

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**“Novas Aplicações de Metaheurísticas na Solução do
Problema de Planejamento da Expansão do Sistema de
Transmissão de Energia Elétrica”**

SILVIA LOPES DE SENA TAGLIALENHA

Orientador: Prof. Dr. Rubén Augusto Romero Lázaro

Tese apresentada à Faculdade de
Engenharia - UNESP – Campus de Ilha
Solteira, para obtenção do título de
Doutor em Engenharia Elétrica.
Área de Conhecimento: Automação.

Ilha Solteira – SP
Abril/2008

FICHA CATALOGRÁFICA

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

T126n Taglialenha, Silvia Lopes de Sena.
Novas aplicações de metaheurísticas na solução do problema de planejamento da expansão do sistema de transmissão de energia / Silvia Lopes de Sena Taglialenha. -- Ilha Solteira : [s.n.], 2008
136 p.

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

Orientador: Rubén Augusto Romero Lázaro
Bibliografia: p. 121-128

1. Energia elétrica – Transmissão. 2. Planejamento. 3. Busca dispersa. 4. Vizinhança variável. 5. Metaheurísticas.

CERTIFICADO DE APROVAÇÃO

TÍTULO: Novas Aplicações de Metaheurísticas na Solução do Problema de Planejamento da Expansão do Sistema de Transmissão de Energia Elétrica.

AUTORA: SILVIA LOPES DE SENA TAGLIALENHA

ORIENTADOR: Prof. Dr. RUBEN AUGUSTO ROMERO LAZARO

Aprovada como parte das exigências para obtenção do Título de DOUTOR em ENGENHARIA ELÉTRICA pela Comissão Examinadora:

Prof. Dr. RUBEN AUGUSTO ROMERO LAZARO

Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. JOSE ROBERTO SANCHES MANTOVANI

Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. ANTONIO PADILHA FELTRIN

Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

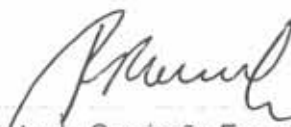
Prof. Dr. LUIZ CARLOS PEREIRA DA SILVA

Departamento de Sistemas de Energia Elétrica / Universidade Estadual de Campinas

Prof. Dr. EDUARDO NOBUHIRO ASADA

Departamento de Engenharia Elétrica / Escola de Engenharia de São Carlos-USP

Data da realização: 18 de abril de 2008.



Presidente da Comissão Examinadora
Prof. Dr. RUBEN AUGUSTO ROMERO LAZARO

Dedicatória:

Ao Junior,

Querido amor, que me inspira e enflora a existência,
todo o meu carinho e reconhecimento...

Agradecimentos

A Deus por tornar possível mais esta vitória.

Ao Professor Dr. Rubén Augusto Romero Lázaro, exemplo de orientador, que, desde a escolha do tema, teve importância ímpar na elaboração deste trabalho.

Ao meu querido amado Junior... Pelo exemplo de persistência e coragem... Obrigada pelo apoio, incentivo e paciência...

Ao Professor Dr. José Roberto Sanches Mantovani, pela confiança e amizade.

A todos os professores do DEE, em especial ao Padilha, Sérgio e Marcelo. E também às secretárias do DEE e PPGEE.

Aos técnicos do laboratório de informática, Deoclécio e Beto. Sem a colaboração de vocês este trabalho ainda estaria sem conclusão.

Às amigas Eliane, Elizete, Elizangela e Maira, pelo carinho e troca de informações.

À Vanuza, por toda sua dedicação...

Também agradeço aos colegas Edgar, Silvio e Edilton.

Ao incentivo da querida amiga Maria Angela...

E ainda a todos aqueles que de uma forma ou de outra contribuíram para a conclusão deste trabalho.

À CAPES pelo suporte financeiro.

*Muitos são os propósitos no coração do homem,
mas é a vontade do Senhor que se realiza.*

(Pv. 19:21)

Resumo

O Problema de Planejamento da Expansão de Sistemas de Transmissão de Energia Elétrica consiste em se escolher, entre um conjunto pré-definido de circuitos candidatos, aqueles que devem ser incorporados ao sistema de forma a minimizar os custos de investimento e operação e atender a demanda de energia futura ao longo de um horizonte de planejamento com confiabilidade, assumindo como conhecido o plano de geração. É considerado um problema muito complexo e difícil por se tratar de um problema não linear inteiro misto, não convexo, multimodal e altamente combinatório.

Este problema tem sido solucionado usando técnicas clássicas como Decomposição de Benders e *Branch and Bound*, assim como também algoritmos heurísticos e metaheurísticas obtendo diversos resultados, mais com uma série de problemas como, por exemplo, alto esforço computacional e problemas de convergência.

Neste trabalho apresentam-se duas novas técnicas de solução para o problema, a saber, as metaheurísticas Busca em Vizinhança Variável e a Busca Dispersa. A Busca em Vizinhança Variável é uma técnica baseada em trocas de estruturas de vizinhança dentro de um algoritmo de busca local, e a metaheurística Busca Dispersa, um método evolutivo que combina sistematicamente conjuntos de soluções para se obter soluções melhores.

Essas técnicas de solução oferecem novas alternativas de solução que oferecem solução aos problemas encontrados com outros métodos, como é um baixo esforço computacional é uma melhor convergência, sendo este o principal aporte do trabalho.

Os algoritmos são apresentados sistematicamente, explicando os seus algoritmos e a forma como são adaptados para resolver o problema do planejamento da expansão de sistemas de transmissão considerando-se a modelagem matemática conhecida com o modelo de transporte e o modelo DC. São realizados testes com os sistemas Sul Brasileiro de 46 barras e o sistema IEEE de 24 barras, e os resultados obtidos são comparados com os obtidos com outras metaheurísticas, verificando seu melhor desempenho tanto em velocidade de processamento como em esforço computacional.

Palavras-chave: Planejamento da Expansão de Redes. Busca Dispersa. Vizinhança Variável. Metaheurísticas.

Abstract

Electric Energy Transmission Network Expansion Problem consist in choose among a set of pre-defined circuits candidates, who must be incorporated into the system so as to minimize the investment costs and operation and meet the future energy demand over a planning horizon with reliability, assuming the generation plan is known. It is a very complex and difficult problem because it is non linear, non convex, multimodal and highly combinatorial.

This problem has been solved using traditional techniques such as Benders decomposition and Branch and Bound, as well as heuristic algorithms and metaheuristics getting different results, but with a series of problems such as high computational effort and convergence problems.

This paper tests out two new techniques for solving the problem as are the metaheuristics Variable Neighborhood Search and Scatter Search.

The Variable Neighborhood Search is a technique based on trading structures within a neighborhood of a local search algorithm, and the Scatter Search metaheuristic is a method which combines systematically sets of solutions in an evolutionary way to achieve better solutions.

These solution techniques offer new alternatives to solve the problems encountered with other methods, such as a low computational effort and better convergence, which is the main contribution of this work.

The techniques are presented systematically, explaining their algorithms and the way they are adapted to solve the network expansion planning problem based on the mathematical model known as the transportation model and the DC model. They are tested with the systems Southern Brazilian with 46 buses and the IEEE 24 buses system, results are compared with those obtained with other metaheuristics, obtaining excellent results with a best performance both in processing speed as in computational effort.

Keywords: Network Expansion Planning. Scatter Search. Neighborhood Search. Metaheuristics.

Lista de Figuras

2.1	Sistema de 3 barras.	p. 23
2.2	Sistema de 3 barras com 2 circuitos na configuração base.	p. 29
3.1	Métodos de Solução.	p. 40
4.1	Algoritmo VND.	p. 57
4.2	Algoritmo VNS reduzida - RVNS.	p. 59
4.3	Algoritmo BVNS.	p. 61
4.4	Algoritmo GVNS.	p. 62
4.5	Algoritmo SVNS.	p. 64
4.6	Algoritmo VNDS.	p. 66
4.7	Algoritmo PVNS.	p. 67
4.8	Algoritmo VND Básico.	p. 72
4.9	Algoritmo VND para o PPEST.	p. 72
4.10	Sistema de 6 barras com seis circuitos na configuração base e seis circuitos adicionais.	p. 73
4.11	Codificação de uma proposta de solução para o PPEST.	p. 73
4.12	Vizinhos em $N_1(n)$	p. 76
4.13	Vizinhos em $N_2(n)$	p. 77
4.14	Vizinhos em $N_3(n)$	p. 77
4.15	Solução VGS para o sistema de 6 barras de Garver.	p. 80
4.16	Codificação da solução inicial do sistema de 6 barras de Garver.	p. 80
4.17	Vizinhos candidatos em $N_2(n)$	p. 81
4.18	Solução ótima (VND/DC) do sistema de 6 barras de Garver.	p. 82
4.19	Sistema Sul Brasileiro - Rede Inicial.	p. 86

4.20	Sistema Sul Brasileiro - Configuração Ótima DC com Reprogramação.	p. 90
4.21	Sistema Sul Brasileiro - Configuração Ótima DC sem Reprogramação.	p. 91
5.1	Algoritmo básico da Busca Dispersa.	p. 100
5.2	Principais passos do algoritmo BD para o PPEST.	p. 103
5.3	Passos do Algoritmo de Melhoria Local.	p. 106
5.4	Algoritmo para construir <i>RefSet</i>	p. 108
5.5	<i>Path Relinking</i>	p. 109
5.6	Sistema de 6 barras de Garver.	p. 111
5.7	Conjunto de Referência.	p. 112
5.8	Primeira solução combinada com PR e solução guia.	p. 112
5.9	Segunda solução combinada com PR e solução guia.	p. 112
5.10	Solução ótima (BD/DC) do sistema de 6 barras de Garver.	p. 113

Lista de Tabelas

4.1	VND: IEEE 24 barras - Modelo de Transportes.	p. 84
4.2	VND: IEEE 24 barras - Modelo DC.	p. 85
4.3	VND: Sul Brasileiro - Modelo de Transportes.	p. 87
4.4	VND: Sul Brasileiro - Modelo DC.	p. 88
5.1	BD: IEEE 24 barras - Modelo de Transportes.	p. 115
5.2	BD: IEEE 24 barras - Modelo DC.	p. 115
A.1	Dados de Linhas - IEEE 24 barras.	p. 129
A.1	Dados de Linhas - IEEE 24 barras (continuação).	p. 130
A.2	Dados de Barras - IEEE 24 barras.	p. 131
A.4	Dados de Linhas - Sul Brasileiro.	p. 132
A.4	Dados de Linhas - Sul Brasileiro (continuação).	p. 133
A.4	Dados de Linhas - Sul Brasileiro (continuação).	p. 134
A.3	Dados de Barras - Sul Brasileiro.	p. 135

Sumário

1	Introdução	p. 15
2	O Problema de Planejamento da Expansão de Sistemas de Transmissão	p. 18
2.1	Modelagem Matemática	p. 19
2.1.1	Modelo DC	p. 20
2.1.1.1	Exemplo: Sistema de 3 barras com modelo DC	p. 22
2.1.2	Modelo de Transportes	p. 24
2.1.2.1	Exemplo: Sistema de 3 barras com modelo de transportes . . .	p. 26
2.1.3	Modelo Híbrido	p. 27
2.1.3.1	Exemplo: Sistema de 3 barras com modelo híbrido linear . . .	p. 29
2.2	O Estado da Arte em Planejamento da Expansão de Sistemas de Transmissão . .	p. 31
2.3	Sistemas Considerados para Testes	p. 37
2.4	Outros Aspectos Relevantes	p. 38
3	Técnicas de Solução	p. 40
3.1	Métodos Exatos	p. 40
3.1.1	<i>Branch and Bound</i>	p. 41
3.1.2	Decomposição de Benders	p. 43
3.2	Algoritmos Aproximados	p. 43
3.2.1	Algoritmo Heurístico Construtivo de Garver	p. 44
3.2.2	AHC de Villasana-Garver-Salon	p. 45
3.2.3	Algoritmos Metaheurísticos	p. 48
3.2.3.1	Recozimento Simulado	p. 48
3.2.3.2	Algoritmos Genéticos	p. 49

3.2.3.3	Busca Tabu	p. 51
4	Busca em Vizinhança Variável	p. 54
4.1	Fundamentos da Busca em Vizinhança Variável	p. 54
4.1.1	VNS de Descida: Algoritmo VND	p. 56
4.1.2	VNS Reduzida: Algoritmo RVNS	p. 58
4.1.3	VNS Básica: Algoritmo BVNS	p. 60
4.1.4	VNS Geral: Algoritmo GVNS	p. 61
4.1.5	Extensões da VNS	p. 63
4.1.5.1	VNS Inclinada: Algoritmo SVNS	p. 63
4.1.5.2	VNS com Decomposição: Algoritmo VNDS	p. 65
4.1.5.3	VNS Paralela: Algoritmo PVNS	p. 65
4.1.5.4	VNS Híbrida	p. 66
4.2	Algumas aplicações recentes	p. 68
4.2.1	Tarefas clássicas de Inteligência Artificial	p. 68
4.2.2	Teoria dos Grafos	p. 69
4.2.3	Problemas de Otimização	p. 70
4.3	VNS aplicada no PPEST	p. 71
4.3.1	Passo 1: Codificação do PPEST	p. 72
4.3.2	Passo 2: Solução inicial	p. 74
4.3.2.1	Solução inicial com o Modelo de Transportes	p. 74
4.3.2.2	Solução inicial com o Modelo DC	p. 75
4.3.3	Passo 3: Definição das estruturas de vizinhanças	p. 76
4.3.4	Passo 4: Busca Local	p. 77
4.3.4.1	Modelo de Transportes	p. 78
4.3.4.2	Modelo DC	p. 79
4.3.5	Critério de Parada	p. 79
4.4	Exemplo: VND com Modelo DC no Sistema de 6 barras de Garver	p. 80

4.5	Redução de Vizinhaça	p. 82
4.6	Resultados Obtidos	p. 83
4.6.1	Sistema IEEE de 24 barras	p. 83
4.6.1.1	Modelo de Transportes	p. 84
4.6.1.2	Modelo DC	p. 84
4.6.2	Sistema Sul Brasileiro	p. 85
4.6.2.1	Modelo de Transportes	p. 87
4.6.2.2	Modelo DC	p. 88
4.7	VNS entre as Metaheurísticas	p. 91
5	Busca Dispersa	p. 95
5.1	Desenvolvimento da Busca Dispersa	p. 95
5.2	Descrição da Busca Dispersa	p. 98
5.3	Estratégias Avançadas	p. 100
5.4	Busca Dispersa aplicada no PPEST	p. 103
5.4.1	Passo 1: Codificação do PPEST para a BD	p. 104
5.4.2	Passo 2: Gerador de Soluções Diversas	p. 104
5.4.2.1	Soluções para o Modelo de Transportes	p. 104
5.4.2.2	Soluções para o modelo DC	p. 105
5.4.3	Passo 3: Melhoria das Soluções	p. 106
5.4.4	Passo 4: Construção do Conjunto de Referência	p. 108
5.4.5	Passo 5: Geração de Subconjuntos	p. 108
5.4.6	Passo 6: Combinação de Soluções	p. 109
5.4.7	Passo 7: Atualizar o conjunto de referência	p. 110
5.4.8	Passo 8: Critério de Parada	p. 111
5.5	Exemplo: BD no sistema de 6 barras de Garver	p. 111
5.6	Outras Estratégias	p. 113
5.7	Reconstrução do Conjunto de Referência	p. 114

5.8	Resultados Obtidos	p. 114
5.8.1	Sistema IEEE de 24 Barras	p. 114
5.8.1.1	Modelo de Transportes	p. 114
5.8.1.2	Modelo DC	p. 115
5.8.2	Sistema Sul Brasileiro	p. 115
5.8.2.1	Modelo de Transportes	p. 115
5.8.2.2	Modelo DC	p. 116
5.9	BD entre as Metaheurísticas	p. 116
6	Considerações Finais	p. 117
6.1	O Problema de Planejamento da Expansão de Sistemas de Transmissão	p. 117
6.2	Busca em Vizinhança Variável	p. 117
6.3	Busca Dispersa	p. 118
6.4	Busca em Vizinhança Variável ou Busca Dispersa?	p. 119
6.5	Trabalhos Futuros	p. 119
	Referências	p. 121
	Apêndice A - Dados dos Sistemas Testes	p. 129
A.1	Sistema IEEE (24 Barras/41 Caminhos)	p. 129
A.2	Sistema Sul Brasileiro (46 Barras/79 Caminhos)	p. 132
	Apêndice B - Artigos Publicados	p. 136

1 Introdução

Frente às recentes reestruturações impostas ao setor de energia elétrica e a necessidade de suprir a crescente demanda de energia elétrica que ocorre com o passar dos anos, surge a tendência dominante de competição de mercados. O novo entendimento para o setor identifica, pelo menos, três etapas distintas nesta cadeia de produção e consumo: geração, transmissão e distribuição. Neste sentido, as atividades de planejamento da expansão do sistema assumem uma importância ainda maior, em função da necessidade de conciliar interesses comerciais dos diversos agentes envolvidos. Então surge o problema de planejamento da expansão dos sistemas de geração e transmissão, o qual precisa ser realizado de forma conjunta e em um horizonte de longo prazo, mesmo que os agentes responsáveis sejam diversos.

Neste trabalho considera-se o planejamento da expansão do sistema de transmissão a longo prazo, um problema considerado clássico no setor de energia elétrica e cuja modelagem matemática ideal corresponde a um problema de programação não-linear inteira mista, e que além disso, apresenta o fenômeno de explosão combinatória. As principais dificuldades na resolução deste problema estão relacionadas com a natureza combinatória 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 e às vezes de qualidade pobre. Devido estas e outras dificuldades, existem inúmeras propostas de soluções para este problema, como aparece na literatura especializada (ROMERO; MONTICELLI, 1994a; ROMERO; MONTICELLI, 1994b; BINATO; OLIVEIRA, 1995; ROMERO; GALLEGOS; MONTICELLI, 1996; BINATO; PEREIRA; GRANVILLE, 2001; ROMERO et al., 2002; ROMERO et al., 2003; LATORRE et al., 2003; FARIA et al., 2005; SILVA et al., 2006; RIDER; GARCIA; ROMERO, 2007).

Inicialmente apresenta-se uma revisão do estado da arte do problema estático da expansão dos sistemas de transmissão de energia elétrica. Neste tipo de planejamento, dada uma configuração inicial, os dados de geração e de demanda do horizonte de planejamento, e outros dados como limites de operação, custos e restrições de investimento, procura-se determinar o plano de expansão com um custo mínimo, ou seja, deve-se determinar onde e que tipos de equipamentos devem ser instalados para que o sistema opere adequadamente, atendendo às necessidades do mercado de energia elétrica.

Ao se tratar de um problema de otimização matemática deve se especificar claramente dois aspectos importantes: a modelagem matemática e a técnica de solução escolhida para resolver o modelo matemático. Pode-se considerar pelo menos três modelos matemáticos para representar o problema de planejamento de sistemas de transmissão: modelo DC, modelo de transportes e modelo híbrido. O modelo DC é considerado como sendo o modelo mais indicado para representar o problema de planejamento, sendo os outros modelos versões relaxadas ou simplificadas do modelo DC. Na solução destes modelos são encontradas na literatura várias técnicas de otimização, que podem ser classificadas em duas categorias: métodos exatos (analíticos ou de otimização clássica) como Decomposição de Benders e *Branch and Bound*; e os métodos aproximados (heurísticas e metaheurísticas). Entretanto, existem ferramentas que apresentam características destas duas categorias técnicas, e em geral também são chamadas de metaheurísticas.

Em ambas categorias, deve-se encontrar um plano de expansão ótimo utilizando procedimentos de cálculo que resolvem a formulação matemática do problema. Devido a impossibilidade de se considerar todos os aspectos do problema de planejamento de sistemas de transmissão, o plano obtido é ótimo somente considerando algumas simplificações, e pode ser técnica, financeira e ambientalmente verificado através de outras técnicas, antes do planejador tomar uma decisão. Na formulação destes modelos, o problema de planejamento da expansão de sistemas de transmissão (PPEST) é descrito como um problema de otimização com uma função objetivo (um critério de medida para avaliar da mesma maneira a eficiência de cada opção de expansão), sujeita a um conjunto de restrições. Estas restrições procuram modelar grande parte dos critérios técnicos, econômicos, e de confiabilidade necessários para a expansão do sistema de potência.

Os métodos analíticos, geralmente utilizando técnicas de decomposição matemática e técnicas de otimização clássica como programação linear, programação dinâmica, programação não-linear e programação inteira mista, apresentam a característica de que determinam a solução do problema de planejamento. São muito eficientes em sistemas de pequeno e médio porte, mas para sistemas de grande porte ainda apresentam problemas de convergência e de elevado esforço computacional.

As técnicas heurísticas são as alternativas atuais para os modelos de otimização matemática. O termo heurística (um método de resolver problemas através de técnicas práticas aprendidas em experiências passadas; formada irregularmente da palavra grega 'heuriskein', que significa descobrir, inventar, ter uma idéia), é utilizado para descrever todas essas técnicas que, ao invés de usar uma aproximação da otimização clássica, gera, passo a passo, uma solução avaliando e selecionando opções de expansão, com ou sem a ajuda do planejador. Para isto, nos modelos heurísticos realizam-se buscas locais com a orientação lógica ou empírica de índices de sensibilidade (regras heurísticas). Estas regras são utilizadas para gerar e classificar as opções de construção de uma solução de boa qualidade durante a busca. Em geral, estes métodos apresentam a vantagem de fornecer soluções de boa qualidade com esforços computacionais pequenos, mas também não garantem que se encontre

a solução ótima de sistemas reais e não fornecem informações sobre a qualidade da solução obtida. Entretanto, estes métodos ainda são os mais utilizados pelas concessionárias de energia elétrica nos trabalhos de planejamento.

Surgiram também os algoritmos metaheurísticos, que descrevem como se explorar o espaço de busca sem se prender a um problema específico. Estes algoritmos coordenam heurísticas mais simples com uma busca local, com o propósito de encontrar soluções de melhor qualidade do que as obtidas utilizando as heurísticas isoladamente.

O objetivo deste trabalho é apresentar duas metaheurísticas que ainda não foram utilizadas na solução do PPEST, a saber, a Busca Dispersa (*Scatter Search*) e a Busca em Vizinhança Variável (*Variable Neighborhood Search*), aplicar estas técnicas na solução deste problema e realizar uma comparação entre o desempenho obtido com estas novas aplicações e outras encontradas na literatura. A organização deste trabalho é descrita a seguir.

No Capítulo 2 apresenta-se o problema de planejamento da expansão do sistema de transmissão a longo prazo e descrevem-se os modelos matemáticos e o estado da arte do PPEST.

No Capítulo 3 são apresentadas as principais técnicas de solução já aplicadas no PPEST.

A apresentação da metaheurística Busca em Vizinhança Variável, uma técnica baseada em mudanças sistemáticas de estruturas de vizinhança dentro de um algoritmo de busca local, é realizada no Capítulo 4. São apontados os fundamentos desta metaheurística, algumas de suas variações e uma revisão das principais publicações. Descreve-se, em detalhes, aplicação desta metaheurística no PPEST e comparam-se os resultados obtidos com outras metaheurísticas conhecidas. Também, neste capítulo, é discutida a importância de implementações considerando técnicas de redução de vizinhança.

No Capítulo 5 realiza-se uma revisão da metaheurística Busca Dispersa, pertencente à família dos algoritmos evolutivos e que se baseia no fato de combinar soluções de qualidade para se obter soluções melhores. Ilustra-se sua aplicação ao Problema de Planejamento da Expansão do Sistema de Transmissão de Energia Elétrica. Propõe-se uma heurística construtiva utilizada para encontrar as soluções iniciais, uma heurística de melhoria e técnicas de combinação por *Path Relinking*. Também são apresentadas algumas variações da Busca Dispersa e as principais aplicações já realizadas. Os resultados obtidos com a aplicação desta técnica na solução do PPEST também aparecem no final deste capítulo.

No Capítulo 6 são apresentados as conclusões sobre o trabalho e ainda algumas considerações sobre trabalhos futuros.

Os dados dos sistemas testados aparecem no Apêndice A e os artigos publicados com os resultados deste trabalho são relacionados no Apêndice B.

2 O Problema de Planejamento da Expansão de Sistemas de Transmissão

O problema de planejamento da expansão de sistemas elétricos de potência é um problema de grande importância no setor elétrico que deve ser solucionado para garantir que os consumidores sejam atendidos de forma econômica e confiável com o aumento da demanda de energia com o passar dos anos.

Apesar do sistema elétrico estar passando por um processo de desregulamentação, migrando de uma estrutura centralizada para uma descentralizada, que tem como objetivo obter uma maior eficiência dos agentes participantes do setor, (agentes de geração, agentes de transmissão, agentes de distribuição, entre outros) que decidirão onde e quando investir seus recursos para expandir o sistema, este processo deverá sofrer a interferência de um agente central que deve funcionar como um plano de referência de forma a garantir uma expansão ótima global do sistema, otimizando a utilização dos recursos disponíveis e os custos para os consumidores. Novos parques geradores e novas rotas de transmissão devem ser construídos para atender esta nova carga do sistema.

O Brasil passou recentemente por um grave problema de racionamento de energia devido à falta de investimentos na área de geração e transmissão de energia, agravado por um período de seca que baixou o nível dos reservatórios. Como maior parte da energia elétrica produzida no país é de origem hidrelétrica e as usinas estão distantes dos grandes centros consumidores, torna-se necessária a construção de novos circuitos de transmissão com a finalidade de transmitir a potência elétrica produzida nestas usinas, para aumentar a confiabilidade do sistema, otimizar recursos hídricos, etc.

As decisões do processo de planejamento estão relacionadas à seleção das melhores unidades geradoras, das melhores rotas de transmissão e das melhores malhas para garantir um suprimento de energia de forma econômica e confiável. Este processo de tomada de decisões dá origem a um problema de otimização complexo que deve ser resolvido pelos engenheiros de planejamento. É necessário o desenvolvimento de estratégias e técnicas de solução que assegurem que as decisões tomadas durante o processo de planejamento sejam as decisões ótimas, ou as melhores decisões possíveis. Este problema não pode ser resolvido sem que sejam feitas simplificações. Normalmente, o problema de planejamento é separado com relação aos seus principais agentes. O problema de planejamento da geração, que não leva em conta os custos da expansão da transmissão, o problema

de planejamento da transmissão, que supõe conhecidas as estimativas de crescimento da demanda e programas alternativos de expansão da geração, até o ano horizonte de planejamento e o planejamento da distribuição.

No planejamento de sistemas de transmissão ainda é possível separar o problema em dois tipos: (1) planejamento estático e (2) planejamento multiestágio. No planejamento estático existe apenas um estágio de planejamento, onde o planejador procura conhecer o circuito ótimo para ser adicionado em um único ano no horizonte de planejamento, ou seja o planejador não está interessado em saber quando o circuito deverá ser instalado, mas encontrar a topologia final ótima para uma futura situação definida. Por outro lado, se múltiplos anos são considerados e a estratégia de expansão ótima abrange todo o período, o planejamento é classificado como multiestágio. Neste caso, o modelo matemático deve conter restrições de tempo para considerar a expansão ao longo dos anos de tal forma que o valor presente dos custos ao longo do planejamento seja minimizado enquanto que as restrições impostas sejam atendidas.

O planejamento multiestágio é muito complexo pois deve levar em consideração não só as especificações técnicas e a alocação dos circuitos planejados, mas também considerações sobre o tempo. Poucos trabalhos sobre planejamento multiestágio para problemas reais de planejamento podem ser encontrados na literatura. Em (LATORRE et al., 2003) são apontados alguns destes trabalhos.

Neste trabalho, interessa-se pelo planejamento da expansão de sistemas de transmissão nos horizontes de médio e longo prazo (10 anos ou mais) que consiste em determinar onde novos equipamentos de transmissão (linhas de transmissão, transformadores, etc.) devem ser instalados de forma a atender a carga de forma econômica e confiável. Devido tanto às incertezas como também às dimensões inerentes a este tipo de problema, métodos rápidos e aproximados de análise são requeridos.

2.1 Modelagem Matemática

O problema de planejamento da expansão de sistemas de transmissão de energia elétrica consiste em se escolher, entre um conjunto pré-definido de circuitos candidatos, aqueles que devem ser incorporados ao sistema de forma a minimizar os custos de investimento e operação e atender a demanda de energia futura ao longo de um horizonte de planejamento com confiabilidade, assumindo como conhecido o plano de geração. Esse problema tem uma natureza dinâmica, isto é, requer a consideração de múltiplos períodos de tempo, determinandose uma seqüência (estágios) de planos de expansão do sistema. Quando o horizonte de planejamento reduz-se a apenas um estágio, o problema multiestágio se transforma em um problema estático, em que o objetivo é determinar onde e que tipo de novos equipamentos de transmissão devem ser instalados de forma a minimizar os custos de investimento sujeito a uma conjunto de restrições técnicas e operativas.

Os dados para este problema são a previsão de carga futura, bem como o despacho dos geradores para atender ao mercado. Além disso, são necessários dados para a rede existente, também chamada de rede básica, e dados para os novos circuitos que podem ser adicionados à rede básica. Note que a rede básica não tem capacidade suficiente para o atendimento da semana no mercado futuro.

O modelo matemático mais indicado para representar a operação adequada do sistema seria a representação do problema por meio de relações matemáticas de fluxo de carga AC (*Alternating Current*), tipicamente usada na análise da operação do sistema elétrico. Entretanto o uso desta modelagem é muito recente e, portanto, sua aplicação ainda se encontra restrita e em fase de desenvolvimento. Assim, considera-se que a modelagem matemática mais indicada em trabalhos de planejamento de sistemas de transmissão, é o chamado modelo DC (*Direct Current*) que leva em conta as duas leis de Kirchhoff apenas para o balanço e o fluxo de potência ativa. Nesse caso, o problema resultante é do tipo de programação não-linear inteiro misto de elevada complexidade para sistemas de grande porte.

2.1.1 Modelo DC

Como foi mencionado anteriormente, o modelo DC é atualmente considerado o modelo matemático mais indicado para representar o problema de planejamento da expansão de sistemas de transmissão. Os principais motivos para essa opção são os seguintes: (1) É a modelagem mais aceita por pesquisadores e especialistas em planejamento das empresas de energia elétrica; (2) Existem várias técnicas de solução (algoritmos) que resolvem de maneira adequada os problemas de planejamento que usam o modelo DC; E requerem tempo de execução reduzido.

Neste modelo, o sistema completo deve satisfazer as duas leis de Kirchhoff, ou seja, todas as barras do sistema devem satisfazer a primeira lei de Kirchhoff (a primeira lei de Kirchhoff simplesmente especifica que o somatório dos fluxos de potência que entram numa barra do sistema deve ser igual ao somatório do fluxo de potência que saem dessa barra do sistema), e todos os laços existentes devem satisfazer à segunda lei de Kirchhoff (a segunda lei de Kirchhoff especifica que a queda de tensão em cada laço deve ser igual a zero).

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

s.a.

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

$$f_{ij} - \gamma_{ij}(n_{ij}^0 + n_{ij})(\theta_i - \theta_j) = 0 \quad (2.3)$$

$$|f_{ij}| \leq (n_{ij}^0 + n_{ij}) \bar{f}_{ij} \quad (2.4)$$

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

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

$$n_{ij} \text{ inteiro, } f_{ij} \text{ irrestrito, } \theta_j \text{ irrestrito} \quad (2.7)$$

$$\forall (i, j) \in \Omega \quad (2.8)$$

O modelo DC assume a forma apresentada em (2.1)-(2.8), sendo:

v : Investimento devido as adições de circuitos.

c_{ij} : Custo de um circuito que pode ser adicionado no caminho ij .

n_{ij} : Número de circuitos adicionados no processo de otimização.

n_{ij}^0 : Número de circuitos existentes na topologia base.

$\bar{n}_{ij}$: Número máximo de circuitos que podem ser adicionados no caminho ij .

γ_{ij} : Suceptância de um circuito no caminho ij .

f : Vetor de fluxos com elementos f_{ij} (o fluxo de potência total através dos circuitos no caminho ij).

$\bar{f}_{ij}$: Capacidade de transmissão de um circuito no caminho ij .

g : Vetor geração com elementos g_k (geração na barra k).

$\bar{g}$: Vetor de geração máxima.

S : Matriz de incidência nó-ramo transposta do sistema elétrico.

θ_i : Ângulo de tensão na barra i .

Ω : Representa o conjunto de caminhos em que é possível adicionar circuitos.

O conjunto de restrições (2.2) representa as equações correspondentes à primeira lei de Kirchhoff, uma equação para cada barra do sistema, e as restrições (2.3) representam as equações correspondentes à segunda lei de Kirchhoff. As restrições (2.4) 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 nos dois sentidos. As restrições (2.5) e (2.6) são triviais e representam apenas restrições de limite de geração e de circuitos adicionados em cada caminho candidato ij . Finalmente, as variáveis f_{ij} e θ_i são irrestritas em valor e as variáveis n_{ij} devem ser inteiras, representando a maior fonte de complexidade no problema.

A presença de todas as equações correspondentes à segunda lei de Kirchhoff no modelo DC transforma este modelo num problema não linear inteiro misto e produzem um alto nível de complexidade no processo de solução.

2.1.1.1 Exemplo: Sistema de 3 barras com modelo DC

Considerando-se o sistema de 3 barras apresentado na Figura 2.1, cujos dados deste sistema, que também são mostrados na Figura 2.1, são os seguintes:

Custo dos circuitos: $c_{12} = 3$, $c_{13} = 2$ e $c_{23} = 2$ (u.m.).

Susceptâncias: $\gamma_{12} = \frac{1}{3}$, $\gamma_{13} = \frac{1}{2}$ e $\gamma_{23} = \frac{1}{2}$ (p.u., para uma base de 100 MW).

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

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

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

Inicialmente deve-se escolher um sentido para os fluxos f_{ij} . Embora esta escolha seja arbitrária, em todos os exemplos apresentados neste trabalho 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 fornecer o sentido correto dos fluxos para cada exemplo.

As equações correspondentes à primeira lei de Kirchhoff aplicadas em cada barra do sistema, considerando positivo o fluxo que entra na barra, produzem as seguintes equações:

1. Equações da primeira lei de Kirchhoff:

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

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

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

2. Equações da segunda lei de Kirchhoff:

$$\text{Circuito 1-2:} \quad f_{12} = \frac{1}{3}(1 + n_{12})(\theta_1 - \theta_2)$$

$$\text{Circuito 1-3:} \quad f_{13} = \frac{1}{2}(1 + n_{13})(\theta_1 - \theta_3)$$

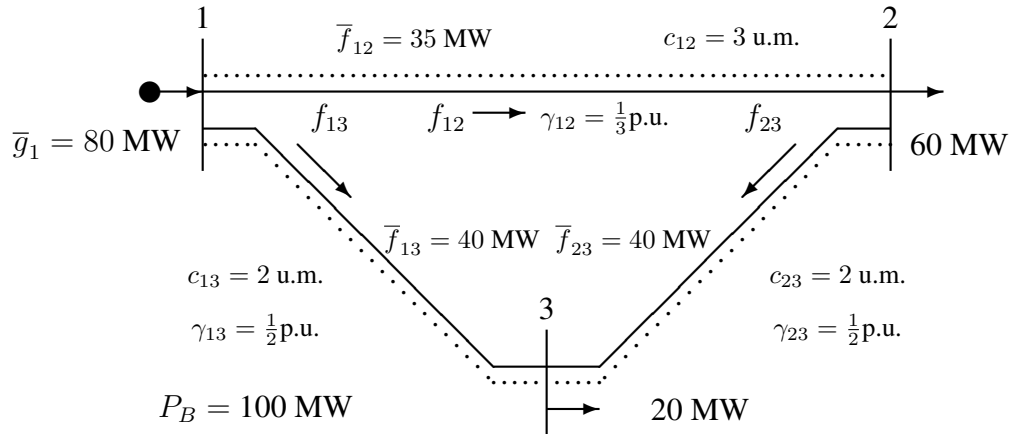


Figura 2.1: Sistema de 3 barras.

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

3. Inequações correspondentes aos limites de fluxos:

Circuito 1-2: $|f_{12}| \leq 0,35(1 + n_{12})$

Circuito 1-3: $|f_{13}| \leq 0,40(1 + n_{13})$

Circuito 2-3: $|f_{23}| \leq 0,40(1 + n_{23})$

As outras restrições são triviais e assim, para o exemplo da Figura 2.1, a modelagem matemática do modelo DC assume a seguinte forma:

$$\min v = 3n_{12} + 2n_{13} + 2n_{23} \quad (2.9)$$

s.a.

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

As incógnitas no problema são: $v, n_{12}, n_{13}, n_{23}, f_{12}, f_{13}, f_{23}, \theta_1, \theta_2, \theta_3$ e g_1 .

2.1.2 Modelo de Transportes

Também pode-se considerar uma modelagem mais simplificada, conhecida como modelo de transportes, que considera apenas a primeira lei de Kirchhoff. Nesse caso, o problema resultante é do tipo linear inteiro misto. Mesmo sendo linear, ainda não é possível encontrar a solução ótima para o modelo de transportes para sistemas de grande porte e de elevada complexidade.

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 em (GARVER, 1970) 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 utilizado 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 (LKC), isto é, a modelagem matemática proposta não leva em conta a segunda lei de Kirchhoff (LKT). 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 produza o menor investimento no plano de expansão do sistema elétrico e condições adequadas de operação desse sistema elétrico. Condições adequadas de operação significam 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.

Levando em conta as observações anteriores, o modelo de transportes para o problema de planejamento de sistemas de transmissão pode ser formulado por (2.10)-(2.17):

$$\min v = \sum_{(i,j) \in \Omega} c_{ij} n_{ij} \quad (2.10)$$

s.a.

$$Sf + g = d \quad (2.11)$$

$$|f_{ij}| \leq (n_{ij}^0 + n_{ij}) \bar{f}_{ij} \quad (2.12)$$

$$0 \leq g \leq \bar{g} \quad (2.13)$$

$$0 \leq n_{ij} \leq \bar{n}_{ij} \quad (2.14)$$

$$f_{ij} \text{ irrestrito} \quad (2.15)$$

$$n_{ij} \text{ inteiro} \quad (2.16)$$

$$(i, j) \in \Omega \quad (2.17)$$

em que os parâmetros e as incógnitas são os mesmos do modelo DC. Na verdade, o modelo de transportes pode ser obtido a partir do modelo DC após relaxar (eliminar) as restrições da segunda lei de Kirchhoff (2.3).

Do ponto de vista de pesquisa operacional o problema (2.10)-(2.17) é um problema de programação linear inteiro misto (PLIM). 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 fossem permitido que os n_{ij} assumissem valores reais, então o sistema (2.10)-(2.17) se transformaria num simples problema de programação linear (PL), fácil de resolver mesmo para o caso de sistemas de grande porte.

É evidente que a restrição n_{ij} inteiro produz a maior complexidade no problema (2.10)-(2.17).

Estas características são 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 e a característica linear facilita o processo de resolução. A desvantagem principal é que a solução apresentada pelo modelo de transportes pode estar distante da solução correspondente ao modelo DC, considerado como modelo mais indicado.

2.1.2.1 Exemplo: Sistema de 3 barras com modelo de transportes

Considerando-se novamente o sistema de 3 barras da Figura 2.1, pode-se dar a representação deste sistema para o modelo de transportes.

1. As equações correspondentes à primeira lei de Kirchhoff aplicadas para cada barra, considerando positivo o fluxo que entra na barra, produzem as seguintes equações:

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

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

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

2. As desigualdades correspondentes aos limites de capacidade de transmissão produzem as seguintes relações:

$$\text{Circuito 1-2: } |f_{12}| \leq 0,35(1 + n_{12})$$

$$\text{Circuito 1-3: } |f_{13}| \leq 0,40(1 + n_{13})$$

$$\text{Circuito 2-3: } |f_{23}| \leq 0,40(1 + n_{23})$$

3. A restrição sobre capacidade de geração produz a seguinte restrição:

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

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

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

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

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

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

exemplo da Figura 2.1, o modelo matemático pode ser escrito como:

$$\min v = 3n_{12} + 2n_{13} + 2n_{23} \quad (2.18)$$

s.a.

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

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

2.1.3 Modelo Híbrido

Outro modelo considerado no PPEST é o modelo híbrido, que combina características do modelo DC e do modelo de transportes. Esta modelagem, numa formulação mais simples preserva as propriedades lineares do modelo de transportes, considerando a primeira lei de Kirchhoff em todos os nós da rede, e a segunda lei de Kirchhoff **somente nos circuitos existentes na rede base** (e não necessariamente nos circuitos que serão adicionados).

O modelo híbrido foi proposto originalmente em (VILLASANA; GARVER; SALON, 1985), sendo que uma modelagem matemática especifica o seguinte: a parcela do sistema elétrico correspondente aos circuitos existentes na configuração base devem satisfazer as duas leis de Kirchhoff e a outra parcela correspondente aos circuitos novos devem satisfazer unicamente a primeira lei de Kirchhoff. Portanto, o modelo híbrido é uma mistura entre o modelo de transportes e o modelo DC.

É claro que uma vez definida a modelagem matemática desta maneira, a solução ótima encontrada também deve satisfazer as duas leis de Kirchhoff na parte do sistema correspondente aos circuitos da configuração base e somente a primeira lei de Kirchhoff para os novos circuitos adicionados. Em outras palavras, no modelo híbrido linear, deve-se satisfazer a primeira lei de Kirchhoff em todas as barras do sistema e a segunda lei de Kirchhoff somente naqueles laços formados por circuitos existentes na configuração base. Assim, por exemplo, se no processo de planejamento for adicionado um circuito, então os laços que eventualmente podem aparecer como consequência da

adição deste circuito não estão obrigados a satisfazer a segunda lei de Kirchhoff. Esta modelagem, chamada de modelo híbrido linear, pode ser usada como estratégia de otimização para encontrar soluções factíveis para o modelo DC. Esta proposta foi apresentada por (GLOVER, 1989), onde a modelagem é apenas utilizada como uma forma de auxílio para o indicador de sensibilidade do algoritmo heurístico proposto. Existe ainda um modelo híbrido não-linear, o qual não será considerado neste trabalho.

A idéia de utilizar 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 é preferido por sua natureza linear, pois para esse tipo de problema existem algoritmos relativamente eficientes, inclusive com provas de otimalidade. Mas, em contrapartida, as soluções encontradas podem ficar muito afastadas da solução ótima do modelo DC. Por outro lado, o modelo DC pode apresentar sérios problemas devido sua natureza não-linear. Assim, o modelo híbrido permite encontrar soluções mais próximas da solução ótima do modelo DC e com a vantagem de trabalhar com técnicas de otimização para problemas lineares. A versão do modelo híbrido linear, que está sendo apresentada e as diferentes variantes que aparecem na literatura especializada, são utilizadas apenas para auxiliar no processo de resolução do modelo DC em algoritmos de planejamento de sistemas de transmissão (ROMERO; MONTICELLI, 1994a; VILLASANA; GARVER; SALON, 1985).

Levando em conta estas observações o modelo híbrido linear pode ser formulado por (2.19)-(2.27),

$$\min v = \sum_{(i,j) \in \Omega} c_{ij} n_{ij} \quad (2.19)$$

s.a.

$$Sf + S^0 f^0 + g = d \quad (2.20)$$

$$f_{ij}^0 - \gamma_{ij}(n_{ij}^0)(\theta_i - \theta_j) = 0, \forall (i, j) \in \Omega_0 \quad (2.21)$$

$$|f_{ij}^0| \leq n_{ij}^0 \bar{f}_{ij}, \forall (i, j) \in \Omega_0 \quad (2.22)$$

$$|f_{ij}| \leq n_{ij} \bar{f}_{ij}, \forall (i, j) \in \Omega \quad (2.23)$$

$$0 \leq g \leq \bar{g} \quad (2.24)$$

$$0 \leq n_{ij} \leq \bar{n}_{ij} \quad (2.25)$$

$$f_{ij}^0 \text{ irrestritos } \forall (i, j) \in \Omega_0 \quad (2.26)$$

$$f_{ij} \text{ e } \theta_j \text{ irrestritos } \forall (i, j) \in \Omega \quad (2.27)$$

Pode-se verificar que o modelo aqui apresentado é um problema de programação linear inteiro misto. Deve-se observar também, que no modelo mostrado em (2.19)-(2.27) os fluxos nos circuitos estão separados em dois grupos (fluxos em circuitos existentes na topologia base, f_{ij}^0 e os fluxos nos

circuitos que não estão presentes na topologia base, f_{ij}). O mesmo acontece com os circuitos adicionados: n_{ij}^0, n_{ij} que representam, respectivamente, o número de circuitos presentes no caminho ij na configuração base e os circuitos que podem ser adicionados no processo de otimização. Também, Ω_0 representa o conjunto dos circuitos presentes na configuração base e Ω representa o conjunto dos circuitos que podem ser adicionados. S^0 é a matriz de incidência nó-ramo dos circuitos existentes na topologia base. As demais variáveis são como em (2.1)-(2.8).

As restrições (2.20) representam a primeira lei de Kirchhoff e o conjunto de restrições (2.21) 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 $f_{ij}^0 - \gamma_{ij}(n_{ij} + n_{ij}^0)(\theta_i - \theta_j) = 0$, simplesmente não aparece, já no modelo híbrido aparece somente uma parcela dessas equações constituídas pelos circuitos existentes na configuração base e, finalmente, no modelo DC aparecem todas as equações desse tipo, uma para cada caminho existente e/ou novos caminhos candidatos à adição de circuitos.

2.1.3.1 Exemplo: Sistema de 3 barras com modelo híbrido linear

Considerando-se o sistema da Figura 2.2, de 3 barras com dois circuitos na configuração base, tem-se:

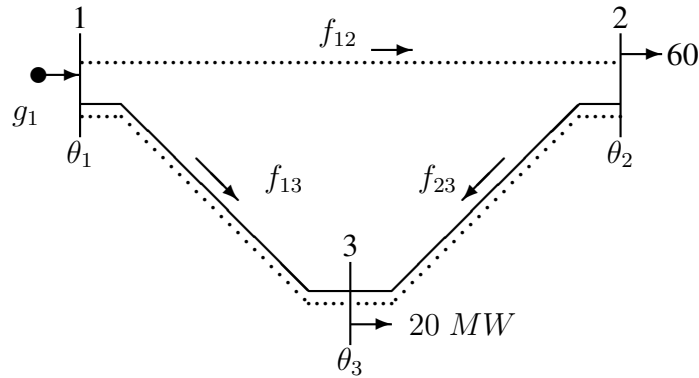


Figura 2.2: Sistema de 3 barras com 2 circuitos na configuração base.

1. Equações da primeira lei de Kirchhoff:

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

$$\text{Barra 2: } -f_{23}^0 + f_{12} - f_{23} = 0, 60$$

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

2. Equações da segunda lei de Kirchhoff:

$$\text{Circuito 1-3: } f_{13}^0 = \frac{1}{2}(\theta_1 - \theta_3)$$

Circuito 2-3: $f_{23}^0 = \frac{1}{2}(\theta_2 - \theta_3)$

3. Desigualdades correspondentes aos limites de fluxos para circuitos da configuração base:

Circuito 1-3: $|f_{13}^0| \leq 0,40$

Circuito 2-3: $|f_{23}^0| \leq 0,40$

4. Desigualdades correspondentes aos limites de fluxos para circuitos fora da configuração base:

Circuito 1-2: $|f_{12}| \leq 0,35 n_{12}$

Circuito 1-3: $|f_{13}| \leq 0,40 n_{13}$

Circuito 2-3: $|f_{23}| \leq 0,40 n_{23}$

As outras restrições são triviais e assim, para o exemplo da Figura 2.2, e utilizando a formulação (2.19)-(2.27), a modelagem matemática do modelo híbrido assume a seguinte forma:

$$\min v = 3n_{12} + 2n_{13} + 2n_{23} \quad (2.28)$$

s.a.

$$\begin{aligned} -f_{13}^0 - f_{12} - f_{13} + g_1 &= 0 \\ -f_{23}^0 + f_{12} - f_{23} &= 0,60 \\ f_{13}^0 + f_{23}^0 + f_{13} + f_{23} &= 0,20 \\ f_{13}^0 &= \frac{1}{2}(\theta_1 - \theta_3) \\ f_{23}^0 &= \frac{1}{2}(\theta_2 - \theta_3) \\ |f_{13}^0| &\leq 0,40 \\ |f_{23}^0| &\leq 0,40 \\ |f_{12}| &\leq 0,35 n_{12} \\ |f_{13}| &\leq 0,40 n_{13} \\ |f_{23}| &\leq 0,40 n_{23} \end{aligned}$$

$$0 \leq g_1 \leq 0,80$$

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

$$f_{13}^0, f_{23}^0, f_{12}, f_{13} \text{ e } f_{23} \text{ irrestritos}$$

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

As incógnitas no problema são: $v, n_{12}, n_{13}, n_{23}, f_{12}, f_{13}, f_{23}, \theta_1, \theta_2, \theta_3$ e g_1 .

2.2 O Estado da Arte em Planejamento da Expansão de Sistemas de Transmissão

Nesta seção considerou-se como base os trabalhos (LATORRE et al., 2003) e (FARIA, 2005).

Até a década passada, as únicas ferramentas disponíveis para a síntese de redes de transmissão eram os *softwares* de análise como os utilizados no cálculo de fluxo de carga, estudos de estabilidade, curto-circuito, etc. O planejador do sistema de energia elétrica era o responsável por determinar onde instalar novos equipamentos para suprir as novas cargas do sistema, resultando em uma configuração que deveria ser analisada através dos métodos mencionados anteriormente. Com o crescimento das dimensões das redes de transmissão, este procedimento se torna inviável.

Mas as pesquisas na área de planejamento de sistemas de transmissão experimentaram uma expansão e novos desenvolvimentos de modelos e técnicas de solução. Muitos artigos e relatos sobre novos modelos foram publicados na literatura especializada devido principalmente à melhoria das ferramentas tecnológicas disponíveis, novos algoritmos de otimização, e o maior nível de incerteza introduzidos pela desverticalização do setor de energia.

O trabalho pioneiro (KNIGHT, 1960) teve o mérito de propor a distinção entre os métodos de análise e métodos matemáticos de projeto (síntese) de sistemas de transmissão de energia elétrica.

Um dos primeiros trabalhos propostos para a solução deste problema é (GARVER, 1970). O autor formulou o problema considerando apenas a Primeira Lei de Kirchhoff e resolveu este modelo matemático usando um Algoritmo Heurístico Construtivo (AHC) que em cada passo era escolhido o circuito mais interessante para ser incorporado ao sistema identificado após resolver um problema de programação linear. Na seção 3.2.1 este método será detalhado.

Em (KALTENBATCH; PESHON; GEHRIG, 1970), também no ano de 1970, propuseram combinar programação linear com programação dinâmica. Programação linear era usada para encontrar o mínimo incremento da capacidade da rede para atender às variações de demanda e geração nas barras do sistema. Após essa etapa, era utilizada programação dinâmica para achar a melhor sequência de investimentos (contínuos) para o período de planejamento. Este trabalho é pioneiro para problemas de planejamento de expansão de redes de transmissão considerando múltiplos estágios.

Um algoritmo ‘puro’ de programação dinâmica foi proposto (DUSONCHET; EL-ABIAD, 1973). Esta proposta parecia contornar as dificuldades em obter a solução ótima dos trabalhos anteriores. Contudo, devido aos altos recursos computacionais requeridos, resultado do formalismo da programação dinâmica, simplificações ou relaxações de importantes restrições eram necessárias em aplicações práticas.

Tendo em vista as desvantagens da programação dinâmica, foi proposto, em (GONZAGA, 1973), um algoritmo de busca em grafos. Este algoritmo, uma versão de algoritmo dual, procura encontrar

um caminho de custo mínimo em grafos de expansão utilizando heurísticas para reduzir o número de alternativas a serem analisadas. Com base em tal algoritmo, foi implementado um programa computacional chamado *Tania*, que foi muito utilizado na solução de problemas de planejamento da expansão de redes com sistemas Brasileiros.

O conceito de rede adjunta combinada com o modelo de fluxo linearizado foi a proposta apresentada em (FISCHL; PUNTEL, 1973). Este trabalho procurava pela variação contínua das susceptâncias dos circuitos que minimiza o custo de reforços na rede de transmissão. Posteriormente, um procedimento heurístico chamado método do vizinho mais próximo seria utilizado para obter os valores discretos das susceptâncias dos circuitos.

A primeira proposta de algoritmos do tipo *Branch and Bound* para este problema apareceu em (LEE; HOCKS; HNYLICZA, 1974). Contudo, assim como nos métodos de programação dinâmica, a utilização de algoritmos combinatórios tipo *Branch and Bound* fica restrita a aplicações a sistemas de pequeno porte face aos recursos computacionais exigidos.

Em (MONTICELLI et al., 1985) propuseram o uso de ferramentas interativas para o planejamento da transmissão. Para ordenar as possibilidades de adições era utilizado o índice de 'Mínimo Esforço', que consiste de uma análise de sensibilidade em relação às susceptâncias dos circuitos em um problema de otimização correlato cujo resultado é idêntico ao modelo de fluxo de carga linearizado.

O uso de análise de sensibilidade no problema de planejamento da sistemas de transmissão foi inicialmente proposto no trabalho (CHAMPS; VANKELECOM; AMOULLE, 1979). Eles utilizaram análise de sensibilidade em relação às susceptâncias a partir de um problema de programação linear cujas restrições são as equações do modelo de fluxo de carga linearizado em conjunto com limites de transporte nos circuitos e de capacidade nos geradores. O objetivo do problema era obter o mínimo corte de carga necessário para eliminar todas as violações operacionais na rede elétrica. O uso de análise de sensibilidade também foi proposto por em (PEREIRA et al., 1987).

Em (BENNON; JUVES, 1982) utilizaram análise de sensibilidade com relação às susceptâncias dos circuitos em conjunto com o modelo linearizado de fluxo de potência, com o objetivo de determinar o caminho mais efetivo para a minimização de um índice de desempenho do sistema.

Em (VILLASANA, 1984) e depois em (VILLASANA; GARVER; SALON, 1985), são apresentadas duas diferentes metodologias para serem aplicadas ao planejamento da expansão de sistemas de transmissão. Estas propostas consistiam de um aperfeiçoamento do trabalho feito por (GARVER, 1970), que propôs o modelo de transportes, representando uma técnica fundamental na pesquisa em planejamento da expansão de sistemas de transmissão, pois naquela época era a única forma disponível de otimizar este problema. Esse modelo relaxado, diferente dos usados em análise de operação, foi chamado de modelo de síntese de sistemas de transmissão. O modelo de transportes, assim como todo modelo de síntese, faz apenas o planejamento considerando o fluxo de potência

ativa. Portanto, resolve apenas o problema de capacidade de transmissão. O problema de planejamento de reativos era resolvido na fase seguinte. Na tentativa de aperfeiçoar estas propostas e diminuir a complexidade do problema, Villasana determinou soluções viáveis para o modelo DC resolvendo modelos híbridos lineares, em que apenas os circuitos existentes na configuração base devem, obrigatoriamente, satisfazer as duas leis de Kirchhoff. Esta metodologia será considerada neste trabalho como estratégia de otimização, e será detalhada na seção 3.2.2.

Em contraste, o modelo híbrido não-linear é um problema de programação não-linear inteiro misto (PNLIM) de complexidade muito parecida como o modelo DC. Esse modelo foi pouco usado por pesquisadores em planejamento de sistemas de transmissão porque devem ser utilizadas as mesmas técnicas utilizadas para o modelo DC e, portanto, pode ser preferível trabalhar diretamente com o modelo DC, considerado ideal. Mesmo assim, o modelo híbrido não-linear deve ser mais fácil de resolver que o modelo DC.

O uso de esquemas de decomposição para este problema teve início com o trabalho (PEREIRA, 1985). Naquele trabalho, um esquema de decomposição de Benders foi aplicado para decompor o problema global de planejamento de redes em dois subproblemas: um de investimento, que tem por objetivo propor um plano de expansão; e outro de operação, que deve analisar o plano proposto e expressar as restrições operacionais em termos das variáveis de investimento através de restrições lineares chamadas de cortes de Benders. Esta nova restrição deve ser adicionada ao subproblema de investimento e novas iterações de Benders são repetidas até a obtenção da convergência. O modelo adotado para formular o problema de planejamento da expansão de redes de transmissão é não-linear e não-convexo, o que pode trazer sérias dificuldades para métodos de cortes como o algoritmo de decomposição de Benders. A aplicação de métodos de planos cortantes a um problema não-linear e não-convexo pode não ser bem sucedida pois os cortes produzidos podem excluir partes da região de viabilidade do problema, inclusive a região que contém a solução ótima.

Com o objetivo de contornar esta deficiência do método de decomposição de Benders, em relação ao modelo não-linear e não-convexo, foi proposto na tese de doutorado de (ROMERO, 1993), aparecendo também em (ROMERO; MONTICELLI, 1994a), uma metodologia de decomposição hierárquica composta por três fases distintas. Na primeira fase o problema de planejamento deve ser resolvido por decomposição de Benders considerando somente o modelo de transporte para o subproblema de operação. Além disso, a integralidade das variáveis de investimento deveriam ser relaxadas. Na segunda fase o modelo do subproblema de operação deve ser trocado por um modelo Híbrido (mais apurado) que consiste do modelo de fluxo de potência linearizado para os circuitos existentes e um modelo de transporte para computar o fluxo nos circuitos planejados. Finalmente, na terceira fase deste trabalho, o modelo de fluxo de carga linearizado era utilizado para o cálculo do fluxo de carga em todos os circuitos da rede de transmissão. O subproblema de investimento considera as variáveis de investimento discretas e utiliza um algoritmo especializado de enumeração

implícita (ROMERO, 1993; ROMERO; MONTICELLI, 1994b).

Em (PINTO; NUNES, 1990) usaram o esquema de decomposição de Benders combinado com um algoritmo de enumeração implícita. Com o objetivo de reduzir o esforço computacional, que pode ser muito grande, eles utilizaram duas técnicas para redução do espaço de busca do problema: redução por inviabilidade e por custo.

Outro método de decomposição proposto para o problema de planejamento, foi proposto em (LEVI; CALOVIC, 1991). Neste trabalho propôs-se dividir o problema de planejamento em dois problemas menores, um tratando somente com questões de investimento e outro considerando somente problemas relacionados à operação. O problema de investimento foi especificado como um especificado como um problema de fluxo de custo mínimo em rede. Este problema era decomposto novamente em dois subproblemas, o primeiro para computar o fluxo inicial que utiliza um algoritmo de programação linear para calcular o mínimo corte de carga. O segundo subproblema utilizava o modelo de rede marginal para obter a solução do modelo de fluxo de carga 'sobrecarregado'.

Em (LATORRE-BAYONA; PÉRES, 1994) propuseram uma metodologia heurística que utiliza a vantagem da decomposição natural do problema em subproblemas de operação e investimento. O subproblema de investimento era resolvido utilizando-se um procedimento heurístico de busca em árvore iniciada a partir de uma solução viável obtida por outros modelos. As variáveis de investimento (ramos da árvore de busca) poderiam ser classificadas de três maneiras: as variáveis questionáveis (circuitos incluídos na solução viável inicial, mas que o usuário pensa não pertencer ao plano ótimo), as variáveis atrativas (circuitos que o usuário pensa pertencer ao planejamento ótimo) e as variáveis congeladas (circuitos que não serão testados no processo de busca). Esta classificação das variáveis já consiste de um critério de truncamento utilizado por este trabalho com o objetivo de redução do tempo computacional. Os outros critérios utilizados eram limites na profundidade e na largura do processo de busca na árvore, limite no número de resoluções do subproblema de operação e limite no número de 'passos errados' do processo de busca na árvore.

Em (BINATO; OLIVEIRA, 1995) propuseram um método de busca, *backward/forward* para o problema de planejamento de expansão de redes de transmissão multiestágio. Neste método são definidos passos para uma análise de planejamento a dois estágios: o passo *backward*, que consiste de um planejamento retornando no tempo buscando antecipações de circuitos já definidos para os anos seguintes e o passo *forward*, que faz uma análise no sentido correto do tempo. Utilizando de uma maneira organizada estes dois passos, o método explora a região de viabilidade do problema em busca de economias de escala quando são considerados vários estágios durante o horizonte de planejamento.

Também em (OLIVEIRA; COSTA; BINATO, 1995) foi utilizado um esquema de decomposição hierárquica, mas composto por duas fases ao invés de três fases como no trabalho de Romero. A

primeira fase, da mesma forma que no trabalho de Romero, considera somente o modelo de transporte, porém não relaxa a integralidade das variáveis de investimento, enquanto que a segunda fase é igual à terceira do trabalho de Romero. A maior diferença entre estes dois trabalhos não vem da decomposição hierárquica utilizada e sim da maneira como o subproblema de investimento era solucionado. Enquanto que o trabalho anterior resolvia o subproblema de investimento até obter a solução ótima utilizando um algoritmo de enumeração implícita especializado, neste trabalho, utiliza-se um algoritmo de *branch and bound* com o objetivo de achar somente a primeira solução viável. Com isso, é possível se obter considerável redução do esforço computacional.

Na tese de doutorado de (BINATO, 2000), foi proposta uma aplicação computacional utilizando decomposição de Benders que assegura que a solução ótima, obtida pelo método de decomposição, é o plano ótimo de expansão da rede de transmissão. Isso está diretamente relacionado com a utilização do modelo linear (0-1) disjuntivo que pôde ser aplicado a problemas testes com sistemas reais devido à obtenção de valores mínimos para a constante disjuntiva. Uma nova heurística para determinar a convergência do problema mestre da decomposição de Benders resultou também em grandes economias em termos de tempo computacional gasto. Entretanto, muitas vezes, o número elevado de candidatos de um caso de planejamento da expansão impede a aplicação com sucesso de técnicas de decomposição. Portanto, é necessário o desenvolvimento de técnicas heurísticas que sejam capazes de prover 'boas' soluções para o problema.

Na década de 90 apareceram novos algoritmos heurísticos e metaheurísticos, diferentes dos algoritmos tradicionais, geralmente mais eficientes e com uma grande variedade de tempo de processamento que pode ser calibrado para cada tipo de aplicação. Pertence a esse tipo de algoritmos técnicas de otimização como algoritmos genéticos e evolutivos em geral, *tabu search*, GRASP, *particle swarm*, *ant colony*, etc.

As metaheurísticas apresentam a grande vantagem de que a forma de resolver um problema varia muito pouco quando se muda a modelagem matemática do problema. Assim, por exemplo, em planejamento de sistemas de transmissão, a forma usada para resolver os modelos de transporte, híbridos e o modelo DC é praticamente a mesma. Em cada caso, deve-se resolver apenas um PL sob diferentes formas. Por esse motivo, todas as aplicações de metaheurísticas em planejamento de sistemas de transmissão foram aplicadas diretamente no modelo DC. As principais aplicações de metaheurísticas no problema de planejamento são apresentadas em (ROMERO; GALLEGO; MONTICELLI, 1996; GALLEGO et al., 1997; GALLEGO; MONTICELLI; ROMERO, 1998a; GALLEGO; MONTICELLI; ROMERO, 1998b; GALLEGO; MONTICELLI; ROMERO, 2000; SILVA; GIL; AREIZA, 2000; SILVA et al., 2001).

Recozimento Simulado (*Simulated Annealing*) foi aplicado em (ROMERO; GALLEGO; MONTICELLI, 1996) e posteriormente foi paralelizado em (GALLEGO et al., 1997). A qualidade dos resultados publicados nestes dois artigos mostraram que tais métodos têm um excelente potencial para a

solução deste problema.

As pesquisas apresentadas, utilizando metaheurísticas, indicam que, no momento, esses tipos de algoritmos são os mais competitivos para encontrar soluções de excelente qualidade de sistemas complexos. As metaheurísticas apresentam a vantagem de que são relativamente fáceis de implementar e, geralmente, apresentam excelente desempenho para todos os tipos de sistemas elétricos. Apresentam a grande desvantagem de que geralmente requerem tempos de processamento elevados para encontrar soluções de excelente qualidade. No entanto, deve-se observar que o tempo de processamento não é um problema crucial em planejamento de sistemas de transmissão. Nos próximos anos deve continuar muito ativa a pesquisa em metaheurísticas aplicadas ao problema de planejamento de sistemas de transmissão.

Praticamente todas as propostas de metaheurísticas apresentadas na literatura especializadas foram aplicadas ao planejamento estático. Em (ESCOBAR; GALLEGO; ROMERO, 2004) foi apresentada a primeira metaheurística aplicada ao planejamento multiestágio de sistemas de transmissão.

Mais tarde, outras metaheurísticas também foram propostas. GRASP (*Greedy Randomized Adaptive Search Procedure*) foi utilizado em (BINATO; PEREIRA; GRANVILLE, 2001). As melhores soluções já conhecidas para dois sistemas testes brasileiros reais utilizados no estudo foram obtidas pela metaheurística, assim como melhoramentos na solução do caso do planejamento da expansão do sistema sudeste brasileiro, mostrando o potencial do método.

Algoritmos de Busca Tabu (*Tabu Search*) foram utilizados nos trabalhos (GALLEGO; MONTICELLI; ROMERO, 1998a; ORTIZ, 1997). Dois Algoritmos Genéticos foram propostos para resolver problemas de planejamento em (*Extended Genetic Algorithms*) (GALLEGO; MONTICELLI; ROMERO, 1998b) e (*Hybrid Genetic Algorithms*) (BINATO; ROMERO, 1998).

Em (SILVA et al., 2001) utilizaram Busca Tabu para resolver o problema de planejamento estático de redes de transmissão. Foram utilizados vários conceitos da Busca Tabu como memória de curto-prazo, lista tabu e critério de aspiração. Uma fase de intensificação, que explora regiões do espaço de busca onde boas soluções devem existir, foi implementada juntamente com uma fase de diversificação, que direciona a busca para regiões não exploradas, utilizando conceitos de memória de médio e longo prazo.

Uma metodologia híbrida combinando GRASP com *Path Relinking* foi desenvolvida em (FARIA et al., 2002). *Path Relinking* é um método que surgiu como uma estratégia de intensificação para melhorar a qualidade da solução de outras metaheurísticas. Nos poucos trabalhos em que foi utilizado, obteve grande sucesso. Neste estudo foi aprimorada a qualidade da busca por novas soluções, ajudando a obter a solução ótima dos problemas propostos com um número menor de iterações.

Praticamente todas as pesquisas apresentadas em planejamento da expansão de sistemas de transmissão foram realizadas apenas utilizando o modelo de planejamento estático. Existe pouca

bibliografia de modelagem e otimização de modelos matemáticos de planejamento multiestágio. Assim, por exemplo em (HAFFNER, 2000) é apresentada uma discussão interessante de modelagem deste problema e em (ROMERO et al., 2003) é apresentado um algoritmo heurístico construtivo para o planejamento multiestágio utilizando o modelo de transportes.

Todos os algoritmos heurísticos encontram apenas soluções de boa qualidade para sistemas de médio e grande porte, e a qualidade dessas soluções pode ficar muito distante das soluções ótimas ou subótimas como acontece, por exemplo, com o sistema norte-nordeste brasileiro. A vantagem dos algoritmos heurísticos é que são simples de entender, robustos e muito rápidos. No momento, os algoritmos heurísticos ainda representam um campo de pesquisa muito interessante e as soluções encontradas por esses algoritmos podem ser utilizadas como base para encontrar soluções melhores utilizando algoritmos mais demorados como as metaheurísticas.

2.3 Sistemas Considerados para Testes

A literatura especializada apresenta alguns sistemas utilizados com grande frequência por pesquisadores em planejamento da expansão de sistemas de transmissão. Esses sistemas apresentam tamanho e complexidade variada. Esses sistemas são os seguintes: (1) sistema de Garver de 6 barras e 15 circuitos, (2) sistema IEEE 24 barras e 41 circuitos, (3) sistema sul brasileiro de 46 barras e 79 circuitos, (4) sistema colombiano de 93 barras e 155 circuitos e (5) sistema norte-nordeste reduzido de 87 barras e 183 circuitos.

O sistema de Garver foi muito utilizado pelos pesquisadores em planejamento. Representa um sistema para fins acadêmicos, mas a alternativa como reprogramação da geração é difícil de resolver. Este sistema permite trabalhar com e sem reprogramação da geração. Os outros sistemas, exceto o sistema norte-nordeste brasileiro apresentam complexidade média, isto é, poucos circuitos devem ser adicionados para encontrar soluções ótimas ou sub ótimas. No momento, muitos algoritmos trabalham adequadamente com esses sistemas.

Os dados disponíveis do sistema sul brasileiro permitem trabalhar com (CR) e sem reprogramação (SR) da geração. Por outro lado, os dados do sistema colombiano permitem trabalhar também com o planejamento multiestágio, considerando três estágios.

O sistema norte-nordeste brasileiro representa um verdadeiro sistema *benchmark* em planejamento de sistemas de transmissão. Para alguns casos do planejamento deve-se adicionar mais de 100 circuitos para encontrar soluções de qualidade, assim esse sistema é altamente complexo e apresenta um número elevado de soluções de qualidade.

2.4 Outros Aspectos Relevantes

Existem outros tópicos de pesquisa com o problema de planejamento da expansão de sistemas de transmissão em que é possível realizar trabalhos de pesquisa relevante. Analisar-se-á brevemente esses tópicos.

A utilização de FACTS no problema de planejamento de sistemas de transmissão é um tópico relevante de pesquisa. Nessa proposta, além de adicionar ao sistema apenas linhas de transmissão e transformadores convencionais, podem ser adicionados dispositivos FACTS. (MIASAKI, 2006), em sua tese de doutorado, desenvolveu um algoritmo genético utilizando controladores FACTS na resolução do PPEST.

O planejamento de sistemas de transmissão considerando contingências é um tópico de pesquisa que foi amplamente sugerido, desde o aparecimento das primeiras técnicas de otimização no problema de planejamento. Nesse tipo de problema, deve-se expandir, de forma ótima, o sistema com a exigência de que quando é simulada a retirada de um circuito, o sistema restante ainda opere adequadamente, isto é, sem corte de carga. Aparentemente, o novo problema parece ser uma simples extensão do planejamento convencional, mas na verdade é muito mais complexo. Em (SILVA et al., 2006) é apresentada uma metaheurística para o problema considerando contingência $(n - 1)$.

O planejamento da expansão de sistemas de transmissão em um ambiente de mercado elétrico aberto é outro tópico de pesquisa que ainda não atingiu um nível relevante. Nesse novo tipo de mercado de energia elétrica existem donos diferentes na geração, na transmissão e na distribuição. Existem também novas restrições como os contratos em *pool*, os contratos bilaterais e a necessidade de que cada gerador tenha a possibilidade de gerar, a qualquer momento, a geração máxima do centro gerador. Nesse novo contexto, o fato marcante é que o sistema deve ser planejado para que seja possível operar com muitos esquemas de geração. O problema de planejamento em mercado aberto ainda não atingiu a maturidade, porque nos últimos anos foram realizados muitos discursos e poucas propostas consistentes. Devido a este fato, ainda não existe uma proposta de modelagem matemática consolidada para este problema. Entretanto, uma proposta inicial foi realizada em (FANG; HILL, 2003) e essa pesquisa deve ser usada como base para caracterizar de forma adequada a modelagem matemática desse tipo de problema para depois identificar a proposta de otimização mais indicada.

Em praticamente todas as técnicas de otimização utilizadas no problema de planejamento, o trabalho pesado em relação a esforço computacional de processamento está relacionado como as sub-rotinas que devem resolver problemas de programação linear (PL) ou problemas de programação não-linear (PNL). Assim, é crucial para reduzir tempo de processamento formular e implementar computacionalmente algoritmos eficientes de PL e PNL. Para problemas de PL uma alternativa é desenvolver técnicas especializadas do simplex como a apresentada em (SCOTT; MARINHO, 1979; HASHIMOTO; ROMERO; MANTOVANI, 2003) para um tipo de aplicação do modelo DC. Nestes traba-

lhos foi desenvolvido um algoritmo simplex especializado que usa um processo de transformação eficiente e o algoritmo dual simplex canalizado com relaxação. Outra alternativa para resolver problemas de PL e PNL é desenvolver algoritmos eficientes e especializados utilizando o método de pontos interiores.

Em (RIDER; GARCIA; ROMERO, 2007) pode ser encontrada uma proposta inicial para otimizar a expansão de sistemas de transmissão utilizando o modelo AC e um algoritmo heurístico construtivo como proposta de otimização.

Em (LATORRE et al., 2003) pode-se encontrar uma bibliografia completa sobre o problema de planejamento da expansão de sistemas de transmissão e em (ROMERO et al., 2002) uma análise detalhada de modelagem matemática. Finalmente, o desafio mais importante consiste em usar o modelo AC. Essa proposta tem a grande vantagem que permite integrar o planejamento de potência ativa e reativa, encontrando propostas de planejamento compatíveis com a operação.

3 *Técnicas de Solução*

O PPEST pode ser representado por, pelo menos, três modelos matemáticos como observa-se no capítulo anterior. Quando utiliza-se o modelo de transportes ou o modelo híbrido linear, obtém-se um problema de programação linear inteiro misto, e quando utiliza-se os modelos híbrido não-lineares ou DC, obtém-se um problema de programação não-linear inteiro misto, que já foi mencionado serem problemas de difícil solução, em especial quando se trata de sistemas de grande porte. As técnicas utilizadas, para resolver este problema, podem ser classificadas em duas categorias (Figura 3.1): métodos exatos (analíticos e de otimização clássica) como *Branch and Bound* e Decomposição de Benders; e os métodos aproximados (heurísticas e metaheurísticas). Neste capítulo serão abordadas algumas destas técnicas.

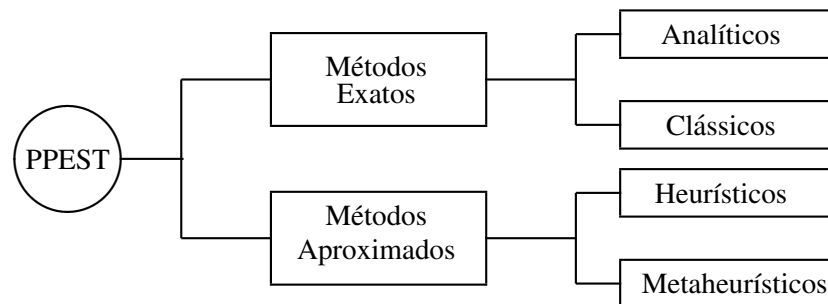


Figura 3.1: Métodos de Solução.

3.1 Métodos Exatos

Os métodos de otimização clássica, geralmente usando técnicas de decomposição matemática, apresentam a característica de encontrar 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 um esforço computacional elevado e problemas de convergência. A seguir descreve-se ligeiramente os métodos *Branch and Bound* e Decomposição de Benders.

3.1.1 *Branch and Bound*

O algoritmo *Branch and Bound* é um algoritmo enumerativo, cuja estrutura de resolução baseia-se na construção de uma árvore onde os nós representam os problemas candidatos e os caminhos representam as novas restrições que devem ser consideradas. Por intermédio dessa árvore, todas as soluções inteiras da região viável do problema são enumeradas de modo implícito ou explícito o que garante que todas as soluções ótimas sejam encontradas. A estrutura geral do algoritmo *Branch and Bound* apresenta três elementos fundamentais, que serão detalhados a seguir: separação, relaxação e sondagem.

Na etapa de separação, o problema original (P) é separado em q subproblemas $(P^1), (P^2), \dots, (P^q)$ sujeitos às seguintes condições:

- (S1) Toda solução viável de (P) é uma solução de somente um dos subproblemas (P^i) , $i = 1, 2, \dots, q$.
- (S2) Uma solução viável de qualquer um dos subproblemas (P^i) , $i = 1, 2, \dots, q$ é também, uma solução viável de (P) .

Estas condições asseguram que o conjunto das soluções viáveis de cada um dos subproblemas (P^i) , $i = 1, 2, \dots, q$, é uma partição do conjunto das soluções viáveis de (P) . Os subproblemas (P^i) , $i = 1, 2, \dots, q$, são denominados descendentes de (P) e podem, sucessivamente, gerar seus próprios descendentes.

O interesse na separação (*branching*) é utilizar a estratégia de 'dividir para conquistar' para resolver o problema (P) . Pode-se descrever, sucintamente, esta estratégia do seguinte modo: enquanto a solução de (P) não é possível, separa-se (P) em dois ou mais subproblemas descendentes, gerando uma lista de problemas candidatos (PC). Então, seleciona-se um dos candidatos dessa lista e tenta-se resolvê-lo. Se a solução não é possível, o problema é novamente separado e seus descendentes são adicionados à lista dos candidatos; caso contrário, o problema é resolvido e uma nova solução é obtida. O valor da função objetivo dessa nova solução é, então, comparado com o valor da solução incumbente, que é a melhor solução viável conhecida até o momento. Caso a nova solução seja melhor do que a solução incumbente, ela se torna a nova incumbente. A seguir, retorna-se à lista e seleciona-se o próximo candidato. Isto é repetido até que a lista esteja vazia, quando pode-se afirmar que a solução do problema é dada pela solução incumbente final.

A forma usual de separação (*branching*) de um problema de programação inteira é através de restrições contraditórias em uma única variável inteira (variável de separação ou de ramificação). Assim, a partir do problema original, denominado nó zero, originam-se dois novos subproblemas (cada um dos subproblemas descendentes é mais fácil de resolver que o subproblema candidato

de origem uma vez que foi acrescentada uma restrição na variável de separação), representados pelos nós 1 e 2 que são sucessivamente divididos formando uma árvore. A cada nó se associa um subproblema candidato e cada ramo indica o acréscimo de uma restrição relacionada à variável empregada na separação. Portanto, à medida que se desce na árvore a região viável dos descendentes gerados vai ficando cada vez mais restrita.

A relaxação consiste em, temporariamente, ignorar algumas restrições do problema (P) visando torná-lo mais fácil de resolver. A condição que deve ser satisfeita é que o conjunto de soluções viáveis do problema original (P) esteja contido no conjunto de soluções viáveis do problema relaxado (PR). Isto implica que:

(R1) Se (PR) não tem solução viável, então o mesmo é verdadeiro para (P).

(R2) O valor mínimo de (P) não é menor que o valor mínimo de (PR).

(R3) Se uma solução ótima de (PR) é viável em (P), então ela é uma solução ótima de (P).

Dentre as formas possíveis de relaxação, destaca-se a eliminação das restrições de integridade das variáveis, o que transforma o problema misto em um problema linear (PL) padrão.

Na análise dos problemas candidatos existe a necessidade de determinar quais são promissores e, portanto, devem ser examinados, e quais podem ser descartados. Isto é realizado na etapa de sondagem onde o problema candidato (PC) é eliminado (descartado para análises futuras), juntamente com todos os seus descendentes, se satisfizer a pelo menos um dos seguintes critérios:

(S1) O problema candidato relaxado (PCR) não tem solução viável. Devido a (R1), isto significa que o problema candidato (PC) também não tem solução viável.

(S2) A solução ótima do problema candidato relaxado (PCR) é pior (*bounding*) do que a melhor solução atualmente conhecida para (P) (solução incumbente). Observar que a solução ótima do problema candidato relaxado é sempre melhor ou igual à solução do problema candidato e de seus descendentes.

(S3) Uma solução ótima do problema relaxado (PCR) é viável, também, em (PC). Neste caso, devido a (R3), ela é ótima em (PC) e, devido a (S2) ela é também factível em (P). Caso seja melhor que a incumbente, a solução deste problema candidato passa a ser a nova solução incumbente.

O algoritmo básico, embora seja conceitualmente simples, apresenta complexidade na implementação computacional. Este algoritmo apresenta excelente desempenho para sistema pequenos, mas em sistemas reais apresenta várias limitações relacionadas com o esforço computacional.

3.1.2 Decomposição de Benders

A Decomposição de Benders é a técnica de decomposição matemática mais adequada para ser utilizada no PPEST. Esta técnica permite decompor o problema 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, o qual é um problema de programação linear que pode ser facilmente resolvido utilizando algoritmos eficientes de PL. Por outro lado, as variáveis de investimento fazem parte do subproblema de investimento, o qual é um problema de programação linear inteira mista com uma única variável contínua que aparece como parte da decomposição. Assim, a decomposição de Benders resolve o PPEST através de uma solução iterativa dos subproblemas de operação e investimento.

A otimização global é atingida através de uma resolução iterativa das soluções separadas dos subproblemas de operação e investimento. Esta técnica não foi capaz de resolver sistemas de grande porte, como o sistema Norte-Nordeste Brasileiro.

3.2 Algoritmos Aproximados

Numa tentativa de contornar as diversas dificuldades enfrentadas para resolver o problema de planejamento utilizando ferramentas de otimização matemática, devido principalmente a sua natureza não-linear e não-convexa, surgiram os algoritmos aproximados. Sendo uma classe bastante ampla, são exemplos de algoritmos aproximados: algoritmos heurísticos construtivos e algoritmos metaheurísticos.

Um algoritmo heurístico construtivo (AHC) é um procedimento iterativo de escolhas ou decisões tomadas passo a passo que, de maneira sistemática, determina uma boa proposta de solução para um problema complexo. No caso do planejamento da expansão dos sistemas de transmissão de energia elétrica, a partir de uma configuração base, em cada passo é adicionado ao sistema um ou vários circuitos até que o conjunto de adições realizadas permita uma operação adequada do sistema elétrico. Portanto, em cada passo, a configuração do sistema é modificada pela adição de um ou vários circuitos, e esta configuração obtida passa a ser denominada configuração corrente. O circuito escolhido em cada passo para ser adicionado à chamada configuração corrente é um circuito que corresponde ao caminho mais atrativo identificado por um critério de sensibilidade, indicador de sensibilidade ou índice de desempenho. Assim a diferença fundamental entre os algoritmos heurísticos construtivos está no indicador de sensibilidade escolhido e, obviamente, no modelo matemático escolhido (transporte, híbrido ou DC).

3.2.1 Algoritmo Heurístico Construtivo de Garver

O algoritmo heurístico de Garver é um AHC que utiliza o modelo de transportes (neste modelo, apenas as restrições da primeira lei de Kirchhoff devem ser obrigatoriamente satisfeitas), e foi o primeiro algoritmo de grande difusão utilizado no planejamento de sistemas de transmissão (GARVER, 1970). Nesta proposta, Garver formulou o problema como um problema de fluxo de potência e utilizou algoritmos de programação linear para identificar as rotas mais diretas entre os geradores e as cargas.

Utilizando o modelo (2.10)-(2.17), e relaxando a integralidade das variáveis inteiras n_{ij} , Garver encontrou a solução ótima contínua, ou seja, variáveis assumindo valores fracionários, para a configuração corrente n_{ij}^0 , resolvendo apenas um problema de PL. Tendo conhecidas as incógnitas n_{ij} , pôde encontrar os fluxos de potência em todos os circuitos antigos (n_{ij}^0) 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 esta proposta. Um processo repetitivo desta 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 mais necessário realizar adições de circuitos ao sistema. O conjunto de adições realizadas representa a proposta de solução do algoritmo de Garver. Em cada passo o algoritmo de Garver resolve o PL abaixo, em que o vetor n_{ij}^1 armazena os circuitos adicionados em cada passo do processo de otimização.

$$\min v = \sum_{(i,j) \in \Omega} c_{ij} n_{ij} \quad (3.1)$$

s.a.

$$Sf + g = d \quad (3.2)$$

$$|f_{ij}| \leq (n_{ij}^0 + n_{ij}^1 + n_{ij}) \bar{f}_{ij} \quad (3.3)$$

$$0 \leq g \leq \bar{g} \quad (3.4)$$

$$0 \leq n_{ij} \leq (\bar{n}_{ij} - \bar{n}_{ij}^1) \quad (3.5)$$

$$f_{ij} \text{ irrestrito} \quad (3.6)$$

$$\forall (i, j) \in \Omega \quad (3.7)$$

O algoritmo de Garver pode ser resumido nos seguintes passos:

Passo 1: Assumir a configuração base n_{ij}^0 e $n_{ij}^1 = 0$ como configuração corrente.

Passo 2: Resolver o PL correspondente ao sistema (3.1)-(3.7) para a configuração corrente. Se todos os $n_{ij} = 0$ então pare, pois foi encontrada uma boa configuração factível. Caso contrário, ir ao passo 3.

Passo 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 novo caminho ij com o maior valor de f_{ij}^v e atualizar a configuração corrente n_{ij}^1 , adicionando um circuito naquele caminho ij . Voltar ao passo 2.

A grande vantagem do modelo de transportes é que praticamente não existe diferença entre resolver problemas de sistemas conexos e altamente ilhados. A desvantagem está no fato de que a solução ótima encontrada por este modelo algumas vezes pode ficar longe da solução ótima do modelo DC, pois a solução do modelo de transportes não satisfaz, necessariamente, a segunda lei de Kirchhoff.

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 solução ótima global.

3.2.2 AHC de Villasana-Garver-Salon

O algoritmo proposto em (VILLASANA; GARVER; SALON, 1985) é um AHC que utiliza o modelo híbrido linear com a estratégia de resolver apenas um PL em cada passo do processo de otimização. A estratégia consiste em resolver um PL de tal forma que seja possível verificar se os circuitos já adicionados permitem que o sistema opere adequadamente para o modelo DC ou, caso contrário, identificar o circuito mais promissor que deve ser adicionado ao sistema. Esses objetivos são atingidos resolvendo um PL que é um caso especial do modelo híbrido linear após relaxar a integralidade das variáveis de investimento n_{ij} . Assim, em cada passo, o algoritmo VGS resolve o PL (3.8)-(3.17).

$$\min v = \sum_{(i,j) \in \Omega} c_{ij} n_{ij} \quad (3.8)$$

s.a.

$$Sf + S^0 f^0 + S^1 f^1 + g = d \quad (3.9)$$

$$f_{ij}^0 - \gamma_{ij} n_{ij}^0 (\theta_i - \theta_j) = 0, \forall (i, j) \in \Omega_0 \quad (3.10)$$

$$f_{ij}^1 - \gamma_{ij} n_{ij}^1 (\theta_i - \theta_j) = 0, \forall (i, j) \in \Omega_1 \quad (3.11)$$

$$|f_{ij}^0| \leq n_{ij}^0 \bar{f}_{ij}, \forall (i, j) \in \Omega_0 \quad (3.12)$$

$$|f_{ij}^1| \leq n_{ij}^1 \bar{f}_{ij}, \forall (i, j) \in \Omega_1 \quad (3.13)$$

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

$$0 \leq g \leq \bar{g} \quad (3.14)$$

$$0 \leq n_{ij} \leq (\bar{n}_{ij} - n_{ij}^1) \quad (3.15)$$

$$f_{ij}^0, f_{ij}^1 \text{ e } f_{ij} \text{ irrestritos} \quad (3.16)$$

$$\theta_j \text{ irrestrito} \quad (3.17)$$

Em que:

S : Matriz de incidência nó-ramo transposta para os circuitos candidatos à adição.

S^0 : Matriz de incidência nó-ramo transposta para os circuitos da topologia base.

S^1 : Matriz de incidência nó-ramo transposta para os circuitos adicionados e dos nós conectados a esses caminhos no processo iterativo do algoritmo VGS.

f : Vetor de fluxos totais nos circuitos novos que devem ser adicionados na resolução do PL com elementos f_{ij} .

f^0 : Vetor de fluxos totais através dos circuitos da topologia base, com elementos f_{ij}^0 .

f^1 : Vetor de fluxos totais através dos caminhos adicionados no processo iterativo e com elementos f_{ij}^1 (fluxo nos circuitos já adicionados no caminho ij).

Ω : Conjunto de índices dos circuitos candidatos.

Ω_0 : Conjunto de índices dos circuitos presentes na topologia base.

Ω_1 : Conjunto de índices dos circuitos adicionados no processo de otimização pelo VGS.

Deve-se observar que no PL mostrado em (3.8)-(3.17) os fluxos em cada caminho estão separados em três grupos (fluxos em circuitos existentes na topologia base: f_{ij}^0 , fluxos nos circuitos já adicionados no processo iterativo do algoritmo VGS: f_{ij}^1 e os fluxos nos circuitos adicionados na resolução do PL, f_{ij}). O mesmo acontece com os circuitos n_{ij}^0 , n_{ij}^1 e n_{ij} representando, respectivamente, o número de circuitos presentes no caminho ij na configuração base, circuitos já adicionados no processo iterativo pelo algoritmo VGS e os circuitos adicionados na resolução do PL. E ainda, Ω_0 representa o conjunto dos circuitos presentes na configuração base e Ω_1 representa o conjunto dos circuitos já adicionados pelo VGS.

Se na solução do PL (3.8)-(3.17) tivermos $v = 0$ e, portanto, $n_{ij} = 0$, $\forall (i, j) \in \Omega$ então, o sistema opera adequadamente apenas com os circuitos da topologia base e os já adicionados no processo iterativo. Como esses circuitos obedecem as duas leis de Kirchhoff, então, a proposta de solução, isto é, os circuitos que foram adicionados, é uma proposta factível para o modelo DC. Caso contrário, a solução encontrada nos permite identificar o circuito mais atrativo que deve ser adicionado ao sistema elétrico no próximo passo.

O AHC proposto em (VILLASANA; GARVER; SALON, 1985) utiliza um indicador de sensibilidade definido a partir da solução ótima do modelo híbrido linear (MHL) correspondente à topologia corrente (circuitos da topologia base e os adicionados no processo iterativo) e relaxando a integralidade das variáveis n_{ij} , isto é, a resolução do modelo (3.8)-(3.17).

Considerando que relaxando a integralidade das variáveis n_{ij} , o problema (3.8)-(3.17) se torna um problema de programação linear (PL), o índice de sensibilidade é definido como sendo o fluxo

de potência pelos circuitos com $n_{ij} \neq 0$ na solução do PL. Em cada passo do AHC, o circuito selecionado para adição é aquele identificado pelo seguinte índice de sensibilidade:

$$IS = \max\{IS_{ij} = n_{ij}\bar{f}_{ij} : n_{ij} \neq 0\}, \quad (3.18)$$

em que n_{ij} é a solução do PL depois de relaxar a condição de integralidade no AHC. Em cada passo a solução corrente é, então, atualizada, e assim a topologia corrente é formada pela topologia base juntamente com os circuitos adicionados durante o processo iterativo.

A característica mais interessante apresentada no algoritmo VGS é que cada circuito adicionado durante o processo deve satisfazer as duas leis de Kirchhoff, simultaneamente, o que significa que a solução determinada pelo VGS é uma solução factível para o modelo DC.

O algoritmo VGS pode ser resumido nos passos a seguir.

Passo 1: Assumir a topologia base como solução corrente e usar o modelo híbrido linear modificado (3.8)-(3.17). Todos os circuitos presentes na solução corrente devem satisfazer as leis de Kirchhoff da corrente e da tensão.

Passo 2: Resolver o PL (3.8)-(3.17) para a topologia corrente. Se a solução indicar que o sistema está operando adequadamente com a nova adição e, portanto, $v = 0$, pare. Uma nova solução para o modelo DC foi encontrada.

Passo 3: Utilizar o índice de sensibilidade (3.18) para identificar o circuito mais atrativo que deve ser adicionado ao sistema. Atualizar a topologia com o circuito adicionado, acrescentando ij em Ω_1 e ir ao passo 2.

A vantagem de se utilizar esta estratégia é que em cada passo se resolve apenas um PL mas no final do processo encontra-se uma solução para o modelo DC. Outra possibilidade é resolver o próprio modelo DC após relaxar a integralidade das variáveis n_{ij} mas, nesse caso, deve-se resolver um problema de programação não-linear em cada passo do AHC.

Do ponto de vista de otimização matemática o VGS 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 solução ótima global.

Um significado importante no algoritmo VGS sobre o índice de sensibilidade é que a solução ótima do PL apresenta um conjunto de $n_{ij} \neq 0$ que identificam o melhor investimento proposto satisfazendo somente a primeira lei de Kirchhoff. Quando o circuito é incorporado no sistema, ele passa a cumprir ambas as leis de Kirchhoff. Assim, o índice de sensibilidade utilizado pode não apresentar o desempenho desejado como o índice utilizado no modelo de transportes. Para outros índices, ver (ROMERO et al., 2003).

3.2.3 Algoritmos Metaheurísticos

As metaheurísticas (O termo metaheurística foi cunhado por Fred Glover em 1986 e foi amplamente aplicado na literatura) são métodos de busca que combinam métodos heurísticos. Estes métodos têm-se mostrado muito efetivos na solução de problemas difíceis de grande porte, inclusive no problema de planejamento da expansão de sistemas de transmissão. Exemplos de Metaheurística são os métodos de Recozimento Simulado, Algoritmos Genéticos, Grasp, Busca Tabu, entre outros. A seguir será descrita a lógica de algumas destas técnicas metaheurísticas utilizadas até agora no PPEST.

3.2.3.1 Recozimento Simulado

Um problema de otimização combinatória pode ser resolvido usando Recozimento Simulado (*Simulated Annealing*, SA), sendo formulado por um par (G, v) em que G representa um conjunto finito, embora muito grande, de configurações e v é o custo associado a cada configuração. Assim, resolver um problema de natureza combinatória deste tipo consiste em encontrar a configuração de custo mínimo dentre todas as configurações possíveis. Assim, SA parte de uma configuração inicial e depois gera um conjunto de configurações candidatas controladas pelo programa de esfriamento. 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 de aceitar configurações com custo mais elevado, permite ao SA sair de eventuais ótimos locais. Este processo é controlado pelo programa de esfriamento.

Para explicar esta técnica faz-se uma breve descrição do algoritmo.

Dada uma substância específica, as moléculas não se encontram com a mesma energia e pode-se afirmar que as partículas se encontram em distintos níveis de energia. O menor nível de energia acontece na temperatura do zero absoluto e esse estado é conhecido como “Estado fundamental de energia mínima”. Cada nível de energia é conhecido como microestado e cada microestado tem um nível de ocupação definido por n_i que é o número de partículas que existem no nível i . Assim, n_i ficaria determinado da seguinte forma:

$$n_i = P \cdot e^{\frac{-E_i}{kT}},$$

sendo que P é um fator de proporcionalidade, k é a constante de “Boltzman”, T é a temperatura da substância e E_i é a energia do micro-estado i .

Claramente observa-se que se E_i aumenta, então, n_i diminui. Se T aumenta, então, n_i aumenta.

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

- Uma adequada representação do problema;
- Um mecanismo de transição;
- Um programa de esfriamento (*cooling schedule*).

Para cada tipo de problema, deve ser definido 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.

3.2.3.2 Algoritmos Genéticos

O algoritmo genético usa um conjunto de configurações (população de indivíduos) para resolver um problema complexo, simulando o que ocorre na genética natural.

Um algoritmo genético simples deve seguir os seguintes passos:

1. Gerar uma população inicial após escolher a forma de codificação;
2. Calcular a função objetivo de cada configuração e armazenar a solução incumbente;
3. Realizar a seleção;
4. Implementar a recombinação (*crossover*);
5. Implementar a mutação e obter a nova geração;
6. Estabelecer um critério de parada, se o critério de parada for satisfeito, pare. Caso contrário, voltar ao passo 2.

O ciclo entre os passos 2 e 5 é chamado “ciclo geracional”.

O mecanismo de codificação depende da natureza das variáveis do problema. Existem problemas com variáveis binárias que são mais simples de implementar. Outros problemas podem apresentar variáveis inteiras ou contínuas. Os primeiros algoritmos genéticos utilizavam a codificação binária, assim, para esses algoritmos, as variáveis que originalmente eram inteiras ou contínuas eram transformadas em binárias.

Deve ser criada uma estratégia para calcular a função objetivo para cada configuração. Às vezes, deve ser utilizada uma estratégia para formular um equivalente da função objetivo que pode ser mais adequado para o algoritmo genético. Às vezes é necessário padronizar a função objetivo.

Por último, este parâmetro que identifica a qualidade de uma configuração (que pode ser a própria função objetivo ou seu equivalente) é chamada de “*fitness function*” (função de adaptação).

Para a seleção, tipicamente é usada a seleção baseada em torneio em que em cada jogo é escolhida a configuração ou topologia ganhadora. Tipicamente em cada jogo participam um número reduzido de configurações, isto é, de duas a quatro configurações. A seleção baseada em torneio tem a vantagem de tratar da mesma forma problemas de maximização e minimização.

Uma vez selecionadas as configurações, deve-se escolher duas propostas de solução e realizar a recombinação. Na recombinação, um par de configurações misturam seu conteúdo genético e formam dois descendentes. O ponto de recombinação é escolhido aleatoriamente. Para a recombinação de um simples ponto (*single point*) é feita a escolha de um número aleatório entre 1 e $(k - 1)$, em uma configuração com k elementos. Uma vez escolhido o ponto de recombinação, as duas configurações candidatas são geradas trocando-se as parcelas à direita do ponto de recombinação das duas configurações. Nem todas as configurações são recombinadas. Uma parcela pequena pode passar para a próxima etapa sem recombinação. Assim é definido a taxa de recombinação ρ_c .

Assim, para que as configurações selecionadas sejam submetidas à recombinação, deve-se gerar um número aleatório $\rho \in [0,1]$. Se ρ for menor que a taxa de recombinação ρ_c , então, deve-se proceder à recombinação; caso contrário as duas configurações selecionadas não são recombinadas.

Neste processo de recombinação, novamente, estão presentes três decisões de caráter aleatório: (1) As duas configurações que devem ser submetidas a recombinação são escolhidas aleatoriamente; (2) O ponto de recombinação é escolhido aleatoriamente e; (3) Deve-se gerar um número aleatório ρ que determina se as configurações selecionadas devem ser submetidas à recombinação. Assim, como no caso da seleção, algumas das decisões anteriores podem ser submetidas por decisões de caráter determinístico como, por exemplo, recombinar as melhores configurações com direito a gerar descendentes.

Na codificação binária, a mutação consiste em transformar o valor de uma posição do arranjo de 0 para 1 ou vice-versa. Na maioria das pesquisas, a mutação é considerada um operador secundário. Uma taxa de mutação é definida como sendo o seguinte: para cada *bit* de cada configuração, gera-se um número aleatório $\rho \in [0, 1]$ e se $\rho < \rho_m$ existirá mutação da posição. Se $\rho_m = 0,05$, significa dizer que existe uma probabilidade de 5% de se realizar uma mutação.

O ciclo geracional é o processo de seleção-recombinação-mutação, que permite encontrar as configurações da nova geração a partir da população corrente. O ciclo geracional é controlado pelo chamado Programa de Controle do Algoritmo Genético. Este programa é composto de parâmetros que definem o tamanho da população, a taxa de recombinação e a taxa de mutação, definindo a qualidade do algoritmo.

Existem vários critérios de parada, entre eles pode-se citar: parar após a execução de um

número determinado de gerações; parar quando a solução incumbente atingir um valor especificado; parar se as configurações da população ficarem muito homogêneas.

A população inicial do algoritmo pode ser gerada aleatoriamente, aleatoriamente controlada ou através de um algoritmo heurístico.

Em (BINATO; ROMERO, 1998; GALLEG0; MONTICELLI; ROMERO, 1998b; SILVA; GIL; AREIZA, 2000) foram desenvolvidos algoritmos genéticos especializados, levando em conta as características específicas do PPEST, apresentando soluções ótimas para sistemas de pequeno e médio porte, e soluções quase ótimas para sistemas de grande porte.

3.2.3.3 Busca Tabu

Busca Tabu (*Tabu Search*, TS) é também um algoritmo adequado para resolver problemas de natureza combinatória. Basicamente, é um procedimento metaheurístico usado para gerenciar um algoritmo heurístico de busca local, evitando que o problema pare em um ótimo local.

TS foi proposto por Fred Glover na década de 80, com conceitos baseados em inteligência artificial. Segundo Glover, TS surgiu da premissa de que a resolução de um problema pode ser considerado inteligente se esse processo incorpora memória adaptativa e exploração sensível.

O algoritmo de busca tabu apresenta as seguintes partes:

- i. Um mecanismo de geração de movimentos, isto é, dada a configuração corrente, deve-se definir e caracterizar a *vizinhança da configuração corrente* e, portanto, de acordo com a lógica TS, deve-se passar para a melhor configuração vizinha (ou para o menos pior).
- ii. A formação da *Lista Tabu* que armazena atributos das configurações vizinhas que estão proibidas. Este mecanismo evita retornar a configurações já visitadas (ciclagem). A escolha do tamanho da lista tabu é crítica no desenvolvimento do algoritmo TS. Normalmente a lista tabu é montada usando informação de memória de curto prazo.
- iii. A definição dos *critérios de aspiração* que tenta contornar as proibições impostas pela lista tabu. Neste contexto, o critério de aspiração anula a proibição de uma configuração vizinha se for verificada que essa configuração candidata é altamente atrativa de acordo com os critérios de aspiração definidos.
- iv. Implementação de *estratégias de intensificação e diversificação*. Estas estratégias são basicamente implementadas usando as memórias baseadas em frequência. A intensificação tenta explorar, mais intensamente, regiões onde foram encontradas excelentes configurações, e a diversificação tenta levar o processo de otimização a regiões distantes ou pouco exploradas na tentativa de encontrar novas configurações de excelente qualidade.

Um algoritmo avançado de busca tabu deve levar em conta os seguintes aspectos:

- a. Montar uma estratégia adequada para gerar uma configuração inicial de boa qualidade ou, ainda melhor, um conjunto de configurações de boa qualidade e de composição diversificada. Assim, existiriam disponíveis configurações atrativas correspondentes a regiões distantes o que facilita a implementação eficiente das estratégias de intensificação e diversificação. Esta parte pode ser implementada usando um algoritmo heurístico muito rápido em termos de esforço computacional.
- b. Armazenamento das chamadas configurações de elite que podem ser posteriormente utilizadas para implementar a intensificação e a diversificação, e também na implementação do chamado *Path Relinking* que é usado para gerar novas configurações de qualidade, misturando adequadamente atributos das configurações de elite.
- c. Criar uma estratégia destinada a reduzir o número de vizinhos candidatos. Em TS, dada uma configuração, existe um número de configurações vizinhas que devem ser analisadas e TS passa para a melhor configuração vizinha. Se o número de vizinhos é elevado, TS pode precisar de um elevado esforço computacional para realizar um simples movimento. Existem várias propostas para reduzir o número de vizinhos que devem ser analisados no TS. As principais propostas são chamadas critério de aspiração plus e a formação de uma lista de vizinhos considerados de elite. Para problemas específicos, pode-se ainda montar uma lista reduzida, aproveitando as características específicas do problema.

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

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

Memória de curto prazo - Um algoritmo TS pode ser implementado simplesmente usando a estratégia de curto prazo (muitas das pesquisas iniciais foram implementadas desta forma). A memória de curto prazo é basicamente a informação de atributos de configurações que foram modificados no passado recente. Esta informação é conhecida como memória baseada em recência (*recency*) e 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.

Memória de longo prazo - 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 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- Estratégia de intensificação; e 3- Estratégia de diversificação.

Memória baseada em frequência consiste basicamente em armazenar informação do número de vezes em que o atributo foi escolhido para gerar ou participar nas configurações durante o processo TS. Existem dois tipos de memória baseadas em frequência: as frequências de transição que armazena 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 o atributo permanece nas novas configurações ou em todas as configurações geradas durante o processo TS.

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 acumulado no processo TS. $N''(x)$ é um subconjunto reduzido de $N(x)$ obtido usando algum critério de redução de configurações candidatas.

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 utilizadas as estruturas de memória consideradas em intensificação ou em memória de curto prazo, mas com critérios de duração modificados.

Em (ORTIZ, 1997; GALLEGO; MONTICELLI; ROMERO, 2000) é apresentado um algoritmo de busca tabu aplicado ao PPEST. Em (GALLEGO; MONTICELLI; ROMERO, 1998b) pode-se observar que boas configurações iniciais diminuem o esforço computacional do processo e, frequentemente, geram configurações de melhor qualidade.

4 *Busca em Vizinhança Variável*

Neste capítulo apresenta-se parte do objetivo deste trabalho, que consiste em aplicar novas metaheurísticas na solução do PPEST, ou seja, aplica-se a metaheurística Busca em Vizinhança Variável na solução do PPEST. Assim, inicialmente na seção 4.1, tem-se o desenvolvimento desta metaheurística que foi proposta em meados da década de 90, (MLADENOVIC, 1995; MLADENOVIC; HANSEN, 1997), e é baseada num princípio simples: explorar o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança durante o processo de busca. Na sequência, algumas extensões são consideradas, incluindo versões híbridas. As aplicações práticas mais importantes desta metaheurística são mencionadas na seção 4.2. Na seção 4.3 descreve-se detalhadamente a aplicação da VNS no PPEST e uma ilustração prática é apresentada na seção 4.4. A importância da utilização de técnicas redução de vizinhança é sintetizada na seção 4.5. Os resultados obtidos são apresentados na seção 4.6. Por fim, na seção 4.7 incluem-se algumas considerações em relação aos aspectos essenciais das metaheurísticas.

4.1 Fundamentos da Busca em Vizinhança Variável

Um algoritmo de busca local, em geral, segue os passos apresentados a seguir.

Passo 1: Gerar uma solução inicial s .

Passo 2: Aplicar uma busca local para encontrar a melhor solução vizinha s' .

Passo 3: Se s' é melhor que s então faça $s = s'$ e vá ao passo 2, se não, pare.

As metaheurísticas oferecem um modo de melhorar consideravelmente o desempenho de procedimentos heurísticos simples, como esses baseados em busca local, que quando consideram problemas complexos e de muitos ótimos locais acabam convergindo para soluções de baixa qualidade. Geralmente consiste de um conjunto de estratégias que guiam e modificam outras heurísticas que determinam soluções que normalmente requerem apenas uma otimalidade local.

As metaheurísticas, em geral, são estruturas para construir heurísticas para problemas de otimização combinatória global. Desde que foi proposta a metaheurística *Simulated Annealing* como um esquema geral para construir heurísticas que saem de mínimos locais, as metaheurísticas foram

assunto de pesquisa intensiva e logo depois foram propostas várias outras metaheurísticas. Discussões e aplicações das melhores metaheurísticas conhecidas podem ser encontradas em (GLOVER; KOCHENBERGER, 2003; REEVES, 1995).

Uma estrutura de vizinhança no espaço de soluções S é uma aplicação:

$$N : X \rightarrow S = \{N(x) : N(x) \subseteq X\}, \quad (4.1)$$

que a cada solução $x \in X$ associa um conjunto de soluções $N(x) \subseteq X$, denominado vizinhança de x .

Denote por N_k , ($k = 1, \dots, k_{max}$), um conjunto finito de estruturas de vizinhança pré-selecionadas e por $N_k(x)$ o conjunto de soluções na k -ésima vizinhança de x . As vizinhanças N_k podem ser caracterizadas por uma ou mais métricas (ou quase-métricas) introduzidas num espaço de solução X .

A Busca em Vizinhança Variável, VNS (do inglês *Variable Neighborhood Search*), é uma metaheurística que foi proposta em meados da década de 90, por (MLADENOVIC, 1995; MLADENOVIC; HANSEN, 1997). É baseada num princípio simples: explorar o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança durante o processo de busca (HANSEN; MLADENOVIC, 1999; HANSEN; MLADENOVIC, 2001d). Ou seja, a metaheurística de busca em vizinhança variável é uma extensão de um algoritmo de busca local que utiliza a estratégia de mudança de tamanho da vizinhança para sair de soluções ótimas locais. Seu desenvolvimento foi rápido, com inúmeras publicações (HANSEN; MLADENOVIC; PÉREZ, 2001; HANSEN; MLADENOVIC, 2001a; HANSEN; MLADENOVIC, 2001b; HANSEN; MLADENOVIC, 2002; HANSEN; MLADENOVIC, 2003a; HANSEN; MLADENOVIC, 2003b).

A VNS, ao contrário de muitas metaheurísticas baseadas em busca local, não permite degradação na função objetivo para realizar um movimento, mas explora, a partir de uma solução inicial, uma seqüência crescente de vizinhanças, e só realiza o movimento até uma nova solução se esta solução for melhor que a solução incumbente.

A metaheurística VNS explora sistematicamente as seguintes observações:

Fato 1: Um mínimo local com relação a uma estrutura de vizinhança não é necessariamente um mínimo local com relação a uma outra;

Fato 2: Um mínimo global é um mínimo local em relação a todas as estruturas possíveis de vizinhança;

Fato 3: Para muitos problemas, um mínimo local com respeito a uma ou a várias vizinhanças são relativamente próximos um do outro.

Os fatos 1 a 3 sugerem, então, o uso de várias estruturas de vizinhança nas buscas locais para abordar um problema de otimização, ou seja, a idéia é definir um conjunto de estruturas de vizinhanças que podem ser utilizadas de forma determinística, de forma aleatória ou determinística e aleatória. Essas formas de utilizar as estruturas de vizinhanças produzem algoritmos VNS de desempenhos diferentes.

O fato 3 é particularmente importante na formulação de um algoritmo VNS, o qual é de caráter empírico, implica que uma solução ótima local fornece informações importantes em relação ao ótimo global especialmente se a solução ótima local for de excelente qualidade. Existe também a observação empírica de que as soluções ótimas locais geralmente estão concentradas em regiões específicas do espaço de busca. Se as soluções ótimas locais estivessem uniformemente distribuídas no espaço de busca todas as heurísticas baseadas em uma única vizinhança se tornariam ineficientes. Portanto, se for encontrado um ótimo local da região em que se encontra o ótimo global então uma metaheurística tipo VNS tem grandes chances de encontrar esse ótimo global. Por outro lado, se o ótimo global se encontra em outra região então a única possibilidade de encontrar o ótimo global é implementar um processo de diversificação. Por esse motivo um equilíbrio entre intensificação e diversificação no processo de busca pode ser importante em uma metaheurística. Na próxima seção pode-se ver a evolução da VNS.

4.1.1 VNS de Descida: Algoritmo VND

A chamada VND (*Variable Neighborhood Descent*) é baseada no Fato 1, ou seja, um ótimo local para um primeiro tipo de vizinhança não é necessariamente um ótimo local para um outro tipo de vizinhança (o ótimo local n' na vizinhança $N_1(x)$ não necessariamente é igual ao ótimo local n'' na vizinhança $N_2(x)$).

O algoritmo VND apresentado em (HANSEN; MLADENOVIC, 2001d) assume a forma mostrada na Figura 4.1.

A solução final proporcionada por este algoritmo é um mínimo local em relação a todas as k_{max} estruturas de vizinhança e, portanto, a probabilidade de se alcançar um mínimo global é maior de que quando se usa somente uma estrutura.

Ao se implementar o algoritmo VND da Figura 4.1, alguns cuidados devem ser tomados, em particular, considerando as seguintes questões:

- (i) Que complexidade tem os diferentes movimentos?
- (ii) Qual a melhor ordem para considerá-los?

Inicialização: Selecione o conjunto de estruturas de vizinhanças N_k , para $k = 1, \dots, k_{max}$, que será utilizado durante o processo; encontre uma solução inicial x (ou aplique a regra a um x conhecido).

Repetir a seqüência até que nenhuma melhoria da solução seja obtida:

(1) Faça $k \leftarrow 1$;

(2) Repetir até que $k = k_{max}$:

(a)**Exploração:** Encontre o melhor vizinho x' de x ($x' \in N_k(x)$);

(b)**Mover ou não:** Se a solução assim obtida é melhor que x , faça $x \leftarrow x'$ e $k \leftarrow 1$; Caso contrário, $k \leftarrow k + 1$.

Figura 4.1: Algoritmo VND.

(iii) Os movimentos são considerados suficientes para garantir uma exploração completa da região que contém x ?

(iv) Qual a precisão desejada para a solução encontrada?

A primeira pergunta refere-se a seleção e classificação dos movimentos: se eles envolvem muitas mudanças elementares, a heurística resultante pode ser muito lenta e freqüentemente pode levar mais tempo que um algoritmo exato em problemas de tamanho pequenos ou médios.

A questão (ii) também influi no esforço computacional empregado para se alcançar soluções de qualidade. Uma implementação freqüente consiste em realizar movimentos por ordem crescente de complexidade (movimentos realizados em vizinhanças de dimensões, $|N_k(x)|$, crescentes), e voltar a realizar a busca considerando o primeiro tipo de movimento (na primeira vizinhança) cada vez que uma direção de descida é encontrada e um passo é realizado naquela direção. Alternativamente, todos os movimentos podem ser aplicados em seqüência contanto que seja feita uma descida em alguma das vizinhanças considerada.

A terceira questão é crucial: para alguns problemas movimentos elementares não são suficientes para sair de um ótimo local, e heurísticas que só os utilizam podem oferecer resultados muito pobres.

Finalmente, a precisão desejada, como questionado em (iv) dependerá se a VND é usada isoladamente, ou dentro de alguma estrutura maior, como a própria VNS. No primeiro caso, pode-se realizar um esforço para obter a melhor solução possível dentro do tempo de processamento considerado; no outro caso, prefere-se determinar uma solução boa mais rapidamente através da VND e melhorá-la depois através da VNS básica, a qual será descrita na seção 4.1.3.

Ainda em relação à qualidade de um ótimo local, existe um outro aspecto importante que deve fazer parte da lógica de implementação de um algoritmo VNS. Um ótimo local de função objetivo de melhor qualidade não necessariamente pode ser mais adequado para tentar encontrar um

ótimo global. Supor que existem duas soluções ótimas locais x_a e x_b em que $f(x_a) < f(x_b)$ para o problema de minimização. Na análise tradicional pode-se concluir que x_a é um ótimo local de melhor qualidade que x_b . Entretanto, se essas soluções forem utilizadas para iniciar (ou reiniciar) o processo de busca então pode-se afirmar que aquela solução com características internas mais próximas da solução ótima global é a mais adequada para iniciar (ou reiniciar) a busca e, portanto, não necessariamente a solução x_a deve ser escolhida.

Assim, por exemplo, para o PPEST a solução ótima local que tiver o maior número de elementos n_{ij} iguais aos da solução ótima é a mais adequada para iniciar (ou reiniciar) a busca. Logicamente, em muitos casos considerados não se conhece a solução ótima. Entretanto, existem problemas onde a solução ótima é conhecida e existem vários algoritmos heurísticos para encontrar soluções ótimas locais para esse problema. Nesse contexto, pode-se usar a observação anterior para identificar o algoritmo heurístico que produz soluções ótimas locais de melhor qualidade para iniciar a busca usando o algoritmo VNS. Esse tipo de comportamento acontece no PPEST em que existem instâncias (sistemas elétricos) cujas soluções ótimas são conhecidas e existem muitos algoritmos heurísticos construtivos para encontrar excelentes soluções ótimas locais. Assim, pode-se identificar o melhor algoritmo heurístico construtivo para ser incorporado na estrutura de solução de um algoritmo VNS.

Existem várias formas de implementar o algoritmo VNS e, portanto, podem ser implementados uma família de algoritmos VNS. Em (HANSEN; MLADENOVIC, 2001d) são analisados os diversos tipos de algoritmos VNS que podem ser implementados. Diversas propostas de algoritmos VNS podem ser utilizadas de forma independente ou integradas às estruturas VNS mais complexas.

Além disso, da ordem sequencial das estruturas de vizinhança na VND pode-se desenvolver uma estratégia aninhada. Por exemplo, se $k_{max} = 3$, então, uma possível estratégia aninhada é executar a VND descrita acima para as duas primeiras estruturas de vizinhança em cada x' que pertença a terceira vizinhança de x , isto é, $\forall x' \in N_3(x)$. Aplicações da VND podem ser encontradas em (BELACEL; HANSEN; MLADENOVIC, 2002; BRINBERG et al., 2000; HANSEN; MLADENOVIC, 2001c; HANSEN; MLADENOVIC; PÉREZ, 2003).

4.1.2 VNS Reduzida: Algoritmo RVNS

Assumindo que se tenha encontrado um mínimo local x e que se pretenda então encontrar outro de melhor qualidade. Nas versões básicas da VNS, assume-se que não há nenhum conhecimento prévio do espaço de busca. Então, as questões a seguir devem ser respondidas:

- (i) Em qual direção realizar a busca?
- (ii) Qual a distância a ser percorrida?

(iii) Como modificar movimentos se eles não oferecem êxito?

A pergunta (i) relaciona a possibilidade de alcançar qualquer ponto factível $x \in X$, ou qualquer vale; a resposta mais simples é escolher uma direção ao acaso. Para problemas em variáveis 0-1 isto equivalerá a complementar algumas variáveis; para problemas Euclidianos contínuos, considerar um coeficiente angular ao acaso leva em conta todos os pontos de X .

A pergunta (ii) é crucial. De fato, ao se explorar ao limite o Fato 2 da seção 4.1, i.e., que em muitos problemas de otimização combinatória global, os ótimos locais tendem a ser próximos um do outro e estarem situados em uma região pequena (ou às vezes várias regiões) de X . Assim, uma vez encontrado um ótimo local, sempre se encontra informações implícitas sobre outros ótimos, e talvez até de ótimos globais. É, então, natural explorar primeiro sua vizinhança. Mas, se o vale que cerca um ótimo local for grande, isto pode não ser suficiente, e o que fazer é questionado em (iii). Novamente uma resposta natural é ir mais adiante.

Estes objetivos são almejados no esquema da RVNS (*Reduced Variable Neighborhood Search*) apresentado na Figura 4.2. Um conjunto de vizinhanças $N_1(x)$, $N_2(x)$, ..., $N_{k_{max}}(x)$ será considerado ao redor da solução atual x , (o qual pode ser ou não um ótimo local). Normalmente, estas vizinhanças são encaixantes, i.e., cada uma contém a anterior. Então, um ponto x' é escolhido aleatoriamente na primeira vizinhança. Se seu valor é melhor que o da incumbente, (i.e., $f(x') < f(x)$), a busca é reiniciada nesta vizinhança vizinhança ($x \leftarrow x'$). Caso contrário, passa-se à próxima vizinhança. Após todas as vizinhanças terem sido consideradas, retoma-se a busca na primeira vizinhança, até que a condição de parada esteja satisfeita (normalmente utiliza-se o tempo máximo de tempo máximo de processamento desde a última melhoria, ou número máximo de iterações).

Inicialização: Selecione o conjunto de estruturas de vizinhanças N_k , para $k = 1, \dots, k_{max}$, que será utilizado durante o processo; encontre uma solução inicial x ; escolha uma condição de parada;

Repetir a seqüência seguinte até que a condição de parada esteja satisfeita:

(1) Faça $k \leftarrow 1$;

(2) Repetir os passos seguintes até que $k = k_{max}$:

(a)**Exploração:** Determine uma solução aleatória x' de x ($x' \in N_k(x)$);

(b)**Mover ou não:** Se a solução assim obtida é melhor que x , faça $x \leftarrow x'$ e continue a busca em N_1 ($k \leftarrow 1$); Caso contrário, $k \leftarrow k + 1$.

Figura 4.2: Algoritmo VNS reduzida - RVNS.

Devido as vizinhanças serem encaixantes, o tamanho das vizinhanças vão aumentando suces-

sivamente. Então, deve-se explorar mais completamente as vizinhanças mais próximas de x do que as mais distantes, e só explorar as mais distantes quando não mais for possível se obter melhorias dentro da primeira vizinhança.

Deve-se observar que o algoritmo RVNS produz uma escolha de vizinhos mais dinâmica escolhendo vizinhos de todas as estruturas de vizinhança (diversificação) e priorizando a primeira estrutura de vizinhança (intensificação) nas fases iniciais da busca. Entretanto, um componente importante da estrutura RVNS é a capacidade de encontrar novas regiões promissoras a partir de um ótimo local. O algoritmo RVNS também pode ser usado de forma independente ou pode ser integrado em uma estrutura mais complexa de algoritmo VNS.

A RVNS é útil em problemas em que uma busca local é muito custosa. Observa-se que o melhor valor para k_{max} é freqüentemente 2. E para a condição de parada, geralmente, é utilizado o número máximo de iterações entre duas melhorias.

4.1.3 VNS Básica: Algoritmo BVNS

Algoritmos VNS mais eficientes podem ser formulados integrando as características do algoritmo VND, que permite encontrar ótimos locais de qualidade, e do algoritmo RVNS que permite encontrar novas regiões promissoras a partir de um ótimo local. Assim, juntando-se estas características podem ser formulados dois tipos de algoritmos VNS que geralmente apresentam excelente desempenho. Esses algoritmos são chamados de *Basic Variable Neighborhood Search* (BVNS) e *General Variable Neighborhood Search* (GVNS).

O algoritmo BVNS combina a busca local com mudanças sistemáticas de estruturas de vizinhança em torno do ótimo local encontrado (HANSEN; MLADENOVIC, 2001d). A estrutura do algoritmo BVNS é mostrada na Figura 4.3.

A lógica de trabalho do algoritmo BVNS é muito interessante. Inicialmente deve-se escolher as k estruturas de vizinhanças. O processo de otimização é iniciado com uma solução $x \in X$ e na primeira vizinhança $N_1(x)$. Na seqüência escolhe-se de forma aleatória um vizinho x' de x em $N_1(x)$. A partir de x' é iniciado um processo de busca local para encontrar um ótimo local x'' . Nesse contexto podem acontecer 3 casos: (1) se $x'' = x$ significa que x já era o ótimo local da vizinhança e, portanto, deve-se mudar para outro nível de vizinhança ($N_2(x)$ neste caso); (2) se x'' é de pior qualidade que x então foi encontrado um ótimo local de pior qualidade que a incumbente x e também deve-se mudar de vizinhança; e (3) se x'' é de melhor qualidade que x , significa que foi encontrada uma solução melhor que a incumbente e, portanto, deve-se atualizar a incumbente e reiniciar a busca permanecendo na vizinhança $N_1(x)$. Em qualquer iteração do processo, sempre

Inicialização: Selecione o conjunto de estruturas de vizinhanças N_k , para $k = 1, \dots, k_{max}$, que será utilizado durante o processo; encontre uma solução inicial x ; escolha uma condição de parada;

Repetir a seqüência seguinte até que a condição de parada esteja satisfeita:

- (1) Faça $k \leftarrow 1$;
 - (2) Repetir os passos seguintes até que $k = k_{max}$:
 - (a)**Agitação:** Gerar aleatoriamente uma solução x' na k -ésima vizinhança de x ($x' \in N_k(x)$);
 - (b)**Busca local:** Aplicar algum método de busca local com x' como solução inicial; denote por x'' o mínimo obtido por esta busca;
 - (c)**Mover ou não:** Se a solução x'' assim obtida é melhor que x , faça $x \leftarrow x''$ e continue a busca em N_1 ($k \leftarrow 1$); Caso contrário, $k \leftarrow k + 1$.
-

Figura 4.3: Algoritmo BVNS.

que a busca local encontra uma nova incumbente volta-se para a vizinhança $N_1(x)$ e sempre que a busca local encontra uma solução de igual ou pior qualidade que a incumbente então passa-se para uma vizinhança maior.

Essa estratégia e a escolha aleatória do vizinho x evita ciclagem e permite encontrar ótimos locais distantes da incumbente corrente. Se a última vizinhança for alcançada sem que seja encontrada uma solução melhor que a incumbente, a busca é iniciada novamente na primeira vizinhança $N_1(x)$ até que uma condição de parada seja cumprida, como por exemplo, o tempo máximo de processamento, ou número máximo de iterações, ou número de máximo iterações desde a última melhoria.

4.1.4 VNS Geral: Algoritmo GVNS

A busca local no algoritmo BVNS pode ser qualquer estratégia heurística. Entretanto, também pode-se usar uma estratégia do algoritmo VNS. Assim, o algoritmo BVNS pode ser transformado em um algoritmo mais geral chamado *General Variable Neighborhood Search* (GVNS). O algoritmo GVNS é obtido generalizando o algoritmo BVNS simplesmente considerando um algoritmo VND como busca local e utilizando o algoritmo RVNS da Figura 4.2 para melhorar a solução corrente. O algoritmo GVNS é mostrado na Figura 4.4.

A GVNS tem sido uma das aplicações que mais obteve êxito recentemente, por exemplo, (ANDREATTA; RIBEIRO, 2002; BRINBERG et al., 2000; CAPOROSSI et al., 1999; CAPOROSSI; HANSEN, 2000; HANSEN; MLADENOVIC, 2001c; RIBEIRO; SOUZA, 2002).

Inicialização: Selecione um conjunto de estruturas de vizinhanças N_k , para $k = 1, \dots, k_{max}$, que será usado durante a agitação; selecione um conjunto de estruturas de vizinhanças N'_j , para $j = 1, \dots, j_{max}$, que será usado durante a descida; encontre uma solução inicial x ; escolha uma condição de parada;

Repetir a seqüência seguinte até que a condição de parada esteja satisfeita:

- (1) Faça $k \leftarrow 1$;
 - (2) Repetir os passos seguintes até que $k = k_{max}$:
 - (a)**Agitação:** Determinar, aleatoriamente, uma solução vizinha x' , na k -ésima vizinhança de x ($x' \in N_k(x)$);
 - (b)**Busca local:** Aplicar a busca VND com as estruturas N'_j , para $j = 1, \dots, j_{max}$; denote por x'' o mínimo assim obtido;
 - (c)**Mover ou não:** Se a solução x'' assim obtida é melhor que x , faça $x \leftarrow x''$ e continue a busca em N_1 ($k \leftarrow 1$); Caso contrário, $k \leftarrow k + 1$.
-

Figura 4.4: Algoritmo GVNS.

Várias questões sobre a seleção das estruturas de vizinhança devem ser consideradas:

- (i) Que propriedades devem ter as vizinhanças para que o esquema resultante possa achar um ótimo global ou uma solução quase ótima?
- (ii) Que propriedades das vizinhanças serão úteis para determinar uma solução quase ótima?
- (iii) As vizinhanças devem ser encaixantes? Caso contrário, como deveriam ser ordenadas?
- (iv) Quais as propriedades desejáveis para os tamanhos das vizinhanças?

As duas primeiras perguntas relacionam a habilidade da metaheurística VNS para determinar os melhores vales, e fazê-lo rapidamente. Para evitar ser bloqueado em um vale, enquanto existir outro melhor, a união das vizinhanças de qualquer solução factível x deve conter todo o conjunto de soluções factíveis:

$$X \subseteq N_1(x) \cup N_2(x) \cup N_3(x) \cup \dots \cup N_{k_{max}}, \forall x \in X. \quad (4.2)$$

Estes conjuntos podem cobrir X sem necessariamente particionar, o que é mais fácil implementar por exemplo, quando utiliza-se estruturas de vizinhanças encaixantes (aninhadas), i.e.,

$$N_1(x) \subset N_2(x) \subset N_3(x) \subset \dots \subset N_{k_{max}}, \quad X \subset N_{k_{max}}, \quad \forall x \in X. \quad (4.3)$$

Se estas propriedades não forem asseguradas, não se pode garantir que o espaço de soluções X seja completamente percorrido.

Vizinhanças aninhadas podem ser facilmente obtidas em muitos problemas de otimização com-

binatória, por exemplo, definindo uma primeira vizinhança $N_1(x)$ por um tipo de movimento (por exemplo 2-opt no problema de caixeiro viajante) e então iterando k para obter as vizinhanças $N_k(x)$, para $k = 2, \dots, k_{max}$. Assim definidas, as vizinhanças, elas têm a propriedade de que os seus tamanhos estão aumentando conforme k aumenta. Assim, ao se percorrer várias vezes a sucessão inteira de vizinhanças, a primeira delas será explorada mais completamente que as demais. Isto é desejável devido ao Fato 3 mencionado na seção 4.1, i.e., ótimos locais tendem a estarem próximos uns dos outros.

4.1.5 Extensões da VNS

Em (HANSEN; MLADENOVIC, 2001a), propuseram várias extensões da VNS para resolver problemas de grandes instâncias. E em (HANSEN; MLADENOVIC, 2003a) é descrito um tutorial sobre VNS, incluindo extensões, exemplos ilustrativos e questões em relação a implementação. A evolução da VNS pode ser vista nas próximas seções.

Na literatura aparecem diversas formas de estender a VNS, às quais geralmente são acrescentadas algumas características adicionais.

As primeiras extensões derivaram-se diretamente da VNS básica. A BVNS é um método do tipo descendente obtido através da primeira melhoria com aleatorização. Sem muito esforço adicional pode ser transformado em um método ascendente-descendente: basta fazer no passo (2c) $x \leftarrow x''$ com alguma probabilidade, ainda que a solução encontrada seja pior que a solução incumbente. Também pode ser transformado em uma busca do melhor movimento: aplicando um movimento na melhor vizinhança k^* entre todas as k_{max} vizinhanças.

Outras extensões da VNS consistem em encontrar a solução x' no passo (2a) como sendo a melhor entre b (um parâmetro) soluções geradas aleatoriamente na k -ésima vizinhança, ou introduzir k_{min} e k_{passo} , dois parâmetros que controlam o processo de troca de vizinhança: no algoritmo anterior, em vez de $k \leftarrow 1$, fazer $k \leftarrow k_{min}$, e em vez de fazer $k \leftarrow k + 1$, fazer $k \leftarrow k + k_{passo}$.

4.1.5.1 VNS Inclinada: Algoritmo SVNS

Geralmente, a VNS encontra soluções melhores ou de igual qualidade que as heurísticas de partidas múltiplas, e soluções muito melhores em casos de problemas com muitos ótimos locais. Isso ocorre devido ao Fato 3 mencionado na seção 4.1: muitos problemas agrupam ótimos locais, geralmente com função globalmente convexas e com pouco ruído, o que significa que possuem muitas soluções distintas com valores parecidos, como é o caso do PPEST.

Porém, pode acontecer que, em alguns casos, os ótimos locais estejam dispersos em várias regiões do espaço de busca, e possivelmente longe dos vales que contêm as soluções quase ótimas.

Uma vez que se tenha encontrado a melhor solução em uma região grande, é necessário muito esforço para que se consiga obter uma solução melhor. Ao se considerar vizinhanças maiores, as soluções geradas aleatoriamente podem diferenciar-se substancialmente da solução atual; e por isso a VNS se degenera em uma heurística de partidas múltiplas (no sentido de que se realizam, iterativamente, descidas com soluções geradas aleatoriamente). Além disso, se a solução incumbente atual não está no vale mais profundo esta informação é em parte irrelevante. Seria então interessante modificar o esquema da VNS para explorar melhor aqueles vales que estão mais longe da solução incumbente. Isto pode ser feito aceitando realizar um movimento para uma solução de valor próximo ao da incumbente atual, mas não necessariamente melhor, desde que esta solução esteja longe da incumbente.

O esquema VNS modificado para esta variante chamada *Skewed Variable Neighborhood Search* (SVNS) é apresentado na Figura 4.5. Portanto a SVNS incorpora uma compensação da distância entre a solução gerada e a solução atual para evitar este inconveniente.

Inicialização: Selecione um conjunto de estruturas de vizinhanças N_k , para $k = 1, \dots, k_{max}$, que será usado durante a busca; encontre uma solução inicial x e seu valor $f(x)$; fazer $n^* \leftarrow x$, $f^* \leftarrow f(x)$; escolha uma condição de parada e um parâmetro α ;

Repetir a seqüência seguinte até que a condição de parada esteja satisfeita:

- (1) Faça $k \leftarrow 1$;
 - (2) Repetir os passos seguintes até que $k = k_{max}$:
 - (a)**Agitação:** Gerar aleatoriamente uma solução x' na k -ésima vizinhança de x ($x' \in N_k(x)$);
 - (b)**Busca local:** Aplicar algum método de busca local utilizando x' como solução inicial; denote por x'' o ótimo local encontrado;
 - (c)**Melhorar ou não:** Se $f(x'') < f^*$ fazer $f^* \leftarrow f(x'')$ e $n^* \leftarrow x''$;
 - (d)**Mover ou não:** Se $f(x'') - \alpha\rho(x, x'') < f(x)$ fazer $x \leftarrow x''$ e $k \leftarrow 1$; Caso contrário, fazer $k \leftarrow k + 1$.
-

Figura 4.5: Algoritmo SVNS.

A regra relaxada para reiniciar a busca utiliza uma função objetivo linear em torno da função considerada, i.e., $f(x)$ é substituída por

$$f(x) - \alpha\rho(x, x''),$$

em que $\rho(x, x'')$ é a distância de x até x'' , e α é um parâmetro. A métrica para a distância entre as soluções pode ser, por exemplo, a distância de Hamming quando soluções são descritas por vetores

booleanos, ou pela distância Euclideana, no caso contínuo. Ou ainda, a distância considerada para definir as vizinhanças N_k pode ser utilizada para este fim. A eleição do parâmetro α deve permitir a exploração de vales distantes de x quando $f(x'')$ é um pouco maior que $f(x)$, mas não muito (caso contrário a solução x sempre seria descartada), e deve evitar ciclagem. O valor de α deve ser determinado experimentalmente, pois depende de cada caso considerado.

Evidentemente, poderiam ser consideradas fórmulas mais complicadas para sair de um vale. E algumas perguntas devem ser respondidas quando aplica-se a SVNS:

(i) O problema considerado apresenta função objetivo quase convexa, ou existem vários vales profundos e separados?

(ii) Como α deveria ser escolhido?

Estas perguntas podem ser respondidas, até certo ponto, utilizando inicialmente uma versão VNS de partidas múltiplas, i.e., iniciando a VNS em vários pontos aleatórios e em um período pequeno. Depois considerar aquela posição que se encontra o melhor ótimo local e observar se eles estão agrupados ou dispersos. Então pode-se calcular os valores em função de distância do ótimo local correspondente para a melhor solução conhecida e escolher α como uma fração da inclinação média.

4.1.5.2 VNS com Decomposição: Algoritmo VNDS

A busca através de vizinhança variável com decomposição (*Variable Neighborhood Decomposition Search*, VNDS) apresentada em (HANSEN; MLADENOVIC; PÉREZ, 2001) estende a VNS em um esquema de vizinhança variável em dois níveis baseados na decomposição do problema. Seus passos são apresentados na Figura 4.6.

Note que a única diferença entre a VNS básica e a VNDS está no passo (2b): em vez de aplicar algum método de busca local no espaço completo X (começando em $x' \in N_k(x)$), na VNDS se resolve, em cada iteração, um subproblema em um subespaço $X_k \subseteq N_k(x)$ com $x' \in X_k$. Quando a busca local utilizada neste passo é também a VNS, aparece então um esquema VNS em dois níveis.

A VNDS pode ser vista com a inclusão do esquema de aproximações sucessivas, que historicamente tem sido muito utilizada em otimização combinatória, na estrutura da VNS.

4.1.5.3 VNS Paralela: Algoritmo PVNS

As buscas através de vizinhança variável paralela (*Parallel Variable Neighborhood Search*, PVNS) constituem a terceira extensão. Foram propostas em (CRAINIC et al., 2001) e (GARCIA et al.,

Inicialização: Selecione um conjunto de estruturas de vizinhanças N_k , para $k = 1, \dots, k_{max}$, que será usado durante o processo de descida; encontre uma solução inicial x ; escolha uma condição de parada;

Repetir a seqüência seguinte até que a condição de parada esteja satisfeita:

- (1) Faça $k \leftarrow 1$;
 - (2) Repetir os passos seguintes até que $k = k_{max}$:
 - (a)**Agitação:** Gerar aleatoriamente uma solução x' na k -ésima vizinhança de x ($x' \in N_k(x)$); denote-se por y o conjunto de atributos da solução presentes em x' , mas não em x , ou seja, $y = x' - x$;
 - (b)**Busca local:** Determinar o ótimo local no espaço de y por inspeção, ou por alguma técnica heurística; denote por y' a melhor solução encontrada e por x'' a solução correspondente no espaço completo X ($x'' = (x' \setminus y) \cup y'$);
 - (c)**Mover ou não:** Se a solução x'' assim obtida é melhor que x , faça $x \leftarrow x''$ e continue a busca em N_1 ($k \leftarrow 1$); Caso contrário, $k \leftarrow k + 1$.
-

Figura 4.6: Algoritmo VNDS.

2002) diversas formas de paralelizar a VNS aplicada ao problema das p -medianas. Em (GARCIA et al., 2002) são analisadas três delas:

- (i) Paralelizar a busca local;
- (ii) Aumentar o número de soluções obtidas na vizinhança atual e realizar buscas locais em paralelo a partir de cada uma das soluções;
- (iii) Fazer o mesmo que em (ii), mas atualizando a informação sobre a melhor solução encontrada.

A segunda paralelização, cujos passos são descritos na Figura 4.7, é a que ofereceu melhores resultados (HANSEN; MLADENOVIC; PÉREZ, 2003).

Foi mostrado em (CRAINIC et al., 2001) que nomeando diferentes vizinhanças em cada processador e interrompendo a busca assim que uma solução seja melhorada, obteve-se resultados muito bons.

4.1.5.4 VNS Híbrida

Considerando que a troca sistemática de estrutura de vizinhanças é uma ferramenta muito simples e robusta, outra forma de estender a VNS é incorporá-la a outras metaheurísticas.

Inicialização: Selecione um conjunto de estruturas de vizinhanças N_k , para $k = 1, \dots, k_{max}$, que será usado durante a busca; encontre uma solução inicial x e escolha uma condição de parada;

Repetir a sequência seguinte até que a condição de parada esteja satisfeita:

- (1) Faça $k \leftarrow 1$;
 - (2) Repetir, em paralelo, até que $k = k_{max}$, para cada processador p os seguintes passos:
 - (a)**Agitação:** Gerar ao azar uma solução x'_p na k -ésima vizinhança de x ($x'_p \in N_k(x)$);
 - (b)**Busca local:** Aplicar algum método de busca local utilizando x'_p como solução inicial; denote por x''_p o mínimo local assim obtido;
 - (c)**Mover ou não:** Se a solução obtida x''_p é melhor que x , fazer $x \leftarrow x''_p$ e $k \leftarrow 1$; Caso contrário, fazer $k \leftarrow k + 1$.
-

Figura 4.7: Algoritmo PVNS.

A busca tabu (*Tabu Search*, TS) (GLOVER, 1989; GLOVER, 1990; GLOVER; LAGUNA, 1997) geralmente utiliza uma estrutura de vizinhança com relação a qual realiza movimentos de descida e subida explorando diferentes tipos de memórias. A princípio, há duas maneiras de se fazer hibridismo com VNS e TS: usar algum tipo de memória para orientar a busca dentro da VNS, ou usar a VNS dentro da TS. Em (BURKE et al., 2001; KOVACEVIĆ et al., 1999; BRÄYSY, 2001) é proposto hibridismo do primeiro tipo, e em (BRINBERG et al., 2000; DAVIDOVIĆ; HANSEN; MLADENOVIC, 2001), do segundo tipo.

A metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*) (FEO; RESENDE, 1995) apresenta duas fases: na primeira fase se constroem soluções utilizando um procedimento guloso aleatório, e na segunda, as soluções são melhoradas por alguma busca local ou um método enumerativo. Uma forma natural de fazer hibridismo com VNS e GRASP é utilizar a VNS na segunda fase de GRASP, como foi feito em (MARTINS et al., 2000; RIBEIRO; UCHOA; WERNEK, 2002; ANDREATTA; RIBEIRO, 2002; CANUTO; RESENDE; RIBEIRO, 2001; OCHI; SILVA; DRUMMOND, 2001; FESTA et al., 2001).

A busca com partidas múltiplas (*Multi Start Search*, MS) (MARTI, 2002) é uma metaheurística clássica que, para evitar a estagnação de um procedimento de descida em um ótimo local, simplesmente reinicia a busca através de outra solução. Em (BELACEL; HANSEN; MLADENOVIC, 2002) é proposta e analisada uma heurística híbrida entre a VNS e a MS, consistindo em reiniciar a VNS a partir de uma solução gerada aleatoriamente no espaço X , quando a busca converge em um mínimo local, por não encontrar nenhuma melhora através das vizinhanças $N_1(x), N_2(x), \dots, N_k(x)$ da solução x .

4.2 Algumas aplicações recentes

As aplicações da metaheurística VNS tem crescido rapidamente em números desde 1995 quando foi apresentado o primeiro trabalho sobre esta metaheurística (MLADENOVIC, 1995). A VNS tem mostrado suas vantagens em diversas áreas importantes de Inteligência Artificial (IA), embora a maioria dos trabalhos sejam aplicados a problemas de otimização.

Os problemas de otimização são o núcleo da Pesquisa Operacional e suas aplicações aparecem em numerosos campos. Em Inteligência Artificial aparecem muitos problemas de satisfação de restrições nos quais as heurísticas devem buscar uma solução factível. Ambos os tipos de problemas diferem menos do que pode parecer à primeira vista, já que os problemas de satisfação de restrições podem ser expressos como problemas de otimização em uma variedade de formas. Por exemplo, a minimização da soma de variáveis artificiais que representem a violação de restrições. Por outro lado, muitos problemas de otimização podem ser formulados como problemas de satisfação de restrições.

Nos últimos anos a programação por restrições (*Constraint Programming*, CP) tem atraído grande atenção em IA por seu potencial ao abordar a resolução prática de problemas da vida real. A característica mais relevante é a forma declarativa de tratar os problemas, já que especificam as restrições que devem cumprir as soluções sem especificar um procedimento computacional para obtê-las. Exemplos de aplicação de busca com VNS neste contexto são (GENDREAU; PESANT; ROUSSEAU, 2002) e (LOUDIN; BOIZUMAULT, 2001).

Nas subseções seguintes descrevem-se os aspectos mais relevantes da aplicação da VNS a distintos campos em IA. Em particular, considera-se, na subseção 4.2.1, a aplicação da VNS a alguns problemas clássicos de IA. Na subseção 4.2.2 incluímos a aplicação em teoria dos grafos, e por fim, na seção 4.2.3, uma aplicação da VNS a diversos problemas relevantes de otimização combinatória, destacando problemas de planejamento logístico e distribuição.

4.2.1 Tarefas clássicas de Inteligência Artificial

Entre as diversas aplicações da VNS em problemas clássicos de Inteligência Artificial (IA), destacam-se os problemas de satisfação, aprendizado em Redes Bayesianas, problemas de classificação e de planejamento.

Os **problemas de satisfação** (DINGZHU; PARDALOS, 1997) têm recebido muita atenção em IA. Este problema, consiste em determinar se um conjunto de cláusulas (todas de forma disjunta ou todas de forma conjunta) construídas sobre n variáveis lógicas tem solução ou não. Em (HANSEN et al., 2000) foram realizadas aplicações da VNS e SVNS no problema de satisfação máxima ponderada, que consiste em, dado um peso positivo a cada cláusula, encontrar a solução que maximize a soma

dos pesos das cláusulas satisfeitas.

As **redes bayesianas** (PEARL, 1998) constituem uma das metodologias mais importantes para tratar as relações de dependência. As redes bayesianas são estruturas que permitem representar a dependência ou independência entre variáveis aleatórias e constituem ferramentas eficientes para realizar inferências através da fórmula de Bayes. Uma rede bayesiana é representada mediante um grafo orientado, com um conjunto de variáveis como vértices e arestas representando dependências. O tipo de rede bayesiana a ser considerado se restringe a um tipo de grafo específico: em muitos casos, a uma árvore ou a uma cadeia. O problema consiste em construir uma rede bayesiana que se ajuste e aprenda de um conjunto de realizações conjuntas das variáveis. Em (CAMPOS; PUERTA, 2001) é usado a VNS baseada em ordenações de redes bayesianas associadas a permutações.

O agrupamento ou *clustering* (HARTIGAN, 1975), é uma técnica relevante de análise exploratória de dados. O problema se caracteriza no contexto da **classificação** não supervisionada, onde não se conhecem a priori, as classes dos objetos disponíveis. Da perspectiva de aprendizagem não supervisionada ou de descoberta, a tarefa se concretiza na obtenção da classificação ótima, por partição e agrupamento, de um conjunto dado de objetos a partir dos atributos que os descrevem. Diversas versões da VNS foram comparadas com grande êxito em (HANSEN; MLADENOVIC, 2001c) às heurísticas clássicas para esta tarefa. Em (BELACEL; HANSEN; MLADENOVIC, 2002) foi usada a estrutura VNS para abordar uma versão deste problema (*Clustering Problem*) onde os pontos são dispostos em diferentes graus e várias classes.

O planejamento de tarefas é um dos problemas de otimização que mais tem recebido atenção em IA (PINEDO, 2001). Em problemas de seqüenciamento, as soluções são dadas por permutações de um conjunto de tarefas. Vários problemas de seqüenciamento em uma máquina com diversas funções objetivos são resolvidos com VNS e VND em (BESTEN; STÜTZLE, 2001). Em (BRINBERG et al., 2000) foi considerado problemas com várias máquinas que podem trabalhar simultaneamente em algumas tarefas, mas respeitando uma tabela de prioridades entre as tarefas. Em (FLESZAR; HINDI, 2004) foi aplicado a VNS em problemas em que também se consideraram restrições de recursos. Em (BURKE et al., 2001) foram resolvidos problemas de planejamento de horário de enfermeiras em um hospital.

4.2.2 Teoria dos Grafos

A metaheurística VNS também tem se mostrado muito útil em teoria dos grafos. Uma invariante da teoria de grafos é uma variável definida em uma família de grafos cujo valor não depende da numeração dos vértices ou arestas. Numerosos problemas da teoria dos grafos podem ser vistos como otimização de invariantes dentro de uma família de grafos. A VNS foi utilizada para aproximar soluções de problemas, negar algumas conjecturas e estabelecer outras, algumas das quais foram

provadas formalmente (CAPOROSSO; HANSEN, 2000).

Em (CAPOROSSO; HANSEN, 2000) também foi mostrado como utilizar esquemas da VNS e VND dentro do sistema **AutoGraphiX**(AGX) para invariantes da teoria do grafo.

4.2.3 Problemas de Otimização

A VNS foi aplicada com grande êxito em muitos problemas de otimização combinatória, entre os quais se encontram os problemas mais relevantes de planejamento logístico: o problema de empacotamento, de localização e de rotas. Adicionalmente, também foram abordados problemas onde as soluções são certos conjuntos de arestas, como as árvores, e alguns problemas de otimização contínua.

Os problemas de **empacotamento** constituem uma das classes de problemas importantes no contexto de logística e distribuição. Em (FLESZAR; HINDI, 2002) foi utilizada a VNS básica em uma das versões básicas: o problema de empacotamento unidimensional (*bin-packing problem*, BPP), em que se tem de empacotar um conjunto de objetos de diferentes tamanhos no menor número de caixas ou segmentos de capacidade fixa.

Os problemas de **localização**, muito relevantes no planejamento logístico, e os problemas de **agrupamento** (*clustering*) têm características comuns, e por isso permitem que sejam tratados de forma similar em buscas metaheurísticas. O problema das p -medianas é o problema mais extensamente estudado em Teoria de Localização. Consiste em determinar p locais, entre um conjunto de m localizações possíveis para um serviço, de forma que se minimize a soma das distâncias de n usuários a seu posto de serviço mais próximo. Este problema foi abordado em (HANSEN; MLADENOVIC, 1997), em (WHITTAKER, 1983) e em (HANSEN; MLADENOVIC; PÉREZ, 2001) utilizando a VNS básica, a VNS reduzida e a VNDS, respectivamente. Diversas variantes da VNS foram aplicadas em (BRINBERG et al., 2000) ao problema de janelas múltiplas, que é uma versão contínua do problema das p -medianas, onde as p localizações podem ser escolhidas em todo o plano, onde também estão localizados os usuários.

Outro tipo de problema de otimização combinatória importante, em relação a todo contexto de planejamento logístico, são os **problemas de rotas**. Tanto as versões clássicas do problema do caixeiro viajante (*Traveling Salesman Problem*, TSP), do problema de rota de veículos (*Vehicle Routing Problem*, VRP), como algumas de suas extensões foram abordadas com a VNS.

O TSP consiste em dadas n cidades com as distâncias e custos entre elas, encontrar uma rota de custo mínimo. Em (HANSEN; MLADENOVIC, 1997) e em (MLADENOVIC; HANSEN, 1997) são aplicados procedimentos do tipo VNS com distintos tipos de movimento para o TSP. Em (OCHI; SILVA; DRUMMOND, 2001) é utilizado a VNS para resolver uma extensão do TSP denominada problema do comprador, e que dada uma partição do conjunto de clientes, deve se visitar pelo menos

um dos conjuntos de cada partição.

Em (SILVA; DRUMMOND; OCHI, 2000) são apresentados algoritmos baseados em GRASP e VNS para resolver generalizações do TSP, onde apenas um conjunto de cidade é visitado, e os autores concluem que a combinação destas metaheurísticas apresenta resultados bem expressivos.

O problema de rotas de veículos (*Vehicle Routing Problem*, VRP) consiste em determinar rotas a partir de um depósito para visitar um conjunto de clientes com um custo mínimo. Os clientes têm demandas conhecidas e os veículos possuem capacidade limitada. Em (BRÄYSY, 2001) foram aplicados a VNS e VND na resolução de uma variante deste problema, onde cada cliente deve ser visitado em certo intervalo de tempo, que se denomina VRPTW (*Time Windows*).

Outros problemas com múltiplas aplicações importantes cuja soluções são **árvores**, como o problema de *Steiner*, onde se deve determinar a árvore de menor distância que conecta todos os nós terminais em um grafo. Foram testadas várias versões da VND para estes problemas em (MARTINS et al., 2000) e (RIBEIRO; UCHOA; WERNEK, 2002). Em outras versões em que o grau de cada vértice no subgrafo está incluído, também foram testados a VNS com sucesso, (RIBEIRO; SOUZA, 2002). Em (CANUTO; RESENDE; RIBEIRO, 2001) foi abordado o problema de determinar a subárvore que minimiza a soma das distâncias das arestas incluídas no grafo e o peso dos vértices conectados. Em conectados. Em (ANDREATTA; RIBEIRO, 2002) foi aplicada a VND ao problema da árvore filogenética (*the Phylogeny Problem*). A efetividade da VNS foi provada em (FESTA et al., 2001) para resolver problemas de corte máximo, onde o objetivo é dividir o grafo em duas partes de forma que a soma dos pesos das arestas que os unem seja máxima.

Em (QU; BURKE, 2005) é apresentada uma versão de hyperheurística com VNS aplicada ao problema de tabela de horários de exames (*Exam Timetabling Problem*) com comparações entre TS e *Iterated Local Search*.

Em (BRINBERG; HANSEN; MLADENOVIC, 2004) foi realizada uma análise das propriedades de convergência da metaheurística VNS em condições gerais. Também é apresentada uma versão híbrida da VNS e de *Random Multistart Local Search* (MLS) aplicada ao conhecido problema de alocação contínua, *Multisource Weber Problem*, e uma explicação do fato de a técnica VNS obter resultados superiores aos obtidos com MLS.

4.3 VNS aplicada no PPEST

Nesta seção descrevem-se os detalhes de implementação da busca em vizinhança variável aplicada ao problema de planejamento da expansão de sistemas de transmissão.

Na primeira implementação, utiliza-se a VNS de descida, VND, cujo algoritmo aparece novamente na Figura 4.8, com o modelo de transportes, cuja solução inicial foi obtida com o Algoritmo

de Garver. Na segunda implementação também utiliza-se a VND, mas com o modelo DC e, nesse caso, a solução inicial é obtida utilizando-se o algoritmo Villasana-Garver-Salon.

Inicialização: Selecione o conjunto de estruturas de vizinhanças N_k , para $k = 1, \dots, k_{max}$, que será utilizado durante o processo; encontre uma solução inicial x (ou aplique a regra a um x conhecido).

Repetir a seqüência até que nenhuma melhoria seja obtida:

- (1) Faça $k \leftarrow 1$;
 - (2) Repetir até que $k = k_{max}$:
 - (a) **Exploração:** Encontre o melhor vizinho x' de x ($x' \in N_k(x)$);
 - (b) **Mover ou não:** Se a solução assim obtida é melhor que x , faça $x \leftarrow x'$ e $k \leftarrow 1$; Caso contrário, $k \leftarrow k + 1$.
-

Figura 4.8: Algoritmo VND Básico.

Em todos os casos considerados, seguiu-se os passos do algoritmo descrito na Figura 4.9, os quais serão explicados em detalhes nas seções seguintes.

Passo 1. Codificação do problema: como representar as propostas de soluções;

Passo 2. Solução inicial: Utilizar um algoritmo heurístico para determinar uma solução inicial;

Passo 3. Definição das vizinhanças: Caracterizar cada vizinhança e determinar seus elementos;

Passo 4. Busca local: Indicar um mecanismo que possa determinar a melhor configuração em cada vizinhança da solução corrente.

Figura 4.9: Algoritmo VND para o PPEST.

4.3.1 Passo 1: Codificação do PPEST

Considere o sistema da Figura 4.10, em que existem seis circuitos na topologia básica (representados pelos traços contínuos na Figura 4.10): $n_{12}^0 = n_{14}^0 = n_{15}^0 = n_{23}^0 = n_{24}^0 = n_{35}^0 = 1$; e seis circuitos adicionados (representados pelos traços pontilhados na Figura 4.10): $n_{12}^1 = n_{23}^1 = n_{26}^1 = n_{35}^1 = 1$ e $n_{46} = 2$. Pode-se adicionar circuitos em todos os caminhos.

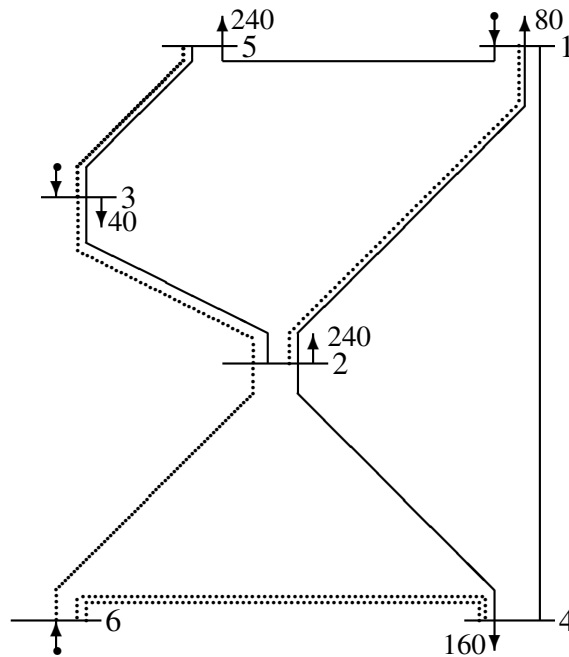


Figura 4.10: Sistema de 6 barras com seis circuitos na configuração base e seis circuitos adicionados.

Esta configuração pode ser representada através de um vetor $n = (n_{11}, n_{12}, \dots)$ de dimensão nl , em que nl indica o número de caminhos (ramos) em que é possível adicionar circuitos, com elementos n_{ij} que indicam a quantidade de circuitos adicionados no caminho ij , como pode ser observado na Figura 4.11.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	→ Ramos
	1-2	1-3	1-4	1-5	1-6	2-3	2-4	2-5	2-6	3-4	3-5	3-6	4-5	4-6	5-6	→ Barras
$n =$	1	0	0	0	0	1	0	0	1	0	1	0	0	2	0	
						↓										
						Circuitos adicionados no										
						processo de otimização										

Figura 4.11: Codificação de uma proposta de solução para o PPEST.

Assim, a configuração n da Figura 4.11 mostra que existe um circuito adicionado no caminho 1 (entre as barras 1 e 2), um circuito adicionado no caminho 6 (entre as barras 2 e 3), outro no caminho 9 (entre as barras 2 e 6), outro no caminho 11 (entre as barras 3 e 5) e outros dois no caminho 14 (entre as barras 4 e 6).

Nos demais caminhos não existe nenhum circuito adicionado.

4.3.2 Passo 2: Solução inicial

Para determinar a solução inicial do problema de planejamento da expansão do sistema de transmissão, em cada passo é escolhido um circuito a ser adicionado ao sistema, onde a escolha é dada por um indicador de sensibilidade especificado pelo AHC. O processo iterativo é finalizado quando uma solução factível e geralmente de boa qualidade foi encontrada.

Embora os AHC sejam robustos e converjam rapidamente, em problemas grandes e complexos apenas convergem em uma solução de boa qualidade (mínimo local).

4.3.2.1 Solução inicial com o Modelo de Transportes

No caso de se considerar o modelo de transportes, utiliza-se o AHC proposto por Garver (GARVER, 1970), descrito na seção 3.2.1, que usa o indicador de sensibilidade definido a partir da solução ótima do modelo (4.4)-(4.10).

$$\min v = \sum_{(i,j) \in \Omega} c_{ij} n_{ij} \quad (4.4)$$

s.a.

$$Sf + g = d \quad (4.5)$$

$$|f_{ij}| \leq (n_{ij}^0 + n_{ij}^1 + n_{ij}) \bar{f}_{ij} \quad (4.6)$$

$$0 \leq g \leq \bar{g} \quad (4.7)$$

$$0 \leq n_{ij} \leq (\bar{n}_{ij} - \bar{n}_{ij}^1) \quad (4.8)$$

$$f_{ij} \text{ irrestrito} \quad (4.9)$$

$$\forall (i, j) \in \Omega \quad (4.10)$$

Neste modelo, apenas as restrições da lei de Kirchhoff das correntes, (4.5), devem ser obrigatoriamente satisfeitas.

Como foi mencionado na seção 3.2.1, a proposta de Garver consiste em se resolver o PL (4.4)-(4.10) para determinar a solução ótima contínua para a configuração corrente n_{ij}^0 , e adicionar um circuito no caminho ij que leva o maior fluxo de potência, ou seja, adicionar um circuito no caminho determinado pelo índice 4.11.

$$IS = \max\{s_{ij} = n_{ij} \bar{f}_{ij} : n_{ij} \neq 0 \text{ na solução do PL (4.4)-(4.10)}\}. \quad (4.11)$$

Um processo repetitivo desta estratégia, adicionando em cada passo um circuito no caminho mais atrativo, constitui o algoritmo de Garver.

O processo termina quando a solução do PL (4.4)-(4.10) correspondente à configuração cor-

rente $n_{ij}^0 + n_{ij}^1$ apresenta uma solução com todos os $n_{ij} = 0$, o que significa que não é mais necessário realizar adições e o conjunto de circuitos adicionados representa a solução inicial para se aplicar a VNS.

4.3.2.2 Solução inicial com o Modelo DC

Para determinar uma solução inicial factível para o modelo DC utiliza-se o AHC proposto por Villasana-Garver-Salon em (VILLASANA; GARVER; SALON, 1985), o qual já foi descrito na seção 3.2.2, e considerando-se o modelo híbrido (4.12)-(4.22) com a estratégia de resolver apenas um PL em cada passo do processo de otimização após relaxar a integralidade das variáveis de investimento n_{ij} .

$$\min v = \sum_{(i,j) \in \Omega} c_{ij} n_{ij} \quad (4.12)$$

s.a.

$$Sf + S^0 f^0 + S^1 f^1 + g = d \quad (4.13)$$

$$f_{ij}^0 - \gamma_{ij} n_{ij}^0 (\theta_i - \theta_j) = 0, \forall (i, j) \in \Omega_0 \quad (4.14)$$

$$f_{ij}^1 - \gamma_{ij} n_{ij}^1 (\theta_i - \theta_j) = 0, \forall (i, j) \in \Omega_1 \quad (4.15)$$

$$|f_{ij}^0| \leq n_{ij}^0 \bar{f}_{ij}, \forall (i, j) \in \Omega_0 \quad (4.16)$$

$$|f_{ij}^1| \leq n_{ij}^1 \bar{f}_{ij}, \forall (i, j) \in \Omega_1 \quad (4.17)$$

$$|f_{ij}| \leq n_{ij} \bar{f}_{ij}, \forall (i, j) \in \Omega \quad (4.18)$$

$$0 \leq g \leq \bar{g} \quad (4.19)$$

$$0 \leq n_{ij} \leq (\bar{n}_{ij} - n_{ij}^1) \quad (4.20)$$

$$f_{ij}^0, f_{ij}^1, \text{ e } f_{ij} \text{ irrestritos} \quad (4.21)$$

$$\theta_j \text{ irrestrito} \quad (4.22)$$

Assim, em cada passo, no algoritmo VGS resolve-se o PL (4.12)-(4.22) e verifica se os circuitos já adicionados permitem uma operação adequada do sistema ou, caso contrário, identifica o circuito mais promissor que deve ser adicionado ao sistema, que é determinado através do índice de sensibilidade definido como sendo o fluxo de potência pelos circuitos com $n_{ij} \neq 0$ na solução do PL:

$$IS = \max\{s_{ij} = n_{ij} \bar{f}_{ij} : n_{ij} \neq 0 \text{ na solução do PL (4.12)-(4.22)}\}, \quad (4.23)$$

Do ponto de vista de otimização matemática, o algoritmo de Garver e o algoritmo Villasana-Garver-Salon são algoritmos heurísticos construtivos que na prática encontram configurações de boa qualidade, mas do ponto de vista teórico, não existe garantia de encontrarem a solução ótima global.

4.3.3 Passo 3: Definição das estruturas de vizinhanças

Dada uma solução n , utilizando a definição (4.1), pode-se definir a seguinte estrutura de vizinhanças no espaço de solução S :

$$N_k(n) = \{n' \in S : d(n, n') = k, k = 1, \dots, k_{max}\}, \quad (4.24)$$

em que $d(n, n') = k$ é a quantidade de posições com atributos distintos em n e n' . Assim, considerando $k_{max} = 6$, os elementos de cada vizinhança são obtidos, respectivamente, realizando os seguintes movimentos:

$N_1(n)$: Retirada de circuitos em um caminho;

$N_2(n)$: Troca de dois caminhos (entram circuitos em um caminho e saem circuitos de outro caminho);

$N_3(n)$: Troca de até três caminhos (entram circuitos em um caminho e saem circuitos de dois outros caminhos, ou entram circuitos em dois caminhos e saem circuitos de outro caminho);

$N_4(n)$: Troca de até quatro caminhos (entram circuitos em dois caminhos e saem circuitos de dois caminhos);

$N_5(n)$: Troca de até cinco caminhos (entram circuitos em dois caminhos e saem circuitos de três caminhos, ou entram circuitos em três caminhos e saem circuitos de dois caminhos);

$N_6(n)$: Troca de até seis caminhos (entram circuitos em três caminhos e saem circuitos de três caminhos).

Note que pode-se retirar ou acrescentar mais que um circuito no mesmo caminho considerado.

Para exemplificar, considere-se a configuração n da Figura 4.11.

As configurações n_1^1 e n_1^2 da Figura 4.12 são vizinhos de n em N_1 . O vizinho $n_1^1 \in N_1(n)$ foi obtido a partir de n com a retirada de um circuito no caminho 9, e o vizinho $n_1^2 \in N_1(n)$, foi obtido a partir de n com a retirada de um circuito no caminho 14.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$n_1^1 =$	1	0	0	0	0	1	0	0	0	0	1	0	0	2	0
$n_1^2 =$	1	0	0	0	0	1	0	0	1	0	1	0	0	1	0

Figura 4.12: Vizinhos em $N_1(n)$.

As configurações n_2^1 e n_2^2 na Figura 4.13 são vizinhos de n em $N_2(n)$. O vizinho $n_2^1 \in N_2(n)$ foi obtido a partir de n com a entrada de dois circuitos no caminho 8 e a retirada de um circuito no

caminho 11. O vizinho $n_2^2 \in N_2(n)$ foi obtido de n com a retirada de um circuito no caminho 6 e com a entrada de dois circuitos no caminho 12.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$n_2^1 =$	1	0	0	0	0	1	0	2	1	0	0	0	0	2	0
$n_2^2 =$	1	0	0	0	0	0	0	0	1	0	1	2	0	2	0

Figura 4.13: Vizinhos em $N_2(n)$.

As configurações n_3^1 e n_3^2 na Figura 4.14 são vizinhos de n em $N_3(n)$. O vizinho $n_3^1 \in N_3(n)$ foi obtido a partir de n com a adição de um circuito no caminho 3 e a adição de 2 circuitos no caminho 12 e com a retirada de 2 circuitos no caminho 14. O vizinho $n_3^2 \in N_3(n)$ foi obtido de n com a retirada de um circuito no caminho 6, com a adição de dois circuitos no caminho 9 e com a retirada de um circuito no caminho 14.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$n_3^1 =$	1	0	1	0	0	1	0	0	1	0	1	2	0	0	0
$n_3^2 =$	1	0	0	0	0	0	0	0	3	0	1	0	0	1	0

Figura 4.14: Vizinhos em $N_3(n)$.

Da mesma forma pode-se obter os demais vizinhos através do critério de vizinhança definido.

4.3.4 Passo 4: Busca Local

A busca local também é feita utilizando-se os dois modelos considerados.

A partir de uma solução factível inicial, pretende-se determinar outras soluções factíveis de melhor qualidade que se encontrem na vizinhança desta solução. E isso deve ser feito em todas as estruturas de vizinhanças predefinidas, sendo que existe a necessidade de verificar se essas propostas de solução são factíveis ou infactíveis.

A factibilidade de uma proposta de solução (neste caso, as soluções vizinhas da topologia corrente) é verificada resolvendo um PL que verifica apenas se o sistema opera adequadamente. Assim, deve-se apenas verificar se com a proposta de solução candidata o sistema opera com corte ou sem corte de carga.

4.3.4.1 Modelo de Transportes

Quando se considera a VND com o modelo de transportes, a busca local em cada vizinhança é feita utilizando o modelo (4.25)-(4.32) para verificar a factibilidade das pro-

$$\min w = \sum_{k \in \Omega} r_k \quad (4.25)$$

$$\text{s.a.} \quad (4.26)$$

$$Sf + g + r = d \quad (4.27)$$

$$|f_{ij}| \leq (n_{ij}^0 + n_{ij}^1) \bar{f}_{ij} \quad (4.28)$$

$$0 \leq r \leq d \quad (4.29)$$

$$0 \leq g \leq \bar{g} \quad (4.30)$$

$$f_{ij} \text{ irrestrito} \quad (4.31)$$

$$\forall (i, j) \in \Omega \quad (4.32)$$

postas de soluções candidatas, em que w representa o corte de carga do sistema e os r_k representam os cortes de carga nas barras de demanda (que é equivalente a geração artificial):

Na vizinhança $N_1(n)$, ordena-se os circuitos por ordem decrescente de custo e tenta-se retirar aqueles que, uma vez simulada sua saída (resolvendo os respectivos PL's, (4.25)-(4.32)), não produzam corte de carga no sistema, ou seja, retornem o valor $\omega = 0$. Como, em geral, o número de caminhos adicionados é pequeno, a busca em N_1 leva a um esforço computacional muito pequeno.

Nas demais vizinhanças, a busca local é feita através da troca de caminhos. Neste caso, deve-se simular a **retirada** de circuitos em um caminho (um ou mais circuitos no mesmo caminho considerado) que tem adição na configuração corrente e a **adição** de circuitos em cada um dos caminhos candidatos. Para isso, calculam-se as variações de custos devido às trocas (diferença do custo das linhas que são adicionadas no sistema e das linhas que saem do sistema) e só se verifica a operação do sistema para as trocas de linhas que apresentam variação negativa, ou seja, só resolve-se PL's para as propostas de solução que acarretam uma redução no valor da função objetivo. Se a simulação indicar, ao resolver o PL (4.25)-(4.32), que se trata de uma configuração factível, essa configuração é, então, armazenada para, no fim da busca local, se escolher a melhor configuração candidata para realizar o melhor movimento. Ou seja, durante a busca local, para decidir se a configuração candidata é uma proposta factível para o modelo de transportes, é verificado se na solução do PL (4.25)-(4.32) tem-se $w = 0$. Caso contrário, a proposta representa uma solução infactível e é então descartada, ou seja, a simulação de trocas de circuitos é cancelada e passa-se para a próxima tentativa.

Como as trocas que incrementam o investimento são automaticamente rejeitadas, a busca lo-

cal implica pouco esforço computacional nas primeiras estruturas de vizinhança, o que não ocorre principalmente em N_5 e N_6 , onde há muitas combinações para serem testadas.

Cabe esclarecer aqui que o movimento só é realizado se a nova configuração é melhor que a incumbente.

4.3.4.2 Modelo DC

Quando utiliza-se o modelo DC, a busca local nas vizinhanças consideradas é realizada da mesma forma que no modelo de transportes, mas considerando que o PL que permite verificar a viabilidade de uma proposta de solução quando se considera o modelo DC, assume a forma (4.33)-(4.41).

$$\min w = \sum_{k \in \Omega} r_k \quad (4.33)$$

$$\text{s.a.} \quad (4.34)$$

$$Sf + g + r = d \quad (4.35)$$

$$f_{ij} - \gamma_{ij}(n_{ij}^1 + n_{ij}^0)(\theta_i - \theta_j) = 0 \quad (4.36)$$

$$|f_{ij}| \leq (n_{ij}^0 + n_{ij}^1)\bar{f}_{ij} \quad (4.37)$$

$$0 \leq r \leq d \quad (4.38)$$

$$0 \leq g \leq \bar{g} \quad (4.39)$$

$$f_{ij}, \theta_j \text{ irrestritos} \quad (4.40)$$

$$\forall (i, j) \in \Omega \quad (4.41)$$

Igualmente ao caso anterior, se $w = 0$ em (4.33), significa que a topologia avaliada representa uma solução factível para o modelo DC. Caso contrário, a proposta representa uma solução infactível, e é então descartada, e passa-se à avaliação da próxima proposta de solução candidata.

4.3.5 Critério de Parada

O critério de parada utilizado foi parar o processo de busca ao se determinar a solução ótima ou parar ao se alcançar $N_{k_{max}}$.

4.4 Exemplo: VND com Modelo DC no Sistema de 6 barras de Garver

Considerando-se o sistema de 6 barras de Garver com a rede básica representada pelos circuitos de traços contínuos e respectivos custos, c_{ij} , descritos na Figura 4.15, para se determinar uma solução inicial factível para o modelo DC, aplicou-se o algoritmo VGS da seção 3.2.2 e após se resolver seis PL's, obteve-se a solução inicial n com um investimento de 130,000 u.m. e cinco adições em quatro caminhos: $n_{23} = n_{26} = n_{35} = 1$ e $n_{46} = 2$ (traços pontilhados na Figura 4.15).

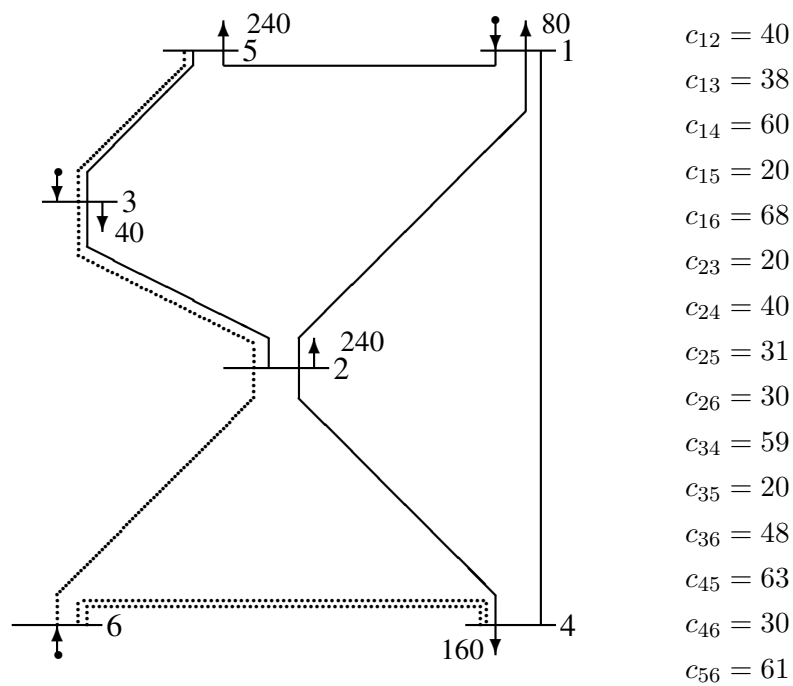


Figura 4.15: Solução VGS para o sistema de 6 barras de Garver.

A representação desta proposta de solução, desconsiderando-se a rede básica, pode ser vista na Figura 4.16.

	1-2	1-3	1-4	1-5	1-6	2-3	2-4	2-5	2-6	3-4	3-5	3-6	4-5	4-6	5-6
$n =$	0	0	0	0	0	1	0	0	1	0	1	0	0	2	0

Figura 4.16: Codificação da solução inicial do sistema de 6 barras de Garver.

Utilizando-se esta solução inicial e a VND proposta para o modelo DC, inicialmente em $N_1(n)$, resolvem-se quatro PL's (tem-se adições em quatro caminhos na configuração corrente, ou seja,

$|N_1| = 4$) para simular a retirada das linhas que foram adicionadas, e verifica-se que não há nenhuma possibilidade de retirada de linhas sem corte de carga no sistema.

Passando-se para a segunda estrutura de vizinhança, ou seja, fazendo $k = 2$, tem-se um total de $15 \times 4 = 60$ (todos os caminhos, com e sem adição na configuração base, são candidatos à adição de circuitos, e os caminhos com adição de circuitos na configuração base são candidatos à retirada) situações de troca de circuitos, mas, como não se resolve PL's para trocas que aumentam o valor da função objetivo, resolvem-se apenas 12 PL's, como pode ser visto abaixo, ou seja, $|N_2| = 12$, uma vez que se um caminho ij , com custo c_{ij} , é candidato a sair do sistema, os únicos caminhos que podem ser candidatos a entrarem no sistema são aqueles que possuem custo menores ou iguais a c_{ij} :

$$\begin{aligned}
n_{23}^0(c_{23} = 20) : n_{15}(c_{15} = 20), n_{35}(c_{35} = 20) &\Rightarrow 2\text{PL's} \\
n_{26}^0(c_{26} = 30) : n_{15}(c_{15} = 20), n_{23}(c_{23} = 20), n_{35}(c_{35} = 20), n_{46}(c_{46} = 30) &\Rightarrow 4\text{PL's} \\
n_{35}^0(c_{35} = 20) : n_{15}(c_{15} = 20), n_{23}(c_{23} = 20) &\Rightarrow 2\text{PL's} \\
n_{46}^0(c_{46} = 30) : n_{15}(c_{15} = 20), n_{23}(c_{23} = 20), n_{26}(c_{26} = 30), n_{35}(c_{35} = 20) &\Rightarrow 4\text{PL's}
\end{aligned}$$

Mas, dentre as 12 soluções candidatas, apenas duas soluções são factíveis: uma com a troca que retira um circuito no caminho 2-3 e acrescenta outro no caminho 3-5 (representada pela configuração n_2^1 da Figura 4.17), e outra solução que retira um circuito no caminho 2-6 e acrescenta outro no caminho 4-6 (representada pela configuração n_2^2 da Figura 4.17).

	1-2	1-3	1-4	1-5	1-6	2-3	2-4	2-5	2-6	3-4	3-5	3-6	4-5	4-6	5-6
$n_2^1 =$	0	0	0	0	0	0	0	0	1	0	2	0	0	2	0
$n_2^2 =$	0	0	0	0	0	1	0	0	0	0	1	0	0	3	0

Figura 4.17: Vizinhos candidatos em $N_2(n)$.

O algoritmo escolhe a solução n_2^2 , pois é a última opção dentre duas com o mesmo valor de incremento na função objetivo: $-c_{23} + c_{35} = -20 + 20 = 0$ e $-c_{26} + c_{46} = -30 + 30 = 0$.

Retomando-se a busca em $N_1(n_2^2)$, resolve-se mais 3 PL's ($|N_1(n_2^2)| = 3$), e é então retirado, sem causar corte de carga no sistema, o circuito do caminho 2-3, obtendo-se a solução ótima do sistema para o modelo DC, a qual pode ser vista na Figura 4.18, com um investimento de 110,000 u.m., resolvendo-se apenas 25 PL's (sendo que destes, 6 foram utilizados para determinar a solução inicial) e utilizando-se apenas duas estruturas de vizinhança.

Este sistema foi resolvido em (GALLEGO; MONTICELLI; ROMERO, 2000) utilizando a meta-heurística *Tabu Search* e o modelo de transportes, mas resolvendo um número de PL's bem maior, a saber, 97 PL's.

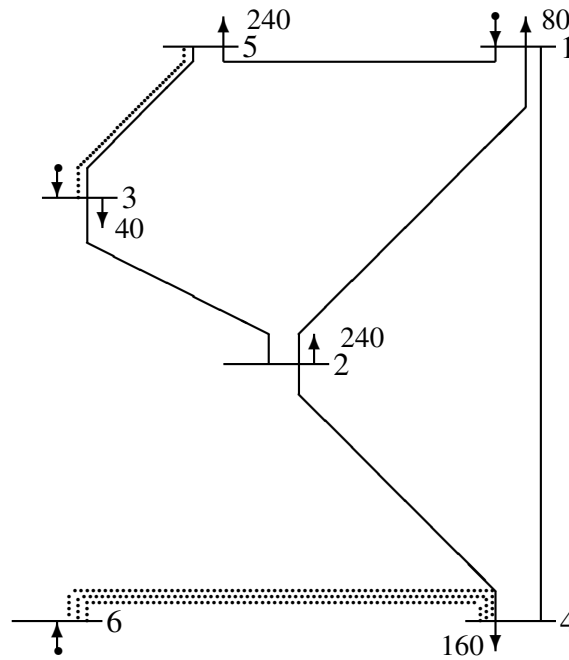


Figura 4.18: Solução ótima (VND/DC) do sistema de 6 barras de Garver.

4.5 Redução de Vizinhança

Um ponto crítico na VND é a caracterização da estrutura de vizinhança $N_k(n)$.

Como na VND avaliam-se todos os vizinhos em cada vizinhança considerada a fim de realizar o melhor movimento, se a vizinhança considerada é muito grande exige-se um esforço computacional muito elevado para a realização de cada movimento. Assim, faz-se necessária a aplicação de técnicas de redução de vizinhança.

Considerando que as estruturas de vizinhança foram definidas através de trocas (entrada/retirada) de circuitos, para determinar quais são os vizinhos da configuração corrente, deve-se identificar os circuitos candidatos à adição ou remoção. Então, após resolver o PL (4.25)-(4.32), ou o PL (4.33)-(4.41), dependendo do caso considerado, para a topologia corrente n , e após se calcular os fluxos nas linhas de transmissão, pode-se determinar os vizinhos candidatos nas respectivas vizinhanças, observando-se os seguintes fatos:

1. Dentre os caminhos existentes, são candidatos à adição de circuitos no sistema os caminhos com circuitos que estão operando em capacidade superior a 50% ou 60%.
2. Todos os caminhos novos que ainda não possuem circuitos adicionados são candidatos à adição de circuitos no sistema.
3. As linha presentes na configuração corrente que estão operando em capacidade inferior a 40%

são candidatas à retirada.

4. Todas as linhas presentes na configuração corrente podem ser consideradas candidatas à adição.

Em cada uma das estruturas de vizinhanças apresentadas, considerando-se os circuitos candidatos à adição e os circuitos candidatos à retirada, foram calculadas as variações de custo devido às trocas de circuitos (diferença dos custos dos circuitos que entraram e circuitos que saíram) e somente após verificar o seu valor negativo (o que acarreta uma diminuição no valor da função objetivo), foi verificada a operação do sistema para a configuração candidata (resolvendo o PL (4.25)-(4.32), ou o PL (4.33)-(4.41)).

No caso em que o sistema deixa de operar adequadamente, ou seja, quando a troca simulada de circuitos indicar que se trata de uma configuração infactível (ocasiona corte de carga no sistema), a simulação deve ser cancelada e passa-se à próxima tentativa. Se a solução do PL indicar que a solução candidata é factível, ou seja, $w_k = 0$, então, a configuração é armazenada a fim de se escolher a melhor configuração factível após avaliar todos os vizinhos candidatos para se atualizar a topologia corrente e a incumbente.

4.6 Resultados Obtidos

O algoritmo proposto foi implementado utilizando-se a linguagem de programação FORTRAN e o MINOS 5.0, (MURTAGH; SAUNDERS, 1995), como uma sub-rotina de resolução de PL.

Apresentam-se os resultados para os sistemas IEEE de 24 barras e para o sistema Sul Brasileiro de 46 barras, considerando-se os modelos de transportes e DC, e planejamentos em que se pode ou não reprogramar a geração. No caso em que é permitida a reprogramação da geração, a geração em cada barra pode variar de zero até o valor máximo. Portanto, os níveis de geração são determinados pelo processo de otimização. Quando não é permitida a reprogramação da geração a soma das gerações nas barras de geração é igual a demanda total do sistema. Assim, nas configurações factíveis, a soma da geração é igual a soma de demandas. Em geral o planejamento SR é mais difícil de ser resolvido do que o planejamento CR.

4.6.1 Sistema IEEE de 24 barras

O Sistema IEEE de 24 barras e 41 circuitos possui uma demanda 8.550 MW e capacidade de geração de 10.215 MW. Os dados das barras e linhas deste sistema se encontram nas Tabelas A.2 e A.1 do Apêndice A.

O planejamento da expansão deste sistema pode ser feita com ou sem reprogramação da geração. Apresentam-se os resultados para dois planos de expansão sem reprogramação da geração:

Planos P1 e P2 da Tabela A.1 do Apêndice A.

4.6.1.1 Modelo de Transportes

Utilizando-se três estruturas de vizinhanças, realizando os seguintes movimentos: N_1 , N_2 , N_1 , N_2 , N_1 , N_2 , N_1 , N_2 , N_3 , N_1 , N_2 e N_3 e resolvendo-se 2.859 PL's (considerando-se todas as linhas como candidatas à entrada e/ou adição) alcança-se a melhor solução conhecida para o plano P1 e o modelo de transportes, conforme (MIASAKI, 2006), a qual aparece na Tabela 4.1. Quando se considera as técnicas de redução de vizinhança, o número de PL's cai para 430.

Tabela 4.1: VND: IEEE 24 barras - Modelo de Transportes.

Plano	Investimento (US\$)	Circuitos Adicionados	No. de PL's Sem redução	No. de PL's Com redução
P1	310,000.00	$n_{03-24} = 1$ $n_{06-10} = 1$ $n_{07-08} = 1$ $n_{10-12} = 1$ $n_{14-16} = 1$ $n_{16-17} = 2$ $n_{16-19} = 1$ $n_{17-18} = 1$	2.859	430
P2	188,000.00	$n_{06-10} = 1$ $n_{07-08} = 2$ $n_{10-12} = 1$ $n_{14-16} = 1$ $n_{16-17} = 1$	141	74

Para se alcançar a melhor solução conhecida para o plano P2, conforme (MIASAKI, 2006), foram necessárias apenas duas estruturas de vizinhanças e a solução de 141 PL's, a qual aparece na Tabela 4.1, realizando-se as seguintes trocas de estruturas de vizinhança: N_1 , N_2 . Com redução nas vizinhanças, foram necessárias as soluções de 74 PL's.

Em (MIASAKI, 2006) foi desenvolvido um Algoritmo Genético especializado para resolver o sistema IEEE de 24 barras, mas que considera a utilização de transformadores defasadores e/ou controladores FACTS, o qual necessita a solução de mais de 15.000 PL's para os dois planejamentos aqui considerados. Vale observar que naquele trabalho é resolvido um problema com modelagem diferente da apresentada aqui.

4.6.1.2 Modelo DC

Considerando-se o modelo DC e o plano P1, para alcançar a melhor solução conhecida (vide (MIASAKI, 2006)), foram necessárias três estruturas de vizinhanças (N_1 , N_2 , N_1 , N_2 , N_1 , N_2 , N_3) e a solução de 4.779 PL's ao se considerar todas as linhas como candidatas à troca, e 1.776 PL's ao se considerar as técnicas de redução da seção 4.5. Os resultados são apresentados na Tabela 4.2.

Para o plano P2, o algoritmo encontra uma configuração melhor do que a apresentada em (MIASAKI, 2006), resolvendo 361 PL's em duas estruturas de vizinhanças, (N_1 , N_1 , N_2 , N_1 , N_2)

considerando-se todas as linhas como candidatas à troca. Ao se aplicar as técnicas de redução da seção 4.5 o algoritmo não convergiu.

Novamente o algoritmo aqui apresentado exigiu um número de PL's muito menor do que o algoritmo proposto em (MIASAKI, 2006).

Em (FARIA, 2005), este problema foi resolvido para o planejamento P2, utilizando-se um algoritmo híbrido de *Path Relinking* com GRASP, também atingindo as soluções ótimas, mas o autor não apresenta o número de PL's utilizados, e sim, o tempo médio de CPU.

Tabela 4.2: VND: IEEE 24 barras - Modelo DC.

Plano	Investimento (US\$)	Circuitos Adicionados	No. de PL's Sem redução	No. de PL's Com redução
P1	392,000.00	$n_{01-05} = 1$ $n_{03-24} = 1$ $n_{06-10} = 1$ $n_{07-08} = 1$ $n_{10-12} = 1$ $n_{14-16} = 1$ $n_{15-24} = 1$ $n_{16-17} = 2$ $n_{17-18} = 2$	4.779	1.776
P2	342,000.00	$n_{03-24} = 1$ $n_{06-10} = 1$ $n_{07-08} = 2$ $n_{09-11} = 1$ $n_{10-12} = 1$ $n_{14-16} = 2$ $n_{16-17} = 1$	361	-

4.6.2 Sistema Sul Brasileiro

O sistema Sul Brasileiro é constituído por 46 barras (duas delas são novas unidades geradoras que devem ser conectadas à rede, barras 28 e 31), 62 caminhos existentes e 17 novos caminhos com uma demanda de 6.880 MW. Assim, existem 79 variáveis inteiras. Pode-se adicionar até três circuitos por caminho, isto é, $n_{ij} = 3$ para $(i, j) \in \Omega$.

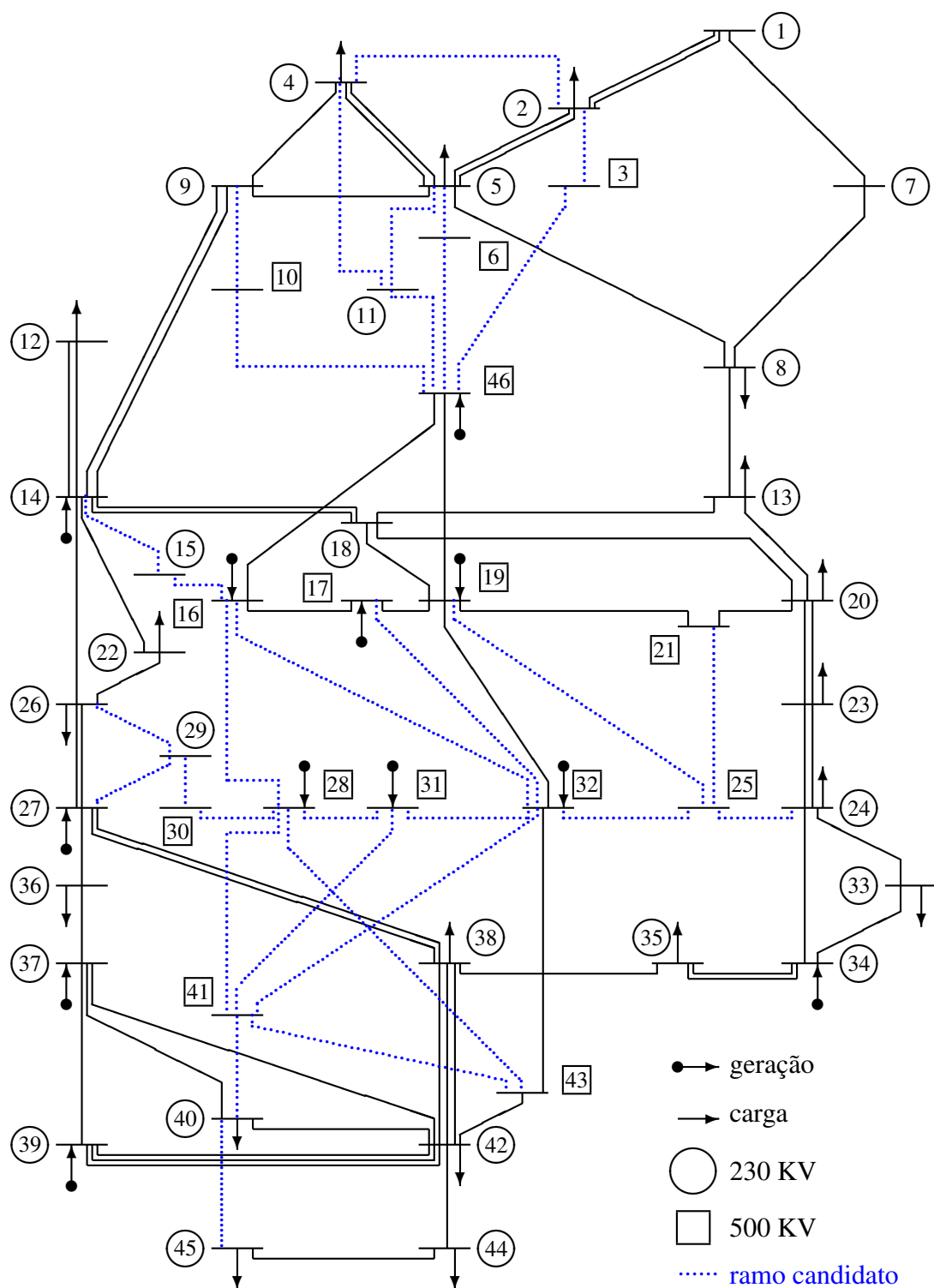


Figura 4.19: Sistema Sul Brasileiro - Rede Inicial.

A Figura 4.19 fornece uma idéia da topologia base deste sistema de potência e ilustra os circuitos existentes (linhas sólidas) e candidatos. Os detalhes dos dados deste sistema aparecem no

Apêndice A.

Formulando-se este problema utilizando o modelo (2.1)-(2.8), ter-se-iam 79 variáveis inteiras, 137 variáveis contínuas e 383 restrições, excluindo os limites nas variáveis. Este sistema é considerado um sistema de médio porte de complexidade mediana (HAFFNER et al., 2001; ROMERO et al., 2002).

Apresentam-se resultados para os modelos de transportes e DC, cuja execução do planejamento pode ser feita com (CR), ou sem (SR) reprogramação da geração.

4.6.2.1 Modelo de Transportes

No planejamento CR, com quatro estruturas de vizinhança e resolvendo 19.623 PL's, o algoritmo atinge a solução ótima global, conforme Tabela 4.3, realizando as seguintes trocas de estruturas de vizinhança: $N_1, N_2, N_3, N_1, N_2, N_3$ e N_4 , com um investimento $v = \text{US\$ } 53,334,000.00$.

Tabela 4.3: VND: Sul Brasileiro - Modelo de Transportes.

Plano	Investimento (US\$)	Circuitos Adicionados	No. de PL's Sem redução	No. de PL's Com redução
CR	53,334,000.00	$n_{20-21} = 2$ $n_{33-34} = 1$ $n_{42-43} = 1$ $n_{05-11} = 2$ $n_{46-11} = 1$	19.623	2.496
SR	127,272,000.00	$n_{14-16} = 1$ $n_{18-19} = 1$ $n_{20-21} = 1$ $n_{42-43} = 2$ $n_{05-11} = 2$ $n_{25-32} = 1$ $n_{31-32} = 1$ $n_{28-31} = 1$ $n_{46-11} = 1$ $n_{24-25} = 2$	940.135	155.920

Já no planejamento SR, para se alcançar o ótimo global de investimento $v = \text{US\$ } 127,272,000.00$, foram necessárias seis estruturas de vizinhança, nas quais se resolveram 940.135 PL's considerando-se a seguinte ordem de trocas de estruturas de vizinhanças: $N_1, N_1, N_2, N_1, N_2, N_3, N_4, N_1, N_1, N_2, N_3, N_4, N_1, N_2, N_3, N_4, N_5$ e N_6 .

Ao se considerar as técnicas de redução de vizinhança apontadas na seção 4.5, o número de PL's resolvidos diminuem consideravelmente nos dois casos considerados: 2.496 e 155.920 PL's, respectivamente, para alcançar as soluções ótimas globais.

Vale ressaltar que o número de PL's no planejamento SR é muito alto pelo fato de se estar caminhando somente por soluções factíveis. Para sair da solução ótima local n_{loc} : $n_{20-21} = 1$, $n_{42-43} = 1$, $n_{05-11} = 2$, $n_{25-32} = 1$, $n_{31-32} = 1$, $n_{28-31} = 1$, $n_{28-30} = 1$, $n_{26-29} = 2$, $n_{46-11} = 1$, $n_{24-25} = 2$, $n_{29-30} = 1$, deve-se retirar os circuitos $n_{28-30} = 1$, $n_{26-29} = 2$ e $n_{29-30} = 1$, e adicionar os circuitos $n_{14-16} = 1$, $n_{18-19} = 1$ e $n_{42-43} = 1$, que para serem determinados, exigiu-se a análise

de mais de 145.000 soluções candidatas, ou seja, a solução de mais de 145.000 PL's, já que não há nenhuma outra solução factível com função objetivo entre $v = \text{US\$ } 129,000,000.00$ e $v = \text{US\$ } 127,272,000.00$, com menos de seis caminhos distintos, ou seja, somente em N_6 é que é possível alcançá-la.

Assim, a busca realizada em N_5 foi irrelevante, e se fosse desprezada, redefinindo N_5 como sendo o conjunto das soluções que diferem de n_{loc} em seis caminhos, em vez de cinco, seria possível se determinar a solução ótima com um número de PL's bem menor.

4.6.2.2 Modelo DC

Quando se considerou o modelo DC e a mesma definição de estruturas de vizinhança, o algoritmo VND conseguiu determinar a solução ótima global para o planejamento CR e uma solução de excelente qualidade para o planejamento SR, conforme Tabela 4.4.

Tabela 4.4: VND: Sul Brasileiro - Modelo DC.

Plano	Investimento (US\$)	Adições	No. de PL's Sem redução	No. de PL's Com redução
CR Ótimo global	72,870,000.00	$n_{02-05} = 1, n_{13-20} = 1$ $n_{20-23} = 1, n_{20-21} = 2$ $n_{42-43} = 1, n_{46-06} = 2$ $n_{05-06} = 2$	2.263	497
SR Ótimo local	163,171,000.00	$n_{20-21} = 1, n_{46-06} = 1$ $n_{19-25} = 1, n_{42-43} = 1$ $n_{46-06} = 1, n_{25-32} = 1$ $n_{31-32} = 1, n_{28-31} = 1$ $n_{31-41} = 1, n_{24-25} = 2$ $n_{40-41} = 1, n_{05-06} = 2$	176	-
SR Ótimo global	154,420,000.00	$n_{20-21} = 1, n_{42-43} = 2$ $n_{46-06} = 1, n_{19-25} = 1$ $n_{31-32} = 1, n_{28-30} = 1$ $n_{26-29} = 3, n_{24-25} = 2$ $n_{29-30} = 2, n_{05-06} = 2$	-	-

Para o planejamento CR, foram necessários resolver 2.263 PL's em 3 estruturas de vizinhanças, seguindo a seguinte seqüência de mudanças de estruturas de vizinhança: N_1, N_2, N_1, N_2 e N_3 . Quando se considera as técnicas de redução de vizinhança, houve uma diminuição de 78% no número de PL's resolvidos, ou seja, com apenas 497 PL's o algoritmo atingiu a solução ótima global.

A solução ótima, para o caso do planejamento sem reprogramação da geração, foi publicada, pela primeira vez, com o melhor limite superior conhecido em (ROMERO; MONTICELLI, 1994a), e a prova da otimalidade foi publicada em (BINATO; PEREIRA; GRANVILLE, 2001). Seu custo de investimento é de US\$154,420,000.00, o que corresponde à adição de 16 circuitos candidatos, os

quais aparecem na Tabela 4.4. Em (GALLEGO; MONTICELLI; ROMERO, 1998a), aparecem o número de PL's necessários para se alcançar a solução ótima para este caso, ao se utilizar as metaheurísticas *Simulated Annealing*, Algoritmo Genético e *Tabu Search*.

O algoritmo proposto para este caso, determinou uma solução de investimento $v_{loc} = \text{US\$ } 163,171,000.00$ em N_2 , resolvendo 176 PL's . Continuou-se as buscas locais em N_3 , N_4 , N_5 e N_6 , mas não foi encontrada nenhuma solução factível de melhor qualidade. Sendo assim, com a definição de vizinhança 4.1, não é possível se chegar a solução ótima para o modelo DC e o planejamento sem reprogramação da geração, já que existem 10 caminhos distintos entre n_{loc} e n^* .

Então, utilizando n_{loc} como solução inicial, e definindo:

$N'_1(n)$: troca de sete caminhos;

$N'_2(n)$: troca de oito caminhos;

$N'_3(n)$: troca de nove caminhos;

$N'_4(n)$: troca de dez caminhos,

pode-se alcançar a solução ótima, mas, evidentemente, com um esforço computacional muito maior.

Na Figura 4.20 ilustra-se a configuração ótima obtida para o modelo DC e o planejamento CR, em que as linhas sólidas representam os circuitos existentes na configuração base. A Figura 4.21 representa a configuração ótima para o modelo DC no planejamento SR.

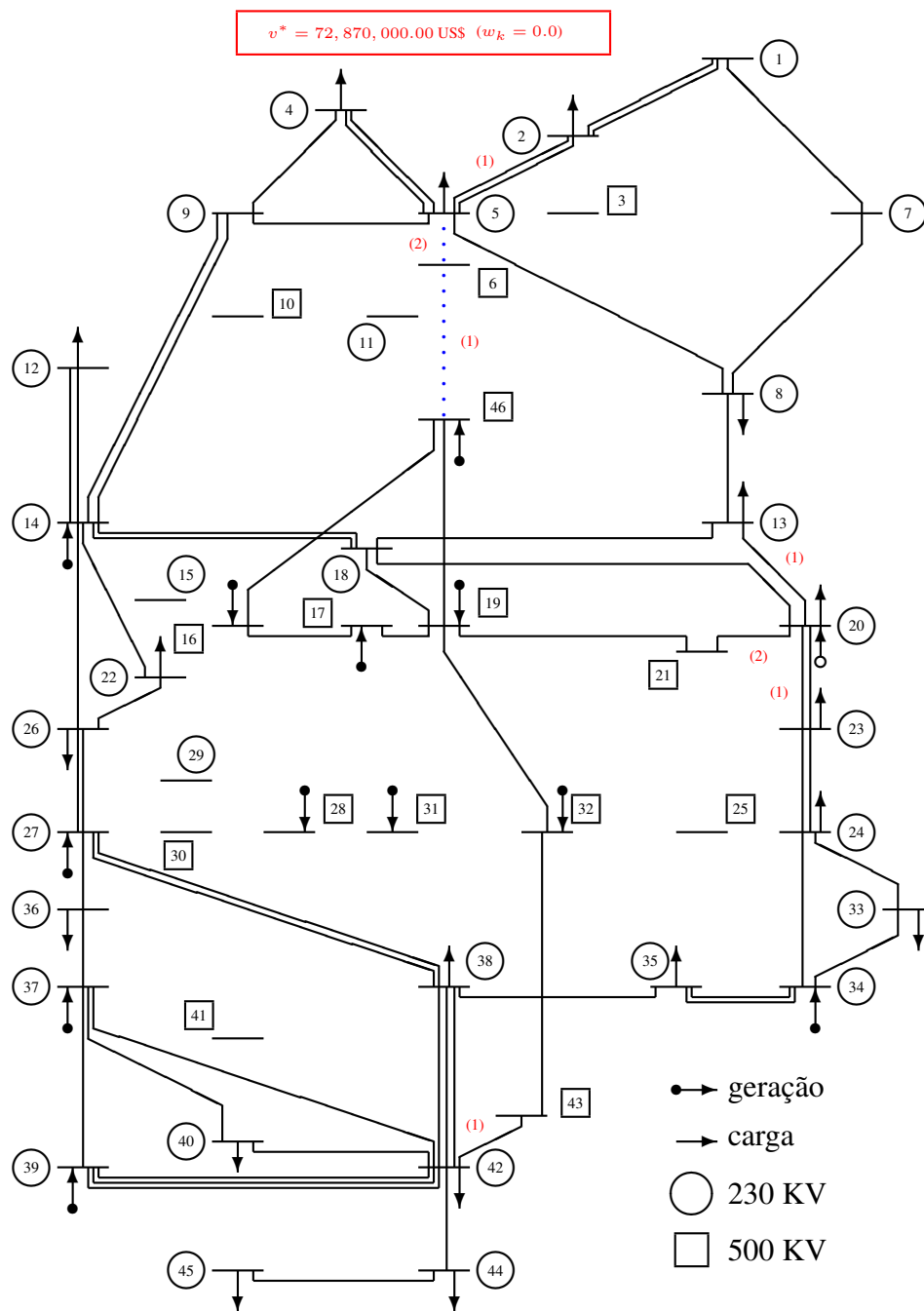


Figura 4.20: Sistema Sul Brasileiro - Configuração Ótima DC com Reprogramação.

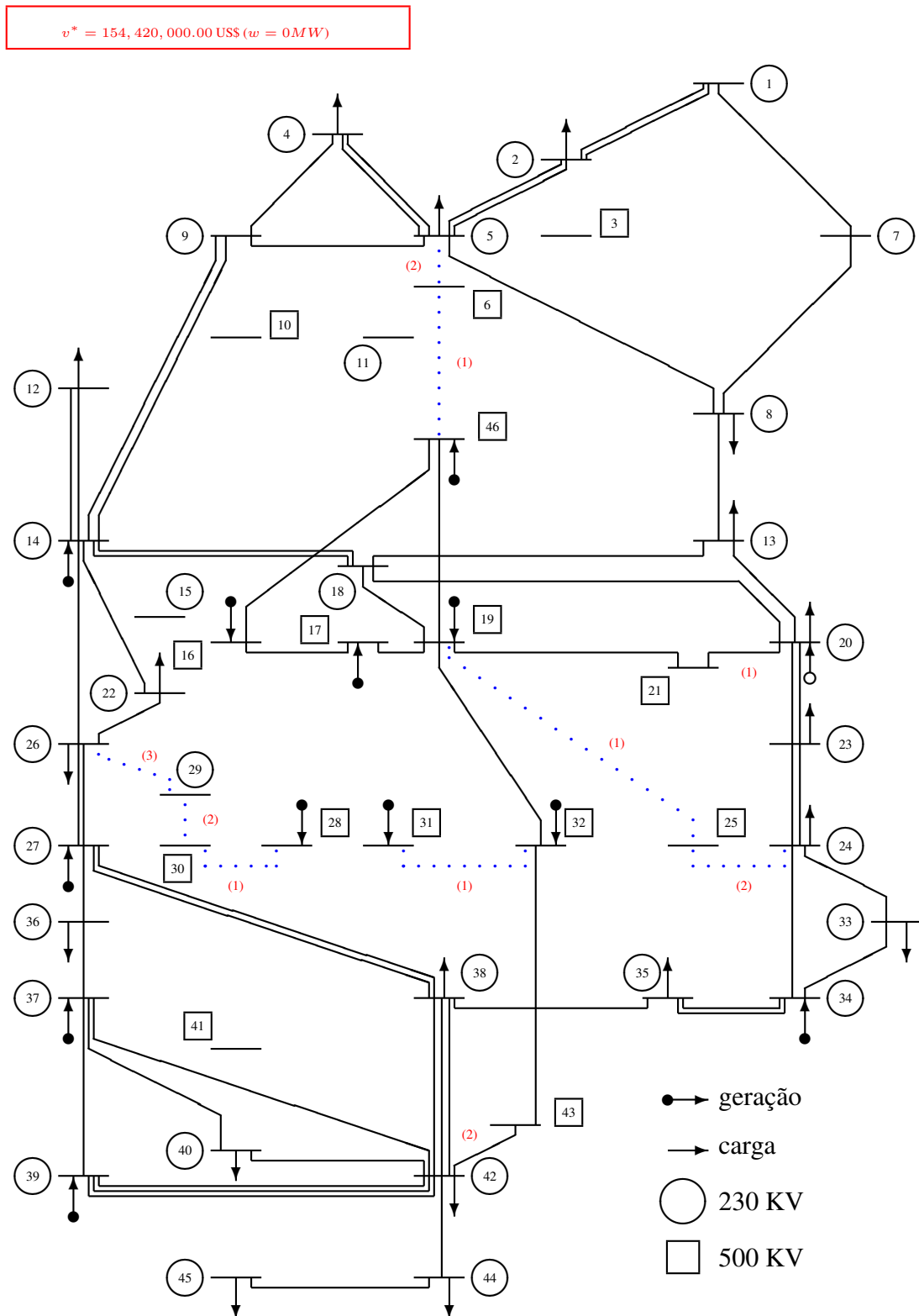


Figura 4.21: Sistema Sul Brasileiro - Configuração Ótima DC sem Reprogramação.

4.7 VNS entre as Metaheurísticas

As heurísticas tradicionais em otimização realizam buscas locais de descida, e por isso são bloqueadas no primeiro ótimo local encontrado. As metaheurísticas proporcionam métodos, deter-

minísticos ou aleatórios, para escapar de ótimos locais de baixa qualidade. Como o valor de tais ótimos locais frequentemente diferem consideravelmente do valor do ótimo global, especialmente se existem muitos ótimos locais (problemas multimodais), o impacto das metaheurísticas tem sido imenso.

Em contrapartida, a teoria das metaheurísticas está mais atrasada. Frequentemente são desenvolvidas boas heurísticas, com algumas inovações e vários parâmetros, mas as razões pelas quais funcionam tão bem são desconhecidas. A situação é ainda mais grave para as metaheurísticas híbridas: é difícil dizer se o mérito é de um dos componentes, ou se está obtendo benefício da interação.

Para compreender melhor estas questões, em (HANSEN et al., 2000) foram utilizadas três idéias para comparar outras metaheurísticas de busca com a VNS: o estudo da topografia dos ótimos locais; das trajetórias seguidas pelos processos de busca e as fases de melhoria ou deterioração das soluções.

O estudo da topografia dos mínimos locais pode descrever o perfil das montanhas ou vales encontrados durante a busca. Quando se aplica a VNS, a descida, a partir de uma solução x' selecionada aleatoriamente, pode retornar ao ótimo local x , ao redor dos quais estão centradas as vizinhanças atuais, ou encontrar outro ótimo local x'' , cujo valor pode ser melhor ou não que o de x . Foi estudado em (HANSEN et al., 2000) a probabilidade destes três eventos como uma função da distância de x a x' aplicadas ao problema da satisfação máxima ponderada. Convém também observar que se x'' está mais perto de x que x' (o que pode ser interpretado que x'' pertence ao mesmo vale que x) ou não (o que pode ser interpretado como um indício de que foi encontrado um grande vale). O relevo indica esta informação com base em uma determinada quantidade de descidas a partir de pontos em vizinhanças sucessivamente mais amplas. Os perfis variam consideravelmente com a qualidade do ótimo local x . Quando x é uma solução ruim é suficiente se afastar um pouco para se obter, com alta probabilidade, um ótimo local melhor. Quando o ótimo local x é de boa qualidade, deve-se afastar bastante para encontrar um novo grande vale e, além disso, a probabilidade de encontrar uma solução melhor que a atual é então baixa. Isto ilustra uma deficiência do esquema básico da VNS que tende a degenerar em uma heurística de partidas múltiplas quando a distância de x a x' é muito grande. A sugestão para contornar este ponto, é aplicar a VNS inclinada: SVNS.

Para alguns problemas são disponíveis instâncias em que se conhecem o ótimo global. Para estes casos, é correto analisar a busca heurística para determinar a frequência que se alcança o ótimo global ou a média da diferença relativa entre o valor ótimo do objetivo e ao alcançado pela busca. Certamente, assim se obtém muito mais informações ao se considerar as próprias soluções e não apenas os seus valores. Uma ferramenta para por em prática esta idéia foi desenvolvida para aplicar uma VNS no problema do vendedor (GENDREAU et al., 2001).

Além de utilizar ferramentas de visualização para guiar a busca, pode-se também avaliar o tra-

balho realizado passo a passo por uma VNS ou outra metaheurística. Podem ser estudadas variantes em cada um dos passos e guardar informações detalhadas para se obter conclusões que podem não ser evidentes na perspectiva global da heurística. Isto pode dar lugar a fenômenos inesperados, e a profundidade proporcionada por sua explicação, pode por sua vez, dar lugar a princípios para melhorar a heurística. Em (HANSEN; MLADENOVIC, 1999) foi observado que selecionando a **primeira melhoria** (ou seja, o primeiro movimento que reduz o valor da função objetivo), se obtém melhores resultados do que selecionando a **melhor melhoria** (ou seja, o movimento que mais reduz o valor da função objetivo). O estudo sistemático deste fenômeno e sua explicação poderia dar lugar a muitas melhorias em uma grande variedade de metaheurísticas.

É conveniente realizar algumas reflexões sobre as propriedades desejáveis das metaheurísticas. Estas boas qualidades são as que propiciam ou garantem interesse prático e teórico das propostas. Em (HANSEN; MLADENOVIC, 2003b) se analisa a medida em que a VNS em relação a outras metaheurísticas se ajustam a tais propriedades. Na lista de tais propriedades incluem-se as seguintes:

- i. Simplicidade:** a metaheurística deve estar baseada num princípio simples e claro;
- ii. Precisão:** os passos da metaheurística devem ser formulados em termos precisos;
- iii. Coerência:** as operações do algoritmo devem se derivar naturalmente dos princípios da metaheurística;
- iv. Eficácia:** os procedimentos devem proporcionar soluções ótimas ou quase ótimas para a maioria dos problemas reais;
- v. Eficiência:** os algoritmos devem empregar um esforço computacional moderado para determinar soluções ótimas;
- vi. Robustez:** o desempenho da metaheurística deve ser de qualidade garantida em vários exemplos e em várias instâncias;
- vii. Facilidade:** deve se expressar claramente, ser fácil de entender e de usar;
- viii. Inovação:** os princípios das metaheurísticas, e/ou eficiência e eficácia das heurísticas derivadas delas, devem levar a novas aplicações.

A seguir examina-se brevemente como a VNS se ajusta a cada uma das propriedades acima e comenta-se as vantagens que ela apresenta.

A VNS está baseada em um princípio simples e relativamente não explorado: a troca sistemática de estruturas de vizinhança durante a busca. A **simplicidade** a torna muito aplicável, como se reflete nas inúmeras aplicações já comentadas. As regras **precisas** na forma de efetuar as trocas

de vizinhanças são cruciais. Todos os esquemas básicos e extensões da VNS se traduzem **coerentemente** dos princípios em que se inspira.

Para validar a **eficácia** de uma metaheurística, estas deveriam encontrar soluções ótimas para a maioria dos problemas de um banco de casos para os quais se conheçam as soluções. A VNS tem mostrado bastante eficácia na resolução de vários bancos de prova com resultados ótimos ou muito próximos dos ótimos conhecidos, e com tempo computacional moderado.

A metaheurística VNS também tem se mostrado muito **eficiente** em muitos experimentos, já que tem obtido tão bons ou melhores resultados que a maioria das metaheurísticas em muitos problemas e de uma forma muito mais rápida. Além disso, a VNS se mostra **robusta** já que tem provado seu desempenho em uma grande variedade de problemas.

A **facilidade** dos algoritmos baseados em VNS contam com a vantagem de que os princípios básicos em que se fundamentam são fáceis de se aplicar, e muito fáceis de usar. Esta qualidade se traduz no fato de que nas aplicações realizadas, o número de parâmetros se mantém a um mínimo ou não aparecem.

O potencial da VNS quanto à **inovação** já foi ilustrado em vários tipos de novas investigações: (ANDREATTA; RIBEIRO, 2002; BRINBERG et al., 2000; GLOVER, 1999; CAPOROSSI et al., 1999; CAPOROSSI; HANSEN, 2000; HANSEN; MLADENOVIC, 2001c; RIBEIRO; SOUZA, 2002).

A busca através de vizinhança variável, é portanto, uma metaheurística recente com o mérito de se ajustar às qualidades da lista anterior. Isto, juntamente com a frequência com a qual a metaheurística incorpora as idéias em que se baseia a VNS justifica o papel destacado da VNS no campo das metaheurísticas.

Ainda que a VNS e outras metaheurísticas têm se mostrado claramente úteis na resolução aproximada de muitos problemas difíceis, é muito custoso resolver problemas de grande dimensão.

O tamanho dos problemas abordados se limitam na prática pelas ferramentas disponíveis para resolvê-lo, mais que pelo potencial destas ferramentas. Quando as metaheurísticas são aplicadas a instâncias realmente grandes, suas forças e debilidades aparecem mais claramente.

As melhoras neste sentido são então altamente desejáveis, tanto para as metaheurísticas em geral, como para a VNS em particular.

No Capítulo seguinte, apresenta-se a Metaheurística Busca Dispersa.

5 *Busca Dispersa*

O segundo objetivo deste trabalho, a aplicação inédita da Metaheurística Busca Dispersa na solução do PPEST, é apresentado neste capítulo.

A metaheurística Busca Dispersa (BD), do inglês *Scatter Search*, foi introduzida em 1977 por Fred Glover (GLOVER, 1977), como uma heurística para programação inteira e foi baseada em estratégias apresentadas na *Management Science and Engineering Conference* em Austin, Texas, em Setembro de 1967 (LAGUNA; ARMENTANO, 2005).

A BD opera em um conjunto de soluções, chamado conjunto de referência, combinando as soluções para gerar outras soluções novas, de modo que estas novas soluções sejam melhores que as soluções que as originaram. A combinação de estratégias foi criada com a crença de que melhores informações podem ser encontradas quando são tratadas de forma integrada, do que quando tratadas isoladamente. Em contraste com outros algoritmos evolutivos como os algoritmos genéticos, a BD é principalmente baseada em métodos sistemáticos com o propósito de criar novas soluções. A BD utiliza estratégias de intensificação e diversificação na busca. Sua estrutura é flexível, permitindo o desenvolvimento de implementações alternativas em vários graus de sofisticação.

Neste capítulo apresentam-se as idéias essenciais da busca dispersa, sua evolução e principais aplicações. Na seção 5.1 será abordado o desenvolvimento da técnica. A descrição dos elementos da BD será feita na seção 5.2. As estratégias avançadas serão tratados na seção 5.3. E na seção 5.4 serão detalhados os passos da BD aplicada na solução do PPEST.

5.1 **Desenvolvimento da Busca Dispersa**

Apresenta-se aqui uma breve versão da BD descrita em (GLOVER, 1977), a qual foi a primeira publicação desta metodologia. A BD utiliza uma sucessão de soluções iniciais coordenadas para gerar outras soluções. As soluções são sistematicamente (e não aleatoriamente) geradas levando em conta características contidas em várias regiões do espaço de busca. A BD opera sobre um conjunto de soluções, chamado conjunto de referência, combinando estas soluções para gerar novas soluções de modo que melhorem as que as originaram. Neste sentido, diz-se que é um método evolutivo. Mas, difere de outros métodos evolutivos, como é o caso dos Algoritmos Genéticos, em pontos

fundamentais, pois a BD não está fundamentada na combinação aleatória em um grande conjunto de soluções, mas sim em seleções sistemáticas e estratégicas em um conjunto pequeno. Como ilustração, basta dizer que os algoritmos genéticos usualmente consideram uma população inicial de 100 soluções, enquanto que em BD é habitual se trabalhar com um conjunto equivalente de somente 10 soluções. A BD se baseia no princípio de que as informações sobre a qualidade, ou sobre um conjunto de regras, restrições ou soluções podem ser melhores utilizadas mediante a combinação conjunta destas informações, do que utilizando informações isoladas.

Uma das características mais notáveis da BD se baseia em integrar a combinação de soluções com uma busca local. Ainda que em estratégias avançadas esta busca local possa conter uma estrutura de memória (*tabu search*), não é necessário que assim seja, limitando-se, na maioria dos casos, em uma busca local convencional.

Como foi mencionado, a primeira descrição do método foi publicada por (GLOVER, 1977), onde estabeleceu-se seus princípios. Neste primeiro artigo se determina que a BD realize uma exploração sistemática sobre uma série de boas soluções chamadas conjunto de referência. Os seguintes comentários resumem os principais aspectos deste trabalho:

- O método se baseia em combinar duas ou mais soluções do conjunto de referência. A combinação de mais de duas soluções tem como objetivo gerar melhores direções de busca.
- Gerar soluções através de combinações lineares de suas soluções pode ser considerado uma forma reduzida do método.
- Ao combinar soluções devem ser selecionados pesos apropriados e não escolhidos à sorte.
- Deve se realizar combinações convexas e não-convexas das soluções.
- A distribuição dos pontos é considerada importante, e devem ser tomados dispersos.

Do nosso conhecimento, a Busca Dispersa não foi novamente aplicada ou discutida até 1990, quando foi apresentada no *Seminar on Operations Research and Artificial Intelligence Search Methods* (Lausanne, Suíça).

Um artigo baseado nesta apresentação foi publicado depois em (GLOVER, 1994), no qual é introduzido a combinação ponderada como mecanismo principal para gerar novas soluções. Nesta versão se enfatiza as buscas lineares entre duas soluções e uso de pesos para apontar direções de qualidade. Introduce o conceito de combinar soluções de qualidade com soluções de diversidade. Além disso, o método inclui uma componente de intensificação que consiste em tomar uma mostra maior de soluções na vizinhança que tiver produzido melhores soluções. Neste artigo, o autor menciona que para trabalhar com problemas de variáveis inteiras, ou que formam uma permutação, é

necessário criar métodos específicos de combinação. Ainda introduz o conceito de combinação baseada em voto, onde se devem definir regras mediante as quais cada solução 'vota' para que suas características apareçam na solução que está sendo construída. Este método dos votos tem sido freqüentemente utilizado nas rotinas de combinação dos algoritmos de BD e constituem uma das chaves do êxito do método.

Em (GLOVER, 1998) publica uma versão mais específica do método, onde se esclarecem e simplificam muitas das idéias expostas em trabalhos anteriores. Esta publicação teve grande impacto no ambiente científico e como tal é referência padrão da Busca Dispersa. Vários pesquisadores passaram então a aplicar o método na solução de inúmeros problemas, obtendo resultados de excelente qualidade (LAGUNA, 1999).

Em (GLOVER; LAGUNA; MARTI, 2000) são apresentadas as aplicações mais recentes do método na resolução de problemas de otimização combinatória. Além disso, é realizada uma conexão deste método com o método de religamento de trajetórias (*Path Relinking*, PR). Assim, do ponto de vista espacial, o processo de gerar combinações lineares de um conjunto de referência de soluções, pode ser visto como a geração de caminhos entre estas soluções. Isto leva a uma concepção mais ampla do significado de combinar soluções do que a introduzida em BD. PR se baseia no fato de que entre duas soluções se pode traçar um caminho que as une, de modo que as soluções no caminho produzido contenham atributos delas. As soluções iniciais podem ter sido geradas mediante um método baseado em busca local e estarem unidas por um caminho, ou terem sido geradas por outro método e não estarem unidas de nenhum modo; em qualquer caso, PR gera um novo caminho que as une. As características do caminho produzido são especificadas por atributos que são adicionados ou eliminados, ou pelos movimentos realizados para alcançar uma solução a partir de outra. Isto constitui uma extensão do conceito de combinação, podendo se obter várias soluções a partir de duas ou mais originais.

Em (CAMPOS et al., 2001) aparece uma implementação da BD orientada para encontrar soluções de alta qualidade para problemas de ordenação linear.

Em 2003 é publicado, por Laguna e Marti, o primeiro livro sobre este método (LAGUNA; MARTI, 2003). Os autores realizam um estudo exaustivo de BD, iniciando por tutoriais básicos, revisando estratégias avançadas e chegando até as linhas de pesquisas atuais. O texto acompanha o código fonte em C para aplicar as técnicas descritas.

Em (GLOVER; LAGUNA; MARTI, 2004) aparecem novas idéias e aplicações de BD e PR, e é discutido o desempenho do método em recentes aplicações, como *Neural Network Training*, *Multi-Objective Routing Problem*, *OpQuest: A Commercial Implementation*, *A Context-Independent Method for Permutation Problems*, *Classical Vehicle Routing*, *Matrix Bandwidth Minimization* e *Arc Crossing Minimization*.

Em (LAGUNA; ARMENTANO, 2005) são apresentadas treze lições para aplicar a BD em problemas de otimização combinatória e de otimização não-linear que podem beneficiar futuras aplicações.

(MARTI; LAGUNA; CAMPOS, 2005) realizam uma comparação do desempenho da implementação do método BD e do Algoritmo Genético no contexto de busca de soluções ótimas para problemas de permutação e observam que, usando as mesmas instâncias e problemas, a técnica BD encontrou soluções de qualidade média superior às obtidas pelo Algoritmo Genético.

A Busca Dispersa tem demonstrado que é muito efetiva na solução de problemas de otimização como, por exemplo, o Problema Quadrático de Alocação (CUNG; MICHELON; TAVARES, 1997), Ordenação Linear (MARTI; CAMPOS, 1999), Coloração de Grafos (HAMIEZ; HAO, 2002), Problema das p -medianas (GARCIA et al., 2003), Problema da Mochila (MARTI; LAGUNA, 2003) e Roteamento de Veículos, (SOSA; GALVÃO, 2005).

Na próxima seção apresenta-se a descrição da Busca Dispersa.

5.2 Descrição da Busca Dispersa

A Busca Dispersa tem por base combinar soluções que aparecem em um conjunto denominado conjunto de referência. Este conjunto guarda "boas" soluções encontradas durante o processo de busca. Vale ressaltar aqui que o significado de "boa" não se restringe apenas à qualidade da solução, mas também a sua diversidade em relação a outras soluções do conjunto de referência. A BD é composta basicamente, por cinco etapas as quais serão descritas a seguir de forma resumida. Posteriormente será explicada, em detalhes, cada etapa no contexto de uma heurística desenvolvida visando determinar a solução do PPEST.

- **Gerador de Soluções Diversas:** esta etapa gera um conjunto P com $Psize$ soluções, de onde extrae-se um subconjunto menor que se denomina conjunto de referência $RefSet$. O processo para gerar $RefSet$ pode ser totalmente determinístico ou parcialmente aleatório.
- **Etapas de Melhoria:** tipicamente consiste de uma busca local para melhorar as $Psize$ soluções inicialmente encontradas. Se a solução que está sendo avaliada é infactível, o método tenta transformá-la em uma solução factível. Se a solução é factível, o método tenta melhorá-la pela mudança dos valores de algumas de suas variáveis, utilizando um procedimento guloso em relação à função objetivo.
- **Geração do Conjunto de Referência:** é um método para construir e atualizar o conjunto de referência $RefSet$.

Construção - A partir do conjunto P se extrai $RefSet$, combinando soluções de alta qualidade com soluções com diversidade; as soluções de $RefSet$ são utilizadas, durante o

processo de busca, para gerar novas soluções mediante a combinação de soluções contidas em *RefSet*.

Atualização - A atualização de *RefSet* acontece quando novos elementos (soluções) que cumprem os requisitos para ingressar em *RefSet* forem encontrados. Assim *RefSet* mantém sua dimensão b constante (b é geralmente pequeno, em torno de 20), mas os valores das soluções vão melhorando durante o processo de busca. Em implementações simples, a atualização de *RefSet* se realiza unicamente por qualidade (em relação à função objetivo) e somente depois de se ter realizado todas as combinações possíveis entre as soluções do conjunto de referência (atualização estática).

- **Geração de Subconjuntos:** opera no conjunto de referência, consistindo em criar diferentes subconjuntos de soluções referência que serão utilizadas na etapa de combinação.
- **Combinação de Soluções:** esta etapa utiliza os subconjuntos gerados através do método de geração de subconjuntos para combinar as soluções em cada subconjunto com o objetivo de encontrar novas soluções. Esta etapa de combinação é um mecanismo específico para cada problema, porque está diretamente relacionado com a representação da solução. Os métodos de combinação tipicamente podem ser sistemáticos ou probabilísticos e podem gerar uma ou mais novas soluções.

O algoritmo descrito na Figura 5.1 mostra como agem os elementos descritos em um esquema básico da Busca Dispersa para um problema de minimização.

Passo 1. Inicie com $P \leftarrow \emptyset$. Use a etapa geradora de soluções diversas e construa uma solução y . Aplique o método de melhoria a y e encontre a solução melhorada x . Se $x \notin P$ então, adicione x a P , caso contrário descarte x . Repita este passo até $|P| = Psize$.

Passo 2. Construa o conjunto de referência $RefSet = \{n^1, \dots, n^b\}$ com as $b/2$ melhores soluções de P e as $b/2$ soluções de P com maior diversidade em relação às soluções que estão em $RefSet$. Ordene essas b soluções de forma crescente em relação à função objetivo. Faça SoluçãoNova $\leftarrow$ Verdade.

Enquanto (SoluçãoNova = Verdade) SoluçãoNova = Falso.

Passo 3. Mediante a geração de subconjuntos gere os subconjuntos nos quais exista pelo menos uma nova solução.

Enquanto (existem subconjuntos sem examinar)

Passo 4. Aplique o método de combinação nas soluções do subconjunto e guarde as soluções obtidas em uma lista denominada *Pool*.

Passo 5. Aplique o método de melhoria a cada solução obtida no Passo 4.

Passo 6. Atualize $RefSet$. Seja x a solução melhorada. Se $f(x) < f(n^b)$ e x não está em $RefSet$ então $RefSet = RefSet \cup x$. (b melhores soluções em $RefSet \cup Pool$); Reordene $RefSet$; SoluçãoNova = Verdade.

Fim Enquanto

Fim Enquanto

Figura 5.1: Algoritmo básico da Busca Dispersa.

Note que o algoritmo básico da BD termina quando não existem novos elementos a serem combinados em $RefSet$. Este esquema pode ser estendido e melhorado em diferentes formas. A metodologia BD é muito flexível, já que cada um de seus elementos podem ser implementados em uma variedade de formas e graus de sofisticação. Diversas formas de melhoria podem ser encontradas em (GLOVER, 1998; GLOVER; LAGUNA; MARTI, 1999; GLOVER; LAGUNA; MARTI, 2000; LAGUNA; ARMENTANO, 2005; MARTI; LAGUNA, 2003).

5.3 Estratégias Avançadas

Embora a Busca Dispersa seja um método cujos princípios foram propostos há quase três décadas, suas aplicações e desenvolvimento são recentes. Durante os últimos anos foram realizadas

novas contribuições ao se aplicar esta metodologia na solução de problemas clássicos de otimização. Por exemplo, o problema do roteamento de ônibus escolares com objetivo múltiplo é tratado em 2002 em (CORBERÁN et al., 2002); o problema da mochila é resolvido por Laguna em 1999, em (LAGUNA, 1999), onde se ilustra, em detalhes, todos os passos da BD. Também em (GLOVER, 1994), é feito o desenvolvimento teórico da BD. Estas aplicações abriram novos campos de estudo, oferecendo novas alternativas aos desenhos básicos. Laguna e Martí realizaram em (LAGUNA; MARTI, 2003) uma revisão dos aspectos essenciais do método BD desenvolvidos durante estes anos. A seguir apresentam-se, sucintamente, estes aspectos.

- a. Considerar o **uso de memória** baseada em frequência para desenvolver métodos de diversificação eficientes. Em vez de gerar soluções aleatórias, propõe-se construir um método que, baseado no número de aparições de atributos significativos na solução, evite a repetição de estruturas similares.
- b. A **dimensão ótima** do conjunto de referência está por determinar. É necessário continuar com experimentação para determinar se deve-se manter este número b relativamente pequeno, ou se no caso de problemas em que a BD não tenha proporcionado bons resultados deve-se aumentá-lo. Também deve-se estudar os casos em que b seja variável em função do estado da busca. Iniciar o conjunto de referência a partir de um grande conjunto de soluções criadas com o gerador mencionado acima, proporcionaria um conjunto com maior diversidade que poderia gerar soluções melhores em um horizonte maior.
- c. Aplicar a rotina de **melhoria** de forma **seletiva**. Os resultados indicam que aplicar o método de melhoria a todas as soluções geradas e combinadas não garantem o melhor resultado final. Utilizar limites de qualidade para não aplicar a rotina de melhoria em soluções que dificilmente irão proporcionar a melhor solução, ocasionando um gasto desnecessário de tempo computacional (MARTI; LAGUNA, 2003). Por outro lado, ao aplicar o método de melhora em todas as soluções se acelera a convergência do método, o que pode ser desejável em alguns casos, mas deve-se evitá-lo ao se executar o método em problemas cujo espaço de busca apresenta grande dimensão.
- d. É necessário estudar a porcentagem de tempo que o método está gerando soluções e o tempo em que está combinando soluções. Em essência, esta é a questão que se esboça em muitos métodos heurísticos: O **equilíbrio** entre a **intensificação** e a **diversificação**.
- e. Iniciar o conjunto de referência com metade das soluções escolhidas por qualidade e a outra metade escolhida por diversidade. Foram realizados experimentos com distintos tamanhos e parece que esta proporção foi a que ofereceu melhores resultados.
- f. A qualidade é mais importante que a diversidade ao atualizar o conjunto de referência. Notar que, mesmo que o método inicie com um conjunto de referência com soluções de qualidade

e diversidade, ao realizar as combinações, só entram no conjunto de referência as soluções que satisfazem o critério de qualidade (até chegar na etapa de degenerá-lo). Em (GLOVER; LAGUNA; MARTI, 2000) foram feitas atualizações do conjunto de referência por critério de diversidade, mas os resultados finais obtidos foram piores.

g. Comparar as atualizações do conjunto de referência realizadas de forma estática e dinâmica.

Notar que para atualizar o conjunto de referência pode-se aplicar duas estratégias: introduzir as soluções combinadas no conjunto de referência logo após sua combinação (forma dinâmica), sempre que cumpram com os requisitos necessários para ingressarem no conjunto, ou guardá-las em uma lista e depois que se tiver realizado todas as combinações (forma estática), proceder as atualizações. A estratégia dinâmica é mais agressiva, no sentido de que as boas soluções entram rapidamente no conjunto de referência, no entanto, como o conjunto de referência é um conjunto de tamanho constante, isto implica que existem soluções que saem do conjunto de referência sem terem sido utilizadas para fazer combinações. É necessário então comparar ambas estratégias para ver qual proporciona melhores resultados. Um segundo aspecto refere-se aos critérios de aceitação ou eliminação de uma solução pertencente ao conjunto de referência (só por qualidade, ou por qualidade e diversidade). Aceita-se uma solução por qualidade sempre que a mesma melhore a pior solução em *RefSet* em relação ao valor da função objetivo; aceita-se uma solução por diversidade se esta solução tiver maior diversidade em relação às que estão no conjunto de referência. Assim, a atualização do conjunto de referência será dada de forma: (i) estática por qualidade e diversidade; ou (ii) dinâmica por qualidade; ou (iii) dinâmica por qualidade e diversidade.

h. A maior parte das soluções de qualidade procedem da combinação de duas soluções. Também, as boas soluções costumam ser geradas da combinação de boas soluções. Em ((CAMPOS; LAGUNA; MARTI, 1999) e em (CAMPOS et al., 2001)), foram realizados numerosos experimentos realizando uma busca ao longo do caminho das soluções de qualidade em um problema concreto.

i. O uso de múltiplos métodos de combinação deve ser considerado. (CAMPOS; LAGUNA; MARTI, 2005) fizeram uma análise de diferentes métodos de combinação, alguns com elementos aleatórios e outros determinísticos, de modo que o algoritmo seleciona o método de combinação probabilisticamente, de acordo com os êxitos obtidos por estes. Da mesma forma que os métodos de busca local baseados em trocas sistemáticas de vizinhança (VNS) estão proporcionando muitos bons resultados, tem-se que considerar a definição de várias 'vizinhanças' de combinação para realizar uma busca mais completa no espaço de soluções.

j. Uma alternativa para o esquema básico da BD consiste em não encerrar o algoritmo quando não existem mais soluções a serem combinadas no conjunto de referência. Neste caso, pode-se reconstruir *RefSet*; para isto mantém-se a metade das soluções com melhor valor em

RefSet e a outra metade é substituída por novas soluções que introduzam diversidade no conjunto. Isto pode ser feito criando um novo conjunto *P* e aplicando o mesmo mecanismo utilizado quando *Refset* foi construído. Uma vez reconstruído *RefSet*, aplica-se novamente o método de combinação e atualização de *Refset*.

Outras estratégias avançadas podem ser encontradas em (LAGUNA; MARTI, 2003), incluindo uma revisão das principais aplicações de BD e PR na resolução de problemas de otimização, tais como: *The Graph Coloring Problem, Capacitated Multicommodity Network Design, The Maximum Clique Problem, Assigning Proctor to Exams, Periodic Vehicle Loading, The Linear Ordering Problem, The Bipartite Drawing Problem, Job Shop Scheduling, The Arc Routing Problem, Multiple Criteria Scatter Search, Metaheuristic Use of Scatter Search via OpQuest, Generation of Diverse MIP Solutions e Path Relinking to Improve Iterated Start Procedures.*

5.4 Busca Dispersa aplicada no PPEST

Nesta seção são apresentadas as diferentes etapas da Busca Dispersa aplicada na solução do PPEST, seguindo os passos da Figura 5.2.

-
- Passo 1.** Codificar o problema para aplicar a BD;
 - Passo 2.** Gerar várias soluções diversas;
 - Passo 3.** Melhorar as soluções obtidas;
 - Passo 4.** Construir o conjunto de referência;
 - Passo 4.** Escolher as soluções a serem combinadas;
 - Passo 6.** Aplicar o método de combinação;
 - Passo 7.** Atualizar o conjunto de referência;
 - Passo 8.** Aplicar um critério de parada.
-

Figura 5.2: Principais passos do algoritmo BD para o PPEST.

Para o método gerador de soluções diversas é apresentada uma heurística construtiva utilizando o algoritmo construtivo de Garver, quando se considera o modelo de transporte, ou o algoritmo construtivo de Villasana-Garver-Salon, quando se considera o modelo DC.

A seguir explica-se cada passo da BD no PPEST.

5.4.1 Passo 1: Codificação do PPEST para a BD

Utiliza-se a mesma codificação do capítulo anterior.

5.4.2 Passo 2: Gerador de Soluções Diversas

Para gerar as soluções iniciais, é utilizado um algoritmo heurístico construtivo, que em cada passo escolhe um circuito a ser adicionado ao sistema, considerando-se os modelos de transportes e DC.

5.4.2.1 Soluções para o Modelo de Transportes

Quando se considera o modelo de transportes, aplica-se o algoritmo de Garver descrito na seção 3.2.1, que considera o modelo (5.1)-(5.7) e utiliza o índice de sensibilidade definido como sendo o fluxo de potência pelos circuitos com $n_{ij} \neq 0$ na solução do PL, ou seja, o índice definido em (5.8).

Lembrando que ao relaxar a integralidade das variáveis n_{ij} , no modelo (5.1)-(5.7), em cada passo resolve-se um PL e acrescenta-se um circuito no caminho que permite o maior fluxo de potência, ou seja, no caminho indicado pelo índice (5.8).

$$\min v = \sum_{(i,j) \in \Omega} c_{ij} n_{ij} \quad (5.1)$$

s.a.

$$Sf + g = d \quad (5.2)$$

$$|f_{ij}| \leq (n_{ij}^0 + n_{ij}^1 + n_{ij}) \bar{f}_{ij} \quad (5.3)$$

$$0 \leq g \leq \bar{g} \quad (5.4)$$

$$0 \leq n_{ij} \leq (\bar{n}_{ij} - \bar{n}_{ij}^1) \quad (5.5)$$

$$f_{ij} \text{ irrestrito} \quad (5.6)$$

$$\forall (i, j) \in \Omega \quad (5.7)$$

Em cada passo a solução corrente é então atualizada, e assim, a topologia corrente é formada pela topologia base, n_{ij}^0 , juntamente com os circuitos adicionados durante o processo iterativo, n_{ij}^1 . Quando a solução do PL indicar que $v = 0$, e portanto $n_{ij} = 0$, significa que o sistema está operando adequadamente e que foi então encontrada uma solução factível.

$$I = \max\{f_{ij}^v = n_{ij}\bar{f}_{ij} : n_{ij} \neq 0 \text{ na solução do PL (5.1)-(5.7)}\} \quad (5.8)$$

Para determinar as demais soluções, proíbe-se, um ou mais dos circuitos acrescentados na primeira solução, e repete-se o processo acima, até se obter as $PSize$ soluções.

5.4.2.2 Soluções para o modelo DC

Da mesma forma, quando se considera o modelo DC, utiliza-se o AHC proposto por Villasana-Garver-Salon descrito na seção 3.2.2, que considera um indicador de sensibilidade definido a partir da solução ótima do modelo híbrido linear relaxado descrito em (5.9)-(5.18), o qual é um problema de programação linear (PL).

$$\min v = \sum_{(i,j) \in \Omega} c_{ij}n_{ij} \quad (5.9)$$

s.a.

$$Sf + S^0 f^0 + S^1 f^1 + g = d \quad (5.10)$$

$$f_{ij}^0 - \gamma_{ij}n_{ij}^0(\theta_i - \theta_j) = 0, \forall (i,j) \in \Omega_0 \quad (5.11)$$

$$f_{ij}^1 - \gamma_{ij}n_{ij}^1(\theta_i - \theta_j) = 0, \forall (i,j) \in \Omega_1 \quad (5.12)$$

$$|f_{ij}^0| \leq n_{ij}^0 \bar{f}_{ij}, \forall (i,j) \in \Omega_0 \quad (5.13)$$

$$|f_{ij}^1| \leq n_{ij}^1 \bar{f}_{ij}, \forall (i,j) \in \Omega_1 \quad (5.14)$$

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

$$0 \leq g \leq \bar{g} \quad (5.15)$$

$$0 \leq n_{ij} \leq (\bar{n}_{ij} - n_{ij}^1) \quad (5.16)$$

$$f_{ij}^0, f_{ij}^1, \text{ e } f_{ij} \text{ irrestritos} \quad (5.17)$$

$$\theta_j \text{ irrestrito} \quad (5.18)$$

Ou seja, em cada passo do AHC seleciona-se um circuito para ser adicionado ao sistema, o qual é identificado pelo índice de sensibilidade (5.19).

$$IS = \max\{IS_{ij} = n_{ij}\bar{f}_{ij} : n_{ij} \neq 0 \text{ na solução do PL (5.9)-(5.18)}\} \quad (5.19)$$

Em cada passo a solução corrente é então atualizada, e assim a topologia corrente é formada pela topologia base, n_{ij}^0 , juntamente com os circuitos adicionados durante o processo iterativo, n_{ij}^1 .

Para se obter o conjunto de soluções diversas deve se repetir este processo $Psize$ vezes, proibindo-se um ou mais dos circuitos adicionados na primeira topologia encontrada.

Após ter encontrado as $Psize$ soluções iniciais, as mesmas devem ser melhoradas por meio do Método de Melhoria, que será explicado na próxima seção.

5.4.3 Passo 3: Melhoria das Soluções

Como o Método Gerador de soluções não fornece soluções infactíveis, sugere-se inserir os passos da Figura 5.3 no algoritmo da Figura 5.2.

Passo 3.1. Ordenar os circuitos adicionados em ordem decrescente de custo. Verificar resolvendo o PL (5.20)-(5.25), (ou (5.26)-(5.32), para o modelo DC), se a retirada de um circuito mantém o sistema operando adequadamente. Se sim, remover o circuito, caso contrário, manter o circuito. Repetir o processo simulando a retirada de circuitos até que o último seja testado. Todos os circuitos que não forem retirados representam a solução melhorada;

Passo 3.2. Definir uma vizinhança e aplicar uma busca local na vizinhança de cada solução a ser melhorada.

Figura 5.3: Passos do Algoritmo de Melhoria Local.

Em ambos modelos, w representa o corte de carga do sistema e os r_k representam os cortes de carga nas barras de demanda que é equivalente a geração artificial.

Se $w = 0$ em (5.20)-(5.25), (ou em (5.26)-(5.32)), significa que a topologia avaliada representa uma solução factível para o modelo de transporte, ou DC.

$$\min w = \sum_{k \in \Omega} r_k \quad (5.20)$$

s.a.

$$Sf + g + r = d \quad (5.21)$$

$$|f_{ij}| \leq (n_{ij}^0 + n_{ij}^1) \bar{f}_{ij}, \quad \forall (i, j) \in \Omega \quad (5.22)$$

$$0 \leq r \leq d \quad (5.23)$$

$$0 \leq g \leq \bar{g} \quad (5.24)$$

$$f_{ij} \text{ irrestrito}, \quad \forall (i, j) \in \Omega \quad (5.25)$$

Caso contrário, a proposta representa uma solução infactível, e o circuito retirado deve então ser devolvido ao sistema.

$$\min w = \sum_{k \in \Omega} r_k \quad (5.26)$$

s.a.

$$Sf + g + r = d \quad (5.27)$$

$$f_{ij} - \gamma_{ij}(n_{ij}^1 + n_{ij}^0)(\theta_i - \theta_j) = 0 \quad (5.28)$$

$$|f_{ij}| \leq (n_{ij}^0 + n_{ij}^1) \bar{f}_{ij}, \quad \forall (i, j) \quad (5.29)$$

$$0 \leq r \leq d \quad (5.30)$$

$$0 \leq g \leq \bar{g} \quad (5.31)$$

$$f_{ij} \text{ irrestrito}, \theta_j \text{ irrestrito}, (i, j) \in \Omega \quad (5.32)$$

No caso do Método de Combinação fornecer uma solução infactível, o Método de Melhoria deve atuar no sentido de retomar sua viabilidade. Isto pode ser feito utilizando um AHC do tipo descrito no Método Gerador de Soluções, onde se adiciona linhas até não existir mais corte de carga, e depois, ainda pode se realizar a fase de retirada descrita no Passo 3.1.

Uma vez construído o conjunto P com as $Psize$ soluções, e após ter aplicado o método de melhoria a cada uma dessas soluções, o conjunto de referência ($RefSet$) deve ser construído. Na seção 5.4.4 explica-se a construção de $RefSet$.

5.4.4 Passo 4: Construção do Conjunto de Referência

Esta etapa da BD é aplicada a qualquer tipo de problema de otimização em que a BD atua. Nesta seção será explicada, em detalhes, a construção de *RefSet* e sua atualização estática por qualidade.

O conjunto de referência *RefSet* é constituído por b_1 soluções de alta qualidade $\{n^1, \dots, n^{b_1}\}$ e b_2 soluções com diversidade $\{n^{b_1+1}, \dots, n^b\}$. A dimensão de *RefSet* é dada por $b = b_1 + b_2$, e o algoritmo para sua construção está descrito na Figura 5.4.

-
- P1.** Inicie os parâmetros $b_1 \leftarrow \alpha$ e $b_2 \leftarrow \beta$; Faça $b \leftarrow 0$.
- P2.** Selecione as b_1 melhores soluções de P em relação à função objetivo e imediatamente as adicione em *RefSet* e elimine estas soluções de P . Faça $b \leftarrow b_1$.
- P3.** Defina a distância entre duas soluções n^1 e n^2 ,

$$d(n^1, n^2) = \sum_{k=1}^{nl} l_k,$$

em que nl é o número de linhas e

$$l_k = \begin{cases} 0, & \text{se } n \in n^1 \cap n^2 \\ 1, & \text{caso contrário} \end{cases}.$$

- P4.** Se $b \leq \alpha + \beta$, então calcular a distância $d_{min}(n) = \min\{d(n, n') : n' \in \text{RefSet}\}$ de uma solução $n \in P$ em relação aos elementos de *RefSet*. Ir então ao passo P5.
- P5.** Selecione a solução n que maximiza a distância mínima e a insira em *RefSet*. Elimine n de P . Caso $b = \alpha + \beta$, *RefSet* foi criado; Ir então ao passo P6. Caso contrário, faça $b \leftarrow b + 1$ e vá ao passo P4.
- P6.** Ordene *RefSet* da melhor à pior solução em relação ao valor da função objetivo.
-

Figura 5.4: Algoritmo para construir *RefSet*.

Terminada a construção de *RefSet*, é necessário gerar subconjuntos de soluções que pertençam a *RefSet* para aplicar o método de combinação de soluções.

5.4.5 Passo 5: Geração de Subconjuntos

Considera-se o caso mais simples do método de geração de subconjuntos, que consiste em realizar unicamente combinações de duas soluções; isto é, em cada fase da combinação são considerados pares de soluções de *RefSet* que não tenham sido combinadas. Após ter obtido os pares de

soluções a serem combinadas aplica-se o método de combinação de soluções. Por exemplo, se os subconjuntos são de tamanho dois, tem-se $(b^2 - b)/2$ *NewSubsets* a serem combinados.

5.4.6 Passo 6: Combinação de Soluções

Considera-se o método de combinação por *Path Relinking*. Neste tipo de combinação, dadas duas soluções n^1 e n^2 , explora-se o caminho que conecta estas duas soluções, produzindo uma sucessão de soluções

$$n^1 = n(1), n(2), \dots, n(r) = n^2,$$

como pode ser visto na Figura 5.5. O ponto inicial n^1 deste caminho é chamada solução inicial, e o ponto final, n^2 , é chamada solução guia.

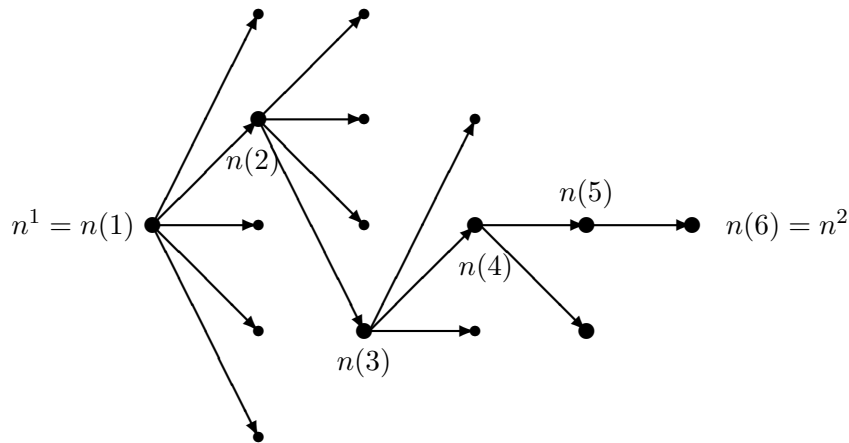


Figura 5.5: *Path Relinking*.

Este caminho pode ser tomado de diversas formas e utilizando vários critérios.

Por exemplo, num primeiro passo, compara-se as soluções n^1 e n^2 obtendo-se dois vetores Δ_{ad} e Δ_{ret} , identificando, respectivamente, os circuitos que devem ser adicionados e os que devem ser retirados em n^1 para alcançar n^2 . Seja $\hat{n} = n^1$ a solução de trabalho.

Num segundo passo implementa-se a escolha de um movimento, que é feito de forma gulosa em Δ_{ret} , ou seja, retira-se de $\hat{n}$ o circuito mais caro em Δ_{ret} e verifica-se resolvendo o PL (5.20)-(5.25), ou (5.26)-(5.32), se a solução é factível, ou seja, se $\omega = 0$. Neste caso, $\hat{n}$ é candidato a *Path Relinking*, e faz-se $n(2) = \hat{n}$. Se $\hat{n}$ não é factível, ou seja, se $\omega > 0$, deve-se acrescentar circuitos em $\hat{n}$ para retomar sua factibilidade e estes circuitos são escolhidos inicialmente em Δ_{ad} .

Se os circuitos de Δ_{ad} não são suficientes para retomar esta factibilidade, deve-se escolher circuitos como no AHC do método gerador de soluções diversas. Feita a adição do circuito, verifica-

se a factibilidade novamente, e se ainda $\omega > 0$, deve se realizar um novo movimento de adição até que $\omega = 0$. Neste caso, $\hat{n}$ é candidato a PR. O processo PR se reinicia, agora em $\hat{n}$ como solução inicial, sendo necessário realizar uma nova comparação com a solução guia. Ao se alcançar a solução guia, toma-se $n^r = \min \hat{n}$ a solução combinada por PR, ou seja, obtém-se apenas uma solução combinada para se armazenar em *Pool*.

Outra forma de se fazer o *Path Relinking* é realizar os movimentos permitindo infactibilidades nas soluções obtidas durante a trajetória. Assim, por exemplo, num primeiro passo, compara-se a solução base com a solução guia, obtendo-se o vetor Δ_k formado pelos k atributos que devem ser modificados para que a solução base alcance a solução guia. Se $n(1) = n^1$ é a solução de trabalho, deve-se simular a cópia de cada um dos atributos de Δ_k , e verificar a factibilidade das k soluções candidatas. O movimento escolhido, ou seja a solução $n(2)$, será a melhor (menor custo) solução factível obtida, ou, caso não se obtenha soluções factíveis, será a solução com menor infactibilidade. Para se realizar o próximo movimento, novamente deve ser simulada a cópia dos $k - 1$ atributos restantes, e se escolher dentre as $k - 1$ opções, a melhor opção (a melhor factível ou a menos infactível). O processo deve ser repetido até que o último atributo de Δ_k tenha sido copiado, ou seja, $n(k) = n_2$. Aqui, todas as soluções factíveis distintas devem ser armazenadas em *Pool*, ou seja, se obtém várias soluções combinadas. Se não existir soluções factíveis, deve-se armazenar em uma lista auxiliar *Pool'* as soluções infactíveis distintas, para que depois parte destas soluções possam ser factibilizadas e aproveitadas.

O PR também pode ser realizado em trajetória dupla, ou seja, considerando-se ambas soluções como base e guia, respectivamente.

O Método de melhoria deve ser aplicado nas soluções armazenadas em *Pool*.

5.4.7 Passo 7: Atualizar o conjunto de referência

O conjunto de referência é atualizado de forma estática por qualidade. Assim, após todas as soluções terem sido combinadas, compara-se o valor da função objetivo de cada solução armazenada em *Pool* com o valor da função objetivo da pior solução de *RefSet*, ou seja, com o investimento da solução n^b . Se a solução comparada tiver investimento menor do que o investimento de n^b , e possuir valor de função objetivo distinto das demais soluções $\{n^1, \dots, n^{b-1}\}$ de *RefSet*, a solução n_b é então substituída pela solução comparada e *RefSet* é novamente reordenado. Passa-se para o próximo elemento de *Pool* e novamente se compara com o pior elemento de *RefSet*, repetindo o processo até que o último elemento de *Pool* seja comparado com o pior elemento de *RefSet*. Se ocorrer de algum elemento de *Pool* tiver o mesmo valor da função objetivo de algum elemento de *RefSet*, deve-se verificar, comparando os atributos destas soluções, se estas soluções são diferentes, antes de inserí-la em *RefSet*. Caso sejam iguais, a solução é descartada.

	1-2	1-3	1-4	1-5	1-6	2-3	2-4	2-5	2-6	3-4	3-5	3-6	4-5	4-6	5-6	
$n^1 =$	0	0	0	0	0	1	0	0	1	0	1	0	0	2	0	$v = 130$
$n^2 =$	0	0	0	0	0	0	0	0	2	0	1	0	0	2	0	$v = 140$
$n^3 =$	0	0	0	0	0	1	0	0	3	0	1	0	0	1	0	$v = 160$
$n^4 =$	0	0	0	0	0	1	0	0	0	0	1	2	0	2	0	$v = 196$
$n^5 =$	0	0	0	0	0	0	1	1	0	0	2	0	0	3	0	$v = 201$
$n^6 =$	0	3	0	1	0	1	0	0	0	0	0	0	0	3	0	$v = 244$

Figura 5.7: Conjunto de Referência.

Inicialmente retira-se o circuito mais caro de Δ_{ret} , $n_{35} = 1$, e resolve-se o PL (5.26)-(5.32), cuja solução indica que ocorreu um corte de carga no sistema. Tenta-se então acrescentar o circuito mais barato em Δ_{ad} , ou seja, o circuito $n_{13} = 3$ e resolve-se novamente o PL (5.26)-(5.32), obtendo-se então $\omega = 0$. E assim, obtém-se a solução $n(2)$ da Figura 5.8, com $v = 150$.

	1-2	1-3	1-4	1-5	1-6	2-3	2-4	2-5	2-6	3-4	3-5	3-6	4-5	4-6	5-6	
$n(2) =$	0	0	0	1	0	1	0	0	0	0	1	0	0	3	0	
$n^1 =$	0	0	0	0	0	1	0	0	1	0	1	0	0	2	0	

Figura 5.8: Primeira solução combinada com PR e solução guia.

Agora, comparando as soluções $n(2)$ e n^1 , obtém-se $\Delta_{ad} = \{n_{26} = 1\}$ e $\Delta_{ret} = \{n_{15} = 1, n_{46} = 1\}$. Como feito anteriormente, o algoritmo retira o circuito $n_{15} = 1$ de Δ_{ret} , e acrescenta o circuito $n_{26} = 1$ de Δ_{ad} , obtendo-se a solução $n(3)$ da Figura 5.9, com $\omega = 0$ e $v = 160$.

	1-2	1-3	1-4	1-5	1-6	2-3	2-4	2-5	2-6	3-4	3-5	3-6	4-5	4-6	5-6	
$n^1 =$	0	0	0	0	0	1	0	0	1	0	1	0	0	2	0	
$n(3) =$	0	0	0	0	0	1	0	0	1	0	0	0	0	3	0	

Figura 5.9: Segunda solução combinada com PR e solução guia.

Ao se aplicar o método de melhoria, o algoritmo identifica que o circuito $n_{23} = 1$ é redundante no sistema, ou seja, sua saída do sistema não causa corte de carga.

Assim, retirando-se este circuito, obtém-se a solução $n(4)$, com $n_{26} = 1$ e $n_{46} = 3$, cujo investimento é de $v = 110$, representada na Figura 5.10, a qual é a solução ótima deste sistema para o modelo DC.

Portanto o algoritmo converge na primeira iteração, resolvendo 87 PL's.

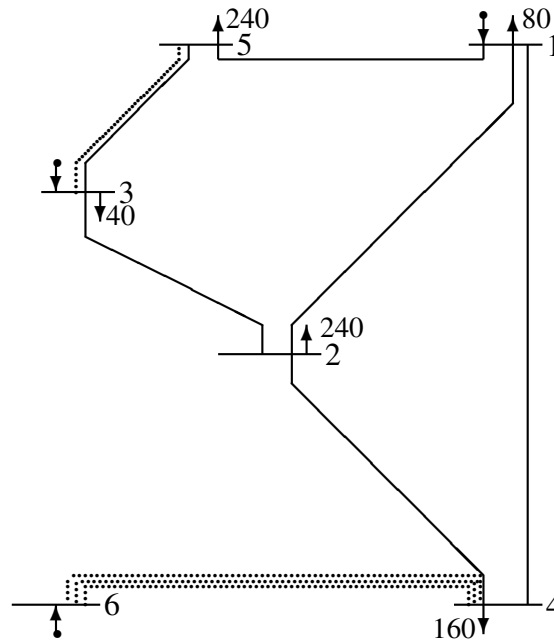


Figura 5.10: Solução ótima (BD/DC) do sistema de 6 barras de Garver.

5.6 Outras Estratégias

No caso da atualização de *RefSet* ser estática (após realizar todas as combinações possíveis), as soluções geradas pelo método de combinação de soluções de *RefSet* são armazenadas em uma lista denominada *Pool*. Se a atualização for dinâmica, após obter cada solução combinada, ela pode ser inserida em *RefSet* sempre que cumprir com um critério de qualidade e/ou diversidade.

Se a atualização é estática por qualidade e diversidade, então escolhe-se as $b/2$ melhores soluções armazenadas em $RefSet \cup Pool$ e as $b/2$ soluções com maior diversidade de $RefSet \cup Pool$.

Quando a atualização é dinâmica por qualidade a solução obtida pelo método de combinação é inserida em *RefSet* sempre que melhorar a pior solução de *RefSet*. Se a atualização for dinâmica por qualidade e diversidade, a solução obtida pelo método de combinação é inserida em *RefSet* sempre que melhorar a pior solução em relação ao valor da função objetivo ou se esta solução aumentar a distância mínima dos elementos em *RefSet* (isto é, se tiver maior diversidade).

Se na atualização de *RefSet* novas soluções não forem introduzidas, ativa-se novamente o método gerador de subconjuntos para aplicar o método da combinação de soluções. Se durante a atualização de *RefSet* não são geradas novas soluções então *RefSet* é reconstruído.

5.7 Reconstrução do Conjunto de Referência

Mantém-se a metade das soluções com melhor valor da função objetivo em *RefSet* e a outra metade é substituída por novas soluções de P que introduzam diversidade. Aplica-se o mesmo mecanismo utilizado quando *RefSet* foi construído. Uma vez reconstruído *RefSet* aplica-se novamente o método de combinação de soluções e a atualização de *RefSet*. A BD termina quando se atinge um número determinado de reconstruções de *RefSet* ou um número preespecificado de soluções avaliadas, ou seja de PL's resolvidos.

5.8 Resultados Obtidos

O algoritmo proposto foi implementado utilizando a linguagem de programação FORTRAN, e o MINOS 5.0 como uma sub-rotina de resolução de PL.

O tamanho de *RefSet*, em ambos os sistemas considerados, foi $b = 10$ soluções, com $b_1 = b_2 = 5$ e $|P| = 15$.

5.8.1 Sistema IEEE de 24 Barras

Foram realizados testes para o Sistema IEEE de 24 barras considerando-se os dois planejamentos utilizados no Capítulo 4, cujos dados aparecem no Apêndice A.

Os dois métodos de recombinação por *Path Relinking* foram utilizados durante a realização dos testes, mas os resultados que seguem foram obtidos com o PR do segundo tipo, já que com este método, os resultados foram melhores.

5.8.1.1 Modelo de Transportes

Ao se aplicar o algoritmo BD proposto no planejamento P1 utilizando o modelo de transportes, atingiu-se, na primeira iteração (primeira atualização de *RefSet*) e resolvendo 157 PL's, a melhor solução conhecida para este planejamento, com um investimento de U\$310,000.00, a qual pode ser vista na Tabela 5.1.

O mesmo acontece ao considerarmos o Plano P2, cujos dados aparecem nas tabelas A.2 e A.1. Ou seja, na primeira iteração e resolvendo 196 PL's, o algoritmo converge para a melhor solução conhecida apresentada na Tabela 4.1, com investimento de U\$188,000.00.

Tabela 5.1: BD: IEEE 24 barras - Modelo de Transportes.

Plano	Investimento (US\$)	Circuitos Adicionados	No. de PL's
P1	310,000.00	$n_{03-24} = 1$ $n_{06-10} = 1$ $n_{07-08} = 1$ $n_{10-12} = 1$ $n_{14-16} = 1$ $n_{16-17} = 2$ $n_{16-19} = 1$ $n_{17-18} = 1$	157
P2	188,000.00	$n_{06-10} = 1$ $n_{07-08} = 2$ $n_{10-12} = 1$ $n_{14-16} = 1$ $n_{16-17} = 1$	196

5.8.1.2 Modelo DC

Quando considerou-se o algoritmo proposto com o modelo DC, o algoritmo converge para as soluções que aparecem na Tabela 5.2, resolvendo, respectivamente, 522 e 329 PL's, em três e duas iterações, respectivamente. Assim, para o Plano P1, o algoritmo converge para a melhor solução conhecida, e para o plano P2, para uma solução de excelente qualidade, diferindo em apenas dois atributos da melhor configuração conhecida.

Tabela 5.2: BD: IEEE 24 barras - Modelo DC.

Plano	Investimento (US\$)	Circuitos Adicionados	No. de PL's
P1	392,000.00	$n_{01-05} = 1$ $n_{03-24} = 1$ $n_{06-10} = 1$ $n_{07-08} = 1$ $n_{10-12} = 1$ $n_{14-16} = 1$ $n_{15-24} = 1$ $n_{16-17} = 2$ $n_{17-18} = 2$	522
P2	354,000.00	$n_{03-24} = 1$ $n_{06-10} = 1$ $n_{07-08} = 2$ $n_{09-12} = 1$ $n_{10-12} = 1$ $n_{12-13} = 1$ $n_{14-16} = 1$ $n_{16-17} = 1$	329

5.8.2 Sistema Sul Brasileiro

Apresentam-se resultados para os modelos de transportes e DC, cuja execução do planejamento pode ser feita com (CR), ou sem (SR) reprogramação da geração.

5.8.2.1 Modelo de Transportes

Quando se utiliza o modelo de transportes, o algoritmo BD proposto encontra as soluções ótimas apresentadas na Tabela 4.3 para os dois casos considerados, resolvendo, respectivamente,

421 e 2.611 PL's, e realizando uma e três atualizações do conjunto de referência, respectivamente.

Para o Plano P2, o algoritmo proposto determina uma solução de investimento U\$ 130,943, 000.00, exigindo a solução de 179 PL's na primeira iteração.

5.8.2.2 Modelo DC

Quando foi utilizado o modelo DC com reprogramação da geração, o algoritmo alcançou a solução ótima apresentada na Tabela 4.4 com um investimento de $v = \text{US\$ } 72,870,000.00$, resolvendo-se 773 PL's e duas atualizações de *RefSet*.

Considerando-se o planejamento SR, também se alcançou o ótimo global apresentado na Tabela 4.4, mas exigindo-se a solução de 7.044 PL's e cinco atualizações de *RefSet*.

5.9 BD entre as Metaheurísticas

Analogamente ao que foi feito coma a VNS, examina-se brevemente como a BD se ajusta a cada uma das propriedades que aparecem na seção 4.7.

A BD é um algoritmo evolutivo, e a **simplicidade** de sua implementação o torna muito aplicável, como se reflete nas inúmeras aplicações já comentadas. As regras **precisas** na forma de efetuar seus passos são fundamentais, principalmente no que diz respeito ao método de combinação de soluções. Os esquemas básicos e extensões da BD se traduzem **coerentemente** dos princípios em que se inspira.

A aplicação da BD tem mostrado bastante **eficácia e robustez** na resolução de vários problemas conhecidos, e com tempo computacional moderado.

A **facilidade** dos algoritmos baseados na BD contam com a vantagem de que os princípios básicos em que se fundamentam são fáceis de se aplicar, e muito fáceis de usar.

A **inovação** deste método depende do grau de sofisticação da implementação de cada passo do algoritmo.

6 *Considerações Finais*

Neste capítulo aponta-se algumas conclusões obtidas sobre o trabalho, bem como as dificuldades encontradas e ainda algumas sugestões para trabalhos futuros.

6.1 O Problema de Planejamento da Expansão de Sistemas de Transmissão

Apresentou-se neste trabalho, uma ampla abordagem do Problema de Planejamento da Expansão de Sistemas de Transmissão de Energia Elétrica, um problema clássico e muito importante na área de otimização. Foram apontadas as principais dificuldades na resolução deste problema, as quais estão relacionadas com a natureza combinatória do processo de planejamento que normalmente leva a um número explosivo de alternativas, inclusive no caso de sistemas de médio porte, que, além disso, 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 e às vezes de pobre qualidade.

Para contornar estas e outras dificuldades na resolução deste problema, neste trabalho fez-se uso inédito de duas metaheurísticas, a Busca em Vizinhança Variável e a Busca Dispersa.

6.2 Busca em Vizinhança Variável

Realizou-se neste trabalho uma rigorosa investigação desta recente metaheurística, a qual foi proposta na década de 90 em (MLADENOVIC, 1995; MLADENOVIC; HANSEN, 1997), e explora o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança durante o processo de busca.

É uma metaheurística que, mesmo sendo recente, conta com uma extensa literatura, o que demonstra seu alto interesse, tanto teórico como prático, e várias áreas de otimização, especialmente no campo de Localização. Além disso, a VNS se ajusta a maioria das propriedades, que segundo autores relevantes, que devem ter as metaheurísticas.

Ao contrário de muitas metaheurísticas, o esquema básico da VNS e suas extensões são simples e requerem poucos, e algumas vezes, nenhum parâmetro. Sempre determina boas soluções de modo mais simples que outros métodos, e também possibilita implementações mais eficientes e sofisticadas.

Foi ilustrada sua aplicação na solução do PPEST, mostrando que os resultados obtidos foram bastantes satisfatórios, em todas as instâncias consideradas.

Um ponto crítico na estratégia de busca em vizinhanças é a escolha da estrutura da vizinhança, ou seja, a maneira pela qual a vizinhança é definida, já que as vizinhanças crescem muito crescem muito com a dimensão do problema no caso do PPEST.

A escolha da solução inicial também é um fator importante nesta estratégia. Neste trabalho foram consideradas outras soluções iniciais, como por exemplo, no modelo DC, tomou-se a solução obtida pelo método de Garver, ou seja, considerou-se uma solução inicial factível apenas para o modelo de transportes e continuou-se a busca local com o modelo DC. Entretanto, aplicando-se esta solução inicial, o número de PL's aumentou consideravelmente para se atingir os mesmos resultados.

Também se realizou testes para a VNS, RVNS, BVNS e SVNS, mas os componentes aleatórios fizeram com que o número de PL's aumentasse muito para atingir os resultados obtidos pela VND, e algumas vezes, resultados inferiores.

Ainda que a VNS tenha se mostrado claramente útil na resolução aproximada de muitos problemas de difícil solução, é muito custoso se resolver problemas muito grandes.

Quando as metaheurísticas são aplicadas à instâncias realmente grandes, suas forças e debilidades aparecem mais claramente. As melhorias neste sentido são então altamente desejáveis, tanto para as metaheurísticas em geral, como para a VNS em particular.

6.3 Busca Dispersa

A metaheurística Busca Dispersa é um método evolutivo que ainda se encontra em desenvolvimento. Embora tenha sido proposto nos anos 70 e seja menos conhecida que os Algoritmos Genéticos, tem sido aplicada com êxito em numerosos problemas difíceis de otimização.

Não existe um esquema único para se aplicar a BD. Apresentou-se aqui os aspectos básicos mais conhecidos do método e revisou-se as principais aplicações implementadas na literatura e as questões que são atualmente objeto de estudo.

A aplicação da BD no PPEST foi cuidadosamente apresentada, detalhando cada passo do algoritmo. Foi ilustrado, através de um exemplo, os passos da BD aplicada na solução de um sistema real de pequeno porte.

Na combinação por *Path Relinking* a escolha da solução guia como sendo a solução de melhor qualidade proporcionou melhores resultados.

Fez-se testes considerando-se tamanhos distintos para *RefSet*, e notou-se que o valor $b = 10$ possibilitou melhor performance do algoritmo.

O tamanho do conjunto de referência foi escolhido segundo (LAGUNA; ARMENTANO, 2005), o qual afirma que deve ter entre 10 e 20 soluções.

Os resultados obtidos com a metaheurística Busca Dispersa foram de excelente qualidade em todos os casos considerados .

6.4 Busca em Vizinhança Variável ou Busca Dispersa?

A VNS é uma metaheurística que segue uma trajetória de descida e que não permite degradação no valor da função objetivo para se realizar um movimento. Em vez de permitir tal degradação, a VNS 'prefere' aumentar a dimensão do espaço de busca, o que faz com que a convergência do algoritmo seja muito custosa, ou inviável, dependendo da dimensão do problema considerado. Já a BD é uma metaheurística da família dos algoritmos evolutivos, que pode permitir tal degradação na realização de um movimento, mas em contrapartida, exige um método específico e inteligente de diversificação no processo de geração de soluções bem como um método especializado de combinação de soluções que leva em consideração as características do problema abordado. Tais particularidades são responsáveis pela diferença da quantidade de PL's necessários para se atingir as soluções ótimas em cada caso.

Como a BD é mais flexível, permitindo diversos graus de sofisticação na implementação de cada passo do seu algoritmo, pode-se concluir que esta metaheurística seja mais promissora que a VNS na solução do PPEST.

6.5 Trabalhos Futuros

Melhorar a definição de vizinhança.

Criar técnicas mais eficientes de redução de vizinhança.

Definir novas formas de combinação de soluções para aplicar em BD.

No método gerador de soluções diversas pode-se considerar outros índices de sensibilidade,

em vez de se proibir circuitos.

Paralelizar BD e VNS.

Referências

- ANDREATTA, A.; RIBEIRO, C. Heuristic for the phylogeny problem. *Journal of Heuristics*, Boston, n. 8, p. 429–447, 2002.
- BELACEL, N.; HANSEN, P.; MLADENOVIĆ, N. Fuzzy j-means: a new heuristic for fuzzy clustering. *Pattern Recognition*, New York, v. 35, n. 10, p. 2193–2200, 2002.
- BENNON, R. J.; JUVES, J. A. Use of sensitivity analysis in automated transmission planning. *IEEE Transaction Power Apparatus Systems*, New York, n. PAS-101, p. 319–3925, 1982.
- BESTEN, M.; STÜTZLE, T. Neighborhoods revisited: Investigation into the effectiveness of variable neighborhood descent for scheduling. *MIC'2001*, Porto, p. 545–549, 2001.
- BINATO, S. *Expansão ótima de sistemas de transmissão através de decomposição de Benders e técnicas de planos cortantes*. Tese (Doutorado) — COPPE/UFRJ, Rio de Janeiro, 2000.
- BINATO, S.; OLIVEIRA, C. A heuristic approach to cope with multi-year transmission expansion planning. *Proceedings of the Stockholm Power Tech Conference*, Stockholm, SPT, n. S01-03-277, 1995.
- BINATO, S.; PEREIRA, M. V. F.; GRANVILLE, S. A new benders decomposition approach to solve power transmission network design problems. *IEEE Transaction on Power Systems*, New York, v. 2, n. 16, p. 235–240, 2001.
- BINATO, S.; ROMERO, R. Power transmission network expansion planning by a hybrid genetic algorithm. In: LATIN-IBEROAMERICAN CONGRESS ON OPERATIONS RESEARCH, 9, 1998, Buenos Aires. In: *Proceedings ...*, Buenos Aires,. 1998.
- BRINBERG, J.; HANSEN, P.; MLADENOVIĆ, N. Convergence of variable neighborhood search. *Les Cahiers du GERAD*, Montreal, G, n. 2002-21, 2004.
- BRINBERG, J.; HANSEN, P.; MLADENOVIĆ, N.; TAILLARD, . Improvements and comparison of heuristics for solving the multisource weber problem. *Operations Research*, Baltimore, v. 3, n. 48, p. 444–460, 2000.
- BRÄYSY, O. *A reactive variable neighborhood search algorithm for the vehicle routing problem with time windows*. Vaasa: Acta Wasaensia, 2001.
- BURKE, E. K.; CAUSMAECKER, P.; PETROVIĆ, S.; BERGHE, C. V. Variable neighborhood search for nurse rostering. *MIC'2001*, Porto, p. 755–760, 2001.
- CAMPOS, L. M.; PUERTA, J. M. Stochastic local search algorithms for linear belief networks: searching in the space of orderings. *Lecture Notes en AI*, Berlin, n. 2143, p. 228–239, 2001.
- CAMPOS, V.; GLOVER, F.; LAGUNA, M.; MARTI, R. An experimental evaluation of a scatter search for the linear ordering problems. *Journal of Global Optimization*, Dordrecht, n. 21, p. 397–414, 2001.

CAMPOS, V.; LAGUNA, M.; MARTI, R. Scatter search for the linear ordering problem. *New Ideas in Optimization*, Berlin, p. 331–341, 1999.

CAMPOS, V.; LAGUNA, M.; MARTI, R. Context independent scatter search and tabu search for permutation problems. *INFORMS Journal on Computing*, Linthicum, v. 1, n. 17, p. 111–122, 2005.

CANUTO, S.; RESENDE, M.; RIBEIRO, C. Local search with perturbations for the prize-collection steiner tree problem in graphs. *Networks*, New York, n. 38, p. 38–50, 2001.

CAPOROSSO, G.; CVETKOVIC, D.; GUTMAN, I.; HANSEN, P. Variable neighborhood search for extremal graphs 2: Finding graphs with extremal energy. *Journal of Chemical Information and Computer Science*, Washington, n. 39, p. 984–996, 1999.

CAPOROSSO, G.; HANSEN, P. Variable neighborhood search for extremal graphs 1: The autographix system. *Discrete Mathematics*, Amsterdam, n. 212, p. 24–44, 2000.

CHAMPS, D. C.; VANKELECOM, J.; AMOULLE, E. Tranex - an interactive computer program for transmission expansion. *IEEE Summer Power Meeting*, New York, n. A79-476-3, 1979.

CORBERÁN, A.; FERNÁNDEZ, E.; LAGUNA, M.; MARTI, R. Heuristic solution to the problem of routing school buses with multiple objectives. *Journal of Operational Research Society*, Oxford, v. 4, n. 53, p. 427–435, 2002.

CRAINIC, T.; GENDREAU, M.; HANSEN, P.; HOEB, N.; MALDENOVIĆ, N. Parallel variable neighborhood search for the p-median. *MIC'2001*, Porto, 2001.

CUNG, V.; MICHELON, T. M. P.; TAVARES, A. A scatter search based approach for the quadratic assignment problem. *Proceedings of the IEEE-ICE'97 Conference in Indianapolis*, Indianapolis, p. 13–16, 1997.

DAVIDOVIĆ, T.; HANSEN, P.; MLADENOVIĆ, N. Variable neighborhood search for multiprocessor scheduling with communication delays. *MIC'2001*, Porto, p. 737–741, 2001.

DINGZHU, J. G.; PARDALOS, P. Satisfiability problem: Theory and applications. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Providence, 1997.

DUSONCHET, Y. P.; EL-ABIAD, A. H. Transmission planning using discrete dynamic optimization. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Providence, n. 35, 1973.

ESCOBAR, A.; GALLEGO, R. A.; ROMERO, R. Multistage and coordinated planning of the expansion of transmission systems. *IEEE Transactions on Power Systems*, New York, v. 2, n. 19, p. 735–744, 2004.

FANG, R.; HILL, D. J. A new strategy for transmission expansion in competitive electricity markets. *IEEE Transaction on Power Systems*, New York, v. 1, n. 18, p. 474–380, 2003.

FARIA, H. J. *Uma nova metaheurística para problemas combinatórios aplicado ao planejamento da expansão de sistemas de transmissão de energia elétrica*. Tese (Doutorado em Engenharia Elétrica) — Universidade Federal do Rio de Janeiro-UFRJ, Rio de Janeiro, 2005.

FARIA, H. J.; BINATO, S.; RESENDE, M. G. C.; FALCÃO, D. M. Grasp com path relinking para o planejamento da expansão de redes de transmissão. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, 14, 2002, Natal. In: *Anais ...*, Natal, 2002, 2002.

- FARIA, H. J.; BINATO, S.; RESENDE, M. G. C.; FALCÃO, D. M. Power transmission network design by greedy randomized adaptive path relinking. *IEEE Transactions on Power Systems*, New York, v. 1, n. 20, p. 43–49, 2005.
- FEO, T.; RESENDE, M. Greedy randomized adaptive search. *Journal of Global Optimization*, Dordrecht, n. 6, p. 109–133, 1995.
- FESTA, P.; PARDALOS, P.; RESENDE, M.; RIBEIRO, C. Grasp and vns for max-cut. *MIC'2001*, Porto, p. 371–376, 2001.
- FISCHL, R.; PUNTEL, W. R. Computer aided design of electrical power transmission network. *IEEE winter Power Meeting*, New York, n. C72-168-8, 1973.
- FLESZAR, K.; HINDI, K. S. New heuristics for one-dimensional bin-packing. *Computers and Operations Research*, New York, v. 7, n. 29, p. 821–839, 2002.
- FLESZAR, K.; HINDI, K. S. Solve the resource-constrained project scheduling problem by a variable neighborhood search. *European Journal of Operational Research*, Amsterdam, v. 2, n. 1, p. 402–413, 2004.
- GALLEGO, R. A.; ALVES, A. B.; MONTICELLI, A.; ROMERO, R. Parallel simulated annealing applied to long term transmission expansion planning. *IEEE Transaction on Power Systems*, New York, v. 1, n. 12, p. 181–187, 1997.
- GALLEGO, R. A.; MONTICELLI, A.; ROMERO, R. Comparative studies on non-convex optimization methods for transmission network expansion planning. *IEEE Transaction on Power Systems*, New York, v. 3, n. 13, p. 822–828, 1998.
- GALLEGO, R. A.; MONTICELLI, A.; ROMERO, R. Transmission system expansion planning by extended genetic algorithm. *IEEE Proceeding Generation, Transmission and Distribution*, New York, v. 3, n. 145, p. 329–335, 1998.
- GALLEGO, R. A.; MONTICELLI, A.; ROMERO, R. Tabu search algorithm for network synthesis. *IEEE Transaction on Power Systems*, New York, v. 2, n. 15, p. 490–495, 2000.
- GARCIA, L. F.; MELIÁN, B.; PEREZ, J.; MORENO, V. J. The parallel variable neighborhood search for the p-median problem. *Journal of Heuristics*, Boston, v. 8, n. 3, p. 375–388, 2002.
- GARCIA, L. F.; MELIÁN, B.; PEREZ, J.; MORENO, V. J. Parallelization of the scatter search for the p-median problem. *Parallel Computing*, Amsterdam, n. 29, p. 575–589, 2003.
- GARVER, L. L. Transmission network estimation using linear programming. *IEEE Transaction Apparatus Systems*, New York, n. 89, p. 1688–1697, 1970.
- GENDREAU, M.; HANSEN, P.; LABBÉ, M.; MLADENOVIC, N. *Variable neighborhood search for the traveling salesman problem*. 2001. Disponível em: <http://citeseer.ist.psu.edu/context/1492859/0>. Acesso em: Abril de 2005.
- GENDREAU, M.; PESANT, G.; ROUSSEAU, L. M. Using constraint-based operators to solve the vehicle routing problem with time windows. *Journal of Heuristics*, Boston, v. 1, n. 8, p. 43–58, 2002.
- GLOVER, F. Heuristics for integer programming using surrogate constraints. *Decision Sciences*, Atlanta, v. 8, p. 156–166, 1977.
- GLOVER, F. Tabu search - part i. *ORSA Journal of Computing*, Baltimore, n. 1, p. 190–206, 1989.

- GLOVER, F. Tabu search - part ii. *ORSA Journal of Computing*, Baltimore, n. 2, p. 4–32, 1990.
- GLOVER, F. Tabu search for nonlinear and parametric optimisation (with links to genetic algorithms). *Discrete Applied Mathematics*, Amsterdam, n. 49, p. 231–255, 1994.
- GLOVER, F. A template for scatter search and path relinking. *Artificial Evolution - Lecture Notes in Computer Science*, Berlin, n. 1363, p. 13–54, 1998.
- GLOVER, F. Scatter search and path relinking. In: CONE, D.; DORIGO, M.; GLOVER, F. (Ed.). *New Ideas in Optimization*. New York: McGraw Hill, p. 297–316. 1999.
- GLOVER, F.; KOCHENBERGER, G. A. *Handbook of Metaheuristics*. Boston: Kluwer Academic Publishers, 2003.
- GLOVER, F.; LAGUNA, M. *Tabu search*. Boston: Kluwer, 1997.
- GLOVER, F.; LAGUNA, M.; MARTI, R. Scatter search. In: GHOSH, A.; TSUTSUI, S. (Ed.). *Theory and Applications of Evolutionary Computation: Recent Trends*. Berlin: Springer-Verlag, 1999.
- GLOVER, F.; LAGUNA, M.; MARTI, R. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, v. 3, n. 39, p. 653–684, 2000.
- GLOVER, F.; LAGUNA, M.; MARTI, R. New ideas and applications of scatter search and path relinking. 2004. Disponível em: <http://www.leeds-faculty.colorado.edu/laguna/articles/ss5.pdf>. Acesso em: 10 fev. 2006.
- GONZAGA, C. C. *Estudos de algoritmos de busca em grafos e sua aplicação a problemas de planejamento*. Tese (Doutorado) — COPPE-Universidade Federal do Rio de Janeiro, Rio de Janeiro, 1973.
- HAFFNER, S. *Planejamento de sistemas de transmissão no contexto de livre mercado*. Tese (Doutorado) — UNICAMP, Campinas, 2000.
- HAFFNER, S.; MONTICELLI, A.; GARCIA, A.; ROMERO, R. Specialized branch-and-bound algorithm for transmission network expansion planning. *IEEE Proceedings - Generation, Transmission and Distribution*, New York, v. 5, n. 148, p. 521–526, 2001.
- HAMIEZ, J. P.; HAO, J. K. Scatter search for graph colouring. *Lecture Notes in Computer Science*, Berlin, n. 5, p. 168–179, 2002.
- HANSEN, P.; JAUMARD, B.; MLADENOVIC, N.; PARREIRA, A. D. Variable neighborhood search for maximum weight satisfiability problem. *Les Cahiers du GERAD*, Montreal, n. G-2000-62, 2000.
- HANSEN, P.; MLADENOVIC, N. Variable neighborhood search for the p -midiam. *Location Science*, Oxford, n. 5, p. 207–226, 1997.
- HANSEN, P.; MLADENOVIC, N. An introduction to variable neighborhood search for the p -midiam. In: VOSS, S. et al (Ed.). *Metaheuristics Advances and Trends in Local Search Paradigms for Optimization*. Dordrecht: Kluwer, p. 433–458. 1999.
- HANSEN, P.; MLADENOVIC, N. Developments of variable neighborhood search. *Les Cahiers du GERAD*, Montreal, v. 22, n. G-01-24, 2001.

- HANSEN, P.; MLADENović, N. Industrial applications of the variable neighborhood search metaheuristics. In: ZACCOUR, G. (Ed.). *Decisions and Control in Management Science*. Boston: Kluwer, p. 261-274. 2001.
- HANSEN, P.; MLADENović, N. J-means: A new local search heuristic for minimum sum-of-squares clustering. *Pattern Recognition*, New York, v. 2, n. 34, p. 405–413, 2001.
- HANSEN, P.; MLADENović, N. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, Amsterdam, n. 130, p. 449–467, 2001.
- HANSEN, P.; MLADENović, N. Variable neighborhood search. In: PARDALOS, M. P.; RESENDE, M. G. C. (Ed.) *Handbook of Applied Optimization*. Boston: Kluwer, 2002.
- HANSEN, P.; MLADENović, N. A tutorial on variable neighborhood search. *Les Cahiers du GERAD*, Montreal, v. 2003-46, n. G, 2003.
- HANSEN, P.; MLADENović, N. Variable neighborhood search. In: GLOVER, F.; KOCHENBERGER, G. A. (Ed.). *Handbook of Metaheuristics*. Dordrecht: Kluwer, p. 145-184. 2003.
- HANSEN, P.; MLADENović, N.; PÉREZ, J. A. M. Variable neighborhood decomposition search. *Journal of Heuristics*, Boston, v. 4, n. 7, p. 335–350, 2001.
- HANSEN, P.; MLADENović, N.; PÉREZ, J. A. M. Variable neighborhood search. *Revista Iberoamericana de Inteligência Artificial*, Amsterdam, n. 19, p. 77–92, 2003.
- HARTIGAN, J. A. *Clustering Algorithms*. New York: John Wiley, 1975.
- HASHIMOTO, S.; ROMERO, R.; MANTOVANI, J. R. S. Efficient linear programming algorithm for the transmission network expansion planning problem. *IEE Proceeding in Generation, Transmission and Distribution*, New York, v. 5, n. 150, p. 536–542, 2003.
- KALTENBATCH, J. C.; PESHON, J.; GEHRIG, E. H. A mathematical optimization technique for the expansion of electrical power transmission systems. *IEEE Transaction on Power Apparatus and Systems*, New York, v. 1, n. PAS-89, p. 113–119, 1970.
- KNIGHT, U. G. W. The logical design of electrical networks using linear programming methods. *IEE Proceedings*, New York, A, n. 107, p. 306–314, 1960.
- KOVACEVIĆ, V.; CANGALović, M.; ASIĆ, M.; IVANOVIĆ, L. Tabu search methodology in global optimization. *Computers and Mathematics with Applications*, New York, n. 37, p. 125–133, 1999.
- LAGUNA, M. *Scatter search*. 1999. Disponível em: <http://leeds-faculty.colorado.edu/Laguna/articles/ss2.pdf>. Acesso em: 12 jan. 2007.
- LAGUNA, M.; ARMENTANO, V. A. *Lessons from applying and experimenting with scatter search*. 2005. Disponível em: www.leeds-faculty.colorado.edu/laguna/articles/sslessons.pdf. Acesso em: 04 out. 2006.
- LAGUNA, M.; MARTI, R. *Scatter Search: Methodology and Implementations in C*. Boston: Kluwer Academic Publishers, 2003.
- LATORRE-BAYONA, G.; PÉRES, I. J. Chopin, a heuristic model for long term transmission expansion planning. *IEEE Transaction on Power Systems*, New York, v. 4, n. 9, p. 1886–1894, 1994.

- LATORRE, G.; CRUZ, R. D.; AREIZA, J. M.; VILLEGAS, A. Classification of publication and model on transmission expansion planning. *IEEE Transaction on Power Systems*, New York, v. 2, n. 18, p. 938–946, 2003.
- LEE, S. T.; HOCKS, K. L.; HNYLICZA, E. Transmission expansion by branch and bound integer programming with optimal cost-capacity curves. *IEEE Transaction on Power Apparatus and Systems*, New York, v. 93, n. PAS, p. 1390–1400, 1974.
- LEVI, V. A.; CALOVIC, M. S. A new decomposition based method for optimal expansion planning of large transmission networks. *IEEE Transaction on Power Systems*, New York, v. 3, n. 6, p. 937–943, 1991.
- LOUDIN, S.; BOIZUMAULT, P. Vns/lfs + cp: A hybrid method for constraint optimization in anytime contexts. *MICS'2001*, Porto, p. 761–765, 2001.
- MARTI, M. L. R.; CAMPOS, V. Intensification and diversification with elite tabu search solutions for the linear ordering problem. *Computers and Operations Research*, New York, n. 26, p. 1217–1230, 1999.
- MARTI, R. Multi-start methods. In: GLOVER, F.; KOCHENBERGER, G. A. (Ed.). *Handbook of Metaheuristics*. Dordrecht: Kluwer, 2003. p. 355–368. 2002.
- MARTI, R.; LAGUNA, M. Scatter search: Basic design and advanced strategies. *Revista Iberoamericana de Inteligência Artificial*, Amsterdam, n. 19, p. 123–130, 2003.
- MARTI, R.; LAGUNA, M.; CAMPOS, V. Scatter search vs. genetic algorithms: An experimental evaluation with permutation problems. 2005. Disponível em: www.leeds-faculty.colorado.edu/laguna/articles/ss_vs_ga.pdf. Acesso em: 01 mar. 2003.
- MARTINS, S. L.; RESENDE, M. G. C.; RIBEIRO, C.; PARDALOS, P. A parallel grasp for the steiner tree problem in graphs using a hybrid local search strategy. *Journal of Global Optimization*, Dordrecht, n. 17, p. 267–283, 2000.
- MIASAKI, C. T. *Planejamento da expansão de sistemas de transmissão de energia elétrica utilizando controladores FACTS*. Tese (Doutorado) — DEE-UNESP, Ilha Solteira, 2006.
- MLADENović, N. A variable neighborhood algorithm - a new metaheuristic for combinatorial optimization. *Abstracts of Papers Presented at Optimization Days, 12*, Montreal, 1995.
- MLADENović, N.; HANSEN, P. Variable neighborhood search. *Computers Ops. Research*, New York, v. 11, n. 24, p. 1097–1100, 1997.
- MONTICELLI, A. et al. Interactive transmission network planning using a least-effort criterion. *IEEE Transactions on Power Apparatus and Systems*, New York, v. 10, n. PAS-101, p. 3319–3325, 1985.
- MURTAGH, B. A.; SAUNDERS, M. A. *Minos 5.4 user's guide*. New York: Stanford University, 1995.
- OCHI, L. S.; SILVA, M. B.; DRUMMOND, L. Metaheuristics based on grasp and vns for solving traveling purchaser problem. *MIC'2001*, Porto, p. 489–494, 2001.
- OLIVEIRA, G. C.; COSTA, A. P.; BINATO, S. Large scale transmission network planning using optimization and heuristic techniques. *IEEE Transactions on Power Systems*, New York, v. 4, n. 10, p. 1828–1834, 1995.

- ORTIZ, J. M. A. *Metodologia de expansão automática da transmissão utilizando um algoritmo de busca tabu*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, 1997.
- PEARL, J. *Probabilistic reasoning in intelligent system*. San Francisco: Morgan and Kaufman, 1998.
- PEREIRA, M. V. F. *Análise de sensibilidade no planejamento da expansão dos sistemas de geração-transmissão*. Tese (Doutorado) — COPPE - Universidade Federal do Rio de Janeiro, Rio de Janeiro, 1985.
- PEREIRA, M. V. F.; PINTO, L. M. V. G.; OLIVEIRA, G. C.; CUNHA, S. H. F. *Composite generation-transmission expansion planning*. 1987. Technical Report EPRI. RP 2473-9.
- PINEDO, M. *Scheduling: theory, algorithms, and systems*. 2nd edition. ed. Glenview: Prentice Hall, 2001.
- PINTO, L. M. V. G.; NUNES, A. A model for optimal transmission expansion planning. *Proceedings of the Power System Computing Conference*, Graz, n. 1, p. 13–23, 1990.
- QU, R.; BURKE, E. K. Hybrid variable neighborhood hyperheuristics for exam timetabling problems. *MIC'2005*, Porto, p. 22–26, 2005.
- REEVES, C. *Modern Heuristic Techniques for Combinatorial Problems*. Berkshire: McGraw-Hill Book Company, 1995.
- RIBEIRO, C. C.; SOUZA, M. C. Variable neighborhood descent for the degree-constrained minimum spanning tree problem. *Discrete Applied mathematics*, Amsterdam, n. 118, p. 43–54, 2002.
- RIBEIRO, C. C.; UCHOA, E.; WERNEK, R. A hybrid grasp with perturbation for the steiner problem in graphs. *INFORMS Journal of Computing*, Linthicum, n. 14, p. 228–246, 2002.
- RIDER, M.; GARCIA, A.; ROMERO, R. Power systems network expansion planning using a model. *IEEE Generation, Transmission and Distribution*, New York, n. 1, p. 731–742, 2007.
- ROMERO, R. *Um método de decomposição para o planejamento a longo prazo de sistemas de transmissão*. Tese (Doutorado) — UNICAMP, Campinas, 1993.
- ROMERO, R.; GALLEGO, R. A.; MONTICELLI, A. Transmission system expansion planning by simulated annealing. *IEEE Transaction on Power System*, New York, v. 1, n. 11, p. 364–369, 1996.
- ROMERO, R.; MONTICELLI, A. A hierarchical decomposition approach for transmission network expansion planning. *IEEE Transactions on Power Systems*, New York, v. 1, n. 9, p. 373–380, 1994.
- ROMERO, R.; MONTICELLI, A. A zero-one implicit enumeration method for optimizing investments in transmission expansion planning. *IEEE Transactions on Power Systems*, New York, v. 3, n. 9, 1994.
- ROMERO, R.; MONTICELLI, A.; GARCIA, A.; HAFFNER, S. Test systems and mathematical models for transmission network expansion planning. *IEE Proceedings Generation, Transmission and Distribution*, New York, v. 1, n. 149, 2002.

ROMERO, R.; ROCHA, R.; MANTOVANI, M.; MANTOVANI, J. R. S. Analysis of heuristics algorithms for the transportation model in static and multistage planning in network expansion systems. *IEEE Proceedings - Generation, Transmission and Distribution*, New York, v. 5, n. 150, p. 521–526, 2003.

SCOTT, B.; MARINHO, J. L. Linear programming for power system network security applications. *IEEE Transaction on Power and Apparatus Systems*, New York, v. 3, 1979.

SILVA, E. L.; AREIZA, J. M.; OLIVEIRA, G. C.; BINATO, S. Transmission network expansion planning under a tabu search approach. *IEEE Transaction on Power Systems*, New York, v. 1, n. 16, p. 62–68, 2001.

SILVA, E. L.; GIL, H. A.; AREIZA, A. M. Transmission network expansion planning under an improved genetic algorithm. *IEEE Transaction on Power Systems*, New York, v. 3, n. 15, p. 1168–1175, 2000.

SILVA, I.; RIDER, M.; MURARI, C. A.; ROMERO, R. Expansion planning considering uncertainties in demand. *IEEE Transaction on Power Delivery Systems*, New York, n. 21, p. 1565–1573, 2006.

SILVA, M. B.; DRUMMOND, L. M. A.; OCHI, L. S. Metaheurísticas grasp+vns para a solução de problemas de otimização combinatória. In: CONGRESSO BRASILEIRO DE PESQUISA OPERACIONAL, 32, 2000, Viçosa. In: *Anais ...*, Viçosa,. 2000.

SOSA, N. G. M.; GALVÃO, R. D. Busca dispersa aplicada ao problema de roteamento de veículos. In: CONGRESSO BRASILEIRO DE PESQUISA OPERACIONAL, 37, 2005, Gramado. In: *Anais ...*, Gramado,. 2005.

VILLASANA, R. *Transmission network planning using linear and linear mixed integer programming*. Tese (Doutorado) — Rensselaer Polytechnic Institute, New York, 1984.

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

WHITTAKER, R. A fast algorithm for the greedy interchange for large-scale clustering and median location problems. *INFOR*, Ottawa, n. 21, p. 95–108, 1983.

APÊNDICE A - Dados dos Sistemas Testes

A.1 Sistema IEEE (24 Barras/41 Caminhos)

Tabela A.1: Dados de Linhas - IEEE 24 barras.

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
1	01-02	1	0.0139	175	3000
2	01-03	1	0.2112	175	55000
3	01-05	1	0.0845	175	22000
4	02-04	1	0.1267	175	33000
5	02-06	1	0.1920	175	50000
6	03-09	1	0.1190	175	31000
7	03-24	1	0.0839	400	50000
8	04-09	1	0.1037	175	27000
9	05-10	1	0.0883	175	23000
10	06-10	1	0.0605	175	16000
11	07-08	1	0.0614	175	16000
12	08-09	1	0.1651	175	43000
13	08-10	1	0.1651	175	43000
14	09-11	1	0.0839	400	50000
15	09-12	1	0.0839	400	50000
16	10-11	1	0.0839	400	50000
17	10-12	1	0.0839	400	50000
18	11-13	1	0.0476	500	66000
19	11-14	1	0.0418	500	58000
20	12-13	1	0.0476	500	66000
21	12-23	1	0.0966	500	134000
22	13-23	1	0.0865	500	120000

continua na próxima página

Tabela A.1: Dados de Linhas - IEEE 24 barras
(continuação).

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
23	14-16	1	0.0389	500	54000
24	15-16	1	0.0173	500	24000
25	15-21	2	0.0490	500	68000
26	15-24	1	0.0519	500	72000
27	16-17	1	0.0259	500	36000
28	16-19	1	0.0231	500	32000
29	17-18	1	0.0144	500	20000
30	17-22	1	0.1053	500	146000
31	18-21	2	0.0259	500	36000
32	19-20	2	0.0396	500	55000
33	20-23	2	0.0216	500	30000
34	21-22	1	0.0678	500	94000
35	01-08	0	0.1344	500	35000
36	02-08	0	0.1267	500	33000
37	06-07	0	0.1920	500	50000
38	13-14	0	0.0447	500	62000
39	14-23	0	0.0620	500	86000
40	16-23	0	0.0822	500	114000
41	19-23	0	0.0606	500	84000

Tabela A.2: Dados de Barras - IEEE 24 barras.

Barra	Geração (MW) Plano P1	Geração (MW) Plano P2	Carga (MW)
1	465	521	324
2	576	521	291
3	0	0	540
4	0	0	222
5	0	0	213
6	0	0	408
7	722	812	375
8	0	0	513
9	0	0	525
10	0	0	585
11	0	0	0
12	0	0	0
13	1424	1599	795
14	0	0	582
15	645	581	951
16	465	419	300
17	0	0	0
18	1200	718	999
19	0	0	543
20	0	0	384
21	1200	1077	0
22	900	900	0
23	954	1403	0
24	0	0	0

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

Tabela A.4: Dados de Linhas - Sul Brasileiro.

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
1	01-07	1	0.0616	270	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	20	10505
18	17-19	1	0.0061	20	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

continua na próxima página

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

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
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
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	20	10889
51	04-11	0	0.2246	240	14247
52	05-11	0	0.0915	270	6167
53	46-06	0	0.0128	20	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	20	21121
60	25-32	0	0.0319	1400	37109
61	31-32	0	0.0046	20	7052
62	28-31	0	0.0053	20	7819
63	28-30	0	0.0058	20	8331

continua na próxima página

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

Nº	Ramo	Linhas existentes	Reatância (pu)	Capacidade (MW)	Custo 10 ³ US\$
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	20	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

Tabela A.3: Dados de Barras - Sul Brasileiro.

Barra	Geração (MW)- Plano SR	Geração (MW)-Plano CR	Carga
1	0	0	0
2	0	0	443.1
3	0	0	0
4	0	0	300.7
5	0	0	238
6	0	0	0
7	0	0	0
8	0	0	72.2
9	0	0	0
10	0	0	0
11	0	0	0
12	0	0	511.9
13	0	0	185.8
14	1257	944	0
15	0	0	0
16	2000	1366	0
17	1050	1000	0
18	0	0	0
19	1670	773	0
20	0	0	1091.2
21	0	0	0
22	0	0	81.9
23	0	0	458.1
24	0	0	478.2
25	0	0	0
26	0	0	231.9
27	220	54	0
28	800	730	0
29	0	0	0
30	0	0	0
31	700	310	0
32	500	450	0
33	0	0	229.1
34	748	221	0
35	0	0	216
36	0	0	90.1
37	300	212	0
38	0	0	216
39	600	221	0
40	0	0	262.1
41	0	0	0
42	0	0	1607.9
43	0	0	0
44	0	0	79.1
45	0	0	86.7
46	700	599	0
Total	10545	6880	6.880

APÊNDICE B - Artigos Publicados

TAGLIALENHA, S. L. S.; ROMERO, R.; VENDRAMINI, E. '*Busca em Vizinhança Variável Aplicada ao Problema de Planejamento da Expansão do Sistema de Transmissão com Modelo de Transportes*', In Anales del XIII Congreso Latino-Iberoamericano de Investigación Operativa, Montevideo, Uruguay, 2006.

TAGLIALENHA, S. L. S.; ROMERO, R. '*Busca em Vizinhança Variável para o Problema de Planejamento da Expansão do Sistema de Transmissão de Energia Elétrica*', In Anais do XXXVIII Simpósio Brasileiro de Pesquisa Operacional, Goiânia, Brasil, p. 510-521, 2006.

TAGLIALENHA, S. L. S.; ROMERO, R.; SEGURA, S. C. '*Planificación de la expansión de la red de transmisión con búsqueda dispersa*', In Anales del XIII Congreso Internacional de Ingeniería Electrónica, Eléctrica y de Sistemas, Lima, Peru, 2006.

TAGLIALENHA, S. L. S.; CARREÕ, E. M.; ROMERO, R. '*Busca Dispersa no Problema de Planejamento da Expansão do Sistema de Transmissão de Energia Elétrica*', In Anais do XII ERIAC - Encontro Regional Ibero-americano do CIGRÉ, Foz do Iguaçu, Brasil, 2007.

Classificação de Segurança Uso restrito aos examinadores		Documento no.	
Data (mês e ano)Abril de 2008		Projeto no.	
Título e subtítulo NOVAS APLICAÇÕES DE METAHEURÍSTICAS NA SOLUÇÃO DO PROBLEMA DE PLANEJAMENTO DA EXPANSÃO DO SISTEMA DE TRANSMISSÃO DE ENERGIA ELÉTRICA			No. do volume No. da parte
Título do projeto			
Entidade Executora (autor coletivo)		Autor(es) Silvia Lopes de Sena Taglialenha	
Entidade patrocinada (cliente ou destinatário principal) PPGEE - UNESP Campus de Ilha Solteira			
Resumo Neste trabalho apresenta-se o Problema de Planejamento da Expansão de Sistemas de Transmissão de Energia Elétrica, sua formulação matemática e uma análise da complexidade da solução de cada modelo. Descreve-se o estado da arte em planejamento, abordando as principais técnicas utilizadas para sua resolução e as publicações mais importantes. Realiza-se uma revisão da Metaheurística Busca Dispersa, um método evolutivo que combina sistematicamente conjuntos de soluções para se obter soluções melhores e se ilustra sua aplicação no Problema de Planejamento da Expansão do Sistema de Transmissão de Energia Elétrica. O mesmo faz-se com a Metaheurística Busca em Vizinhança Variável, um técnica baseada em mudanças sistêmicas de estruturas de vizinhança dentro de um algoritmo de busca local. Descreve-se os detalhes de implementação no problema de planejamento da expansão do sistema de transmissão de energia elétrica. Expõe-se testes realizados com o modelo de transportes e com o modelo DC. Nos testes realizados para o modelo DC também são utilizados o modelo de transportes e o modelo híbrido como parte do processo de otimização. Os resultados são comparados com outras metaheurísticas evolutivas e de busca local.			
Palavras-chave Planejamento da expansão de redes. Busca Dispersa. Vizinhança Variável. Metaheurísticas.			
No. da edição	No. de páginas 136	ISSN (para relatórios publicados)	Classificação (CDC ou CDD)
Distribuidor		Número de exemplares 5 (cinco)	Preço -
Observações Trabalho apresentado ao Programa de Pós-Graduação em Engenharia Elétrica, UNESP - Campus de Ilha Solteira, como parte dos requisitos para a obtenção do título de Doutor em Engenharia Elétrica.			