforço computacional.
4. Não se conhecem ainda as soluções ótimas de sistemas de grande porte como o sistema norte-nordeste brasileiro. Este fato é normal em problemas grandes e complexos como o problema de planejamento de sistemas de transmissão assim como acontece em muitos problemas complexos de outras áreas do conhecimento e também em outros problemas complexos em engenharia elétrica.
5. O número elevado de publicações obtidas em pesquisas sobre o problema de planejamento de sistemas de transmissão nos últimos anos é uma prova de nossa contribuição ativa no desenvolvimento de algoritmos eficientes para o problema de planejamento de sistemas de transmissão. Merece especial destaque os sete artigos publicados em revistas internacionais de alto nível.

As pesquisas que estão sendo desenvolvidas atualmente e as que estão em fase de implementação são as seguintes:

1. Desenvolvimento de algoritmos eficientes de PL para cada tipo de subproblema que aparece com cada um dos modelos existentes e com cada técnica de solução escolhida. Merece especial atenção o desenvolvimento de algoritmos baseados em técnicas de pontos interiores e em reotimização.



2. Desenvolvimento de um algoritmo eficiente do tipo Branch and Bound, sem decomposição matemática, para o modelo de transportes com capacidade de encontrar soluções ótimas de sistemas de grande porte.
3. Desenvolvimento de algoritmos eficientes de otimização clássica para o modelo DC, diferentes da técnica de decomposição de Benders.
4. Desenvolvimento de algoritmos híbridos baseados nos algoritmos combinatoriais e os algoritmos heurísticos construtivos.
5. Desenvolvimento de algoritmos baseados em técnicas de inteligência artificial e em redes neurais.
6. Desenvolver linhas de pesquisas relacionados com o problema de planejamento dinâmico de sistemas de transmissão e levando em conta a operação do sistema num mecanismo de livre acesso.



Referências Bibliográficas

- [1] *R. Fischl*: "Optimal System Expansion Planning: A Critical Overview", ERDA & EPRI Conf., Henniker, 1975.
- [2] *L.L. Garver*: "Transmission Network Estimation Using Linear Programming", IEEE Trans. Power App. Syst., Vol. PAS-89, pp. 1688-1697, September-October, 1970.
- [3] *A. Monticelli, A. Santos Jr., M.V.F. Pereira, S.H.. Cunha, B.J. Parker, J.C.G. Praça*: "Interactive Transmission Network Planning Using a Least-Effort Criterion", IEEE Trans. Power App. Systems, Vol. PAS-101, No. 10, October, 1982.
- [4] *M. V. F. Pereira, L. M. V. G. Pinto*: "Application of Sensitivity Analysis of Load Supplying Capability to Interactive Transmission Expansion Planning", IEEE Transaction on Power Apparatus and Systems, Vol. PAS-104, No. 2, pp 381-389, February, 1985.
- [5] *V.A. Levi, M.S. Calovic*: "A New Decomposition Based Method for Optimal Expansion Planning of Large Transmission Networks", IEEE Transactions on Power Systems, Vol. 6, No. 3, pp 937-943, October, 1991.
- [6] *C. Dechamps, A. Jamouille*: "Interactive Computer Program for Planning the Expansion of Meshed Transmission Networks", International Journal of Electrical Power and Energy Systems, Vol. 2, No. 2, pp 103-108, April, 1980.
- [7] *R. Villasana, L.L. Garver, S.J. Salon*: "Transmission Network Planning Using Linear Programming", IEEE Trans. Power App. Systems, Vol. PAS-104, No. 2, February 1985.
- [8] *G. Latorre-Bayona, A. Ramos, J.I. Perez-Arriaga, J.F. Alonso, A. Saiz*: "PERLA: Un Modelo de Planificación Estática a Largo Plazo de la Red de Transporte: Opciones de Modelado y Análisis de Idoneidad", II Jornadas Hispano-Lusos de Ingeniería Eléctrica. Coimbra, Portugal, 1991.
- [9] *G. Latorre-Bayona, J.I. Perez-Arriaga*: "CHOPIN, A Heuristic Model for Long Term Transmission Expansion Planning", IEEE Transactions on Power Systems, Vol. 9, No. 4, November 1994.
- [10] *Pereira M.V.F., Pinto L.M.V.G., Oliveira G.C., Cunha S.H.F.*: "Composite Generation Transmission Expansion Planning"; EPRI Research Project 2473-9, EL-5179, 1987.
- [11] *Granville S., Pereira M.V.F.*: "Analysis of the Linearized Power Flow Model in Benders Decomposition". EPRI-Report RP 2473-6, Stanford University, February 1985.



- [12] *G.C. Oliveira, A.P.C. Costa, S. Binato*: "Large Scale Transmission Network Planning Using Optimization and Heuristic Techniques", IEEE Transactions on Power Systems, Vol. 10, No. 4, pp 1828-1834, November 1995.
- [13] *H. Rudnick, R. Palma, E. Cura and C. Silva*: "Economically Adapted Transmission Systems in Open Access Schemes – Application of Genetic Algorithms", IEEE PES Summer Meeting, July 1995.
- [14] *R. Romero, A. Monticelli*: "A Hierarchical Decomposition Approach for Transmission Network Expansion Planning", IEEE Transactions on Power Systems, Vol. 9, No. 1, pp. 373-380, February 1994.
- [15] *R. Romero - A. Monticelli*: *A Zero-One Implicit Enumeration Method for Optimizing Investments in Transmission Expansion Planning*, IEEE Transactions on Power Systems, Vol. 9, No. 3, pp. 1385-1391, August 1994.
- [16] *R. Romero, R.A. Gallego e A. Monticelli*: *Transmission System Expansion Planning by Simulated Annealing*, IEEE Transactions on Power Systems, Vol. 11, No. 1, pp. 364-369, February 1996.
- [17] *R.A. Gallego, A.B. Alves, A. Monticelli, R. Romero*: *Parallel Simulated Annealing Applied to Long Term Transmission Network Expansion Planning*, IEEE Transactions on Power Systems, Vol. 12, No. 1, pp. 181-188, February 1997.
- [18] *R.A. Gallego, A. Monticelli, R. Romero*: Comparative Studies of Non-Convex Optimization Methods for Transmission Network Expansion Planning, para aparecer em IEEE Transactions on Power Systems, Vol. 13, No. 3, pp 822-828, August 1998.
- [19] *R.A. Gallego, A. Monticelli, R. Romero*: "Transmission System Expansion Planning by Extended Genetic Algorithm", IEE Proceedings in Generation, Transmission and Distribution, Vol. 145, No. 3, pp 329-335, May 1998.
- [20] *R.A. Gallego, A. Monticelli, R. Romero*: "Tabu Search Algorithm for Network Synthesis", para aparecer em IEEE Transactions on Power Systems, Vol. 15, 2000.
- [21] *S. Haffner, A. Garcia, A. Monticelli, R. Romero*: "Planejamento Dinâmico da Expansão de Sistemas de Transmissão de Energia Elétrica Considerando Múltiplos Estágios", XV SNPTEE, Foz de Iguaçu, Outubro de 1999.
- [22] *Aarts E., Korst J.*: "Simulated Annealing and Boltzmann Machines". John Wiley & Sons, 1989.
- [23] *M. Srinivas, L.M. Patnaik*: "Genetic Algorithms", IEEE Computer, 1994.
- [24] *D.E. Goldberg*: "Genetics Algorithms in Search, Optimization and Machine Learning". Addison Wesley, Reading, Mass., 1989.
- [25] *F. Glover*: "Tabu Search Fundamental and Uses", Graduate School of Business, University of Colorado. 1995.
- [26] *A. Diaz, F. Glover, H. M. Ghaziri, J.L. Gonzalez, M. Laguna, P. Moscato, F.T. Tseng*: "Optimización Heurística y Redes Neuronales", Editorial Paraninfo, Madrid. 1996.



- [27] *F. Glover, M. Laguna*: "Tabu Search", Kluwer Academic Publishers, 1996.
- [28] *A. Monticelli*: "Fluxo de Carga em Redes de Energia Elétrica, Editora Edgar Bucher Ltda, 1983.
- [29] *C.J. Baldwin, C.A. De Salvo, C.H. Hoffman, W.S. Ku*: "A Model for Transmission Planning by Logic", AIEE Transactions PAS, Vol. 79, pp 1638-1645, February 1960.
- [30] *A. Monticelli*: "Expansão de Sistemas Elétricos de Potência: Um Algoritmo para o Planejamento do Sistema de Transmissão", Tese de Doutorado, UNICAMP, 1975.
- [31] *A. Santos Jr.*: "Síntese Conceitual Estática de Redes de Transmissão", Tese de Mestrado, UNICAMP, 1981.
- [32] *M. V. F. Pereira*: "Aplicação de Análise de Sensibilidade no Planejamento da Expansão de Sistemas de Geração/Transmissão", Tese de Doutorado, COPPE-UFRJ, 1985.
- [33] *R. Romero*: "Um método de Decomposição para o Planejamento a Longo Prazo de Sistemas de Transmissão", Tese de Doutorado, UNICAMP, 1993.
- [34] *R.A. Gallego*: "Planejamento a Longo Prazo de Sistemas de Transmissão Usando Técnicas de Otimização Combinatorial", Tese de Doutorado, UNICAMP, 1997.
- [35] *C.R. Rocha*: "Algoritmo Heurístico Usado no Planejamento a Longo Prazo de Sistemas de Transmissão", dissertação de Mestrado, UNESP, 1999.
- [36] *S. Haffner, A. Monticelli, A. Garcia, J. Mantovani, R. Romero*: "A Branch and Bound Algorithm for Transmission System Expansion Planning Using a Transportation Model", submetido para IEE Proceeding in Generation, Transmission and Distribution, 1999.
- [37] *R. Romero, R. Kurokawa, J. Mantovani*: "Algoritmo de Branch and Bound Especializado para o Problema de Planejamento da Expansão de Sistemas de Transmissão de Energia Elétrica, anais do XXXI-SBPO, Simpósio Brasileiro de Pesquisa Operacional, pp 1650-1661, Juiz de Fora, Novembro de 1999.
- [38] *S. Marchiori, F. Villarreal, R. Romero*: "Programação Linear Dual Simplex Usada no Planejamento da Expansão de Sistemas de Transmissão", anais do XXXI-SBPO, Simpósio Brasileiro de Pesquisa Operacional, pp 464-478, Juiz de Fora, Novembro de 1999.



Apêndice A

Dados dos Sistemas Testados

A.1 Sistema de 6 Barras de Garver

Níveis de Geração e Carga

Barra	Cap. de Geração	Ger. Atual	Carga (MW)
1	150	50	80
2	0	0	240
3	360	165	40
4	0	0	160
5	0	0	240
6	600	545	0

Características das Linhas

Nº	Linha	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ dol.
1	1-2	1	0.40	100	40
2	1-3	0	0.38	100	38
3	1-4	1	0.60	80	60
4	1-5	1	0.20	100	20
5	1-6	0	0.68	70	68
6	2-3	1	0.20	100	20
7	2-4	1	0.40	100	40
8	2-5	0	0.31	100	31
9	2-6	0	0.30	100	30
10	3-4	0	0.59	82	59
11	3-5	1	0.20	100	20
12	3-6	0	0.48	100	48
13	4-5	0	0.63	75	63
14	4-6	0	0.30	100	30
15	5-6	0	0.61	78	61



A. 2 Sistema Sul Brasileiro de 46 Barras

Níveis de Geração e Carga

Barra	Cap. de Geração	Ger. Atual	Carga (MW)
1	0.0	0.0	0.0
2	0.0	0.0	443.1
3	0.0	0.0	0.0
4	0.0	0.0	300.7
5	0.0	0.0	238.0
6	0.0	0.0	0.0
7	0.0	0.0	0.0
8	0.0	0.0	72.2
9	0.0	0.0	0.0
10	0.0	0.0	0.0
11	0.0	0.0	0.0
12	0.0	0.0	511.9
13	0.0	0.0	185.8
14	1257.0	944.0	0.0
15	0.0	0.0	0.0
16	2000.0	1366.0	0.0
17	1050.0	1000.0	0.0
18	0.0	0.0	0.0
19	1670.0	773.0	0.0
20	0.0	0.0	1091.2
21	0.0	0.0	0.0
22	0.0	0.0	81.9
23	0.0	0.0	458.1
24	0.0	0.0	478.2
25	0.0	0.0	0.0
26	0.0	0.0	231.9
27	220.0	54.0	0.0
28	800.0	730.0	0.0



Níveis de Geração e Carga - Continuação

Barra	Cap. de Geração	Ger. Atual	Carga (MW)
29	0.0	0.0	0.0
30	0.0	0.0	0.0
31	700.0	310.0	0.0
32	500.0	450.0	0.0
33	0.0	0.0	229.1
34	748.0	221.0	0.0
35	0.0	0.0	216.0
36	0.0	0.0	90.1
37	300.0	212.0	0.0
38	0.0	0.0	216.0
39	600.0	221.0	0.0
40	0.0	0.0	262.1
41	0.0	0.0	0.0
42	0.0	0.0	1607.9
43	0.0	0.0	0.0
44	0.0	0.0	79.1
45	0.0	0.0	86.7
46	700.0	599.0	0.0



Características das Linhas

Nº	Linha	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ dol.
1	01-07	1	0.0616	270	4349
2	01-02	2	0.1065	270	7076
3	04-09	1	0.0924	270	6217
4	05-09	1	0.1173	270	7732
5	05-08	1	0.1132	270	7480
6	07-08	1	0.1023	270	6823
7	04-05	2	0.0566	270	4046
8	02-05	2	0.0324	270	2581
9	08-13	1	0.1348	240	8793
10	09-14	2	0.1756	220	11267
11	12-14	2	0.0740	270	5106
12	14-18	2	0.1514	240	9803
13	13-18	1	0.1805	220	11570
14	13-20	1	0.1073	270	7126
15	18-20	1	0.1997	200	12732
16	19-21	1	0.0278	1500	32632
17	16-17	1	0.0078	2000	10505
18	17-19	1	0.0061	2000	8715
19	14-26	1	0.1614	220	10409
20	14-22	1	0.0840	270	5712
21	22-26	1	0.0790	270	5409
22	20-23	2	0.0932	270	6268
23	23-24	2	0.0774	270	5308
24	26-27	2	0.0832	270	5662
25	24-34	1	0.1647	220	10611
26	24-33	1	0.1448	240	9399
27	33-34	1	0.1265	270	8288
28	27-36	1	0.0915	270	6167
29	27-38	2	0.2080	200	13237
30	36-37	1	0.1057	270	7025
31	34-35	2	0.0491	270	3591
32	35-38	1	0.1980	200	12631
33	37-39	1	0.0283	270	2329
34	37-40	1	0.1281	270	8389
35	37-42	1	0.2105	200	13388
36	39-42	3	0.2030	200	12934
37	40-42	1	0.0932	270	6268
38	38-42	3	0.0907	270	6116
39	32-43	1	0.0309	1400	35957
40	42-44	1	0.1206	270	7934



Características das Linhas - Continuação

Nº	Linha	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ dol.
41	44-45	1	0.1864	200	11924
42	19-32	1	0.0195	1800	23423
43	46-19	1	0.0222	1800	26365
44	46-16	1	0.0203	1800	24319
45	18-19	1	0.0125	600	8178
46	20-21	1	0.0125	600	8178
47	42-43	1	0.0125	600	8178
48	02-04	0	0.0882	270	5965
49	14-15	0	0.0374	270	2884
50	46-10	0	0.0081	2000	10889
51	04-11	0	0.2246	240	14247
52	05-11	0	0.0915	270	6167
53	46-06	0	0.0128	2000	16005
54	46-03	0	0.0203	1800	24319
55	16-28	0	0.0222	1800	26365
56	16-32	0	0.0311	1400	36213
57	17-32	0	0.0232	1700	27516
58	19-25	0	0.0325	1400	37748
59	21-25	0	0.0174	2000	21121
60	25-32	0	0.0319	1400	37109
61	31-32	0	0.0046	2000	7052
62	28-31	0	0.0053	2000	7819
63	28-30	0	0.0058	2000	8331
64	27-29	0	0.0998	270	6672
65	26-29	0	0.0541	270	3894
66	28-41	0	0.0339	1300	39283
67	28-43	0	0.0406	1200	46701
68	31-41	0	0.0278	1500	32632
69	32-41	0	0.0309	1400	35957
70	41-43	0	0.0139	2000	17284
71	40-45	0	0.2205	180	13994
72	15-16	0	0.0125	600	8178
73	46-11	0	0.0125	600	8178
74	24-25	0	0.0125	600	8178
75	29-30	0	0.0125	600	8178
76	40-41	0	0.0125	600	8178
77	02-03	0	0.0125	600	8178
78	05-06	0	0.0125	600	8178
79	09-10	0	0.0125	600	8178



A.3 Sistema Nordeste Brasileiro de 87 Barras

Níveis de Geração e Carga

Barra	Geração em 2002 (MW)	Carga em 2002 (MW)	Geração em 2008 (MW)	Carga em 2008 (MW)
1	0	1857	0	2747
2	4048	0	4550	0
3	0	0	0	0
4	517	0	6422	0
5	0	0	0	0
6	0	0	0	0
7	0	31	0	31
8	403	0	82	0
9	465	0	465	0
10	538	0	538	0
11	2200	0	2260	0
12	2257	0	4312	0
13	4510	0	5900	0
14	542	0	542	0
15	0	0	0	0
16	0	0	0	0
17	0	0	0	0
18	0	0	0	0
19	0	86	0	125
20	0	125	0	181
21	0	722	0	1044
22	0	291	0	446
23	0	58	0	84
24	0	159	0	230
25	0	1502	0	2273
26	0	47	0	68
27	0	378	0	546
28	0	189	0	273
29	0	47	0	68
30	0	189	0	273
31	0	110	0	225
32	0	0	0	0
33	0	0	0	0
34	0	28	0	107
35	1635	0	1531	0
36	0	225	0	325
37	169	0	114	0
38	0	0	0	0
39	0	186	0	269
40	0	1201	0	1738



Níveis de Geração e Carga - Continuação

Barra	Geração em 2002 (MW)	Carga em 2002 (MW)	Geração em 2008 (MW)	Carga em 2008 (MW)
41	0	520	0	752
42	0	341	0	494
43	0	0	0	0
44	0	4022	0	5819
45	0	0	0	0
46	0	205	0	297
47	0	0	0	0
48	0	347	0	432
49	0	777	0	1124
50	0	5189	0	7628
51	0	290	0	420
52	0	707	0	1024
53	0	0	0	0
54	0	0	0	0
55	0	0	0	0
56	0	0	0	0
57	0	0	0	0
58	0	0	0	0
59	0	0	0	0
60	0	0	0	0
61	0	0	0	0
62	0	0	0	0
63	0	0	0	0
64	0	0	0	0
65	0	0	0	0
66	0	0	0	0
67	1242	0	1242	0
68	888	0	888	0
69	902	0	902	0
70	0	0	0	0
71	0	0	0	0
72	0	0	0	0
73	0	0	0	0
74	0	0	0	0
75	0	0	0	0
76	0	0	0	0
77	0	0	0	0
78	0	0	0	0
79	0	0	0	0
80	0	0	0	0
81	0	0	0	0
82	0	0	0	0
83	0	0	0	0
84	0	0	0	0
85	0	487	0	705
86	0	0	0	0
87	0	0	0	0

Características das Linhas - Nordeste Brasileiro

Nº	Linha	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ dol.
1	01-02	2	0.0374	1000	44056
2	02-04	0	0.0406	1000	48880
3	02-60	0	0.0435	1000	52230
4	02-87	1	0.0259	1000	31192
5	03-71	0	0.0078	3200	92253
6	03-81	0	0.0049	3200	60153
7	03-83	0	0.0043	3200	53253
8	03-87	0	0.0058	1200	21232
9	04-05	1	0.0435	1000	52230
10	04-06	0	0.0487	1000	58260
11	04-32	0	0.0233	300	7510
12	04-60	0	0.0215	1000	26770
13	04-68	0	0.0070	1000	10020
14	04-69	0	0.0162	1000	20740
15	04-81	0	0.0058	1200	21232
16	04-87	1	0.0218	1000	26502
17	05-06	1	0.0241	1000	29852
18	05-38	2	0.0117	600	8926
19	05-56	0	0.0235	1000	29182
20	05-58	0	0.0220	1000	27440
21	05-60	0	0.0261	1000	32130
22	05-68	0	0.0406	1000	48880
23	05-70	0	0.0464	1000	55580
24	05-80	0	0.0058	1200	21232
25	06-07	1	0.0288	1000	35212
26	06-37	1	0.0233	300	7510
27	06-67	0	0.0464	1000	55580
28	06-68	0	0.0476	1000	56920
29	06-70	0	0.0371	1000	44860
30	06-75	0	0.0058	1200	21232
31	07-08	1	0.0234	1000	29048
32	07-53	0	0.0452	1000	54240
33	07-62	0	0.0255	1000	31460
34	08-09	1	0.0186	1000	23420
35	08-12	0	0.0394	1000	47540
36	08-17	0	0.0447	1000	53570
37	08-53	1	0.0365	1200	44190
38	08-62	0	0.0429	1000	51560
39	08-73	0	0.0058	1200	21232
40	09-10	1	0.0046	1000	7340
41	10-11	1	0.0133	1000	17390
42	11-12	1	0.0041	1200	6670
43	11-15	1	0.0297	1200	36284
44	11-17	1	0.0286	1200	35078
45	11-53	1	0.0254	1000	31326
46	12-13	1	0.0046	1200	7340



Características das Linhas - Continuação

Nº	Linha	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ dol.
47	12-15	1	0.0256	1200	31594
48	12-17	1	0.0246	1200	30388
49	12-35	2	0.0117	600	8926
50	12-84	0	0.0058	1200	21232
51	13-14	0	0.0075	1200	10690
52	13-15	0	0.0215	1200	26770
53	13-17	0	0.0232	1200	28780
54	13-45	1	0.0290	1200	35480
55	13-59	1	0.0232	1200	28780
56	14-17	0	0.0232	1200	28780
57	14-45	0	0.0232	1200	28780
58	14-59	0	0.0157	1200	20070
59	15-16	2	0.0197	1200	24760
60	15-45	0	0.0103	1200	13906
61	15-46	1	0.0117	600	8926
62	15-53	0	0.0423	1000	50890
63	16-44	4	0.0117	600	8926
64	16-45	0	0.0220	1200	27440
65	16-61	0	0.0128	1000	16720
66	16-77	0	0.0058	1200	21232
67	17-18	2	0.0170	1200	21678
68	17-59	0	0.0170	1200	21678
69	18-50	4	0.0117	600	8926
70	18-59	1	0.0331	1200	40170
71	18-74	0	0.0058	1200	21232
72	19-20	1	0.0934	170	5885
73	19-22	1	0.1877	170	11165
74	20-21	1	0.0715	300	6960
75	20-21	1	0.1032	170	6435
76	20-38	2	0.1382	300	12840
77	20-56	0	0.0117	600	8926
78	20-66	0	0.2064	170	12210
79	21-57	0	0.0117	600	8926
80	22-23	1	0.1514	170	9130
81	22-37	2	0.2015	170	11935
82	22-58	0	0.0233	300	7510
83	23-24	1	0.1651	170	9900
84	24-25	1	0.2153	170	12705
85	24-43	0	0.0233	300	7510
86	25-26	2	0.1073	300	29636
87	25-26	3	0.1691	170	10120
88	25-55	0	0.0117	600	8926
89	26-27	2	0.1404	300	25500
90	26-27	3	0.2212	170	12760
91	26-29	1	0.1081	170	6710
92	26-54	0	0.0117	600	8926



Características das Linhas - Continuação

Nº	Linha	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ dol.
93	27-28	3	0.0826	170	5335
94	27-35	2	0.1367	300	25000
95	27-53	1	0.0117	600	8926
96	28-35	3	0.1671	170	9900
97	29-30	1	0.0688	170	4510
98	30-31	1	0.0639	170	4235
99	30-63	0	0.0233	300	7510
100	31-34	1	0.1406	170	8525
101	32-33	0	0.1966	170	11660
102	33-67	0	0.0233	300	7510
103	34-39	2	0.1160	170	7150
104	34-39	2	0.2968	80	6335
105	34-41	2	0.0993	170	6215
106	35-46	4	0.2172	170	12705
107	35-47	2	0.1327	170	8085
108	35-51	3	0.1602	170	9625
109	36-39	2	0.1189	170	7315
110	36-46	2	0.0639	170	4235
111	39-42	1	0.0973	170	6105
112	39-86	0	0.0233	300	7510
113	40-45	1	0.0117	600	8926
114	40-46	3	0.0875	170	5500
115	41-64	0	0.0233	300	7510
116	42-44	2	0.0698	170	4565
117	42-85	2	0.0501	170	3465
118	43-55	0	0.0254	1000	31326
119	43-58	0	0.0313	1000	38160
120	44-46	3	0.1671	170	10010
121	47-48	2	0.1966	170	11660
122	48-49	1	0.0757	170	4895
123	48-50	2	0.0256	170	2090
124	48-51	2	0.2163	170	12760
125	49-50	1	0.0835	170	5335
126	51-52	2	0.0560	170	3795
127	52-59	1	0.0117	600	8926
128	53-54	0	0.0270	1000	32120
129	53-70	0	0.0371	1000	44860
130	53-76	0	0.0058	1200	21232
131	53-86	0	0.0389	1000	46870
132	54-55	0	0.0206	1000	25028
133	54-58	0	0.0510	1000	60940
134	54-63	0	0.0203	1000	25430
135	54-70	0	0.0360	1000	43520
136	54-79	0	0.0058	1200	21232
137	56-57	0	0.0122	1000	16050
138	58-78	0	0.0058	1200	21232



Características das Linhas - Continuação

Nº	Linha	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ dol.
139	60-66	0	0.0233	300	7510
140	60-87	0	0.0377	1000	45530
141	61-64	0	0.0186	1000	23420
142	61-85	0	0.0233	300	7510
143	61-86	0	0.0139	1000	18060
144	62-67	0	0.0464	1000	55580
145	62-68	0	0.0557	1000	66300
146	62-72	0	0.0058	1200	21232
147	63-64	0	0.0290	1000	35480
148	65-66	0	0.3146	170	18260
149	65-87	0	0.0233	300	7510
150	67-68	0	0.0290	1000	35480
151	67-69	0	0.0209	1000	26100
152	67-71	0	0.0058	1200	21232
153	68-69	0	0.0139	1000	18060
154	68-83	0	0.0058	1200	21232
155	68-87	0	0.0186	1000	23240
156	69-87	0	0.0139	1000	18060
157	70-82	0	0.0058	1200	21232
158	71-72	0	0.0108	3200	125253
159	71-75	0	0.0108	3200	125253
160	71-83	0	0.0067	3200	80253
161	72-73	0	0.0100	3200	116253
162	72-83	0	0.0130	3200	149253
163	73-74	0	0.0130	3200	149253
164	73-75	0	0.0130	3200	149253
165	73-84	0	0.0092	3200	107253
166	74-84	0	0.0108	3200	125253
167	75-76	0	0.0162	3200	185253
168	75-81	0	0.0113	3200	131253
169	75-82	0	0.0086	3200	101253
170	75-83	0	0.0111	3200	128253
171	76-77	0	0.0130	3200	149253
172	76-82	0	0.0086	3200	101253
173	76-84	0	0.0059	3200	70953
174	77-79	0	0.0151	3200	173253
175	77-84	0	0.0115	3200	132753
176	78-79	0	0.0119	3200	137253
177	78-80	0	0.0051	3200	62253
178	79-82	0	0.0084	3200	98253
179	80-81	0	0.0101	3200	117753
180	80-82	0	0.0108	3200	125253
181	80-83	0	0.0094	3200	110253
182	81-83	0	0.0016	3200	23253
183	82-84	0	0.0135	3200	155253



