

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE SISTEMAS

Buscas Informadas Baseadas em Grafos para a Minimização das Perdas em Sistemas de Distribuição de Energia Elétrica

Celso Cavellucci

Orientador: Prof. Dr. Christiano Lyra Filho

Este exemplar corresponde a redação final da tese
defendida por Celso Cavellucci
e aprovada pela Comissão
Julgada em 16 / 12 / 99
Orientador

Tese apresentada à Faculdade de Engenharia Elétrica e
de Computação da Universidade de Campinas –
UNICAMP, como parte dos requisitos exigidos para
obtenção do título de Doutor em Engenharia Elétrica.

Dezembro 1998



16/12/98

UNIDADE	BC
N.º CHAMADA:	T/UNICAMP
	C315b
V.	Ex.
TOMBO BC/	39918
PROC.	278/00
C	☐
D	☒
PREÇO	R\$ 11,00
DATA	12/01/00
N.º CPD	

CM-00137B35-B

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

C315b

Cavellucci, Celso

Buscas informadas baseadas em grafos para minimização das perdas em sistemas de distribuição de energia elétrica / Celso Cavellucci.--Campinas, SP: [s.n.], 1998.

Orientador: Christiano Lyra Filho.

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

1. Sistemas de energia elétrica. 2. Otimização combinatória. 3. Inteligência artificial. 4. Energia elétrica - Distribuição. I. Lyra Filho, Christiano. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

‘O computador não erra’. Erra, e muito e, gravemente. Mas, admitindo-se que não erra, esse é o seu maior erro. A humanidade, o pouco que avançou, avançou, porque o cérebro humano não tem certeza. Experiência e erro é o seu destino!

Millör Fernandes

À Lia, Murilo e Bruno

Aos meus pais, Irany e Mercedes

AGRADECIMENTOS

Ao Christiano pela dedicada orientação, apoio e amizade durante todos esses anos.
Ao Welfane, pela amizade e pelas indispensáveis contribuições e discussões.
Ao Gerard Authié, por sua disponibilidade de compartilhar seu conhecimento.
À Lia, Murilo e Bruno pela compreensão, alegria e amor, proporcionando-me paz e tranquilidade.

Ao Simoni e Roberval, pelo apoio incondicional, sem o qual não seria possível a realização deste trabalho.

À Ana e Silvia pela preciosa colaboração na busca das referências bibliográficas.

À Ana pela paciente revisão da bibliografia.

E a todas as pessoas que contribuíram, direta ou indiretamente, para a realização deste trabalho.

RESUMO

Este trabalho apresenta uma nova abordagem para a minimização das perdas em sistemas de distribuição de energia elétrica. A minimização das perdas é obtida por meio da reconfiguração das redes de distribuição. Considerando que essas redes operam com uma configuração radial, busca-se a árvore recobridora do grafo que representa a rede de distribuição, que minimize as perdas de energia e satisfaça as restrições de demanda e os limites do fluxo de corrente nas linhas; em outras palavras, trata-se de uma generalização do problema da árvore recobridora de custo mínimo. A generalização é devida às variações nos custos dos arcos com a mudança da configuração. A solução ótima global desse problema combinatório é obtida por um procedimento recursivo em duas fases, onde se combina as técnicas de fluxo não lineares e estratégias de busca usadas na área de inteligência artificial. Na primeira etapa a restrição de operação radial da rede é relaxada, levando a uma solução otimista para o problema; as informações desta solução são usadas na segunda etapa, onde busca-se a solução factível de custo mínimo. As etapas são repetidas, até que a configuração de rede radial de mínimas perdas seja encontrada. O procedimento recursivo é controlado por estratégias de busca inteligentes para contornar a explosão exponencial do esforço computacional. Três procedimentos de busca informada foram concebidos para obter uma árvore recobridora de perdas mínimas: *backtracking informado*, *backtracking heurístico* e *algoritmo A**. Apresenta-se também procedimentos de paralelização dos algoritmos. Estudos de caso são discutidos, indicando as possibilidades e limitações da abordagem proposta.

ABSTRACT

This thesis presents a new approach to minimize losses in electrical energy distribution systems. The loss minimization is accomplished with the reconfiguration of the distribution network. Since distribution networks must operate radially, the problem can be regarded as a generalization of the minimum spanning tree problem. It seeks a spanning tree for a graph that represents the distribution network which minimizes losses while meeting loads and satisfying constraints on line flow capacities – the generalization is due to variation in the costs as the network configuration changes. A global optimum to this combinatorial problem is found through a recursive two-step procedure, merging non-linear network flow techniques with intelligent search strategies. Radial operation is relaxed in a first step, leading to an optimistic solution; information from the optimistic solution are used to approach feasibility in the second step. Both steps are repeated, driven by intelligent search techniques to cope with computational intractability. Three informed search procedures were conceived to obtain a minimum-loss spanning tree: enhanced backtracking, heuristic backtracking and A* algorithm. Parallel implementations of the algorithms are also presented. Case studies are discussed, providing guidelines about the possibilities and limitations of the proposed approach.

ÍNDICE

RESUMO	iii
ABSTRACT	iv
APRESENTAÇÃO	1
CAPÍTULO 1 - INTRODUÇÃO	5
1.1. Reconfiguração de Rede	5
1.2. O Sistema de Potência	6
1.3. Sistema de Distribuição de Energia Elétrica	8
1.4. Estados do Sistema de Distribuição	9
1.4.1. Operação na Emergência e na Restauração	10
1.4.2. Operação no Estado Normal	10
1.5. Perdas de Energia na Distribuição	11
1.6. Configuração da Rede de Mínimas Perdas	13
1.7. Abordagens para o problema de Reconfiguração	16
1.7.1. Procedimento em duas etapas	17
1.7.2. Branch-Exchange	17
1.7.3. Programação Matemática	18
1.7.4. Otimização Combinatória	19
1.7.5. Abordagem por Construção	19
1.7.6. Rede Neurais	20
1.7.7. Sistema Especialista	20
1.7.8. Heurísticas Mistas	21
1.8. Nova Abordagem para Minimização das Perdas	22
CAPÍTULO 2 - MODELO PARA REDES DE DISTRIBUIÇÃO	23
2.1. Representação da Rede Primária	23
2.2. Modelo de Grafo	23
2.3. Representação da Rede	26
2.3.1. Listas Auxiliares	27
2.3.2. Rede em Operação	27
2.4. Comentários	29
CAPÍTULO 3 - MINIMIZAÇÃO DAS PERDAS E ESTRATÉGIA DE SOLUÇÃO	31
3.1. Introdução	31
3.2. Equações do Fluxo de Potência para Redes Radiais	33
3.3. Fluxo de Potência Aproximado	35
3.4. Formulação do Problema	37
3.5. Estratégia de Solução	38

3.5.1. Solução Otimista	39
3.5.2. Método do Gradiente Reduzido para Fluxos em Redes	40
3.5.3. Comentários	42
CAPÍTULO 4 - CONCEITOS DE IA NA BUSCA DE SOLUÇÕES ÓTIMAS.....	43
4.1. Introdução	43
4.2. Busca no Espaço de Estados	43
4.3. Procedimento de Busca	45
4.3.1. Algoritmo básico da busca irrevogável	46
4.3.2. Algoritmo Backtracking	47
4.3.3. Algoritmo “Best-First”	50
4.3.4. Busca em Grafo.....	51
4.4. Busca da Árvore Recobridora para Redes de Distribuição	52
4.5. Busca Irrevogável.....	53
4.6. Buscas Backtracking Informado	54
4.7. Busca Backtracking Heurístico.....	56
4.8. Algoritmo A*	58
4.8.1. Construção da função de avaliação heurística para o problema P	60
4.8.2. Monotocidade da função heurística	63
4.9. Paralelização dos algoritmos	65
CAPÍTULO 5 - ASPECTOS DE IMPLEMENTAÇÃO E RESULTADOS COMPUTACIONAIS.....	66
5.1. Introdução	66
5.2. Algoritmo para Busca Irrevogável.....	67
5.3. Algoritmo Backtracking Informado	70
5.4. Algoritmo Backtracking Heurístico	72
7.5. Algoritmo A*	74
7.6. Comentários.....	77
CAPÍTULO 6 - MÉTODOS DE BUSCA INFORMADAS E PROCESSAMENTO PARALELO	81
6.1. Introdução	81
6.2. Backtracking Paralelo	81
6.3. Implementação do Algoritmo do Backtracking Paralelo	83
6.4. Estudos de Casos	85
6.5. Análise de Desempenho de Algoritmos Paralelos.....	91
6.6. Paralelização do Algoritmo A*	94
CAPÍTULO 7 - COMENTÁRIOS FINAIS E CONCLUSÕES.....	97
APÊNDICE 1 - OBTENÇÃO DOS CONJUNTOS M_o E M_t	101
A1.1. Introdução	101
A1.2. Construção da Lista P_o	102
A1.3. Construção das Listas L_T e P_T	103
APÊNDICE 2 - FERRAMENTAS COMPUTACIONAIS PARA OPERAÇÃO DA REDE.....	105
A2.1. Introdução	105
A2.2. Configurador de Rede.....	107
A2.3. Simulador de Manobras - SIMA.....	111

A2.4. Sistema Baseado em Conhecimento SBA-MANOBRA118

A2.5. Comentários120

APÊNDICE 3- PUBLICAÇÕES VINCULADAS AO TRABALHO121

BIBLIOGRAFIA123

FIGURAS

Figura 1 Sistema de Potência.....	7
Figura 2. Sistema de Distribuição	8
Figura 3. Estados da rede de distribuição	9
Figura 4. Fluxo e perdas de energia elétrica no sistema de transmissão e distribuição (CODI, 1997)	12
Figura 5. Distribuição do total das perdas de energia (CODI 1997)	13
Figura 6. Unifilar do sistema de distribuição primário	14
Figura 7. Configuração alterada do Unifilar	15
Figura 8. Rede de distribuição exemplo	16
Figura 9. Diagrama simplificado da rede primária	24
Figura 10 Modelo de grafo para redes de distribuição	25
Figura 11. Modelo de grafo para a rede em operação	25
Figura 12. Armazenamento na estrutura arco-orientada	26
Figura 13. Estrutura de armazenamento em preorder da árvore A	28
Figura 14. Diagrama simplificado da rede de distribuição em operação	32
Figura 15. Rede radial de um alimentador	33
Figura 16. Árvore de busca criada pelo processo de expansão	45
Figura 17. Busca em profundidade	48
Figura 18. Árvore de busca gerada para a solução do problema P	53
Figura 19. Função heurística.....	57
Figura 20. Partição G_1'' e G_2''	61
Figura 21. Condição de monotonicidade	63
Figura 22. Transição da configuração n para a n_1	64
Figura 23. Diagrama esquemático da REDE47	66
Figura 24. Diagrama esquemático da rede BWU01	67
Figura 25. Algoritmo Busca Irrevogável	68
Figura 26. Algoritmo Backtracking Informado	71
Figura 27. Algoritmo Busca Heurística	73

Figura 28. Algoritmo A*	75
Figura 29. Árvore de busca parcial do problema de reconfiguração da rede para mínimas perdas	82
Figura 30. Implementação do processamento paralelo na estrutura <i>pai-filho</i>	82
Figura 31. REDE 47 - Backtracking Informado	87
Figura 32. BWU01 Backtracking Informado	87
Figura 33. BAURU 9 Backtracking Informado	88
Figura 34. REDE 47 Backtracking Heurístico	88
Figura 35. BWU01 Backtracking Heurístico	89
Figura 36. BAURU 9 Backtracking Heurístico	89
Figura 37. Árvore de busca parcial gerada pelo algoritmo A*	94
Figura 38. Armazenamento na estrutura arco-orientada	101
Figura 39. Diagrama funcional do COD	106
Figura 40. Principais componentes do Configurador de Rede	108
Figura 41. Diagrama esquemático da rede de distribuição em operação	109
Figura 42. Defeito na subestação	110
Figura 44. Restauração parcial da rede	112
Figura 45. Módulos do SIMA	113
Figura 46. Etapas para validação de uma manobra	113
Figura 47. Desenho do unifilar da rede de distribuição	114
Figura 48. Detalhamento da rede de distribuição	115
Figura 49. Dados dos elementos da rede de distribuição	116
Figura 50. Esboço do plano de manobras	117
Figura 51. Diagrama de blocos do SBA-MANOBRA	119

TABELAS

Tabela 1. Percentual das perdas anuais de energia no setor elétrico (CODI, 1997).	13
Tabela 2 Armazenamento dos parâmetro físicos e elétricos.....	27
Tabela 3. Características das Redes do estudo de casos	67
Tabela 4. Resultados da busca irrevogável pelo valor do fluxo.....	69
Tabela 5. Resultados da busca irrevogável pelo valor do gradiente.....	69
Tabela 6. Resultados para a busca Bactraking Informado	72
Tabela 7. Resultados para a busca Backtracking Heurístico	74
Tabela 8. Resultados para a busca A^*	77
Tabela 9. Resultados do algoritmo <i>backtracking informado</i> usando critérios de ordenação diferentes.....	78
Tabela 10. Valores no topo da lista ABERTO	79
Tabela 11. Resultados para a busca A^*	80
Tabela 12. Exemplos de funções do PVM.....	85
Tabela 13. Características das redes do estudos de casos para o algoritmo paralelo	85
Tabela 14. Resultados da implementação serial	86
Tabela 15. Número de nós explorados em função do parâmetro δ	91
Tabela 16. <i>Speedup</i> para o <i>backtracking informado</i>	92
Tabela 17. <i>Speedup</i> para o <i>backtracking heurístico</i>	93
Tabela 18. Lista de arcos em ordem crescente do nó origem.	102
Tabela 19. Lista P_a de apontadores para os arcos origens da lista de arcos.....	103
Tabela 20. Lista L_T para a ordenação dos arcos pelos nós terminais.....	103
Tabela 21. Lista P_T dos apontadores para os arcos terminais.....	104
Tabela 22 Funções e opções do menu	114
Tabela 23. Simbologia adotada para as entidades da rede	115

APRESENTAÇÃO

Este trabalho propõe procedimentos de busca para obter a configuração de mínimas perdas para redes de distribuição de energia elétrica. Sendo a operação do sistema de distribuição radial, o problema pode ser formulado como uma generalização do problema da árvore recobridora de custo mínimo (Ahuja et al., 1993). A generalização é devido à variação do custo do arco com a mudança da configuração da rede. A abordagem adotada para a solução deste problema utiliza técnicas de otimização de fluxos não lineares em redes (Lyra e Tavares, 1988), integradas ao conceito de estratégias de busca, usados na área de inteligência artificial (Pearl, 1984).

A configuração da rede de distribuição pode ser alterada para satisfazer as restrições de carga (somatório das demandas de energia elétrica dos consumidores do sistema), ou as restrições operacionais, como o valor do limite inferior da tensão ou a capacidade de fluxo das linhas de distribuição. Por exemplo, na ocorrência de um defeito no sistema, alguns consumidores ficam sem o fornecimento de energia elétrica, violando a restrição de carga. Neste caso, configura-se a rede para isolar o defeito e reestabelecer o fornecimento de energia elétrica para a parte do sistema em condições de operação, sem violar as restrições operacionais. O conjunto de manobras necessárias para configurar a rede é denominado plano de manobras.

A configuração de rede pode ser definida usando um simulador de manobras (Cavellucci e Lyra, 1995). Neste caso, o operador fornece um plano de manobras (seqüência de aberturas e fechamentos de chaves) e o simulador apresenta o estado da rede para esta configuração. Este procedimento se repete, até que o operador encontre uma configuração que satisfaça critérios preestabelecidos, por exemplo, uma redução significativa nas perdas de energia. Note que o simulador não realiza nenhum processo de busca da configuração. Depende do operador avaliar se deve

ou não testar uma nova hipótese, caracterizando o simulador como uma *ferramenta passiva*.

A inclusão do armazenamento e tratamento de conhecimento sobre manobras, transforma o simulador numa *ferramenta ativa*, ou seja, além de calcular o estado da rede, é possível sugerir planos de manobras para isolamento do defeito e restabelecimento da rede, que atendam a critérios preestabelecidos. Como exemplo desta classe de ferramentas computacionais, pode-se citar o sistema especialista em manobras de rede de distribuição (Cavellucci, 1989; Cavellucci e Lyra, 1993). Ele foi implementado utilizando conceitos de inteligência artificial (sistemas de produção) para inferir planos de manobras a partir de dados de uma contingência ocorrida na rede.

Caracteriza-se o estado normal do sistema de distribuição quando as restrições de carga e operacionais estão satisfeitas. Neste estado, busca-se operar o sistema com a maior eficiência possível. Por exemplo, pode-se operar o sistema na configuração de mínimas perdas, satisfazendo as restrições de carga e operacionais. Como a demanda dos consumidores de energia elétrica varia com o tempo, atendê-la de maneira eficiente implica em configurar a rede para mínimas perdas a cada variação significativa desta demanda.

Neste trabalho são apresentados os procedimentos de busca *irrevogável*, *backtracking informado*, *backtracking heurístico* e o *algoritmo A**, para resolver o problema da configuração de rede de mínimas perdas. A busca *irrevogável* guia-se por uma heurística, baseada no ótimo local, para decidir a direção a ser tomada no caminho da solução do problema. Uma vez decidido o caminho, esta estratégia não considera a possibilidade de explorar caminhos alternativos.

Na procura da solução ótima do problema adota-se a estratégia de busca por tentativa. Ao contrário do mecanismo da busca irrevogável, a estratégia do controle da busca por tentativa considera todos os caminhos alternativos. A explosão combinatória é inerente a esta abordagem. Para minimizar seus efeitos, o procedimento de busca *backtracking informado* (uma das classes de procedimentos em

IA) utiliza um procedimento de poda dos caminhos menos promissores, que usa conhecimentos do domínio do problema. Os resultados apresentados pelo algoritmo implementado indicam a possibilidade de seu paralelismo, estudado em outros trabalhos (Tão, 1997; Tão, Cavellucci e Lyra 1998)

Para antecipar a poda dos caminhos menos promissores, o procedimento de busca *backtracking heurístico* usa uma estimativa do valor das perdas na configuração radial da rede de distribuição. Esta estimativa é incluída no procedimento de poda, por meio de uma função de avaliação heurística, melhorando sensivelmente os resultados obtidos pela busca *backtracking informado*.

O algoritmo da classe A*, pertencente a uma categoria de algoritmos muito eficientes na área de inteligência artificial, é guiado por uma função de avaliação heurística que garante a obtenção da solução ótima para o problema.

O trabalho está organizado em 7 capítulos. O primeiro capítulo apresenta breve descrição dos sistemas de potência (constituídos pela geração, transmissão e distribuição de energia elétrica), os estados que estes sistemas podem assumir, uma discussão sobre perdas de energia elétrica, abordagens existentes para resolver o problema de configuração da rede para redução de perdas e um resumo da abordagem proposta.

O capítulo 2 descreve o modelo de grafo utilizado para representar o sistema de distribuição de energia elétrica. São apresentadas as estruturas de dados necessárias para o armazenamento do grafo, que representa o sistema de distribuição, e da árvore, que representa a rede radial.

O capítulo 3 apresenta a formulação do problema de minimização das perdas para um sistema de distribuição de energia elétrica e a estratégia de solução adotada. São descritas as equações do fluxo de potência e das perdas para redes radiais, utilizadas no problema de minimização das perdas. Após a discussão da formulação do problema é apresentada a estratégia de solução, como um procedimento recursivo de duas etapas.

Os conceitos de busca da área de inteligência artificial, utilizados na obtenção da solução do problema de minimização das perdas, são discutidos no capítulo 4. A representação no espaço de estados é utilizada para descrever os procedimentos de busca desenvolvidos no trabalho. A partir de um procedimento básico de busca, são desenvolvidos os procedimentos *busca irreversível*, *backtracking informado*, *backtracking heurístico* e o *algoritmo A**.

No capítulo 5 são apresentados e discutidos os resultados obtidos pelos algoritmos implementados, com uma descrição resumida de cada um, seguida de comentários sobre a implementação.

O capítulo 6 descreve a paralelização dos procedimentos de busca *backtracking informado* e *backtracking heurístico*, apresentados por Tão (1997). Apresenta também uma proposta para paralelização do *algoritmo A**.

Comentários finais, conclusões e propostas para novos trabalhos são apresentados no capítulo 7.

No apêndice 1 apresenta-se detalhes sobre as listas auxiliares acrescentadas à estrutura de dados descrita no capítulo 2. Essas listas são utilizadas para percorrer convenientemente os elementos do grafo que representa a rede de distribuição de energia elétrica. No apêndice 2 descreve-se três ferramentas computacionais para auxiliar na operação da rede de distribuição: configurador de rede, simulador de manobras e sistema baseado em conhecimento para sugerir planos de manobras. No apêndice 3 lista-se as publicações vinculadas a este trabalho de tese.

CAPÍTULO 1

INTRODUÇÃO

1.1. RECONFIGURAÇÃO DE REDE

A configuração da rede de distribuição de energia elétrica é objeto de estudo nos planejamentos de longo, médio e curto prazo. O objetivo do planejamento de longo prazo (de 15 a 20 anos) do sistema de distribuição é expandir a rede para atender novas demandas de energia elétrica, minimizando os investimentos e as perdas (Silva, 1990). A expansão da rede é definida pela localização das subestações de distribuição e, posteriormente, pela determinação do traçado dos circuitos da rede primária (alimentadores de distribuição). No estudo do traçado e dimensionamento dos circuitos, deve-se buscar a configuração de rede de mínimas perdas para atender a demanda futura com o menor nível de investimento.

O planejamento a médio prazo (horizonte de 3 a 5 anos) estabelece o detalhamento das etapas definidas no plano de expansão - obtido no planejamento a longo prazo. Ele considera o traçado da rede definido anteriormente, busca adequá-lo às condições atualizadas da carga e faz possíveis correções do nível de tensão do sistema, minimizando os custos de investimentos e as perdas de energia no sistema.

O planejamento de curto prazo procura a configuração da rede com mínimas perdas que atenda a demanda de energia solicitada ao sistema. A configuração da rede é alterada através do fechamento e abertura de chaves instaladas ao longo dos alimentadores. Este processo é conhecido como *reconfiguração da rede*. A utilização

de sistemas de automação da distribuição (DMS - Distribution Management System) disponibiliza, além da monitoração e supervisão, comandos para a operação remota, e em tempo real, de chaves do sistema de distribuição, permitindo atualizar a configuração de rede em função da demanda de energia elétrica. A operação da rede de distribuição na configuração de mínimas perdas diminui a quantidade de energia distribuída e não faturada, maximiza os benefícios dos investimentos feitos no sistema e melhora a qualidade do serviço prestado pela empresa distribuidora de energia elétrica.

Normalmente, opera-se a rede de distribuição de energia elétrica na configuração radial (i.e., uma estrutura em árvore). Esta configuração leva ao compromisso mais adequado entre custo e requisitos de proteção para a operação da rede. Portanto, o problema da reconfiguração da rede pode ser caracterizado como uma generalização do problema de determinação da árvore de custo mínimo (Ahuja *et al.*, 1993) - a generalização resulta da variação do custo com a mudança da configuração. A solução deste problema deve fornecer uma configuração radial para operação da rede de distribuição, cujas perdas de energia sejam mínimas, e as restrições dos limites de queda de tensão e corrente nas linhas sejam satisfeitas.

Para melhor situar o problema da reconfiguração de rede no planejamento operacional, principal contribuição deste trabalho, descreve-se a seguir o sistema de potência, o sistema de distribuição e seus estados de operação. Em seguida, discute-se a distribuição das perdas de energia no sistema, as principais abordagens para o problema da reconfiguração da rede e, finalmente, a abordagem adotada neste trabalho.

1.2. O SISTEMA DE POTÊNCIA

Os sistemas de potência incluem a geração, transmissão e distribuição de energia elétrica (Fortunato *et al.*, 1990). O sistema de geração corresponde ao meio de produção da energia elétrica, normalmente usinas hidroelétricas e termoeletricas. Nas usinas hidroelétricas a energia elétrica é obtida pela transformação da energia

potencial dos cursos d'água. A energia elétrica é gerada nas usinas termoeletricas pela transformação da energia cinética dos gases e vapores em expansão, obtida pela queima de combustível. Não havendo possibilidade significativa de armazenagem, a energia elétrica produzida deve ser imediatamente transportada aos centros urbanos e industriais para sua distribuição e consumo. Os meios de transporte correspondem às linhas e subestações, formando o sistema de transmissão. O sistema de distribuição recebe energia do sistema de transmissão para entregar aos consumidores residenciais, comerciais e industriais.

O diagrama da Figura 1 mostra os principais componentes dos sistemas de geração, transmissão e distribuição (Pansini, 1983). A subestação associada à usina de geração serve para elevar a tensão elétrica, da ordem de centenas de volts, para a tensão elétrica de transmissão, da ordem de centenas de kV.

As linhas de transmissão, usualmente, conectam as subestações das usinas de geração às subestações de distribuição, cuja função é rebaixar a tensão elétrica de

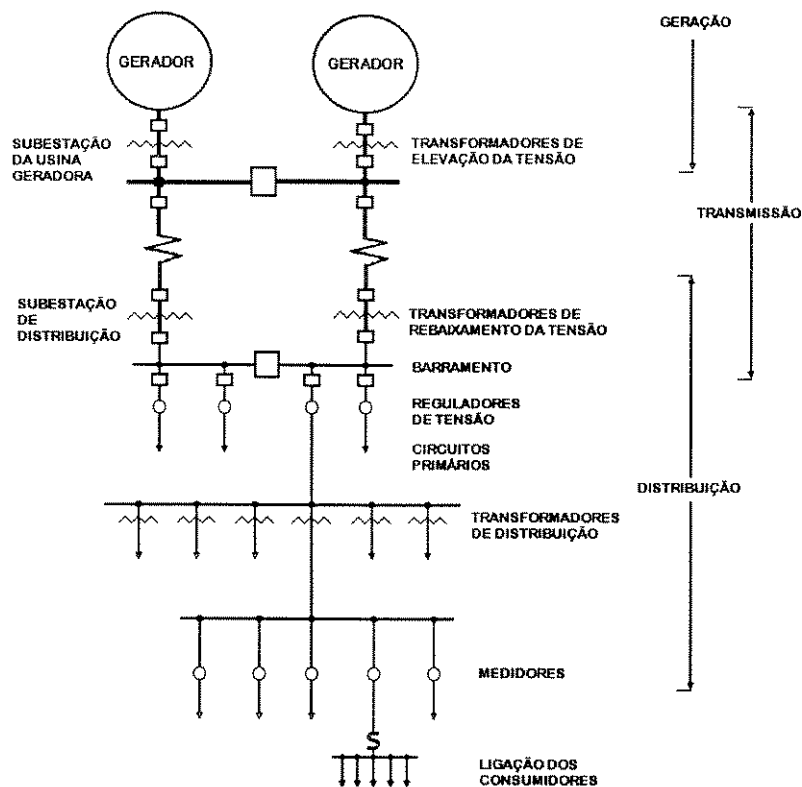


Figura 1 Sistema de Potência

centenas de kV para dezenas de kV. Do barramento da subestação de distribuição saem os circuitos (ou alimentadores) trifásicos. Eles são conectados ao barramento através de disjuntores, dispositivos de proteção contra curtos-circuitos, e, algumas vezes, de reguladores de tensão, que mantêm a tensão elétrica da distribuição dentro de limites predeterminados quando há variação de carga. As subestações da distribuição usualmente possuem vários destes circuitos para suportar a energia total a ser distribuída.

1.3. SISTEMA DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA

O sistema de distribuição é dividido, usualmente, em rede primária e rede secundária, conforme o diagrama simplificado de distribuição na Figura 2.

A rede primária compreende o barramento da subestação de distribuição até o primário do transformador de distribuição. Nos transformadores de distribuição a

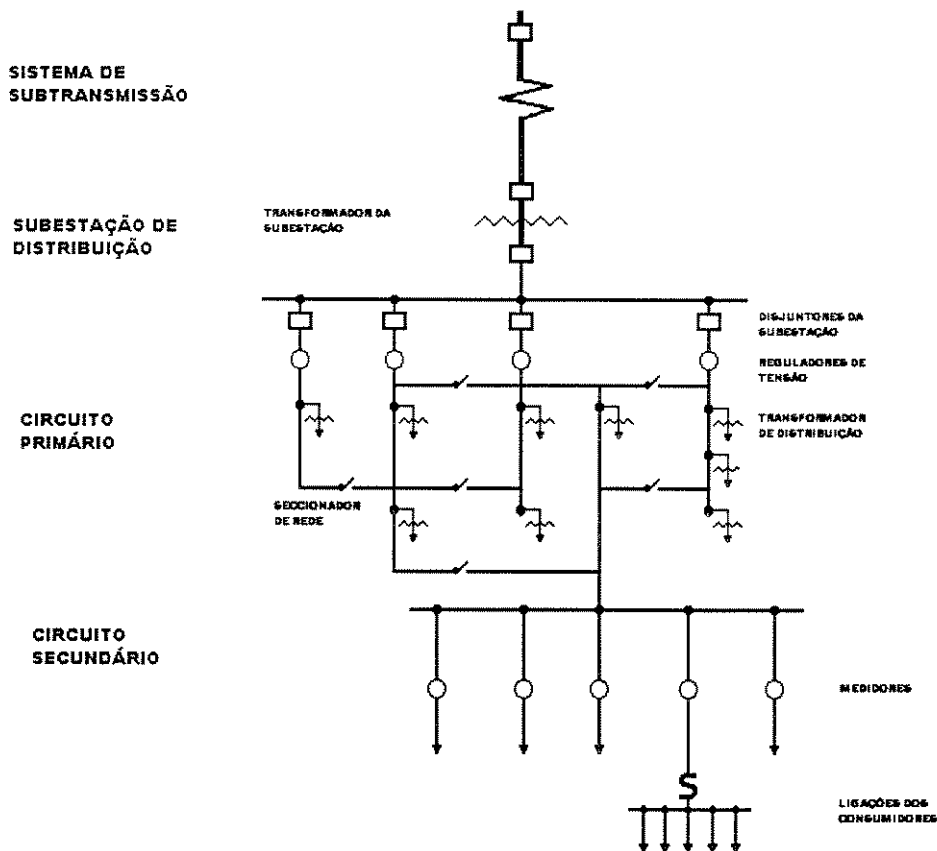


Figura 2. Sistema de Distribuição

tensão elétrica é reduzida de dezenas de kV para centenas de Volts. A energia elétrica flui através de condutores dos transformadores de distribuição até os dispositivos de medição, onde é registrado o consumo de energia a ser faturado pela concessionária.

1.4. ESTADOS DO SISTEMA DE DISTRIBUIÇÃO

O sistema de distribuição deve fornecer energia elétrica com confiabilidade e economia, satisfazendo as restrições de carga e restrições operacionais. As restrições de carga impõem a necessidade de suprir a demanda total dos consumidores sob responsabilidade do sistema, enquanto as restrições operacionais estabelecem limites de tensão e corrente para garantir uma operação segura e eficiente dos equipamentos instalados. Considerando essas restrições, a operação da rede de distribuição pode ser dividida, conceitualmente, em três estados distintos: normal, emergência e restauração (Murphy e Wu, 1990). A Figura 3 mostra o inter-relacionamento destes estados.

As restrições de carga e operacionais estão satisfeitas no estado normal; as ações de operação devem manter a rede operando com a máxima confiabilidade e no menor custo. O estado de emergência é caracterizado pela existência de uma ou mais falhas na rede, violando as restrições operacionais, isto é, os limites de tensão

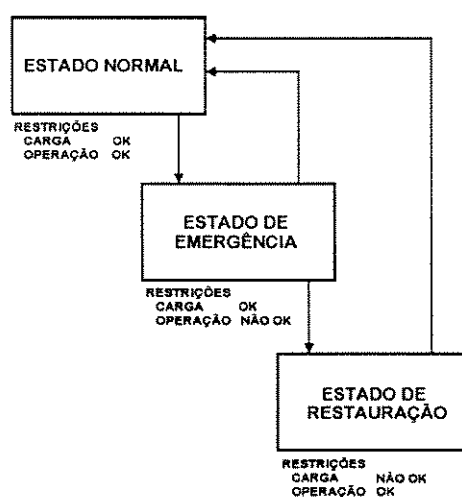


Figura 3. Estados da rede de distribuição

ou corrente não são satisfeitos. Após a identificação do defeito, e isolamento da área atingida pela falha, caracteriza-se o estado de restauração. Neste estado, as restrições operacionais estão satisfeitas, pois no processo de restauração é em geral necessário interromper o fornecimento de energia para uma parte dos consumidores.

1.4.1. Operação na Emergência e na Restauração

No estado de emergência, o sistema de proteção atua isolando os componentes da rede com defeito, visando evitar acidentes e a propagação do defeito para outras áreas da rede. Como consequência, os equipamentos e sua isolação são preservados - seriam degradados se expostos a corrente de curto circuito. Os condutores são também preservados, se submetidos ao mínimo a correntes que excedem sua capacidade física.

Após o sistema de proteção atuar e isolar os equipamentos defeituosos, o sistema de distribuição altera seu estado para restauração. O operador então localiza o equipamento com defeito, para realizar manutenção ou substituí-lo. Em seguida, a parte do sistema em condições de serviço deve ser identificada e restabelecida. As empresas distribuidoras de energia elétrica procuram diminuir, ao máximo, o tempo da indisponibilidade do sistema, para garantir a qualidade do serviço.

1.4.2. Operação no Estado Normal

No estado normal é desejável melhorar a eficiência do sistema, diminuindo seu custo operacional e aumentando sua confiabilidade e segurança. Confiabilidade é uma medida de disponibilidade e qualidade de serviço, enquanto segurança é a medida de robustez do sistema, considerando as possíveis ocorrências de defeito na rede.

As principais funções no estado normal são: o controle integrado de Volt/Var e a reconfiguração da rede para mínimas perdas. O controle da tensão dentro de certos limites preestabelecidos é necessário, pois os equipamentos instalados

pelos consumidores são normalmente projetados para uma operação aproximadamente constante. Variações significativas no nível de tensão podem danificar ou ocasionar o mau funcionamento destes equipamentos, comprometendo a qualidade do serviço oferecido. A potência reativa que flui nos circuitos também deve ser controlada. Sua circulação aumenta as perdas na linha e, conseqüentemente, reduz a capacidade da rede.

Operar a rede com a configuração de mínimas perdas é importante tanto pelo aspecto econômico como da qualidade de fornecimento. A configuração de mínimas perdas é caracterizada por uma melhor distribuição do fluxo de potência nos circuitos da rede; em conseqüência, aumenta a vida útil dos equipamentos instalados na rede e melhora o perfil de tensão dos circuitos de alimentação. Recentes avanços da automação do sistema de distribuição estão permitindo manobrar as chaves em tempo real e remotamente (a partir do centro de operação). Com esta opção, torna-se possível reconfigurar a rede para reduzir as perdas, balancear a carga e melhorar a confiabilidade do sistema em resposta a variações significativas na demanda. Em conseqüência, o problema de minimização de perdas, antes concentrado no planejamento, começa a ser tratado também na operação da rede.

1.5. PERDAS DE ENERGIA NA DISTRIBUIÇÃO

A dissipação da energia elétrica no sistema de potência é causada pela resistência elétrica existente nas linhas e equipamentos dos subsistemas de transmissão e distribuição. As perdas totais no sistema de potência são normalmente estimadas em 7%, sendo 2% na transmissão e 5% na distribuição (Buch *et al.*, 1982). No Brasil, as perdas no sistema de potência são aproximadamente 15,30% (42.517 GWh) da energia total produzida no país (CODI, 1997), onde 7,28% (20.232 GWh) são perdas na transmissão e 8,02% (22.285 GWh) na distribuição. A Figura 4 ilustra essa incidência de perdas nas redes elétricas.

BRASIL - PERDAS DE ENERGIA ELÉTRICA

1994 - DADOS EM GWh

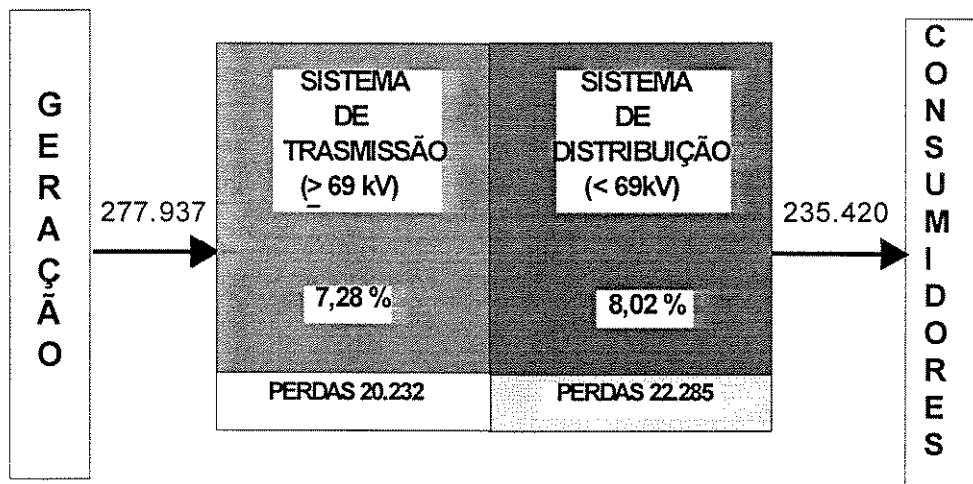


Figura 4. Fluxo e perdas de energia elétrica no sistema de transmissão e distribuição (CODI, 1997)

Supondo um custo unitário médio para as perdas da ordem de R\$ 65,00/MWh, as perdas totais do sistema representam R\$ 2.764,00 milhões de receita anual não efetivada, sendo R\$ 1.315,00 milhões na transmissão e R\$ 1.449,00 milhões na distribuição.

A Tabela 1 mostra os valores percentuais das perdas totais de energia (transmissão e distribuição) de algumas empresas do setor elétrico brasileiro. Observa-se uma variação significativa das perdas apuradas por essas empresas. Uma análise mais detalhada faz-se necessária para verificar as causas das perdas, porém, no geral, pode-se considerar elevados os percentuais de perdas nestes sistemas.

As perdas totais no sistema podem ainda ser divididas em perdas comerciais e perdas técnicas. As perdas comerciais referem-se à energia efetivamente entregue ao consumidor, porém não computadas no faturamento da empresa. As perdas técnicas correspondem à energia perdida no transporte e nos equipamentos instalados na rede, portanto são inerentes ao sistema. A Figura 5 apresenta o fluxo de energia elétrica, destacando as perdas nos subsistemas (transmissão e distribuição) e pela origem (perdas comerciais ou técnicas).

Tabela 1. Percentual das perdas anuais de energia no setor elétrico (CODI, 1997).

Empresa	1988	1989	1990	1991	1992	1993	1994
CEB	5,6	6,8	7,5	7,9	8,9	9,5	10,4
CEEE	12,5	12,3	11,1	11,2	11,4	11,0	11,8
CELESC	7,2	7,2	7,0	7,0	7,1	7,9	7,9
CELG	15,1	12,9	13,1	14,5	16,3	16,9	17,6
CELPE	12,1	12,0	12,7	13,0	13,1	14,8	15,7
CEMAT	17,4	17,6	17,3	19,6	23,0	25,3	25,8
CEMIG	8,9	9,6	8,8	9,5	9,0	10,0	8,2
CERJ	17,3	17,7	16,4	16,5	19,4	23,4	25,2
CESP	4,3	3,4	2,4	3,0	3,3	3,7	3,4
COELBA	11,4	12,2	10,0	14,6	13,8	14,2	15,1
COPEL	7,5	7,4	6,8	8,7	7,4	6,7	6,2
CPFL	6,8	7,0	6,3	6,6	5,5	6,3	6,9
ELETROPAULO	6,4	7,5	6,4	7,4	8,3	7,9	8,7
ENERSUL	15,6	14,6	13,4	16,2	15,3	14,9	14,8
ESCELSA	7,2	7,7	9,4	9,8	10,9	11,4	12,6
LIGHT	12,8	14,5	13,6	14,2	14,1	13,9	15,5
CODI	9,9	10,0	9,7	10,3	10,3	11,1	11,3
Brasil	13,2	13,4	13,0	13,8	13,7	14,6	15,3

Como pode-se observar na Figura 5, 48% das perdas do sistema ocorrem no subsistema de transmissão, enquanto 52% ocorrem no subsistema de distribuição. Estes valores indicam um equilíbrio na distribuição de perdas entre os subsistemas. Por outro lado, considerando a origem destas perdas, os percentuais apresentados são de 83% para as perdas técnicas e de 17% para as perdas comerciais, do total de perdas do sistema. Supondo os mesmos R\$ 65,00/MWh para o custo unitário médio, as perdas comerciais e técnicas, representam, custos aproximados de R\$ 482 milhões e R\$ 2.280 milhões, respectivamente.

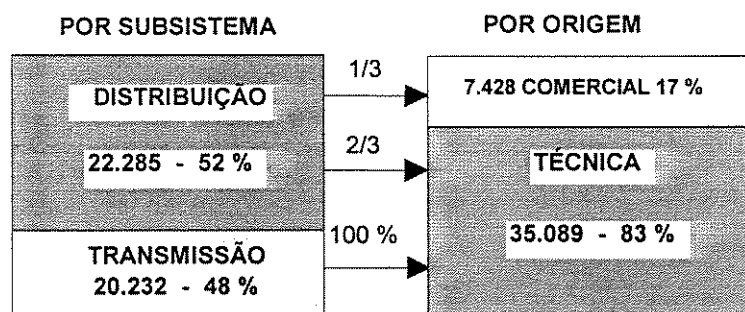


Figura 5. Distribuição do total das perdas de energia (CODI 1997)

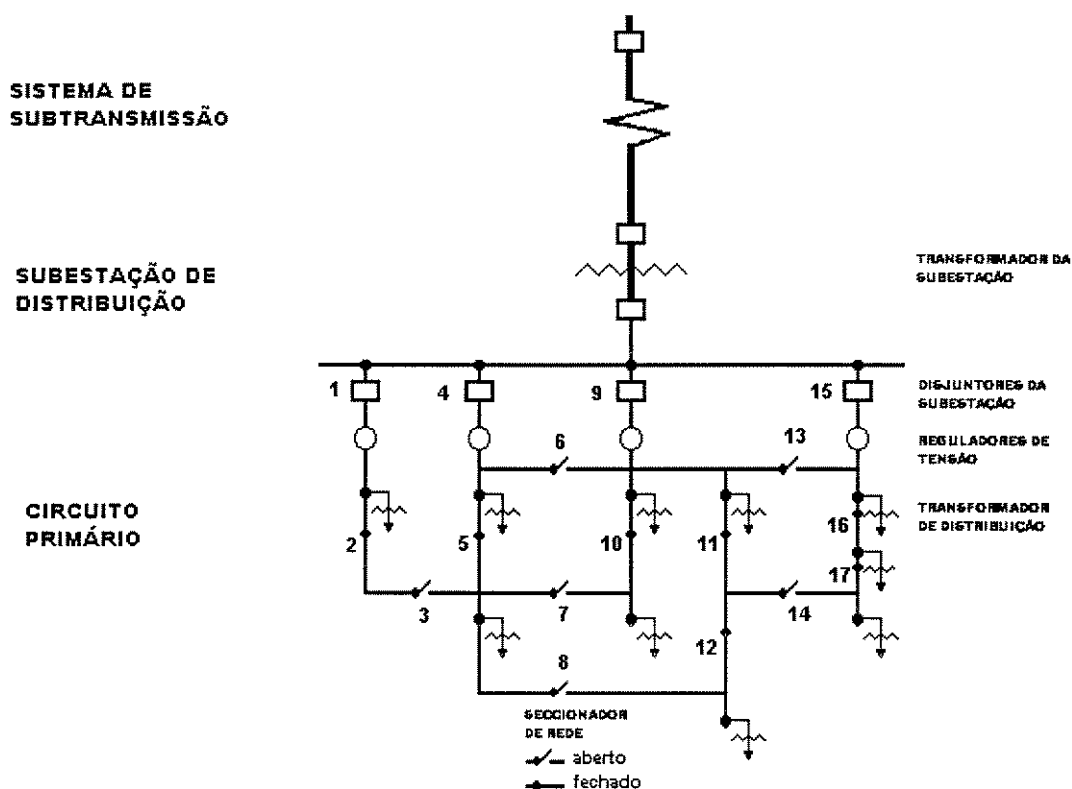


Figura 6. Unifilar do sistema de distribuição primário

1.6. CONFIGURAÇÃO DA REDE DE MÍNIMAS PERDAS

A Figura 6 mostra uma rede distribuição representada pelo seu diagrama unifilar. Esta rede possui uma subestação com 4 (quatro) circuitos independentes para a distribuição de energia elétrica. Os consumidores estão ligados aos transformadores de distribuição, cuja soma das demandas individuais determina a demanda total do sistema. Os seccionadores de rede ou chaves, numerados sequencialmente, definem a configuração da rede.

A configuração do sistema de distribuição pode ser alterada, mudando o estado das chaves, por ação local (pelas equipes de manutenção) ou remota (pelo sistema de automação). A Figura 7 ilustra a mudança da configuração da Figura 6, quando são abertas as chaves de número 5, 10 e 11 e fechadas as chaves de números 3, 7 e 14.

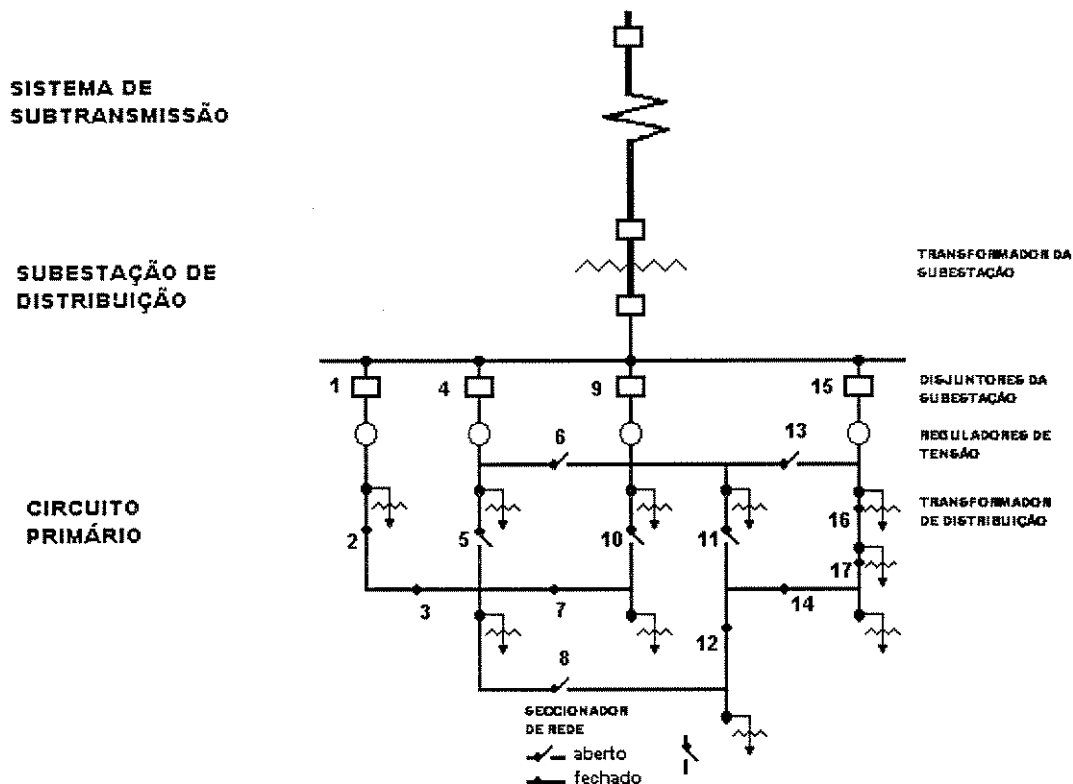


Figura 7. Configuração alterada do Unifilar

No estado normal da rede de distribuição, um dos objetivos para reconfigurar a rede é a minimização das perdas inerentes ao sistema de distribuição. Considerando a rede em operação e a demanda a ser suprida, busca-se a configuração que apresente as menores perdas de energia elétrica, respeitando os limites de queda de tensão e do fluxo de corrente nas linhas. Diferentes configurações de rede são obtidas pela combinação do fechamento e abertura de chaves, selecionadas criteriosamente para manter a estrutura radial da rede de distribuição. Por exemplo, na rede de distribuição representada na Figura 8, com 1 subestação, 4 transformadores de distribuição e 6 chaves (duas abertas), existem 11 (onze) configurações de rede radial diferentes.

O número de configurações possíveis cresce exponencialmente com a dimensão da rede. Em sistemas maiores, este número pode chegar a milhões de configurações diferentes. Teoricamente, para encontrar a configuração de rede de mínimas perdas, pode-se verificar o valor das perdas de todas as configurações de rede radiais diferentes e escolher aquela de menor valor para as perdas. Este

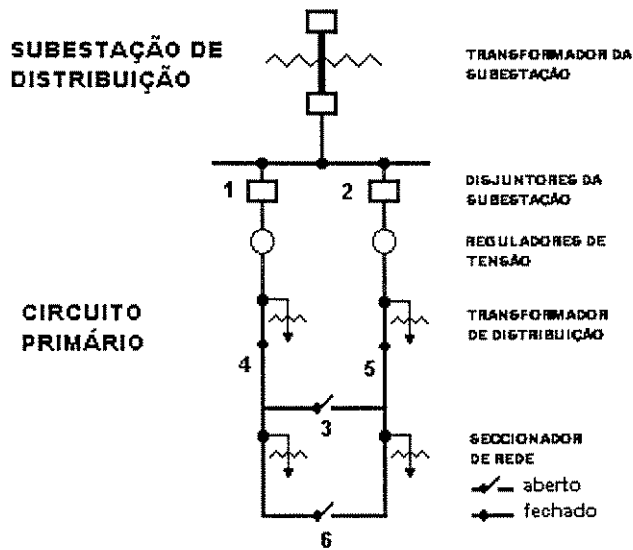


Figura 8. Rede de distribuição exemplo

procedimento é inviável, considerando o tempo de processamento necessário para obter a solução desejada.

Merlin e Back (1975) foram os primeiros a propor a reconfiguração da rede para mínimas perdas, utilizando as chaves abertas existentes. A partir deste trabalho, muitos outros foram publicados, explorando técnicas como programação matemática, heurísticas e inteligência artificial, ou mesmo, uma combinação destas. A seguir são mencionados alguns trabalhos referentes às principais abordagens para o problema de reconfiguração da rede.

1.7. ABORDAGENS PARA O PROBLEMA DE RECONFIGURAÇÃO

As abordagens para redução das perdas no sistema de distribuição de energia elétrica diferem em relação às estratégias utilizadas na solução do problema. Considerando estas diferenças pode-se classificá-las em abordagens de duas etapas, *branch-exchange*, programação matemática, otimização combinatória, por construção, por heurística, rede neurais, inteligência artificial e por partição da rede.

1.7.1. Procedimento em duas etapas

A primeira etapa nesta abordagem consiste no fechamento de todas as chaves abertas do sistema, definindo uma configuração de rede malhada. Em seguida, calcula-se o fluxo de potência ótimo (mínimas perdas) para esta configuração. Na segunda etapa, as *malhas* (ciclos) existentes na rede são removidas, através da abertura de chaves. Usualmente, métodos heurísticos são utilizados para selecionar, sistematicamente, as chaves a serem abertas, até que uma configuração de rede radial seja encontrada. Utilizando esta abordagem, Merlin e Back (1975) descrevem um procedimento heurístico baseado no fluxo de potência e no método da classe *branch and bound* (Garfinkel & Nemhauser, 1972). Os resultados apresentados mostram um método heurístico eficiente, comparando a solução aproximada obtida pela heurística com a solução ótima do *branch and bound*. Posteriormente, Shirmohammadi e Hong (1989) modificaram o método de Merlin e Back para incluir as restrições operacionais - limites de tensão e corrente dos alimentadores.

Observe que a abordagem de duas etapas parte de uma configuração malhada. Esta característica é vantajosa comparada com outras abordagens descritas a seguir, pois a solução obtida independe das características de configurações radiais conhecidas.

1.7.2. Branch-Exchange

A heurística *branch-exchange* é muito utilizada na solução do problema de reconfiguração de rede. O processo tem início com uma configuração de rede radial (solução factível). As perdas elétricas são reduzidas através da mudança da configuração da rede, sem perder a estrutura radial do sistema. A configuração da rede é alterada até que não haja mais variações significativas das perdas do sistema. A maioria dos métodos desenvolvidos, baseiam-se em equações para aproximar a variação de perdas devido à transferência de carga de um circuito para outro. Estas equações guiam o processo iterativo na escolha das configurações candidatas (Ross *et al.*, 1981; Cinvalar *et al.*, 1988; Baran e Wu, 1989; Castro e

no sistema –uma variável 0-1 para cada um dos nós que interligam a chave aberta. As variáveis 0-1 indicam em qual circuito o nó da rede será conectado, em outras palavras, simula a abertura e fechamento das chaves necessárias para transferência da carga, definindo uma nova configuração radial.

1.7.4. Otimização Combinatória

Técnicas de otimização combinatória são também exploradas para obter a solução do problema da reconfiguração da rede de mínimas perdas. Chiang e Jumeau (1990a, 1990b) formulam um problema de otimização multi-objetivo. O método proposto procura a minimização das perdas e o balanceamento ótimo, isto é, a distribuição de fluxo correspondente ao menor carregamento médio da rede, através de algoritmo baseado na técnica *simulated annealing*.

A técnica *busca TABU* (Glover, 1988) é utilizada por Cherkaoui et al. (1993). Cada etapa desta estratégia consiste na geração e seleção de uma das configurações próximas à configuração em exploração (evitando-se certas “configurações TABU”), cujos valores de perdas são inferiores às perdas da configuração considerada nesta etapa. Este processo repete-se até que não seja mais possível gerar configurações próximas; neste caso a configuração em exploração é declarada como uma solução ótima local do problema.

1.7.5. Abordagem por Construção

Nesta abordagem, todas as chaves são inicialmente consideradas abertas. Em seguida são identificadas as chaves que devem ser fechadas, definindo a configuração de rede radial desejada. Um exemplo desta abordagem é apresentada por Glamocanin (1990), onde é proposto um procedimento sistemático para construção da rede de distribuição, fechando-se sucessivamente as chaves fechadas que apresentarem o menor acréscimo de perdas ao sistema de distribuição. Uma aproximação do acréscimo das perdas é calculada como uma função da potência a ser transportada da subestação até o nó da rede considerado. Este procedimento de construção é basicamente uma técnica de busca do tipo

“gulosa” – método que encontra a solução ótima através de um algoritmo iterativo orientado por alguma medida de otimização. Este método em geral não encontra a solução ótima global.

1.7.6. Rede Neurais

As técnicas de redes neurais resolvem problemas de otimização através de uma estrutura de elementos computacionais não lineares interligados, simulando as redes neurais biológicas. As redes neurais são previamente “treinadas” para armazenar regras de classificação ou reconhecimento de padrões (Lippmann 1987).

Na abordagem do problema da minimização de perdas na distribuição através de redes neurais, Kim *et al.* (1993) utilizaram no “treinamento” uma correspondência entre padrões de demanda e boas configurações de operação. Para diminuir o esforço computacional, no trabalho é proposto a divisão da rede de distribuição em zonas de carga. Os resultados mostram que as soluções obtidas, para cada perfil de carga considerado, são próximas das respectivas soluções ótimas globais para o problema.

1.7.7. Sistemas Especialista

A técnica de *sistemas especialistas (SE)*, uma área específica da inteligência artificial, foi também adotada para reduzir as perdas elétricas do sistema de distribuição. No trabalho apresentado por Taylor e Lubkeman (1990), um programa comercial para desenvolvimento de SE foi empregado para implementar 5 (cinco) regras heurísticas. Estas regras guiam a procura de uma solução aproximada do problema. A técnica tem a vantagem de possibilitar a modelagem de aspectos de difícil formulação matemática.

1.7.8. Heurísticas Mistas

O conhecimento das características do problema pode ser usado na formalização de uma busca sistemática. As heurísticas utilizadas tornam possível ao processo de busca decidir por uma direção promissora a ser seguida e, portanto, obter a redução das perdas no sistema de distribuição com o mínimo esforço computacional. Chen e Cho (1993) utilizam um método de busca sistemático para definir um plano de manobras que reduza as perdas de energia, considerando o perfil de carga horária dos alimentadores. O método de busca utiliza uma função de avaliação que estima a redução das perdas de cada chaveamento candidato. Com base no plano horário de manobras, obtém-se um plano de manobras para cada estação do ano.

Uma abordagem mista é proposta por Broadwater *et al.* (1993). O algoritmo descrito utiliza as regras de chaveamento de Cinvalar *et al.* (1988) e um método de busca, guiado pela função de perdas apresentado em Huddleston *et al.* (1990), para selecionar o chaveamento.

Lee *et al.* (1994) descrevem um algoritmo de busca baseado numa técnica de otimização local. Uma função heurística é usada para selecionar a chave aberta a ser considerada na etapa seguinte. As perdas são estimadas para todas as alternativas de configurações obtidas a partir da chave aberta selecionada, e aquela de menor perda é escolhida para a próxima iteração. Repete-se este processo até que não haja mais variações significativas das perdas. Este método não considera caminhos alternativos de busca após escolha da chave a ser aberta, o que não garante a solução ótima global do problema.

Borozan *et al.* (1995) descrevem um algoritmo de busca com uma estratégia irrevogável - *hill-climbing* (Pearl, 1984). Iniciando com uma rede malhada (todas as chaves abertas são fechadas), o algoritmo busca a rede radial de mínimas perdas, abrindo a chave com o menor fluxo a cada iteração. Um procedimento similar, em duas etapas, é proposto por Augugliaro *et al.* (1995). Na primeira etapa, as chaves com o menor fluxo são abertas iterativamente, para transformar a rede malhada

inicial em uma rede radial. A partir desta rede radial os autores buscam melhorar as perdas, aplicando a técnica *branch-exchange* proposta por Cinvalar *et al.* (1988).

1.8. NOVA ABORDAGEM PARA MINIMIZAÇÃO DAS PERDAS

Com o objetivo de obter a solução ótima global para o problema da reconfiguração de rede para mínimas perdas, este trabalho utiliza um procedimento recursivo de duas etapas, combinando as técnicas de fluxo não lineares (Lyra e Tavares, 1988) e estratégias de busca formal usadas na área de inteligência artificial (Pearl, 1994).

Na primeira etapa, a operação radial da rede de distribuição é relaxada e uma solução otimista para o problema é encontrada, utilizando um algoritmo de minimização de fluxos em redes com a função objetivo não-linear (Lyra, 1984). Esta solução é infactível, porém fornece um limitante inferior para as perdas na rede. Informações da solução otimista são utilizadas na segunda etapa para selecionar a chave a ser aberta, eliminando um ciclo existente na rede. As etapas são repetidas até que uma solução radial seja encontrada.

O procedimento recursivo pode ser controlado por estratégias de buscas informadas. Neste trabalho, os procedimentos foram baseados em estratégias de busca irrevogável e buscas informadas do tipo *best-first* (Pearl, 1984). As buscas desta última classe são capazes de garantir a obtenção de configurações ótimas.

A implementação do processo recursivo requer um modelo para representar a rede de distribuição de energia elétrica. No capítulo a seguir é descrito o modelo de grafo adotado neste trabalho.

CAPÍTULO 2

MODELO PARA REDES DE DISTRIBUIÇÃO

2.1. REPRESENTAÇÃO DA REDE PRIMÁRIA

Um diagrama simplificado da rede de distribuição primária é apresentado na Figura 9. Seus componentes mais importantes para o equacionamento do problema de minimização das perdas na rede são os *blocos de carga* ($BC_1, BC_2, \dots, BC_{10}$), as *chaves* ($CH_1, CH_2, \dots, CH_{16}$) e as *subestações de distribuição* (SE_1 e SE_2). Os blocos de carga compreendem um conjunto de consumidores de energia, ligados a transformadores secundários, que por sua vez estão interligados por trechos de rede. Os blocos são delimitados por chaves (i.e., os circuitos internos não podem ser seccionados); a transferência da soma da energia consumida em um bloco só é possível na sua totalidade. As chaves permitem o gerenciamento da configuração da rede, transferindo a energia dos blocos de carga entre alimentadores da mesma subestação, ou entre subestações do sistema de distribuição.

2.2. MODELO DE GRAFO

Um modelo de grafo é adotado para representar essas entidades e seus relacionamentos (Ahuja, 1993; Cavellucci e Lyra, 1993). Esta representação mostra-se conveniente por três aspectos: pela abstração intuitiva da rede de distribuição, pela possibilidade de implementar algoritmos eficientes e pelo armazenamento compacto dos dados.

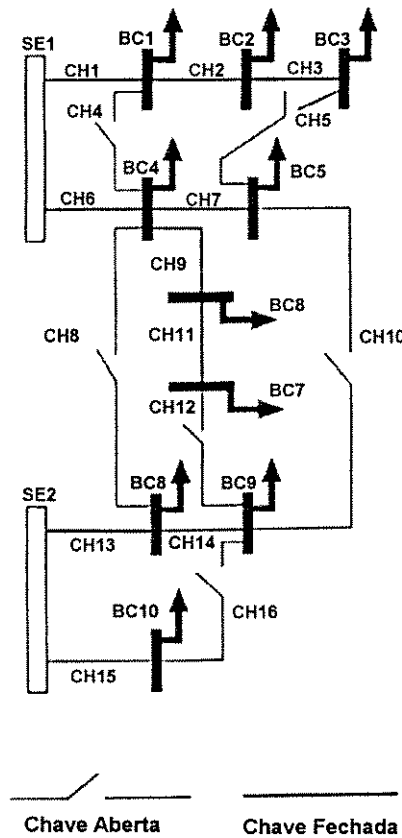


Figura 9. Diagrama simplificado da rede primária

A rede de distribuição é representada pelo grafo orientado $G = [N, A]$. Os nós do conjunto N representam blocos de carga ou subestações - um nó raiz (R) é também incluído em N para evitar dificuldades desnecessárias no tratamento da conectividade da rede. Os arcos do conjunto A denotam dispositivos seccionadores de rede (todos os tipos de chaves e disjuntores), exceto aqueles conectando as subestações ao nó raiz, que podem ser identificados com a rede de transmissão. O modelo de grafo para a rede de distribuição esquematizada na Figura 9 é apresentado na Figura 10.

Em geral, as redes de distribuição são operadas radialmente, isto é, existe um único caminho energizado que liga qualquer nó a raiz. A Figura 11 representa uma rede em operação - os arcos correspondentes as chaves abertas (CH4, CH5, CH8, CH13 e CH15 na Figura 10) não estão representados. Usando a terminologia de grafo (Ahuja, 1993), a rede energizada, ou em operação, define uma árvore recobridora

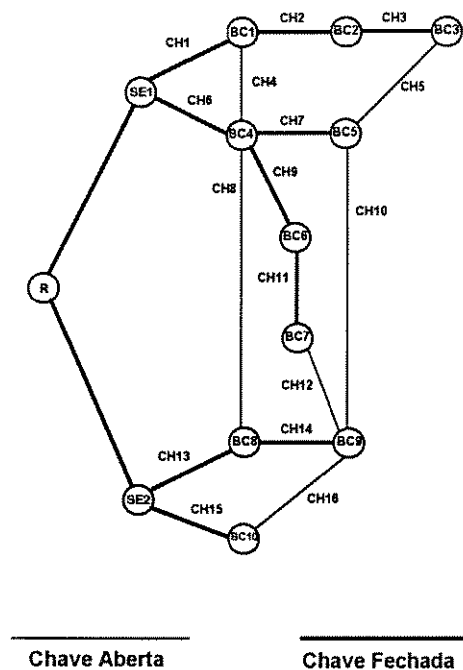


Figura 10 Modelo de grafo para redes de distribuição

$G' = [N, A']$ para o grafo G (um subgrafo acíclico conexo, incluindo todos os nós em G) – A' é o conjunto de todas as chaves fechadas da rede.

Em princípio, a direção dos arcos é irrelevante, pois a corrente elétrica em um dado

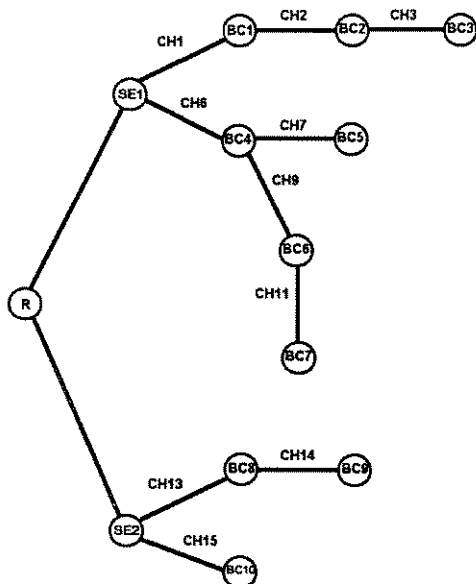
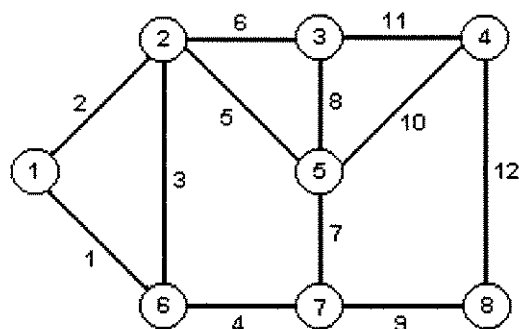


Figura 11. Modelo de grafo para a rede em operação



Arco(k)	1	2	3	4	5	6	7	8	9	10	11	12
o_k	1	1	2	6	2	2	5	3	7	4	3	4
t_k	6	2	6	7	5	3	7	5	8	5	4	8

Figura 12. Armazenamento na estrutura arco-orientada

trecho da rede, representado por um arco no grafo, pode fluir em ambos os sentidos. Porém, sendo a operação da rede radial, o sentido do arco é definido pelo sentido do fluxo de corrente elétrica na rede, indo da fonte (subestação) para a carga (blocos de carga). Possíveis mudanças no sentido do fluxo de corrente, e conseqüente alteração do sentido do arco, podem ser tratadas convenientemente pelos algoritmos, como veremos adiante.

2.3. REPRESENTAÇÃO DA REDE

A representação *arco-orientado* (Jensen e Barnes, 1980)¹ foi adotada para armazenar a rede de distribuição e seus parâmetros físicos e elétricos. Nesta estrutura, os elementos básicos de armazenamento são listas indexadas pelos arcos do grafo. Duas listas são necessárias para armazenar os arcos de um grafo: a lista dos nós origem ($O = \{o_k\}$) e a lista dos nós destinos ($D = \{d_k\}$) - os k -ésimos elementos destas listas são os nós origem e destino, respectivamente, do arco k . A Figura 12

¹ Faz-se referência a Jensen e Barnes (1980) porque as implementações dos algoritmos para tratamento da estrutura de dados foram baseados nos algoritmos apresentados pelos autores, porém considera-se Ahuja et al (1993) uma referência mais completa para estrutura de grafo.

ilustra o armazenamento dos arcos do grafo G nas listas O e D . Por exemplo, o arco 8 é definido pelo nó origem 3 e nó destino 5, os quais estão armazenados no oitavo elemento das listas O e T , respectivamente.

Os demais parâmetros, físicos e elétricos, pertinentes aos arcos, são armazenados em listas semelhantes. A Tabela 2 exemplifica as listas para armazenar o fluxo no arco, x_k , e a resistência, r_k .

Tabela 2 Armazenamento dos parâmetro físicos e elétricos

Arco(k)	1	2	3	4	5	6	7	8	9	10	11	12
o_k	1	1	2	6	2	2	5	3	7	4	3	4
t_k	6	2	6	7	5	3	7	5	8	5	4	8
x_k	20	30	0	15	0	20	0	5	5	0	10	0
r_k	2	3	0	2	0	4	0	1	1	0	2	0

2.3.1. Listas Auxiliares

Freqüentemente é necessário recuperar o conjunto dos arcos que têm origem no nó i (M_{o_i}) e o conjunto dos arcos que terminam no nó i (M_{t_i}). Estes conjuntos são obtidos percorrendo-se convenientemente os elementos das listas O e T . Para tanto, acrescenta-se na estrutura de dados três listas auxiliares (Jensen e Barnes, 1980): lista de apontadores para os arcos com origem no nó i , $p_o(i)$; lista de ordenação dos arcos pelos seus nós destinos, $l_d(i)$; lista de apontadores para os arcos que chegam no nó i , $p_d(i)$ (ver Apêndice I para detalhes de implementação).

2.3.2. Rede em Operação

A rede em operação pode ser representada pela árvore A recobridora do grafo G . Para armazenar a árvore A em preordem (Jensen e Barnes, 1980; Kennington e Helgason, 1980) utiliza-se três listas de apontadores: P_B , P_F e P_P . Para um dado nó i , arbitrário, da árvore temos:

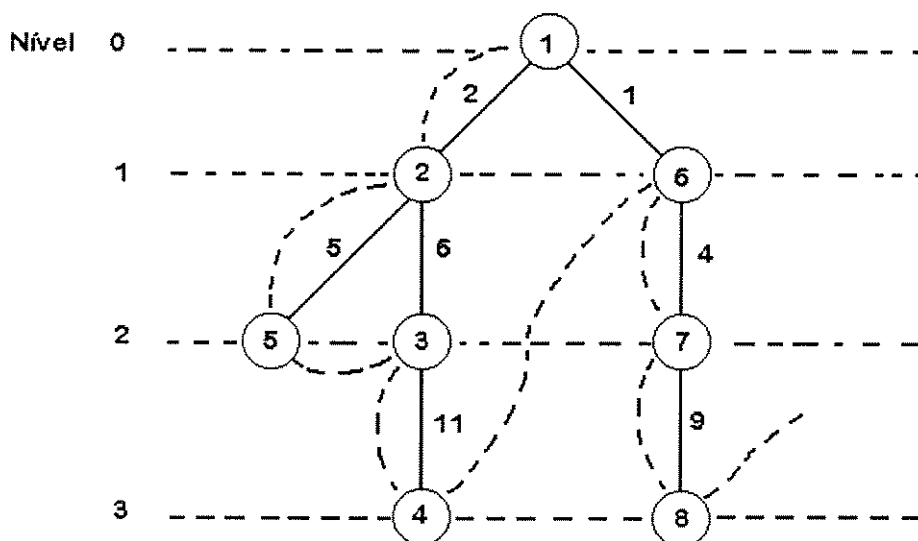
$P_B(i)$: apontador para o arco que liga ao antecessor do nó i (arco pai);

$P_P(i)$: apontador para o nó antecessor de i na árvore (nó pai);

$P_F(i)$: apontador para o próximo nó da árvore quando percorrida em preordem (filho).

A Figura 13 exemplifica o armazenamento de uma árvore em preordem, onde o nó raiz é o nó 1. Por definição, para o nó raiz os apontadores $P_B(i)$ e $P_P(i)$ são iguais a zero, indicando a não existência do arco pai nem do nó pai. A linha tracejada, ligando todos os nós da árvore, é uma representação gráfica da lista de apontadores P_F .

Além dos apontadores P_B , P_P e P_F , pode-se acrescentar a esta estrutura de dados o rótulo $P_D(i)$, que define o nível de profundidade do nó i na árvore - número de



Nó	1	2	3	4	5	6	7	8
$P_B(i)$	0	2	6	11	5	1	4	9
$P_P(i)$	0	1	2	3	2	1	6	7
$P_F(i)$	2	5	4	6	3	7	8	-
$P_D(i)$	0	1	2	3	2	1	2	3

Figura 13. Estrutura de armazenamento em preorder da árvore A

nós existentes entre o nó raiz e o nó i (Jensen e Barnes, 1980). Por definição, o nível do nó raiz é igual a zero. Estes rótulos têm como objetivo melhorar o desempenho computacional dos algoritmos. Porém, é necessário considerar o espaço de memória requerido para o armazenamento do rótulo. Na Figura 13 as linhas horizontais tracejadas ilustram o rótulo nível - número de nós da raiz a um nó i arbitrário.

2.4. COMENTÁRIOS

A estrutura de dados descrita apresenta as características de ser compacta e possibilitar um bom desempenho computacional aos algoritmos que a utilizam. O armazenamento do grafo e sua respectiva árvore recobridora requerem um total de 5 (cinco) listas, duas de tamanho igual ao número de arcos e 3 (três) de dimensão igual ao número de nós. Supondo que cada elemento destas listas ocupe 4 (quatro) Bytes de memória, um grafo de 1000 nós e 1100 arcos é armazenado em 20 kBytes, aproximadamente. Esta quantidade reduzida de memória mostra o quanto a estrutura de dados é compacta, possibilitando o armazenamento de redes de grande porte.

Os algoritmos para tratamento e busca na estrutura de dados foram adaptados da área de fluxos em redes. Isso permitiu importar um processo de aperfeiçoamento de mais de 30 anos, para melhorar seu desempenho. Os algoritmos básicos compreendem: a recuperação dos arcos que chegam e dos arcos que saem de um nó; a inclusão e exclusão de um arco da árvore; obtenção dos nós e arcos que definem uma subárvore e a recuperação dos nós e arcos que pertencem ao caminho entre dois nós da árvore. Estes algoritmos são utilizados, por exemplo, no cálculo da tensão nos nós e corrente nos arcos da rede de distribuição, na recuperação dos ciclos da rede malhada e na simulação da abertura e fechamento de uma chave.

O Apêndice 2 descreve três ferramentas computacionais, que exemplificam o emprego desta estrutura de dados na área de engenharia de distribuição de energia elétrica. Essas ferramentas, elaboradas durante o desenvolvimento deste trabalho,

têm como objetivo auxiliar o operador na visualização do estado da rede de distribuição, e na elaboração de planos de manobras para lidar com ocorrências de emergências ou em situações programadas.

O capítulo a seguir apresenta a formulação do problema da reconfiguração da rede para mínimas perdas, considerando as equações de fluxo de potência e das perdas para uma rede de distribuição radial. Uma estratégia de busca da solução ótima do problema é definida com o auxílio de conceitos da área de inteligência artificial (discutidos no capítulo 4), que possibilita o desenvolvimento de algoritmos para encontrar soluções ótimas para operação da rede de distribuição.

CAPÍTULO 3

MINIMIZAÇÃO DAS PERDAS E ESTRATÉGIA DE SOLUÇÃO

3.1. INTRODUÇÃO

O objetivo de minimizar as perdas no sistema de distribuição de energia elétrica é reduzir desperdícios, melhorando sua eficiência. Alterando o estado de uma chave aberta para fechada e vice-versa, a configuração dos alimentadores é modificada e, conseqüentemente, os fluxos de correntes passantes nas linhas, os perfis de tensões e as perdas são redistribuídos no sistema. A Figura 14 mostra o diagrama simplificado de uma rede de distribuição em operação. A configuração da rede pode ser alterada, por exemplo, fechando-se a chave CH12 e abrindo-se a chave CH11. Esta manobra transfere a carga entre os alimentadores.

Em geral, as redes de distribuição são operadas radialmente, isto é, não existe circuitos que fechem anel. Por exemplo, na Figura 14 se a chave CH4 for fechada cria-se um circuito em anel, incluindo as chaves CH1, CH4 e CH6. Considerando a topologia de rede radial, o problema de reconfiguração da rede pode ser formulado como uma generalização do problema da árvore de recobrimento de custo mínimo (Ahuja et al., 1993). A generalização é devido à mudança dos pesos a cada configuração considerada, sendo os pesos (ou perdas) função do fluxo passante nas linhas. A solução deste problema é a árvore de recobrimento de perdas mínimas (do grafo que representa a rede de distribuição), sujeita às restrições de carga, de

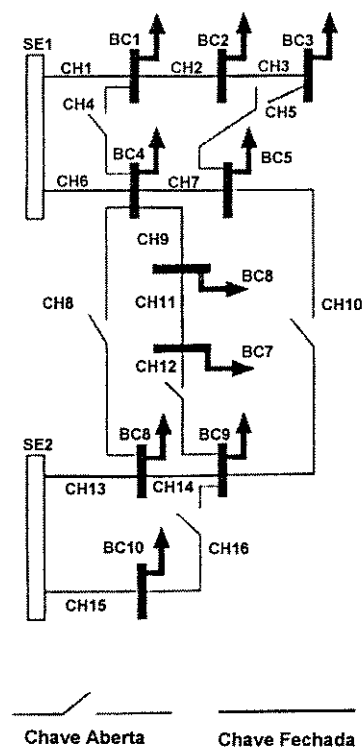


Figura 14. Diagrama simplificado da rede de distribuição em operação

operação e da estrutura topológica da rede. Informalmente, pode-se escrever o seguinte problema de otimização:

Min Perdas do sistema

s.a .

Restrições de carga

Restrições operacionais

Estrutura topológica da rede (rede radial)

As seções a seguir detalham a formulação matemática do problema de reconfiguração da rede de mínima perdas.

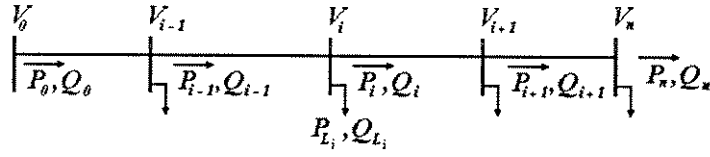


Figura 15. Rede radial de um alimentador

3.2. EQUAÇÕES DE FLUXO DE POTÊNCIA PARA REDES RADIAS

As equações de fluxo de potência para redes radiais são formuladas como um conjunto de equações recursivas (Baran e Wu, 1989a e 1989b). Considere a rede radial com um alimentador, apresentada na Figura 15.

Na figura, V_0 representa o valor da tensão no barramento da subestação, a qual supõe-se constante. As linhas são modeladas como impedâncias série $z_l = r_l + j x_l$, onde r_l e x_l são, respectivamente, a resistência e a reatância da linha l . As cargas, consideradas cargas de potência constante, são representadas por $S_L = P_L + jQ_L$, onde S_L é a potência aparente, P_L a potência ativa e Q_L a potência reativa nas barras de carga L .

Conhecendo-se a potência fornecida na subestação ($S_0 = P_0 + jQ_0$), a potência e a tensão no fim do primeiro trecho da linha podem ser calculadas como segue (Baran e Wu, 1989):

$$S_1 = S_0 - S_{perdas,1} - S_{L_1} = S_0 - z_1 \frac{|S_0|^2}{|V_0|^2} - S_{L_1}$$

$$V_1 = V_0 - z_1 I_0 = V_0 - z_1 \frac{S_0^*}{V_0^*}$$

Generalizando-se, quando o fluxo de potência ativa, o fluxo de potência reativa e a tensão no nó origem i são conhecidos, é possível calcular as mesmas grandezas no nó destino $i + 1$, através do seguinte conjunto de equações recursivas:

$$P_{i+1} = P_i - r_i \frac{P_i^2 + Q_i^2}{V_i^2} - P_{L_{i+1}} \quad (1a)$$

$$Q_{i+1} = Q_i - x_i \frac{P_i^2 + Q_i^2}{V_i^2} - Q_{L_{i+1}} \quad (1b)$$

$$V_{i+1}^2 = V_i^2 - 2(r_i P_i + x_i Q_i) + (r_i^2 + x_i^2) \frac{P_i^2 + Q_i^2}{V_i^2} \quad (1c)$$

As equações 1a, 1b e 1c são conhecidas como *equações dos fluxos nas linhas*. Se P_0, Q_0 e V_0 são conhecidas ou estimadas, então estas grandezas podem ser calculadas para os demais nós, aplicando-se as equações de fluxos nas linhas sucessivamente. Este procedimento é denominado atualização *forward* (Baran e Wu 1989).

De maneira inversa, é definida a atualização *backward*. A atualização tem início nos nós terminais da rede (nó n) e calcula-se a potência ativa, potência reativa e a tensão do nó anterior (nó $n-1$), supondo conhecido P_n, Q_n e V_n . Este procedimento é repetido até o nó 0 (nó que representa a subestação), calculando-se uma nova estimativa para P_0 e Q_0 . A seguir são apresentadas as equações do *fluxo nas linhas backward* (Baran e Wu, 1989).

$$P_{i-1} = P_i + r_i \frac{P_i'^2 + Q_i'^2}{V_i^2} + P_{L_i} \quad (2a)$$

$$Q_{i-1} = Q_i + x_i \frac{P_i'^2 + Q_i'^2}{V_i^2} + Q_{L_i} \quad (2b)$$

$$V_{i-1}^2 = V_i^2 + 2(r_i P_i' + x_i Q_i') + (r_i^2 + x_i^2) \frac{P_i'^2 + Q_i'^2}{V_i^2} \quad (2c)$$

onde $P_i' = P_i + P_{L_i}$ e $Q_i' = Q_i + Q_{L_i}$.

Estabelecidas as equações do fluxo de potência, pode-se calcular as perdas totais do sistema pela expressão a seguir.

$$\text{Total de Perdas} = \sum_{i=0}^{n-1} r_i \frac{P_i^2 + Q_i^2}{V_i^2}$$

3.3. FLUXO DE POTÊNCIA APROXIMADO

O cálculo do fluxo de potência para redes radiais pode ser aproximado (Baran e Wu, 1989b). Essa aproximação é usada para estimar as perdas do sistema a serem minimizadas.

As equações (1) podem ser simplificadas. Os termos quadráticos nas equações, representando as perdas nos arcos, são muito menores que os termos de potência ativa e reativa nos arcos, P_i e Q_i , respectivamente. Eliminando-se os termos de segunda ordem, um novo conjunto de equações é definido.

$$P_{i+1} = P_i - P_{L_{i+1}} \quad (3a)$$

$$Q_{i+1} = Q_i - Q_{L_{i+1}} \quad (3b)$$

$$V_{i+1}^2 = V_i^2 - 2(r_i P_i + x_i Q_i) \quad (3c)$$

Para uma rede radial, a solução do fluxo de carga simplificado é obtida com facilidade. Por exemplo, a solução para a rede da Figura 15 é calculada como segue.

$$P_{i+1} = \sum_{k=i+2}^n P_{L_k} \quad (4a)$$

$$Q_{i+1} = \sum_{k=i+2}^n Q_{L_k} \quad (4b)$$

$$V_{i+1}^2 = V_i^2 - 2(r_i P_i + x_i Q_i) \quad (4c)$$

Considerando $V_i^2 \cong 1 \text{ p.u.}$, pode-se desprezar as equações (3c) e (4c), associadas às quedas de tensões ao longo do alimentador, permitindo a aproximação das perdas de potência no trecho i , conforme equação a seguir.

$$\text{Perdas}_i = r_i \frac{P_i^2 + Q_i^2}{V_i^2} \cong r_i (P_i^2 + Q_i^2)$$

Portanto, o total de perdas estimadas no sistema é calculado pelo somatório das perdas dos trechos da rede de distribuição.

$$\text{Total de Perdas} = \sum_{i=0}^{n-1} r_i (P_i^2 + Q_i^2) \quad (5)$$

A equação (5) é a função objetivo a ser minimizada no problema de reconfiguração para mínimas perdas na rede de distribuição.

3.4. FORMULAÇÃO DO PROBLEMA

Considerando $V_i^2 \cong 1 \text{ p.u.}$ (e implicações associadas), o problema **P** para obtenção da topologia de rede de mínimas perdas de energia em um sistema de distribuição, representado pelo grafo $G = [N, A]$ é formulado a seguir.

$$\mathbf{P} \left\{ \begin{array}{l} \text{Min} \quad \sum_{k \in A'} r_k (P_k^2 + Q_k^2) \\ \text{s.a} \quad \mathbf{A}\mathbf{p} = \mathbf{p}_L \\ \quad \mathbf{A}\mathbf{q} = \mathbf{q}_L \\ \quad \underline{\mathbf{p}} \leq \mathbf{p} \leq \overline{\mathbf{p}} \\ \quad \underline{\mathbf{q}} \leq \mathbf{q} \leq \overline{\mathbf{q}} \\ \quad G' = [N, A'] \quad \text{uma árvore} \end{array} \right.$$

- onde P_k é o fluxo de potência ativa no arco k
 Q_k é o fluxo de potência reativa no arco k
 r_k é a resistência ôhmica atribuída ao arco k
 $\mathbf{A}$ é a matriz de incidência associada ao grafo G (Ahuja et al.,1993)
 $\mathbf{p}$ o vetor dos fluxos de potência ativa
 $\mathbf{q}$ o vetor dos fluxos de potência reativa
 $\mathbf{p}_L$ o vetor das demandas de potências ativas ou potência ativa injetada no nó raiz
 $\mathbf{q}_L$ o vetor das demandas de potências reativas ou potência reativa injetada no nó raiz
 $\underline{\mathbf{p}}$ e $\overline{\mathbf{p}}$ são vetores dos limites para os fluxos de potência ativa
 $\underline{\mathbf{q}}$ e $\overline{\mathbf{q}}$ são vetores dos limites para os fluxos de potência reativa
 A' o conjunto de arcos com fluxo diferente de zero

Cada linha da matriz $\mathbf{A}$ está associada a uma restrição do balanço de fluxo de potência, ativo ou reativo, no nó. Supõe-se a injeção de potência ativa e reativa no nó raiz suficiente para suprir a demanda de potência nos nós, isto é, o fornecimento e demanda dos nós devem satisfazer as condições $\sum_{i \in N} p_i = 0$ e $\sum_{i \in N} q_i = 0$.

As variáveis e restrições do problema **P** podem ser particionadas nos subconjuntos *ativo* e *reativo*. Na maioria das redes há uma proporcionalidade entre as demandas de potências ativas e reativas², permitindo separar **P** em dois subproblemas aproximados, de minimização de perdas de potência ativa e reativa – nesta condição as árvores encontradas para a solução dos problemas de minimização das perdas ativas e reativas são as mesmas. Particiona-se a função objetivo em perdas ativas e perdas reativas.

No caso geral, pode-se escrever a equação (5) como segue.

$$\text{Total de Perdas} = \sum_{k \in A'} r_i P_i^2 + \sum_{k \in A'} r_i Q_i^2 \quad (6)$$

onde $\sum_{k \in A'} r_i P_i^2$ e $\sum_{k \in A'} r_i Q_i^2$ são o total de perdas da potência ativa e potência reativa, respectivamente.

Pode-se então formular estes subproblemas na forma genérica:

$$\text{P1} \begin{cases} \text{Min} & \sum_{k \in A} r_k x_k^2 & (7) \\ \text{s.a} & \mathbf{Ax} = \mathbf{b} & (8) \\ & \underline{\mathbf{x}} \leq \mathbf{x} \leq \bar{\mathbf{x}} & (9) \\ & G' = [N, A'] \text{ uma árvore} & (10) \end{cases}$$

A equação (7) é uma função não linear, representando o total de perdas de energia no sistema. A perda no arco k é calculada pela multiplicação da resistência ôhmica da linha, r_k , pelo quadrado do fluxo de potência ativa ou reativa, x_k . As restrições de carga do sistema estão representadas pela equação (8). A (com dimensão $m \times n$) é a matriz incidência nó-arco para o grafo $G = [N, A]$, $\mathbf{x}$ o vetor dos fluxos de potência ativa ou reativa, $\mathbf{b}$ o vetor das demandas ativa ou reativa nos nós, ou a potência (ativa ou reativa) injetada no nó raiz, $\underline{\mathbf{x}}$ e $\bar{\mathbf{x}}$ são vetores dos limites para os fluxos de potência, ativa ou reativa, e A' o conjunto de arcos com fluxos diferentes de zero.

² Esta condição é válida para redes que tenham fator de potência alto.

De forma análoga ao problema P, supõe-se que o fornecimento e demanda de potência nos nós satisfaz a condição $\sum_{i \in N} b_i = 0$. As equações (9) e (10) correspondem, respectivamente, à capacidade física das linhas e à restrição de operação radial do sistema.

Como consequência da restrição (10), a solução do problema P1 é uma árvore recobridora de perdas mínimas da rede de distribuição. Entretanto, contrariamente ao problema clássico da árvore recobridora de peso mínimo, o custo $r_k x_k^2$, associado a cada arco, varia com a mudança da configuração da árvore G' . Esta generalização conduz a solução do problema P1 para longe dos algoritmos de otimização gulosos, muito simples, que resolvem o problema tradicional de encontrar uma árvore de recobrimento de peso mínimo (Ahuja et. al. 1993).

3.5. ESTRATÉGIA DE SOLUÇÃO

Conforme discutido no Capítulo 1 (seção 1.6), aparentemente a solução de P1 pode ser obtida pela enumeração exaustiva de todas as possíveis configurações de rede, construídas combinando a abertura e fechamento de chaves para manter a estrutura radial da rede. A aplicação deste procedimento é limitado a redes muito pequenas e com poucas chaves. No exemplo do Capítulo 1, para uma rede com apenas 6 chaves, sendo duas abertas, existem 11 configurações de rede radial diferentes. Em redes maiores o número de configurações possíveis inviabiliza este procedimento de *enumeração explícita*.

A maioria dos trabalhos referenciados no Capítulo 1, utiliza heurísticas para contornar o problema da explosão combinatória das configurações de rede. Os métodos desenvolvidos apresentam bons resultados, porém não garantem a obtenção da solução ótima.

A contribuição deste trabalho é sugerir uma metodologia para explorar um número reduzido de configurações de rede e garantir a obtenção da solução ótima do problema de reconfiguração de rede para mínimas perdas. Para isso, foram usados

conjuntamente conceitos da área de inteligência artificial (Pearl, 1994) e técnicas de otimização em redes (Lyra e Tavares, 1988).

Informações no contexto do problema podem guiar um procedimento de busca formal para encontrar a solução ótima de **P1**. Explorando esta idéia, concebeu-se um processo recursivo de duas etapas. Na primeira etapa a restrição de radialidade da rede é relaxada, obtendo-se uma solução aproximada de **P1**. Esta solução, na maioria dos casos, apresenta uma situação onde os arcos com fluxos diferentes de zero formam anéis (ou malhas). Consequentemente, não é uma solução de **P1** (por não satisfazer a restrição de radialidade), mas define um limitante inferior para as perdas da solução ótima de **P1**.

Uma chave é selecionada na segunda etapa para abrir um dos anéis existentes, aproximando-se da rede radial. A escolha da chave a ser aberta é baseada nos valores dos fluxos e gradientes, obtidos na primeira etapa através de um método de otimização especializado em grafo, descrito no item 3.5.2. Estas etapas são repetidas, recursivamente, até que uma rede radial seja encontrada.

3.5.1. Solução Otimista

Um problema de fluxo de custo mínimo convexo, com função não linear quadrática, P_r , é caracterizado quando a restrição (4) é relaxada no problema **P1**,

$$P_r \begin{cases} \text{Min} & \phi(\mathbf{x}) & (11) \\ \text{s.a} & \mathbf{Ax} = \mathbf{b} & (12) \\ & \underline{\mathbf{x}} \leq \mathbf{x} \leq \bar{\mathbf{x}} & (13) \end{cases}$$

onde $\phi(\mathbf{x}) = \sum_{k \in A} r_k x_k^2$.

Teoricamente, P_r pode ser resolvido por qualquer algoritmo de otimização não linear (Luenberger, 1984). Entretanto, como problemas análogos são resolvidos inúmeras vezes no procedimento recursivo de duas etapas, foi adotado um método de gradiente reduzido especializado para grafos (Lyra e Tavares, 1988), trabalhado para explorar os aspectos estruturais do modelo de grafo.

3.5.2. Método do Gradiente Reduzido para Fluxos em Redes

O método do gradiente reduzido particiona as variáveis do problema nos conjuntos das variáveis dependentes e independentes (Luenberger, 1984). Seja $\mathbf{I}$ o conjunto de índices para as variáveis dependentes e $\mathbf{J}$ o conjunto de índices para as variáveis independentes. Considere também a seguinte notação adicional:

$\mathbf{x}_I$ - vetor dos fluxos cujos componentes são x_i , $i \in \mathbf{I}$;

$\mathbf{x}_J$ - vetor dos fluxos cujos componentes são x_j , $j \in \mathbf{J}$;

$\mathbf{A}^k$ - coluna k da matriz $\mathbf{A}$;

$\mathbf{A}^I$ - matriz formada pela união das colunas i , $i \in \mathbf{I}$;

$\nabla_I \phi(x)$ - vetor das derivadas parciais $\frac{\partial \phi(x)}{\partial x_i}$, $i \in \mathbf{I}$.

As variáveis no conjunto dependente devem satisfazer a condição de não-degenerescência (Luenberger, 1984), ou seja, os arcos associados a $\mathbf{x}_I$ devem definir uma árvore recobridora para a rede e satisfazer a condição de $\underline{\mathbf{x}} \leq \mathbf{x} \leq \bar{\mathbf{x}}$.

Dada uma solução factível inicial para fluxos nos arcos (no caso de redes de distribuição, a solução factível inicial adotada é o fluxo de potência nas linhas da rede em operação) e uma partição $(\mathbf{I}, \mathbf{J})$, sendo $\mathbf{I}$ o conjunto dos arcos associados a uma árvore e $\mathbf{J}$ o conjunto dos arcos pertencentes a co-árvore, uma iteração do algoritmo do método do gradiente reduzido especializado para grafo pode ser sintetizado na sequência de passos apresentada a seguir.

Etapa1. Calcular o gradiente da $\phi(x)$, $\nabla \phi(x)$.

Etapa2. Obter as variáveis duais λ , resolvendo o sistema $\lambda \mathbf{A}^I = \nabla_I \phi(x)$.

Etapa3. Calcular todas as componentes do gradiente reduzido ρ_j , para todo $j \in \mathbf{J}$,

$$\rho_j = \nabla_j \phi(x) - (\lambda_p(x) - \lambda_q(x)),$$

onde λ_p e λ_q são, respectivamente, variáveis duais associadas ao nó-origem (p) e nó-destino (q) do arco representado por $\mathbf{A}^j$.

Etapa 4. Projetar o gradiente reduzido sobre as restrições de canalização, ou seja

$$\Delta x_j = \begin{cases} \rho_j, & \text{se } \underline{x}_j \leq x_j \leq \bar{x}_j \\ & \text{ou } x_j = \underline{x}_j \text{ e } \rho \leq 0 \\ & \text{ou } x_j = \bar{x}_j \text{ e } \rho \geq 0 \\ 0, & \text{se } x_j = \underline{x}_j \text{ e } \rho \geq 0 \\ & \text{ou } x_j = \bar{x}_j \text{ e } \rho \leq 0 \end{cases}$$

Etapa 5. Se $\Delta x_j = 0$ pare, o atual vetor de fluxos é ótimo - satisfazendo as condições de Kuhn-Tucker para minimização com restrições (Luenberger, 1984). Caso contrário, obter os componentes dependentes da direção de busca Δx_1 ,

$$\Delta x_1 = \sum_{j \in J} [-\hat{A}^j] \Delta x_j$$

onde $\hat{A}^j$ é solução de $A^j = A^1 \hat{A}^j$

Etapa 6. Calcular

$$\alpha_1 = \max \{ \alpha \text{ tal que } \underline{x}_1 \leq x_1 - \alpha \Delta x_1 \leq \bar{x}_1 \}$$

$$\alpha_2 = \max \{ \alpha \text{ tal que } \underline{x}_J \leq x_J - \alpha \Delta x_J \leq \bar{x}_J \}$$

$$\alpha_3 = \min \{ \phi(x - \alpha \Delta x) \text{ tal que } 0 \leq \alpha \leq \alpha_1, 0 \leq \alpha \leq \alpha_2 \}$$

Atualizar $\mathbf{x}$, $\mathbf{x} \leftarrow (\mathbf{x} - \alpha_3 \Delta \mathbf{x})$.

Etapa 7. Se $\alpha_3 \leq \alpha_1$ voltar para a Etapa 1. Caso contrário, encontrar um arco a_e cujo fluxo satisfaça a condição $\underline{x}_e < x_e < \bar{x}_e$ e que defina uma nova árvore recobridora quando substituir o arco a_s , que chega a um de seus limites; atualize a partição $(\mathbf{I}, \mathbf{J})$,

$$\mathbf{I} \leftarrow \mathbf{I} - \{s\} + \{e\}$$

$$\mathbf{J} \leftarrow \mathbf{J} - \{e\} + \{s\}$$

e retorne a Etapa 1.

A eficiência do método especializado em grafos é caracterizado:

- pela facilidade de encontrar um vetor de fluxos factível e uma partição $(\mathbf{I}, \mathbf{J})$; por exemplo, a configuração de rede em operação para o sistema de distribuição;
- pela estrutura de rede do problema. As etapas 2 e 5 são executadas sem operações de matrizes; as tarefas necessárias são realizadas com a manipulação da rede implícita, auxiliada pelo uso de estrutura de dados baseada nos rótulos *predecessor*, *fio*, e *nível* (Capítulo 2);

- pela inexistência de erros de arredondamento no processo computacional de pivoteamento (nenhuma operação de divisão é requerida).

O método do gradiente reduzido é utilizado para obter uma solução aproximada do problema P1. Esta solução reúne informações usadas no controle da busca recursiva para a seleção da próxima chave a ser aberta, definindo uma nova configuração de rede.

3.5.3. Comentários

A interpretação do problema como uma busca no espaço de estados, associando a todas as possíveis configurações de redes, permite aplicar métodos de busca, que procuram explorar apenas uma parte do espaço de estados com auxílio de informações extraídas do contexto do problema, sem perda do objetivo principal - a configuração de rede radial que minimize as perdas do sistema de distribuição de energia elétrica.

No capítulo a seguir, é apresentada a formalização do espaço de estados para o problema de reconfiguração da rede para mínimas perdas e os procedimentos de busca utilizados na segunda etapa do processo recursivo.

CAPÍTULO 4

CONCEITOS DE IA NA BUSCA DE SOLUÇÕES ÓTIMAS

4.1. INTRODUÇÃO

Os conceitos usados na área de inteligência artificial (IA) para resolver problemas de busca podem ser utilizados para encontrar a configuração da rede radial de mínimas perdas. Uma estratégia de busca sistemática, no conjunto de configurações alternativas, pode encontrar a árvore recobridora factível ótima - solução do problema P. O uso de conhecimento no domínio do problema reduz o número de configurações de rede a serem examinadas. O conhecimento é usado na seleção da próxima configuração a ser examinada nos processos de busca.

Neste capítulo, descreve-se inicialmente uma forma conveniente de representação de problemas de busca no contexto da IA. Com o modelo de representação definido, discute-se procedimentos de busca, distinguindo-se as estratégias de controle irrevogáveis e as estratégias de controle por tentativas. Entre as estratégias de controle por tentativas destacam-se as *backtracking* e *busca em grafo*.

4.2. BUSCA NO ESPAÇO DE ESTADOS

Os processos de busca na área de IA são descritos através de uma estrutura formal. Estes processos podem ser comparados à tarefa de encontrar, ou construir, um

objeto com características bem definidas, observando os seguintes componentes (Pearl, 1989):

- uma estrutura de símbolos ou códigos que possa representar os estados do *espaço de estados*;
- regras ou operadores para transformação de um estado em outro, para varrer sistematicamente o espaço de estados;
- um método para selecionar as transformações, procurando descobrir o estado desejado o mais cedo possível.

Na nomenclatura adotada na área de IA, estes três componentes básicos são conhecidos, como *banco de dados*, *operadores* e *controle de estratégia*, respectivamente (Pearl, 1984):

Para realizar uma busca sistemática no espaço de estados é necessário caracterizar os *estados iniciais* e os *estados finais*. Os estados iniciais são as situações possíveis onde o processo de busca pode ser iniciado. Os estados finais são aqueles aceitos como soluções factíveis do problema.

A representação do espaço de estados é construída a partir das características relevantes dos objetos no domínio do problema. Nem sempre esta representação é evidente. Deve-se trabalhar para obter uma representação que possa corresponder a uma estrutura de dados exequível. Por exemplo, no jogo de xadrez um estado pode ser representado pelas posições das peças no tabuleiro, armazenadas em uma matriz 8x8. O estado inicial é definido pela posição das peças no início do jogo. Os estados finais não necessitam ser enumerados, basta descrever claramente suas características e convertê-las em condições a serem satisfeitas. No caso do jogo de xadrez é impraticável enumerar todas as posições de xeque mate possíveis, porém, pode-se dizer que o objetivo do jogo foi atingido quando não existir alternativas de saída para o Rei, que está sendo atacado. Estas características podem ser convertidas em regras, que devem ser verificadas a cada a posição das peças no tabuleiro, para verificar se um estado final foi encontrado.

As regras possuem uma precondição para aplicação em um certo estado. Se a precondição é satisfeita para uma certa regra, ela pode ser aplicada para produzir um novo estado do espaço de estados. Normalmente, essas regras são denominadas *regras de produção* (Nilson, 1980).

4.3. PROCEDIMENTO DE BUSCA

O resultado da aplicação sistemática das regras de produção sobre os estados é representado por uma estrutura de grafo, mais particularmente, por uma estrutura de árvore. A Figura 16 exemplifica, sem perda de generalidade, uma parcela da árvore de busca gerada por um processo de *busca em grafo* genérico.

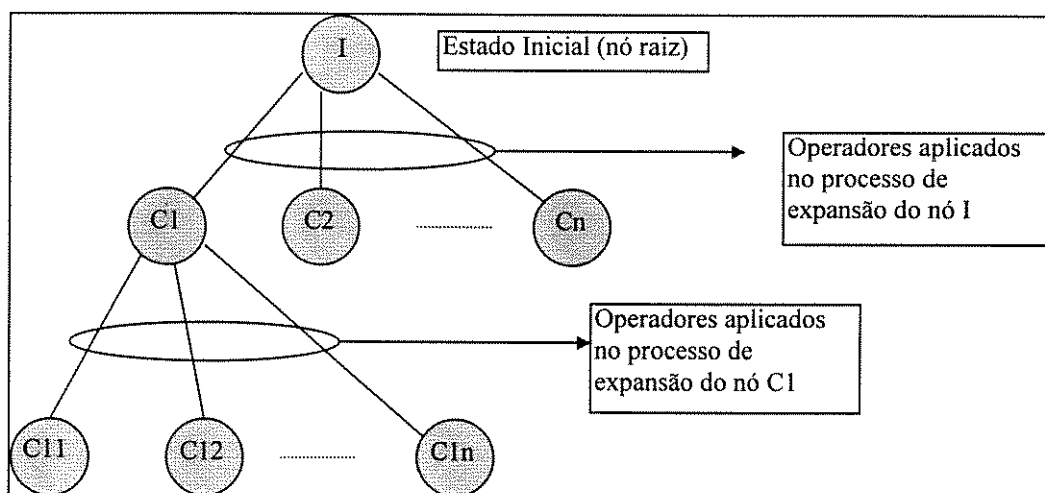


Figura 16. Árvore de busca criada pelo processo de expansão

O processo de busca tem início no nó I, raiz da árvore de busca. Este nó está associado ao estado inicial. No caso do nó I não ser um nó final (solução do problema), o processo gera estados que possam ser alcançados do estado inicial, aplicando-se os operadores (ou regras de produção) selecionados pela estratégia de controle adotada. Após a geração de todos os nós sucessores, o nó pai é então considerado um nó *expandido*; o procedimento é conhecido como *expansão* do nó. Por exemplo, a expansão do nó I, na Figura 16, gera o conjunto dos nó sucessores de I, $I_{suc} = \{C1, C2, \dots, Cn\}$. Após esta operação o nó I é marcado como expandido.

Quando apenas alguns (ou mesmo um) dos sucessores de um certo nó são gerados, diz-se que o nó foi *explorado*; o procedimento é denominado *exploração* do nó.

Um dos nós gerados é escolhido para ser explorado ou expandido. Esta seleção é realizada segundo critérios estabelecidos e adotados pela estratégia de controle do processo de busca. O conjunto dos nós gerados nesta nova exploração ou expansão é reunido, conforme um critério preestabelecido, ao conjunto de nós previamente gerados e ainda não explorados.

Em um dado instante do procedimento de busca descrito, os nós da árvore de busca podem ser divididos em três conjuntos disjuntos: *conjunto de nós expandidos*, *conjunto de nós explorados* e *conjunto de nós gerados e não explorados*. Na Figura 16, o nó C1 foi selecionado para ser explorado (no caso expandido), gerando o conjunto $C1_{suc} = \{C11, C12, \dots, C1n\}$. O conjunto $C1_{suc}$ e conjunto I_{suc} são reunidos. Por exemplo, os nós do conjunto $C1_{suc}$ podem ser incluídos no final do conjunto I_{suc} , e um novo conjunto de nós gerados e não explorados é obtido, $\{C1, C2, \dots, Cn, C11, C12, \dots, C1n\}$. O nó C1 é então marcado como expandido neste caso. O procedimento repete-se até que um nó terminal seja encontrado (ou identifique-se que não é possível encontrar uma solução para o problema).

A busca no espaço de estados pode ser resumida como um processo sistemático de tentativas de aplicação das regras de produção, até encontrar um estado final. A estratégia de controle seleciona as regras, mantém o registro da sequência de regras já experimentadas e armazena os estados já avaliados. Utilizando estratégias de controle diferentes, pode-se obter procedimentos de busca distintos. A seguir são discutidos três algoritmos genéricos baseados nas estratégias de controle *irrevogável*, *backtracking* e *best-first*.

4.3.1. Algoritmo básico da busca irrevogável

A busca irrevogável recebe este nome por não permitir retornar a um ponto anterior na árvore de busca, ou seja, não permite “arrependimento” em relação ao caminho escolhido. Esta estratégia de controle está normalmente baseada numa

otimização local, onde o conhecimento é usado de forma “gulosa” para escolher o que parece ser o melhor caminho, em direção ao nó objetivo.

Um exemplo dessa classe de procedimentos é o processo de busca chamado *hill-climbing*. (Nilson, 1980). Este procedimento usa o conhecimento local para construir a solução global. Ele é útil quando existe uma função com informação suficiente para guiar a busca ao ótimo global (ou próximo a ele).

As etapas de um algoritmo básico que usa a estratégia de controle irrevogável são apresentadas a seguir.

Algoritmo Básico da Busca Irrevogável

- Etapa 1.* Crie uma árvore de busca, com apenas o nó raiz, r . Inclua r na lista ABERTO.
- Etapa 2.* Se a lista ABERTO está vazia, pare. A busca não teve sucesso, isto é, a solução não foi encontrada. Caso contrário, continue na Etapa 3.
- Etapa 3.* Remova o primeiro nó da lista ABERTO. Denomine este nó de n .
- Etapa 4.* Expandir o nó n , gerando o conjunto M dos sucessores de n .
- Etapa 5.* Para cada nó n' pertencente ao conjunto M faça:
 - Etapa 5.1.* Se n' é uma solução para o problema, pare. Mostre a solução encontrada.
 - Etapa 5.2.* Caso contrário, quando n' não é solução do problema, inclua-o na lista ABERTO.
- Etapa 6.* Ordene a lista ABERTO por algum critério predefinido. Volte para a etapa 2.

Neste algoritmo, uma função que avalia cada um dos nós gerados é utilizada para ordenar a lista ABERTA. Na etapa 3, remover o primeiro elemento da lista ABERTA é equivalente a selecionar o melhor estado gerado até o momento.

4.3.2. Algoritmo Backtracking

A estratégia de controle *backtracking* padrão é uma variação da busca em profundidade que possui o esquema *last-in-first-out* - LIFO (Pearl, 1984). A Figura 17

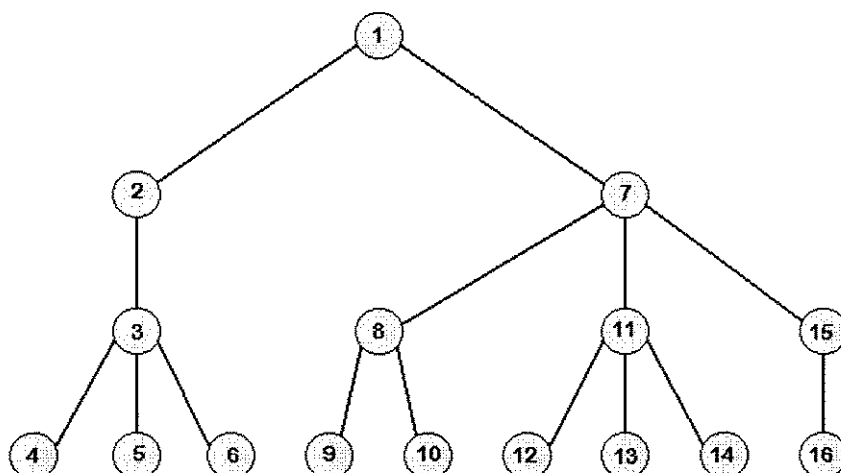


Figura 17. Busca em profundidade

ilustra uma ordem dos nós gerados com um procedimento de busca *backtracking* (os números dos nós indicam a ordem em que foram visitados).

A busca tem início no nó raiz da árvore de busca (nó 1) e procura encontrar o caminho da solução gerando um sucessor a cada nó explorado; nesse processo, afasta-se do nó raiz e, em consequência, aumenta a profundidade da árvore de busca. A profundidade é aumentada até que um nó objetivo, ou nó final (nó sem sucessor), seja atingido. Se o nó objetivo é gerado, a busca termina e o caminho da solução é recuperado. Se um nó final é alcançado, a busca retorna (*backtracking*) para o nó antecessor mais próximo no caminho em exploração, verificando as possibilidades de explorar um outro sucessor. Em caso afirmativo, a busca atualiza o caminho de exploração. Por exemplo, na Figura 17, quando o algoritmo atinge o nó 4, após ter percorrido o caminho $\{1,2,3\}$, verifica que este é um nó final, e retorna ao nó 3. A busca reinicia no nó 5.

A seguir é apresentada a seqüência de etapas do procedimento de busca utilizando a estratégia de controle *backtracking*.

Algoritmo Básico de Busca *Backtracking*

Etapa 1. Inclua o nó inicial na lista ABERTO. Se este nó é um nó objetivo, termina o procedimento.

- Etapa 2.* Se a lista ABERTO está vazia, pare, pois nenhum nó objetivo foi encontrado – problema sem solução. Caso contrário, continue na Etapa 3 (isto é, a busca continua no caminho em exploração).
- Etapa 3.* Seja n o primeiro nó na lista ABERTO. Se a profundidade de n é igual a uma profundidade limite especificada, ou se todos os sucessores de n já foram visitados, remova o nó n da lista ABERTO e retorne à Etapa 2. Caso contrário, continue na Etapa 4.
- Etapa 4.* Gere um novo sucessor de n , chame-o de n' . Inclua n' na lista ABERTO e armazene um apontador para o nó n .
- Etapa 5.* Marque o nó n para indicar que o arco (n, n') da árvore de busca foi visitado.
- Etapa 6.* Se n' é um nó objetivo, termine o procedimento e mostre a solução obtida, utilizando os apontadores armazenados; caso contrário, continue na etapa 7.
- Etapa 7.* Se n' é um nó sem sucessor, retire-o da lista ABERTO.
- Etapa 8.* Volte para a Etapa 2.

Note que o procedimento básico termina a busca ao encontrar um nó objetivo – o primeiro que satisfaça as condições estabelecidas (ou quando a busca é concluída com insucesso). Não existe no algoritmo nenhum cuidado em verificar se a solução encontrada é a solução ótima para o problema. O algoritmo básico descrito pertence à classe dos procedimentos que atendem aos problemas de busca por *satisfação*.

Com pequenas alterações no algoritmo básico é possível definir um algoritmo *backtracking* para busca da solução ótima. Para isto, na Etapa 6 do algoritmo compara-se a solução obtida com a melhor solução encontrada anteriormente. No caso da solução encontrada ser melhor, esta deve atualizar a solução anterior. O processo de busca é encerrado quando não houver mais nós a serem expandidos. Como esta condição é verificada na Etapa 3, deve-se alterá-la para recuperar a solução ótima encontrada, caso exista. As etapas do algoritmo básico para obtenção da solução ótima do problema são relacionadas a seguir.

Algoritmo Básico de Busca *Backtracking* para Obter a Solução Ótima

- Etapa 1.* Inclua o nó inicial na lista ABERTO. Se este nó é um nó objetivo, armazenar a solução.

- Etapa 2.* Se a lista ABERTO está vazia, pare e recupere a solução ótima, se existir. Caso contrário, continue na Etapa 3.
- Etapa 3.* Seja n o primeiro nó na lista ABERTO. Se a profundidade de n é igual a uma profundidade limite especificada, ou se todos os sucessores de n já foram visitados, remova o nó n da lista ABERTO e retorne para a Etapa 2. Caso contrário continue na Etapa 4.
- Etapa 4.* Gere um novo sucessor de n , chame-o de n' . Inclua n' na lista ABERTO e armazene um apontador para o nó n .
- Etapa 5.* Marque o nó n para indicar que o arco (n, n') da árvore de busca foi visitado.
- Etapa 6.* Se n' é um nó que satisfaz as restrições do problema, verifique se a solução obtida é melhor que a solução atual (i.e., a melhor solução encontrada anteriormente); em caso afirmativo, armazene a solução mais recente e continue na Etapa 7.
- Etapa 7.* Se n' é um nó sem sucessor, retire-o da lista ABERTO.
- Etapa 8.* Volte para a Etapa 2.

Este algoritmo pressupõe que o valor da solução atual no início do processo seja igual a ∞ ou, quando possível, igual ao valor de alguma solução conhecida para o problema. O algoritmo realiza uma exploração exaustiva do espaço de estados definido para o problema, inviabilizando sua aplicação para a maioria dos casos práticos. Porém, acrescentando-se conhecimentos do contexto do problema nas etapas de seleção do nó a ser explorado e ordenação da lista ABERTO, é possível diminuir o número de nós explorados, reduzindo a porção do espaço de estados visitada pelo procedimento de busca. Este assunto será retomado na item 4.6 deste capítulo.

4.3.3. Algoritmo “Best-First”

O conhecimento heurístico sobre o contexto do problema pode ser usado no procedimento de busca de maneira formal. Por exemplo, a escolha do próximo nó a ser expandido pode utilizar informações heurísticas que direcione o processo de busca para a solução do problema, o mais cedo possível. A quantidade de informação embutida na heurística utilizada determina a porção explorada do espaço de estados.

A escolha do nó mais promissor na árvore de busca pode ser estimada numericamente por uma *função de avaliação heurística*, $f(n)$. A função heurística, em geral, depende da representação adotada para o nó n , da representação do nó objetivo, das informações obtidas pelo processo de busca até o momento e de outros conhecimentos relacionados com a solução do problema. Algoritmos que utilizam esse tipo de estratégia de controle são chamados *best-first* (Pearl, 1984).

Normalmente, o conceito *best-first* é utilizado com a família de procedimentos de busca denominada *busca em grafo* (Nilson, 1980), discutida adiante. No entanto, este trabalho investiga também as possibilidades do conceito *best-first* em uma busca da família *backtracking* (*backtracking heurístico*)

4.3.4. Busca em Grafo

As etapas de um algoritmo básico da família *busca em grafo*, utilizando o conceito *best-first*, são apresentadas a seguir. Para manter a generalização, não é imposta nenhuma restrição na natureza de $f(\cdot)$ e, se dois caminhos levam ao mesmo nó, aquele que apresentar o maior valor para $f(\cdot)$ será descartado.

Algoritmo Básico de Busca com Estratégia *Best-First*

- Etapa 1.* Crie uma árvore de busca, com apenas o nó raiz, r . Inclua r na lista ABERTO.
- Etapa 2.* Crie uma lista FECHADO (inicialmente esta lista está vazia).
- Etapa 3.* Se a lista ABERTO está vazia, pare; a busca não teve sucesso, isto é, a solução não foi encontrada. Caso contrário, continue na Etapa 4.
- Etapa 4.* Remova o nó n com o melhor valor de $f(n)$ da lista ABERTO (eventuais empates são resolvidos arbitrariamente). Inclua o nó n na lista FECHADO.
- Etapa 5.* Expanda o nó n , gerando o conjunto M dos sucessores de n .
- Etapa 6.* Se algum dos sucessores é um nó objetivo, pare e recupere a solução obtida.
- Etapa 7.* Para cada nó n' pertencente ao conjunto M faça:
 - Etapa 7.1.* Calcule o valor de $f(n')$

Etapa 7.2. Se n' não foi gerado anteriormente (n' não pertence à lista ABERTO ou a lista FECHADO), inclua o nó n' na lista ABERTO. Atribua ao nó n' o valor de $f(n')$, calculado na Etapa 7.1.

Etapa 7.3. Caso o nó n' já tenha sido gerado (n' pertence à lista ABERTO ou FECHADO), verifique se ele é mais promissor que o anteriormente gerado, comparando o valor de $f(n')$ com o valor de f previamente armazenado. Se o valor anterior de f é melhor, descarte o nó n' , recentemente gerado. Se o valor de f para o nó n' é melhor, substitua o valor anterior pelo novo valor calculado (faça n' apontar para n , substituindo seu antecessor anterior). Neste último caso, se o nó n' pertence a lista FECHADO, mova-o para a lista ABERTO.

Etapa 8. Volte à Etapa 3.

4.4. BUSCA DA ÁRVORE RECOBRIDORA PARA REDES DE DISTRIBUIÇÃO

Uma árvore recobridora para o problema de reconfiguração da rede de distribuição pode ser encontrada através da busca no espaço de estados. Um estado é caracterizados pela configuração de rede representada por grafo – o padrão de fluxo associado ao estado corresponde às perdas mínimas para a configuração de rede associada. O nó inicial da árvore de busca está associado ao valor solução ótima do problema relaxado P_r (uma rede com malhas que define um limitante inferior para o total de perdas do sistema de distribuição). Os nós sucessores são gerados pela abertura de ciclos (ou seja, remoção de arcos no grafo) e associados ao valor solução do problema P_r para a nova configuração de rede. Um exemplo da árvore gerada pelo procedimento de busca é apresentado na Figura 18.

O nó inicial, e cada um dos nós internos na árvore de busca, contém a informação de uma solução infactível (aproximada) do problema P . A infactibilidade dessa solução é removida gradualmente, à medida que os nós são gerados, aumentando a distância do nó inicial.

A seguir são apresentados algoritmos de busca para o problema de reconfiguração da rede de distribuição para mínimas perdas, baseados nos algoritmos básicos de busca irrevogável e *backtracking*, discutidos anteriormente.

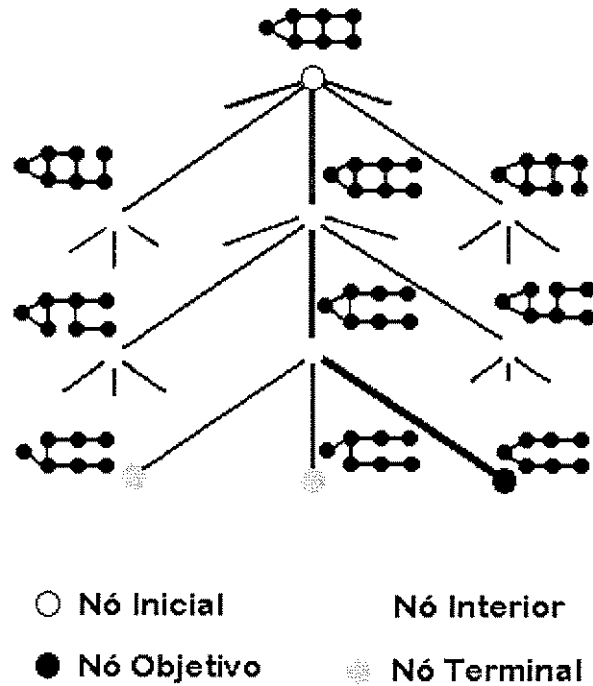


Figura 18. Árvore de busca gerada para a solução do problema P

4.5. BUSCA IRREVOGÁVEL

Um procedimento de busca com estratégia de controle irrevogável para resolver o problema de reconfiguração de rede para mínimas perdas é apresentado a seguir.

Algoritmo Busca Irrevogável

- Etapa 1.* Resolva o problema P_r para a configuração de rede associada ao nó inicial.
- Etapa 2.* Se não existirem ciclos, pare; mostre a solução do problema. Caso contrário, continue na Etapa 3.
- Etapa 3.* Obter o conjunto M_i contendo todos os arcos que pertencem a pelo menos um ciclo.
- Etapa 4.* Remova um arco k pertencente a M_i , conforme um critério predefinido. Inclua o arco k na lista CAMINHO.
- Etapa 5.* Resolva o problema aproximado P_r para a configuração de rede resultante da remoção do arco k .
- Etapa 6.* Volte para a etapa 2.

A lista CAMINHO armazena todos os arcos que foram removidos da configuração de rede inicial, até obter-se a rede radial. A operação de expansão de nós é definida pela resolução do problema P_r e a formação do conjunto M dos arcos pertencentes aos ciclos. A remoção do arco, que atenda um critério predefinido (Etapa 4), é a regra de transformação de um estado a outro utilizada no algoritmo. Sendo o número de arcos a serem retirados finito, este algoritmo sempre encontra uma rede radial. Note que a árvore de busca é gerada implicitamente pelo algoritmo.

Um critério para remoção do arco na Etapa 4 pode ser, por exemplo, a escolha do arco com o menor fluxo. Estratégia semelhante foi utilizada por Merlin (1975), para obter uma solução aproximada do problema de minimização das perdas.

4.6. BUSCAS BACKTRACKING INFORMADO

Conhecimento no domínio do problema pode ser usado para detectar e podar subconjuntos de nós da árvore de busca, sem explorá-los explicitamente. Esse recurso pode melhorar sensivelmente o desempenho do algoritmo *backtracking padrão*, discutido na seção 4.3.2.

A solução do problema aproximado P_r , para a configuração de rede associada ao nó n na árvore de busca, é usada para indicar o próximo nó a ser gerado, incluindo-o no caminho da solução. Além disso, uma solução factível é adotada como referência do melhor valor para as perdas encontradas pelo procedimento de busca, permitindo a *poda por dominância* (Pearl, 1984). Adicionando este conhecimento ao *backtracking padrão*, produz-se o algoritmo que denominaremos *backtracking informado*.

Quando um nó n é gerado no *backtracking informado*, o valor da solução do problema P_r associado ao nó n é comparado com o valor das perdas da solução de referência. Se o valor da solução de referência é excedido, o algoritmo poda o caminho e retorna (executa o *backtracking*), direcionando a busca para um caminho alternativo. Assim, evita a exploração de caminhos que conduziriam à árvores recobridoras não-ótimas para a rede de distribuição. Cada vez que a busca encontrar uma árvore

recobridora factível (um nó final) com um valor melhor para as perdas, a solução de referência é atualizada.

De forma análoga ao *backtracking padrão*, duas listas são usadas na implementação da estratégia *backtracking informado*, ambas estruturadas como pilhas. A lista ABERTO mantém os sucessores do nó em exploração, representados pelos arcos candidatos a serem retirados da configuração associada ao nó. A lista CAMINHO armazena o caminho da solução que está sendo explorada.

Denotando S como o valor das perdas na solução de referência e $f(n)$ o valor da solução de P_r no nó n , a estratégia *backtracking informado* pode ser resumida como segue.

Algoritmo Backtracking Informado

- Etapa 1.* Atribua à variável S o valor das perdas da configuração em operação disponível para a rede de distribuição.
- Etapa 2.* Resolva o problema P_r para a configuração de rede associada ao nó inicial. Se existirem ciclos, inclua os arcos pertencentes aos ciclos na lista ABERTO numa ordem predefinida (por exemplo, em ordem crescente dos valores do gradiente). Caso não existam ciclos, termine o procedimento de busca – a configuração de rede inicial é a solução do problema.
- Etapa 3.* Se a lista ABERTO está vazia, pare e recupere a solução ótima; se existir; caso contrário continue na Etapa 4.
- Etapa 4.* Gere um novo nó n , removendo o primeiro arco da lista ABERTO e resolvendo o problema aproximado P_r para a configuração resultante (a solução de P_r fornece um novo padrão de fluxo ótimo e $f(n)$ o valor associado para as perdas); inclua n na lista CAMINHO.
- Etapa 5.* Compare o mais novo valor calculado para $f(n)$ com S ; se $f(n) \geq S$, descarte o novo nó gerado, removendo-o da lista CAMINHO e retorne à Etapa 3 (em outras palavras, pode o caminho e faça *backtracking*). Caso contrário, se $f(n) < S$, continue na Etapa 6.
- Etapa 6.* Identifique os ciclos no nó n ; inclua no topo da pilha ABERTO a lista dos arcos que pertencem aos ciclos de forma ordenada e retorne a Etapa 2. Caso contrário, se o padrão do fluxo ótimo



associado a n não possuir nenhum ciclo (ou seja, uma rede radial, critério que identifica um nó terminal) continue na Etapa 7.

Etapa 7. Substitua a solução de referência S pelo valor da função $f(n)$, $S \leftarrow f(n)$; salve a solução associada a n e retire o nó do topo da lista CAMINHO (*backtracking*) – n é um nó final com o melhor valor de perdas até este instante. Retorne à Etapa 2.

Note que a estratégia de busca *backtracking informado* não perde a solução de mínimas perdas, isto é, como no *algoritmo backtracking padrão*, esta estratégia é uma busca admissível. De fato, a ordenação da lista ABERTO, na *Etapa 2* e na *Etapa 6*, apenas altera a ordem na qual os nós são gerados. Por outro lado, como o valor das perdas aumenta a cada arco retirado da rede, pode-se garantir que o caminho podado na *Etapa 5* levaria a uma árvore recobridora não-ótima.

4.7. BUSCA BACKTRACKING HEURÍSTICO

O valor da função $f(n)$, utilizada no algoritmo *backtracking informado*, permite avaliar a possibilidade de poda dos caminhos alternativos de solução existentes a partir do nó n . Esta avaliação considera a configuração de rede obtida pelo caminho construído no processo de busca, do nó inicial (nó raiz da árvore de busca) até o nó n . Pode-se incluir também à função $f(n)$ uma estimativa do caminho compreendido entre o nó n e um nó terminal. Para isso, adota-se uma *função de avaliação heurística*, com o objetivo de antecipar a poda por dominância. A função de avaliação heurística $f_h(n)$ é definida como:

$$f_h(n) = g(n) + h(n) \quad (14)$$

onde $g(n)$ é o acréscimo de perdas da configuração de rede obtida a partir do nó inicial ao nó n e $h(n)$ é a estimativa do acréscimo de perdas para obter uma configuração de rede radial, isto é, a estimativa do acréscimo de perdas do nó n a um nó terminal. A Figura 19 ilustra o conceito utilizado na concepção da função $f_h(n)$.

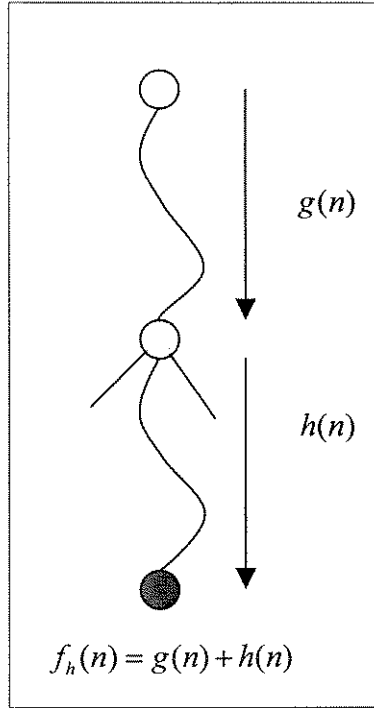


Figura 19. Função heurística

A função $h(n)$ para estimar as perdas é caracterizada pela equação (15).

$$h(n) = c(n) \delta \quad (15)$$

onde o valor de $c(n)$ é igual ao número de ciclos existentes na configuração de rede associada ao nó n e δ é uma estimativa do acréscimo médio das perdas correspondentes à abertura de cada ciclo, no caminho do nó n a um nó terminal.

Na estratégia *backtracking heurístico* a função de avaliação $f_h(n)$ substitui a função $f(n)$ na *Etapa 5* do algoritmo *backtracking informado*, apresentado no item anterior. Note que a função de avaliação $f(n)$, na busca *backtracking informado*, corresponde a $g(n)$ na busca *backtracking heurístico*.

Quando a função $h(n)$ é um bom *estimador* das perdas futuras no caminho em exploração, uma maior antecipação da poda por dominância é obtida. Portanto, consegue-se reduzir o espaço de estados explicitamente gerado, descobrindo-se caminhos de solução menores, do nó inicial a um nó terminal.

Pode-se notar que $f_h(n)$ está baseada nas idéias da função de avaliação que caracteriza a família dos métodos de *busca heurísticas* (Nilson, 1980). Entretanto, $f_h(n)$ é usado no *backtracking heurístico* para melhorar a eficiência da poda por dominância, enquanto que a função de avaliação nos métodos de busca heurística apontam para a ordem dos nós a serem expandidos.

Um aspecto importante a verificarmos no método *backtracking heurístico* é se ele ainda continua um algoritmo admissível; em outras palavras, se a busca ainda garante encontrar a árvore recobridora factível ótima para a rede de distribuição. A admissibilidade depende do valor de $h(n)$. Se $h(n) = 0$, o algoritmo *backtracking heurístico* tem um comportamento igual ao *backtracking informado* e, portanto, o algoritmo é admissível. Por outro lado, se $h(n)$ superestima as perdas, não podemos mais garantir um algoritmo admissível, pois a poda por dominância pode desconsiderar o caminho para a solução ótima do problema. Deve-se então ajustar o valor de $h(n)$ de forma a garantir a admissibilidade do algoritmo. A procura da *melhor* forma para construção da função $h(n)$ conduz naturalmente a um algoritmo da classe A^* , objeto de discussão da próxima seção.

4.8. ALGORITMO A^*

Um algoritmo da classe A^* (Hart et al., 1968; Pearl, 1984), uma categoria de busca em grafo (discutido na seção 4.3.4), utiliza uma função de avaliação heurística para indicar o *melhor sucessor* de um nó da árvore de busca.

Para fins de referência, pode-se definir uma função de avaliação heurística ótima, $f_h^*(n)$,

$$f_h^*(n) = g^*(n) + h^*(n) \quad (16)$$

onde $g^*(n)$ é o menor caminho do nó inicial ao nó n (para o problema em discussão, o caminho de acréscimo mínimo das perdas de energia) e $h^*(n)$ é o

menor custo do nó n até um nó objetivo. Portanto, $f_h^*(n)$ é o valor do custo do caminho ótimo do nó inicial a um nó objetivo, passando pelo nó n .

Se $f_h^*(n)$ é calculado para todos os possíveis sucessores do nó inicial, pode-se selecionar qual deles indica o próximo caminho a ser explorado pelo método de busca, ou seja, o melhor sucessor de n para o nó inicial. O mesmo processo pode ser repetido para este melhor sucessor de n , e assim sucessivamente, construindo-se o caminho de solução ótima até um nó objetivo ser encontrado. Este procedimento caracteriza a idéia do *algoritmo* A^* .

Uma função de avaliação ótima não existe para problemas reais. Em alguns casos, entretanto, é possível conceber uma aproximação de $f_h^*(n)$ que permita construir o caminho de busca do *algoritmo* A^* , não muito longe do caminho ótimo.

Considere, por exemplo, a aproximação $f_h(n)$,

$$f_h(n) = g(n) + h(n) \quad (17)$$

com

$$h(n) \leq h^*(n) \quad (18)$$

Se uma função de avaliação heurística com esta propriedade é adotada no *algoritmo* A^* , ele gera um o caminho de menor custo para a solução ótima, caracterizando um *algoritmo admissível* (Nilson, 1980).

Duas listas são usadas para implementar o *algoritmo* A^* : ABERTO e FECHADO. A lista ABERTO armazena todos os nós gerados e ainda não expandidos, ordenados pelo valor da função de avaliação heurística. A lista FECHADO armazena os nós já expandidos (todos os seus nós sucessores foram gerados). Sendo $f_h(n)$ o valor da função de avaliação heurística para o nó n , as etapas do *algoritmo* A^* são apresentadas a seguir.

Algoritmo A^*

- Etapa 1.* Inclua o nó inicial r na lista ABERTO. Calcule o valor da função $f_h(r)$ associado ao nó r – raiz da árvore de busca.
- Etapa 2.* Crie a lista FECHADO (inicialmente esta lista está vazia).
- Etapa 3.* Se a lista ABERTO está vazia, pare – não há solução para o problema. Caso contrário, continue na *Etapa 4*.
- Etapa 4.* Remova o nó n com o menor valor de $f_h(n)$ da lista ABERTO; inclua-o na lista FECHADO. Se vários nós são qualificados, escolha um deles arbitrariamente.
- Etapa 5.* Expanda o nó n , gerando todos os seus sucessores; ligue os sucessores por apontadores ao nó n .
- Etapa 6.* Se algum dos sucessores é um nó objetivo, pare; recupere a solução obtida.
- Etapa 7.* Para cada um dos nós n' sucessores de n , faça:
- Etapa 7.1.* Estime $h(n')$ e calcule o valor de $f_h(n') = g(n') + h(n')$.
 - Etapa 7.2.* Se n' não foi gerado anteriormente (n' não pertence à lista ABERTO ou a lista FECHADO); inclua o nó n' na lista ABERTO e associe o valor de $f_h(n')$ ao nó n' .
 - Etapa 7.3.* Se n' já foi gerado (n' pertence à lista ABERTO ou à lista FECHADO), direcione seus apontadores para o caminho com o menor valor da função f_h .
 - Etapa 7.4.* Se os apontadores de n' foram alterados e se ele pertencia à lista FECHADO, inclua-o na lista ABERTO, reabrindo o nó n' .
- Etapa 8.* Volte para a *Etapa 3*.

Note que o *algoritmo A^** requer, no processo de expansão, o cálculo de $f_h(n)$ para todos os sucessores de um dado nó, aumentando significativamente o esforço computacional (em relação ao *backtracking informado*) para encontrar o próximo nó a ser explorado.

4.8.1. Construção da função de avaliação heurística para o problema P

A função de avaliação heurística f_h , definida para o algoritmo *backtracking heurístico*, foi utilizada para antecipar a poda por dominância. É possível construir uma função de avaliação heurística satisfazendo as condições (17) e (18) para guiar o algoritmo

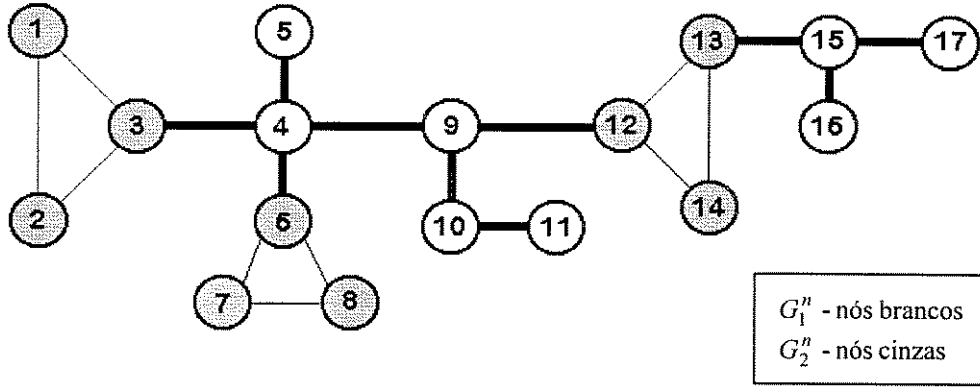


Figura 20. Partição G_1'' e G_2''

A^* à solução do problema **P**, no problema de minimização de perdas em sistemas de distribuição. A função a ser construída pode ser vista como um refinamento de f_h , adotado para o *backtracking informado*.

Considere o grafo $G'' = [N, A'']$, representando a configuração de rede n da árvore de busca. Esta rede pode ser particionada em dois conjuntos de componentes, representados pelos subgrafos $G_1'' = [N_1, A_1'']$ e $G_2'' = [N_2, A_2'']$ ($N_1, N_2 \subset N$ e $A_1'', A_2'' \subset A$, sendo $A_1'' \cap A_2'' = \emptyset$), onde G_1'' possui apenas os arcos da árvore e G_2'' os arcos que compõem os ciclos (ou malhas) do grafo. A Figura 20. exemplifica esta partição.

Na Figura 20 pode-se identificar $G_1'' = [N_1, A_1'']$, os componentes radiais, e $G_2'' = [N_2, A_2'']$, os componentes em malhas, na forma a seguir.

$$N_1 = \{3, 4, 5, 6, 9, 10, 11, 12, 13, 15, 16, 17\}$$

$$A_1'' = \{(3, 4), (4, 5), (4, 6), (4, 9), (9, 10), (9, 12), (10, 11), (13, 15), (15, 16), (15, 17)\}$$

e

$$N_2 = \{1, 2, 3, 6, 7, 8, 12, 13, 14\}$$

$$A_2'' = \{(1, 2), (1, 3), (2, 3), (6, 7), (7, 8), (8, 6), (12, 13), (12, 14), (13, 14)\}$$

Chamando $l(G_1'')$ e $l(G_2'')$ as perdas em G_1'' e G_2'' , respectivamente, temos

$$l_p(G_1'') = \sum_{k \in A_1''} r_k x_k^2 \quad (19)$$

$$l_p(G_2'') = \sum_{k \in A_2''} r_k x_k^2 \quad (20)$$

Atribuindo a $f_h(n)$ o valor da função objetivo da solução ótica do problema P_r , ela passa a totalizar as perdas mínimas dos componentes radiais e malhados para a configuração em estudo, ou seja,

$$f_h(n) = l_p(G_1'') + l_p(G_2'') \quad (21)$$

Comparando a função (17) com a função (21), pode-se associar $g^*(n)$ à $l_p(G_1'')$ (o valor mínimo das perdas dos componentes radiais da configuração de rede associada ao nó n). A função $h(n)$ é associada a $l_p(G_2'')$, uma estimativa das perdas da configuração radial destes componentes (o valor das perdas dos componentes malhados). Portanto, a função (21) define uma *função de avaliação heurística* para o problema de busca da árvore recobridora factível de mínimas perdas, cujo valor é igual ao total das perdas da configuração de rede associada ao nó n (considerando-se as perdas dos componentes radiais e malhados).

Pode-se mostrar que a função heurística $h(n) = l_p(G_2'')$ satisfaz a condição (18), definindo um *algoritmo* A^* admissível para o problema P .

T1. Seja $h(n)$ o somatório das perdas dos arcos pertencentes a G_2'' e $h^(n)$ o valor mínimo das perdas da rede radial associada a G_2'' , então $h(n) \leq h^*(n)$ para $\forall n$*

A distribuição de fluxo da configuração n é a solução ótica do problema P_r . Nesta solução $l(G_2'')$ é uma estimativa otimista para as perdas em uma solução radial. Em outras palavras, a redistribuição de fluxo para uma configuração radial associada a G_2'' implica no aumento das perdas do sistema. Portanto, o valor mínimo das perdas da rede radial associada a G_2'' terá um custo para as perdas, $h^*(n)$, igual, ou maior, ao estimado por $h(n)$. Isto é,

$$h(n) \leq h^*(n) \text{ para } \forall n.$$

4.8.2. Monotocidade da função heurística

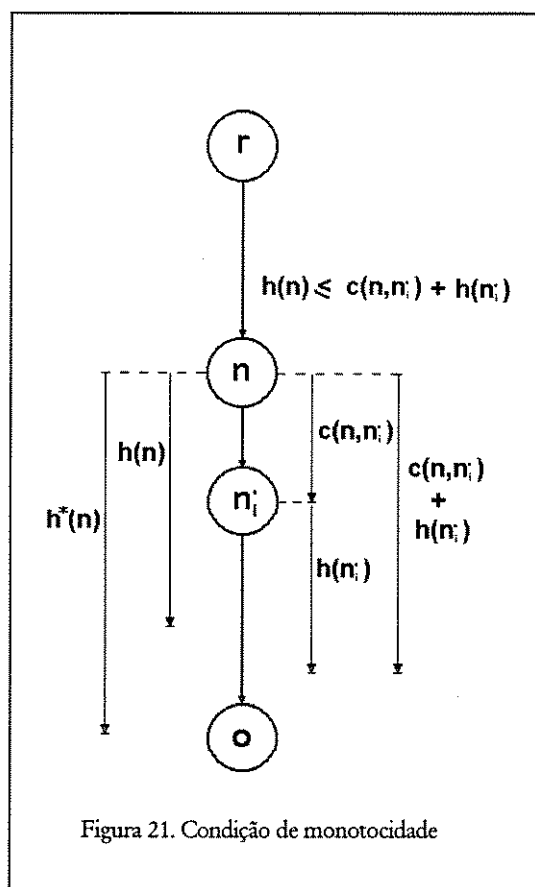
Uma função heurística é monotônica quando para todos os nós n'_i sucessores do nó n a inequação a seguir é verificada (Pearl, 1984).

$$h(n) \leq c(n, n'_i) + h(n'_i) \quad \forall n, n'_i \quad (22)$$

onde n'_i é o i -ésimo nó sucessor de n .

A inequação (22) mostra que o valor estimado para o custo do caminho ótimo de n ao nó objetivo, não é maior que o custo associado ao arco do nó n ao nó n'_i da árvore de busca, adicionado com a estimativa de custo do caminho ótimo do nó n'_i ao nó objetivo. A Figura 21 ilustra esta condição da árvore de busca, isto é, não há degeneração da informação.

Estabelecida a restrição de monotonicidade para a função f_h , pode-se afirmar que quando o algoritmo A^* seleciona o nó n para ser expandido, ele pertence ao caminho ótimo da solução do problema (Nilson 1970). Consequentemente, o algoritmo A^*



pode ser implementado com simplificação da *Etapa 7*. Esta simplificação é detalhada no capítulo 5.

Pode-se mostrar que a função heurística definida na sessão 4.8.1 é monotônica. Considere a transição, na árvore de busca do *algoritmo A**, do nó n para o nó n'_i , conforme ilustra a Figura 22.

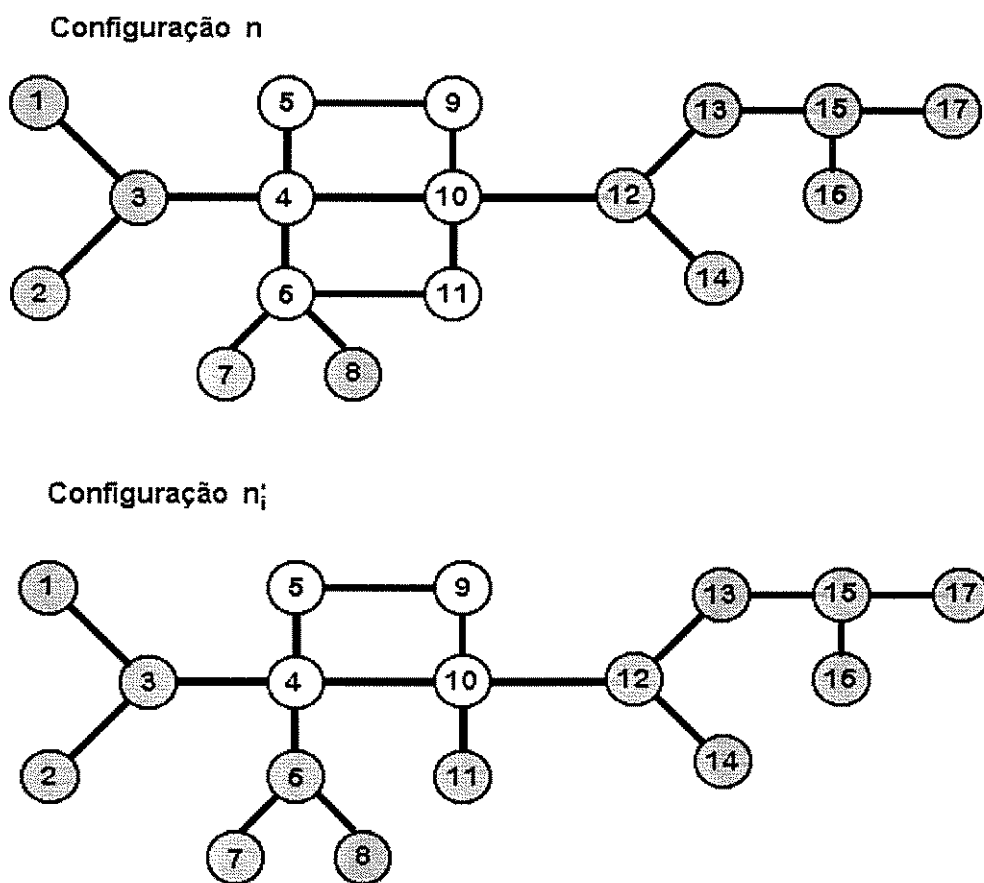


Figura 22. Transição da configuração n para a configuração n'_i

Seja $c(n, n'_i)$ o custo de transformação da configuração de rede associada ao nó n na configuração associada ao nó n'_i , isto é, o valor do acréscimo de perdas pela remoção do arco k na configuração de rede n . Em outras palavras, $c(n, n'_i)$ representa o acréscimo de custo para a radialização da rede.

Considerando a parcela radial e com malhas da rede, tem-se,

$$c(n, n_i') = r(n, n_i') + m(n, n_i') \quad (23)$$

onde $r(n, n_i')$ e $m(n, n_i')$ são os acréscimos de custo das parcelas radiais e com malhas, respectivamente. Logo,

$$r(n, n_i') = g(n_i') - g(n) \text{ e} \quad (24)$$

$$m(n, n_i') = h(n) - h(n_i') \quad (25)$$

Substituindo (24) e (25) em (23), tem-se

$$c(n, n_i') = [g(n_i') - g(n)] + [h(n) - h(n_i')] \quad (26)$$

Sendo a parcela da equação $[g(n_i') - g(n)]$ positiva ou nula para qualquer n, n_i' , pode-se escrever,

$$c(n, n_i') \geq h(n) - h(n_i'), \quad (27)$$

provando a restrição de monotonicidade da função f_h .

Como a função heurística f_h é monotônica, pode-se construir uma versão do *algoritmo* A^* onde elimina-se as etapas 7.2, 7.3 e 7.4. Este aspecto será detalhado no próximo capítulo.

4.9. PARALELIZAÇÃO DOS ALGORITMOS

Observe que a árvore de busca, gerada pelos algoritmos apresentados, pode ser particionadas em sub-árvores independentes, sugerindo a utilização de processamento paralelo. Utilizando esta observação, foi desenvolvido um trabalho de paralelização do algoritmo *backtracking informado* e *backtracking heurístico* (Tão, 1996, 1997, 1998). As principais características do trabalho e uma alternativa para paralelização do *algoritmo* A^* são descritas no Capítulo 6.

A seguir são apresentados, para cada um dos métodos descritos neste capítulo, detalhes de implementação dos algoritmos e seus respectivos resultados computacionais.

CAPÍTULO 5

ASPECTOS DE IMPLEMENTAÇÃO E RESULTADOS COMPUTACIONAIS

5.1. INTRODUÇÃO

Neste capítulo são detalhadas as implementações dos algoritmos, discutidas suas utilizações e apresentados os resultados obtidos. Dois estudos de caso com minimização de perdas associadas ao fluxo de potência ativa foram considerados. O primeiro usa uma pequena rede de distribuição, com 47 nós e 51 arcos (REDE47).

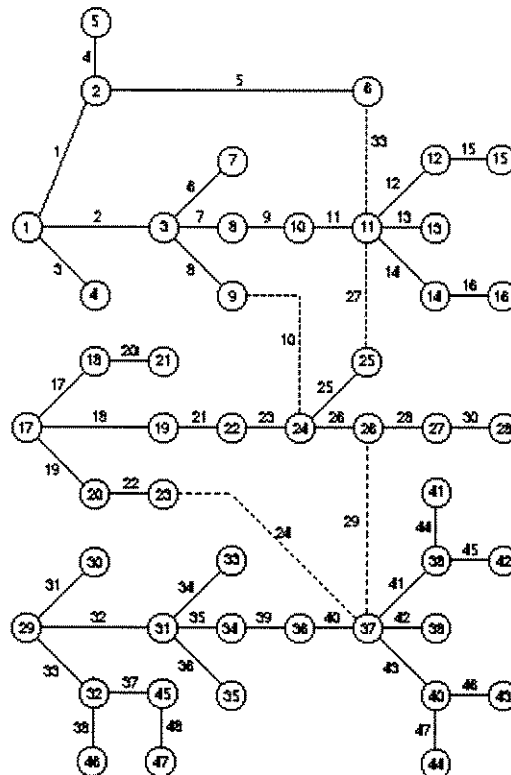


Figura 23. Diagrama esquemático da REDE47

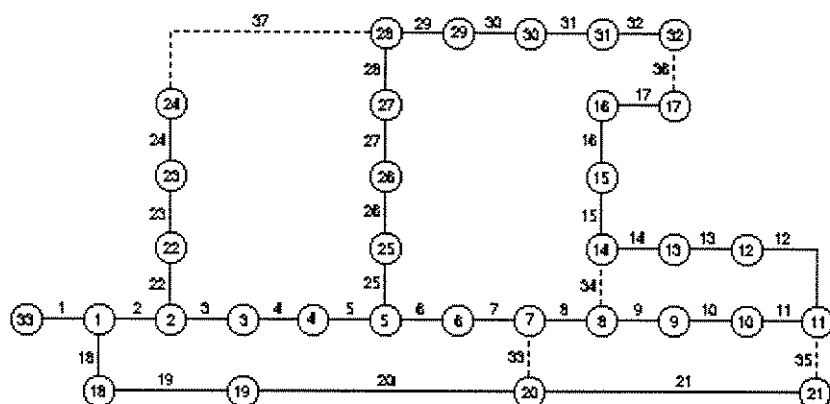


Figura 24. Diagrama esquemático da rede BWU01

No segundo estudo de caso foi considerada uma rede de distribuição usada como sistema teste por Baran e Wu (1989), com 33 nós e 38 arcos (BWU01). Os diagramas esquemáticos destas redes são apresentados nas Figuras 23 e 24, respectivamente.

A Tabela 3 resume os dados das redes REDE47 e BWU01, apresentando os valores totais das perdas associadas à configuração inicial da rede radial de referência (representada nas Figuras 36 e 37, para cada uma das redes, respectivamente). A Tabela 3 também apresenta o valor das perdas quando não se considera a restrição de radialidade na operação do sistema de distribuição, isto é, o valor das perdas do problema relaxado P_r – o valor das perdas dessa solução é utilizado para gerar informações no início do processo de radialização.

Tabela 3. Características das Redes do estudo de casos

	Nós	Arcos	Valor total das perdas na configuração radial Inicial (W)	Valor total das perdas não considerando a restrição de radialidade (W)
REDE47	47	51	2.525	1.823
BWU01	33	38	131.530	78.841

A seguir são apresentados detalhes das implementações dos algoritmos discutidos no Capítulo 4. Para maior clareza, eles são descritos com uma pseudolinguagem de programação. Os algoritmos foram codificados na linguagem C++ (compilador

Turbo C++ 3.1) e os tempos de execução (minutos:segundos) foram obtidos em um micro computador Petium 166 MHz.

5.2. ALGORITMO PARA BUSCA IRREVOGÁVEL

O procedimento *Busca Irrevogável* foi implementado como o algoritmo recursivo apresentado na Figura 25.

Busca Irrevogável

procedimento principal()

- *leitura dos dados da rede*
- *resolva o problema P_r para a configuração de rede associada ao nó inicial n_0*
- *resolva(n_0)*
- *mostre a solução do problema P*

resolva(n)

- *se existe ciclo*
 - *retirar arco k de menor fluxo*
 - *armazene k na lista CAMINHO*
 - *obtenha a solução de P_r para a configuração de rede associada ao nó n'*
 - *resolva(n)*
- *recupere a solução na lista CAMINHO*

Figura 25. Algoritmo de Busca Irrevogável

A condição inicial do processo é associada à solução de P_r - para a configuração inicial da rede. A condição de parada do procedimento recursivo é a não existência de ciclos na rede - configuração radial da rede de distribuição. O valor do fluxo é utilizado para escolher o próximo nó a ser inspecionado. O arco k , que possui o menor valor do fluxo, é escolhido no conjunto compreendido por todos os arcos pertencentes aos ciclos existentes na configuração de rede. Em seguida, o arco k é

Tabela 4. Resultados da busca irrevogável pelo valor do fluxo

	Chaves abertas na solução final	Perdas totais (W)	Tempo de execução (mm:ss)
REDE47	29, 27, 10 e 40	2.117	00:00:05
BWU01	32, 9, 14, 7 e 37	84.076	00:00:05

removido da configuração da rede, armazenado na lista CAMINHO, e uma nova solução é obtida para o problema relaxado P_r . Observe que a árvore de busca é construída implicitamente através do mecanismo de recursividade utilizado no algoritmo. Os resultados obtidos são mostrados na Tabela 4.

Os resultados obtidos são também a solução ótima para o problema P , como pode-se verificar comparando estes resultados com os obtidos pelo algoritmo *backtracking informado* (vide seção 5.3). No entanto, deve-se enfatizar que nem sempre a busca irrevogável encontra a solução ótima.

O valor do menor gradiente pode também ser usado como uma alternativa de escolha do próximo nó a ser inspecionado. Resultados utilizando este critério são apresentados na Tabela 5.

Embora não seja possível garantir a solução ótima do problema P , os resultados apresentaram reduções de perdas significativas: 16,2% e 23,5% para a REDE47 e BWU01, respectivamente. A solução obtida para a REDE47 é também a solução ótima, como poderá ser observado posteriormente. Estes resultados confirmam que a aplicação do algoritmo apresenta bons resultados aproximados para o problema.

Um aspecto positivo deste algoritmo é a simplicidade de implementação e o baixo tempo de processamento para obtenção da solução. Estas características podem ser

Tabela 5. Resultados da busca irrevogável pelo valor do gradiente

	Chaves abertas na solução final	Perdas totais (W)	Tempo de execução (mm:ss)
REDE47	29, 27, 10 e 40	2.117	00:05,94
BWU01	32, 10, 14, 6 e 25	100.655	00:05,22

aproveitadas para obter as soluções iniciais utilizadas nos algoritmos de busca por tentativa (*backtracking informado*, *backtracking heurístico* e *algoritmo A^**).

5.3. ALGORITMO BACKTRACKING INFORMADO

A implementação do algoritmo para o *backtracking informado* é apresentada na Figura 26. Inicialmente, atribui-se à variável S o valor das perdas para uma configuração de rede radial conhecida (por exemplo, uma configuração utilizada na operação de rede). Obtém-se a solução de P_r associado ao nó inicial da árvore de busca (n_0) e faz-se a chamada do procedimento *resolva()*.

No procedimento *resolva()* verifica-se a existência ou não de ciclos na configuração de rede n . Caso existam ciclos, os arcos pertencentes aos ciclos são empilhados na lista ABERTO em ordem decrescente do valor do gradiente, definindo a estratégia de controle de busca em profundidade e o critério de geração do próximo nó a ser explorado na árvore de busca - são retirados da configuração primeiramente os arcos com maior valor de gradiente. No caso de não haver ciclos, um nó terminal foi encontrado. Se o valor das perdas na configuração associada ao nó n é menor do que o das perdas da configuração de referência armazenada em S , o caminho da nova solução é armazenado e o valor da solução é atribuído a S . Em seguida, o procedimento retorna a constante RESOLVIDO, indicando que a busca está concluída para este caminho da árvore de busca.

O *laço eterno* garante que todos os nós promissores do espaço de estados são explorados. A cada iteração, um arco k é retirado do topo da lista ABERTO e incluído na lista CAMINHO, caracterizando o início da exploração de todas as soluções que não incluem o arco k . Cria-se a nova configuração n' , removendo-se o arco k da configuração n . Calcula-se o valor de $f(n')$, resolvendo-se um problema de fluxo de custo mínimo não linear (*pfcnnl*) para a configuração n' .

Backtracking Informado

procedimento principal()

- *leitura dos dados da rede*
- *atribua a S o valor das perdas da configuração em operação*
- *resolva o problema P , para a configuração de rede associada ao nó inicial n_0*
- *resolva (n_0)*
- *mostre a solução do problema P*

resolva (n)

- *se existe ciclo*
 - *inclua todos os arcos pertencentes aos ciclos da configuração n na pilha ABERTO, em ordem crescente do valor do gradiente.*
 - *caso contrário, não existe ciclo*
 - *faça $S \leftarrow f(n)$*
 - *retorne RESOLVIDO*
 - *laço eterno*
 - *remova arco k do topo da pilha ABERTO*
 - *inclua o arco k na pilha CAMINHO*
 - *remova k da configuração n formando configuração n'*
 - *incremente o nível da árvore de busca*
 - *calcule $f(n')$*
 - *se $f(n') < S$*
 - *resolva (n')*
 - *remova arco k da pilha CAMINHO (poda por dominância)*
 - *decremente o nível da árvore de busca*
 - *se não existe mais nós a serem explorados neste nível*
 - *volte ao nível anterior da árvore de busca*
 - *retorne RESOLVIDO*
-

Figura 26. Algoritmo Backtracking Informado

Se o valor de $f(n')$ é menor que o valor armazenado em S a busca continua neste caminho, chamando, recursivamente, o procedimento *resolva()* para o nível seguinte da árvore de busca. Caso contrário, isto é, se a solução encontrada indica que este caminho da árvore de busca não é promissor, executa-se a poda por dominância, retirando-se o arco k da lista CAMINHO, e faz-se o *backtracking*, decrementando de uma unidade o nível da árvore de busca.

O procedimento *resolva()* é repetido até que não haja mais nós a serem explorados, retornando a instrução seguinte a *resolva(n_o)*, para apresentar a solução ótima armazenada na lista CAMINHO e seu valor em S .

A Tabela 6 apresenta os resultados obtidos pelo algoritmo *Backtracking Informado* para a REDE47 e BWU01.

Tabela 6. Resultados para a busca Bactraking Informado				
	Chaves abertas na solução final (ótima)	Perdas Totais (W)	Tempo (mm:ss)	Nós Explorados
REDE47	10, 27, 29 e 40	2.110	00:01,65	236
BWU01	7, 9,14, 32 e 37	84.075	00:18,84	2480

5.4. ALGORITMO BACKTRACKING HEURÍSTICO

O algoritmo *backtracking heurístico* foi implementado usando-se como base o algoritmo *backtracking informado* descrito anteriormente. Nessa estrutura de algoritmo a inclusão da função heurística $f_h(n)$, descrita no Capítulo 4, é trivial; basta alterar o cálculo da função $f(n)$ e a condição da poda por dominância. Isto é, substitui-se as instruções.

calcule $f(n')$
se $f(n') < S$

pelas instruções

calcule $f_h(n') = g(n') + h(n')$
se $f_h(n') < S$

Backtracking Heurístico

procedimento principal()

- leitura dos dados da rede
- atribua a S o valor das perdas da configuração em operação
- resolva o problema P_r para a configuração de rede associada ao nó inicial n_0
- resolva (n_0)
- mostre a solução do problema P

resolva (n)

- se existe ciclo
 - inclua todos os arcos pertencentes aos ciclos da configuração n na pilha ABERTO, em ordem crescente do valor do gradiente.
- caso contrário, não existe ciclo
 - faça $S \leftarrow f(n)$
 - retorne RESOLVIDO
- laço eterno
 - remova arco k do topo da pilha ABERTO
 - inclua o arco k na pilha CAMINHO
 - remova k da configuração n formando configuração n'
 - incremente o nível da árvore de busca
 - calcule $f_h(n') = g(n') + h(n')$
 - se $f_h(n') < S$
 - resolva (n')
 - remova arco k da pilha CAMINHO (poda por dominância)
 - decrémente o nível da árvore de busca
 - se não existe mais nós a serem explorados neste nível
 - volte ao nível anterior da árvore de busca
 - retorne RESOLVIDO

Figura 27. Algoritmo Busca Heurística

Tabela 7. Resultados para a busca Backtracking Heurístico

	Chaves abertas na solução final	Perdas totais (W)	Tempo (mm:ss)	Nós Explorados
REDE47	10, 27, 29 e 40	2.117	00:00,88	102
BWU01	7, 9, 14, 32 e 37	84.075	00:14,61	1774

onde $g(n')$ é igual ao valor da solução de P_r para a configuração de rede associada ao nó n' e $h(n') = c(n')\delta$, como descrito no item 4.7. A Figura 27 apresenta o algoritmo *backtracking heurístico* na sua forma completa. As instruções em negrito destacam as alterações feitas.

O comportamento deste algoritmo é semelhante ao algoritmo *backtracking informado*. Porém, diante de sua capacidade de “previsão”, o número de nós explorados é menor, como mostram os resultados apresentados na Tabela 7.

Estes resultados foram obtidos adotando-se valores de δ iguais a 0,09 e 0,9, para as redes REDE47 e BWU01, respectivamente - os valores foram ajustados para a obtenção da solução ótima.

5.5. ALGORITMO A^*

O algoritmo A^* foi implementado através de dois procedimentos, *resolva* e *expanda*. O procedimento *resolva* realiza o controle da estratégia de busca, verificando se encontrou a solução do problema, se o problema não tem solução ou se há a necessidade de expandir o nó da árvore de busca. O procedimento *expanda* executa o processo de expansão do nó n da árvore de busca e a classificação da pilha ABERTO, em ordem decrescente da função $f(n)$. A Figura 28 apresenta a implementação do algoritmo A^* .

procedimento principal()

- *leitura dos dados da rede*
- *otimiza a configuração inicial, n_0*
- *identifica os ciclos da configuração n_0*
- *se existir ciclo*
 - *cria no inicial da árvore de busca*
 - *resolva()*
 - *se achou solução*
 - *recupera o caminho da solução*
 - *mostra solução*
 - *caso contrário, não existe solução*
 - *mostra “problema sem solução”*
 - *caso contrário, não existe ciclo*
 - *mostre a configuração de rede encontrada*

procedimento resolva()

- *laço eterno*
 - *se pilha ABERTO está vazia*
 - *termine, problema sem solução*
 - *retire o nó n do topo da lista ABERTO*
 - *inclua o nó n na lista FECHADO*
 - *se não existe ciclo na configuração n (rede radial)*
 - *termine, solução do problema encontrada*
 - *expanda (n)*

procedimento expanda(n)

- *recupere lista dos arcos que pertencem aos ciclos da configuração associada ao nó n*
- *para cada um dos i arcos faça*
 - *obtenha a configuração associada ao nó n_i*
 - *se o nó n_i não pertence a lista ABERTO ou FECHADO*
 - *calcule $f_h(n) = g(n) + h(n)$*
 - *identifique e armazene os ciclos associados ao nó n_i*
 - *inclua o nó n_i na árvore de busca*
- *classifique a lista ABERTO em ordem decrescente de $f_h(n)$ (o nó com o menor valor para f_h no topo da lista)*

Inicialmente, os dados da configuração associada ao nó inicial (n_0) são lidos, obtém-se a solução do problema P_r para essa configuração e verifica-se a existência, ou não, de ciclos. A não existência de ciclos na solução de P_r implica numa configuração radial e, portanto, numa solução ótima factível do problema P – não havendo, assim, necessidade do procedimento de busca. No caso de existirem ciclos, cria-se o nó inicial da árvore de busca com as informações da solução de P_r e o processo de busca é iniciado, chamando-se o procedimento *resolva*. No retorno deste procedimento verifica-se a existência ou não da solução ótima factível. No caso de existir, o caminho da solução é recuperado e apresentado. Caso contrário, uma indicação da inexistência da solução é apresentada.

No procedimento *resolva*, o *laço eterno* garante a exploração da árvore de busca, expandindo os nós mais promissores até encontrar um nó objetivo (configuração de rede sem ciclos) ou verificar a falta de nós para expandir e, portanto, identificar a não existência de solução para o problema. O nó mais promissor é retirado do topo da lista ABERTO, condição assegurada por ordenação desta lista pelo valor decrescente de $f(n)$. A lista FECHADO, além de incluir os nós já expandidos, mantém o caminho em exploração, tornando possível a recuperação da solução encontrada.

A expansão dos nós é realizada pelo procedimento *expanda()*. Considerando a configuração de rede associada ao nó n como a configuração base, obtém-se as configurações filhas, removendo-se um a um os arcos pertencentes aos ciclos. Se a configuração filha associada ao nó n_i ainda não foi gerada anteriormente pelo processo de busca³, calcula-se o valor da função heurística de avaliação, $f_h(n_i)$, os arcos pertencentes aos ciclos são identificados e armazenados para serem utilizados na geração dos nós sucessores de n_i , e cria-se um novo nó na árvore de busca incluindo o nó n_i e o respectivo valor de $f_h(n_i)$ na lista ABERTO.

³ Configurações já exploradas são identificadas por um processo de busca nas listas ABERTO e FECHADO, onde procura-se caminhos de mesmo comprimento do caminho em exploração e com os mesmos conjuntos de arcos.

Tabela 8. Resultados para a busca A^*

	Chaves abertas na solução ótima	Perdas totais (W)	Tempo (mm:ss)	Nós Explorados
REDE47	40, 10, 27 e 29	2.110	00:14,34	481
BWU01	37, 7, 14, 9 e 32	84.075	11:39,48	6973

Embora a função f_h seja monotônica (demonstrado no Capítulo 5), manteve-se no algoritmo a verificação da existência de configurações equivalentes nas listas ABERTO e FECHADO, para evitar cálculos desnecessários do valor da função heurística f_h – os valores da função f_h para configurações equivalentes são iguais e o custo computacional para obtê-los é maior do que verificar a existência ou não de configurações equivalentes..

Finalmente, a lista ABERTO é ordenada pelo valor decrescente de $f_h(n)$, colocando o nó mais promissor no topo desta lista. Os resultados obtidos pelo algoritmo A^* são apresentados na Tabela 8.

O número de nós explorados indica a quantidade de nós para os quais o algoritmo calculou o valor da função $f_h(n)$, isto é, indica o número de vezes em que o problema P_r foi resolvido. Assim, mantém-se o mesmo critério de contagem de nós explorados adotados nos algoritmos *backtracking informado* e *heurístico*, tornando possível compará-los.

5.6. COMENTÁRIOS

Os resultados apresentados, para os algoritmos *backtracking informado* e *backtracking heurístico*, foram obtidos ordenando a lista ABERTO pelo valor decrescente do gradiente. Alguns experimentos foram realizados para verificar a influência de diferentes critérios de ordenação da lista ABERTO. Considerou-se os valores de gradiente e fluxo nos arcos em ordem crescente e decrescente, totalizando 4 (quatro) critérios diferentes, que foram testados no algoritmo *backtracking informado*. Os resultados são apresentados na Tabela 9.

Tabela 9. Resultados do algoritmo *backtracking informado* usando critérios de ordenação diferentes

Estudo de Caso	Critério de ordenação		Chaves abertas na solução final	Perdas totais (W)	Tempo de execução (mm:ss)	Nós Explorados
REDE47	Gradiente	Decrescente	40, 10, 27 e 29	2.110	00:01,70	236
		Crescente	29, 27, 10 e 40	2.110	00:02,42	400
	Fluxo	Decrescente	40, 10, 27 e 29	2.110	00:01,70	236
		Crescente	29, 27, 10 e 40	2.110	00:02,36	400
BWU01	Gradiente	Decrescente	37, 7, 14, 9, 32	84.076	00:18,84	2480
		Crescente	32, 9, 14, 7, 37	84.076	00:31,42	4625
	Fluxo	Decrescente	37, 7, 14, 9, 32	84.076	00:53,77	4003
		Crescente	32, 9, 14, 7, 37	84.076	00:39,88	6161

Observa-se na Tabela 9 que a utilização do gradiente ou fluxo para a ordenação dos arcos, não altera o número de nós explorados pelo procedimento de busca para a REDE47. O mesmo ocorrendo com os tempos de execução, onde os resultados são aproximadamente iguais. Os resultados obtidos para rede BWU01 mostram um menor número de nós explorados e, conseqüentemente, um menor tempo de execução, quando utilizado o valor de gradiente em ordem decrescente para ordenação da lista ABERTO. Estes resultados também mostram que a influência do critério de ordenação adotado depende da rede considerada, sugerindo a necessidade de adequá-lo ao problema.

Uma das causas do modesto desempenho do *algoritmo A** é exemplificado na Tabela 10. Nesta tabela são listados os primeiros valores de $f(n)$ na lista ABERTO, armazenados em ordem crescente durante a execução do processo de busca. Pode-se observar que os valores da função $f(n)$ são muito próximos, implicando um gasto de tempo significativo para a seleção do melhor caminho, num conjunto de candidatos que apresentam qualificações aproximadamente iguais. Em tais casos, a admissibilidade torna-se uma propriedade dispendiosa e de pouco valor prático. Ela impõe ao *algoritmo A** um gasto de tempo considerável, não permitindo o algoritmo completar a busca em um subótimo aceitável.

Tabela 10. Valores no topo da lista ABERTO

83.925659	83.981285
83.927483	83.983101
83.931740	83.983643
83.937691	83.984474
83.942780	84.002571
83.945740	84.004288
83.970657	$\vdots$

Essa observação conduz à reconsideração da admissibilidade do *algoritmo* A^* , possibilitando outras perspectivas de aplicação do algoritmo, com o relaxamento do requisito da otimalidade. Algumas propostas existem nesse sentido, sugerindo a ponderação das funções $g(n)$ e $h(n)$, por exemplo (Pearl, 1984),

$$f(n) = (1 - w)g(n) + wh(n) \quad (28)$$

Com a variação de w entre os valores de 0 e 1 obtém-se o grau de confiabilidade desejado para a solução do problema. O valor de w pode ser alterado durante o processo de busca, ponderando o valor da função $h(n)$ à medida que o processo aprofunda-se na árvore de busca, isto é,

$$f(n) = g(n) + h(n) + \varepsilon \left[1 - \frac{d(n)}{N} \right] h(n) \quad (29)$$

onde $d(n)$ é o valor da profundidade do nó n e N a profundidade (antecipada) do nó objetivo. Não é difícil demonstrar que se $h(n) - h^* \leq \varepsilon$, para qualquer nó n , então o algoritmo A^* é ε -admissível, ou seja, $C(t) - C^* \leq \varepsilon$ (Pearl, 1984), onde $C(t)$ é o valor do custo da solução final e C^* o valor do custo ótimo da solução do problema.

Outra abordagem é proposta por Pearl e Kim (1982), por meio de uma variação do algoritmo A^* , denominado A_ε^* , que considera o esforço de busca da solução. Este algoritmo usa duas listas, ABERTO e FOCAL. A lista FOCAL é uma sublista da

Tabela 11. Resultados para a busca A^*

	Chaves abertas na solução ótima	Perdas totais (kW)	Tempo (mm:ss)	Nós Explorados
REDE47	40, 10, 27 e 29	2,110	00:01,54	73
BWU01	37, 7, 14, 9 e 32	84,075	01:42,11	1122

lista ABERTO, contendo apenas aqueles nós cujo o valor de $f(n)$ não seja superior a um fator $(1 + \varepsilon)$ do menor valor de f existente na lista ABERTO, isto é,

$$\text{FOCAL} = \left\{ n \text{ tal que } f(n) \leq (1 + \varepsilon) \min_{n' \in \text{ABERTO}} f(n') \right\} \quad (30)$$

O comportamento do algoritmo A_e^* é idêntico ao A^* , exceto que o próximo nó a ser explorado é selecionado da lista FOCAL, com o menor valor da função heurística $h_F(n)$, onde $h_F(n)$ estima o esforço computacional requerido para completar a busca a partir do nó n . O objetivo desse procedimento é selecionar o nó que apresente a melhor condição para completar a busca o mais rapidamente possível (note que para $\varepsilon = 0$, A_e^* reduz-se ao *algoritmo* A^*).

Com base nesta abordagem, implementou-se uma versão do *algoritmo* A_e^* para o problema de reconfiguração das mínimas perdas, sendo $h_F(n)$ igual à profundidade do nó n na árvore de busca. Os resultados obtidos são apresentados na Tabela 11.

Note que o tempo de execução e o número de nós explorados apresentam reduções consideráveis, sem a perda da otimalidade.

No capítulo a seguir discute-se alguns aspectos sobre a paralelização dos algoritmos de busca inteligentes apresentados. Tendo-se como base o trabalho desenvolvido por Tão (1997). Apresenta-se também uma proposta de paralelização do *algoritmo* A^* .

CAPÍTULO 6

MÉTODOS DE BUSCA INFORMADA E PROCESSAMENTO PARALELO

6.1. INTRODUÇÃO

Os métodos de busca informada exploram um número reduzido de nós em relação ao número total de nós do espaço de estados. No entanto, o tempo computacional para obter a configuração de rede de mínimas perdas é ainda elevado. O processamento paralelo é uma alternativa para reduzir o tempo total de execução dos algoritmos de busca, permitindo tratar redes de distribuição de maior porte.

A paralelização dos algoritmos *backtracking informado* e *backtracking heurístico* foram tratados no trabalho desenvolvido por Tão (1997). O resumo dos resultados deste trabalho e uma proposta de paralelização do algoritmo A^* são apresentados a seguir.

6.2. BACKTRACKING PARALELO

O paralelismo dos processos de busca informada *backtracking informado* e *backtracking heurístico* foram motivados pela observação da árvore de busca ilustrada na Figura 25. Note que a partir do nó inicial são geradas n configurações de rede, onde n é igual ao número de arcos pertencentes aos ciclos existentes na solução otimista inicial. Para cada uma das n configurações de rede caracteriza-se um problema P_i a ser resolvido pelo procedimento de busca. Esta observação permite particionar a árvore de busca em N subárvores a serem exploradas pela busca como tarefas

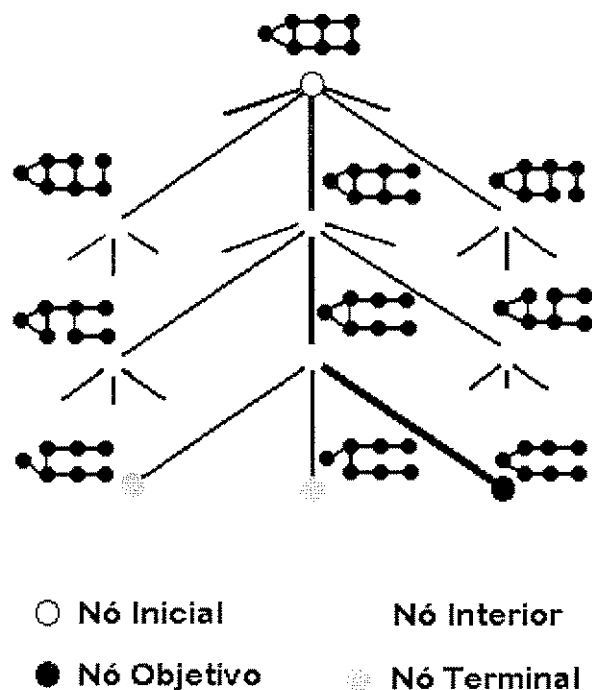


Figura 29. Árvore de busca parcial do problema de reconfiguração da rede para mínimas perdas

independentes, sugerindo uma implementação destes métodos de busca com processamento paralelo. Na implementação paralela adotou-se uma estrutura *pai-filho* (Bertsekas e Tsitsiklis, 1989), como ilustra a Figura 26.

O processo pai resolve o problema P_r para o nó inicial e ativa todos os processos filhos associados aos subproblemas definidos no particionamento da árvore de busca, coordena a comunicação entre os processos para troca de informações e estabelece a estratégia de controle para obtenção da solução do problema P . Cada um dos processos filhos executa um procedimento de busca, procurando a melhor

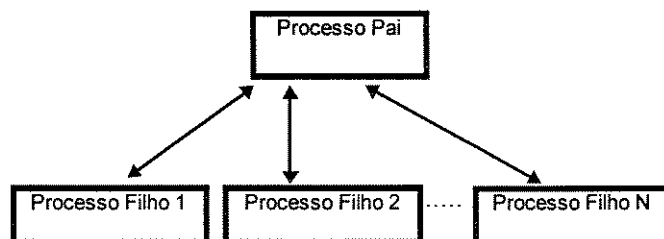


Figura 30. Implementação do processamento paralelo na estrutura *pai-filho*

solução factível para seu espaço de estados (subárvore) e retornando resultados parciais ou finais ao processo pai, conforme o critério de comunicação estabelecido.

No trabalho de Tão (1997) foram investigados três critérios de comunicação entre os processos. Na primeira abordagem o processo pai recebe os resultados dos processos filhos quando estes encontram a melhor solução factível para a sua subárvore, isto é, a comunicação entre os processos pai e filho existe apenas no início e no fim das buscas desenvolvidas nos processos filhos.

No segundo critério, a comunicação é estabelecida cada vez que um processo filho encontra uma solução factível. O valor desta solução é comunicado imediatamente ao processo pai. O processo pai aguarda os demais processos filhos terminarem esta fase, onde cada um deles informa sua respectiva solução factível. Em seguida comunica aos processos filhos a melhor solução factível encontrada nesta fase, para que eles iniciem uma nova fase, usando o valor desta solução como referência para a poda.

Na terceira estratégia, a comunicação para informar ao processo pai a melhor solução factível encontrada até o momento, ocorre após os processos filhos terem visitado um determinado número de nós na árvore de busca. De forma similar à adotada no segundo critério, o processo pai identifica a melhor solução entre todas as soluções factíveis recebidas e comunica-a aos processos filhos. O processo filho, ao receber a nova solução factível, dá continuidade ao processo de busca, iniciando uma outra fase do processamento.

6.3. IMPLEMENTAÇÃO DO ALGORITMO DO BACKTRACKING PARALELO

As principais etapas dos processos pai e filhos são resumidas a seguir.

Algoritmo do Processo Pai

Etapa1. Obter a solução otimista e armazenar em uma pilha os arcos que formam ciclos nesta solução.

Etapa2. Gerar todos os processos filhos.

- Etapa3.* Enviar a cada processo filho um arco da pilha, o valor inicial da solução factível e a lista dos arcos fornecidos aos demais processos.
- Etapa4.* Aguardar cada um dos processos filhos enviar o valor da melhor solução factível encontrada, e a informação sobre a conclusão (ou não) da exploração do respectivo subespaço de estados.
- Etapa5.* Se todos os processos filhos terminaram a exploração dos respectivos subespaços continue na Etapa 7; caso contrário, continue na Etapa 6.
- Etapa6.* Para os processos filhos que não finalizaram a exploração do subespaço, informar o valor da melhor solução factível encontrada e continuar na Etapa 4.
- Etapa7.* Imprimir a solução do problema e finalizar a execução.

Algoritmo do Processo Filho

- Etapa1.* Receber do processo pai o arco a ser retirado da configuração de rede, o valor inicial da solução factível e a lista dos arcos enviados aos demais processos filhos.
- Etapa2.* Executar o procedimento de busca até finalizar uma fase, ou terminar a exploração do respectivo subespaço de estado.
- Etapa3.* Enviar ao processo pai o valor da melhor solução factível encontrada e informar se o procedimento de busca terminou ou não. Caso o procedimento de busca não tenha terminado, continuar na Etapa 4.
- Etapa4.* Aguardar do processo pai o valor da melhor solução factível encontrada nesta fase e continuar na Etapa 2.

Estes algoritmos foram implementados na linguagem C, utilizando o pacote de *software* PVM (Parallel Virtual Machine). PVM é uma biblioteca de funções que permite a uma rede heterogênea de computadores seriais, paralelos e vetoriais, ser utilizada como um único computador paralelo de grande porte (Geist et. al., 1994). Códigos escritos em FORTRAN ou C podem ser paralelizados com a inclusão das funções da biblioteca do PVM. Algumas destas funções são apresentadas na Tabela 12.

Note que a implementação apresentada considera apenas a paralelização do primeiro nível da árvore de busca. Esta abordagem simplifica a descrição sem perda

Tabela 12. Exemplos de funções do PVM

pvm_myid	Retorna o número de identificação do processo
pvm_parent	Retorna o número de identificação do processo pai
pvm_spawn	Gera processos filhos
pvm_initsend	Prepara buffer para envio de dados
pvm_send	Envia dados
pvm_recv	Recebe dados
pvm_pk	Empacota dados para comunicação
pvm_unpk	Desempacota dados

de generalidade – a paralelização pode ser implementada em todos os níveis da árvore de busca, se o número de processadores disponíveis permitir.

6.4. ESTUDOS DE CASOS

No trabalho de Tão (1997) os algoritmos *backtracking informado* e *backtracking heurístico* foram testados em três redes de distribuição: REDE47, BWU01 e BAURU9. As duas primeiras foram apresentadas no Capítulo 6 e utilizadas nos estudos de casos dos algoritmos seriais. A rede BAURU9, com 56 nós e 64 arcos, é parte da rede de distribuição da cidade de Bauru, no interior do estado de São Paulo. A Tabela 13 resume os dados destas redes e os valores das perdas associados a configurações factíveis iniciais.

Os resultados apresentados foram obtidos em um *cluster* heterogêneo de estações de trabalho IBM RISC: quatro IBM RS/6000-25T (de 66.7 MHz com 32 Mbytes de RAM) e dois IBM RS/6000-36T (de 50 MHz com 64 Mbytes de RAM). Os valores dos tempos de execução são as médias dos tempos obtidos em várias execuções

Tabela 13. Características das redes do estudos de casos para o algoritmo paralelo

	Nós	Arcos	Valor das perdas na configuração inicial (W)
REDE47	47	51	2.525
BWU01	33	38	131.269
BAURU9	56	64	101.922

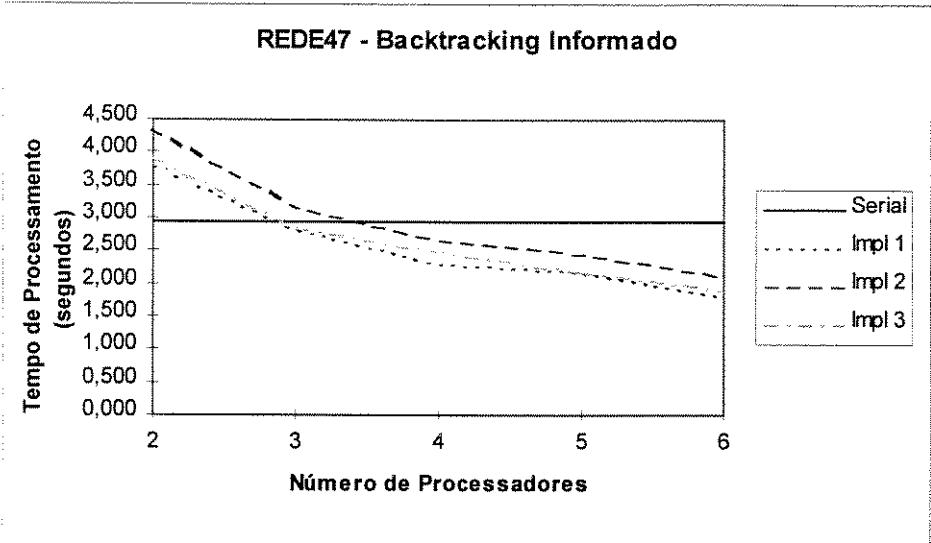
para o mesmo estudo de caso.

As Tabelas 14(a) e (b) mostram os resultados obtidos com implementações seriais (um processador) do *backtracking informado* e *backtracking heurístico*. Estes resultados apresentam pequenas diferenças daqueles apresentados no Capítulo 6, devido a diferenças de compiladores e máquinas.

A seguir apresenta-se um resumo de alguns dos resultados apresentados no trabalho de Tão (1997). Os resultados para o *backtracking informado* e *backtracking heurístico* são mostrados nas Figuras 27 a 29 e Figuras 30 a 32, respectivamente. Nestas figuras são apresentadas as implementações das três estratégias de comunicação: sem interrupção (Implementação 1), interrupção a cada solução ótima obtida (Implementação 2) e interrupção a cada n nós explorados da árvore de busca (Implementação 3).

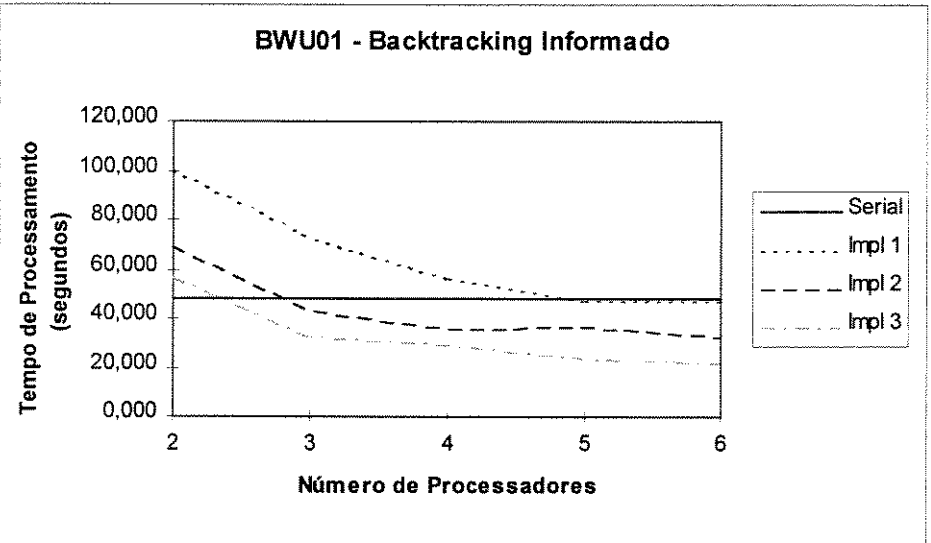
Tabela 14. Resultados da implementação serial

(a) <i>Backtracking Informado</i>			
	Perdas totais na solução ótima (W)	Nós explorados	Tempo de execução (segundos)
REDE47	2.117	237	2,92
BWU01	83.849	2494	48,33
BAURU9	89.030	14667	1218,74
(b) <i>Backtracking Heurístico</i>			
	Perdas totais na solução ótima (W)	Nós explorados	Tempo de execução (segundos)
REDE47	2.117	103	1,54
BWU01	83.849	1788	32,46
BAURU9	89.030	13280	1070,94



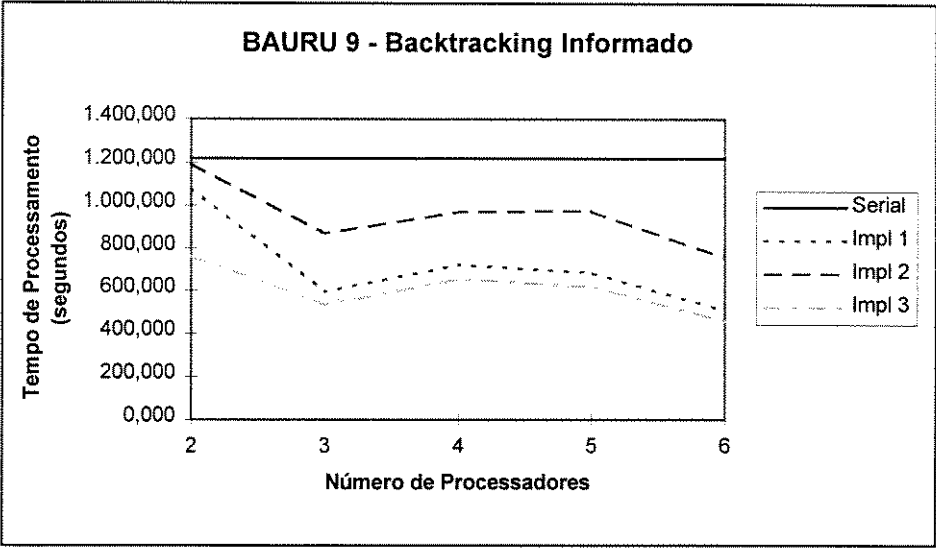
	Nós explorados	Tempo de execução(em segundos) em função do número de processadores (estações de trabalho)				
		2	3	4	5	6
Implementação 1	264	3,79	2,82	2,29	2,16	1,82
Implementação 2	229	4,36	3,16	2,68	2,44	2,11
Implementação 3	201	3,91	2,84	2,49	2,18	1,90

Figura 31. REDE 47 - Backtracking Informado



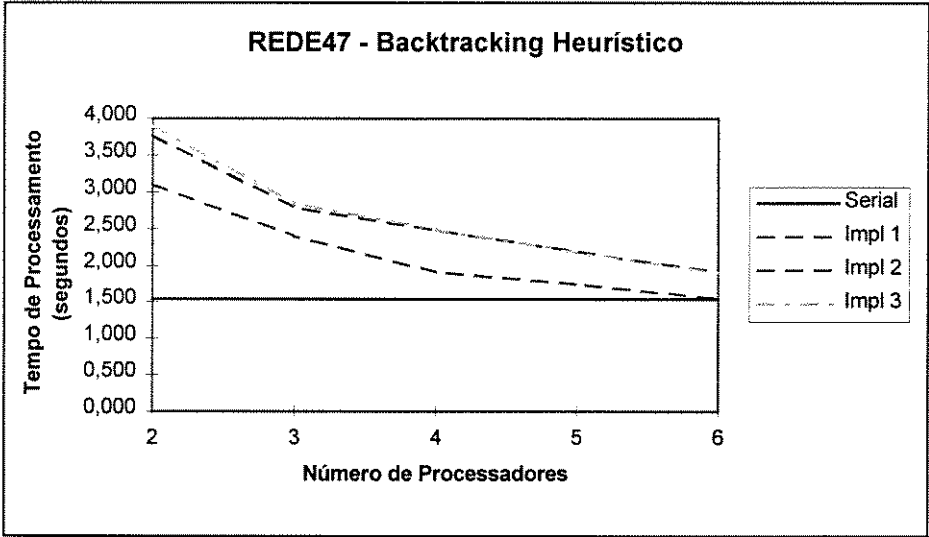
	Nós explorados	Tempo de execução(em segundos) em função do número de processadores (estações de trabalho)				
		2	3	4	5	6
Implementação 1	6662	100,07	71,74	56,55	47,20	47,15
Implementação 2	2725	69,50	43,37	35,79	36,45	32,38
Implementação 3	2500	57,46	32,31	29,31	23,87	22,32

Figura 32. BWU01 Backtracking Informado



	Nós explorados	Tempo de execução(em segundos) em função do número de processadores (estações de trabalho)				
		2	3	4	5	6
Implementação 1	21813	1082,16	594,02	721,92	686,28	507,07
Implementação 2	21434	1193,69	867,52	968,03	971,51	753,34
Implementação 3	11108	763,66	532,19	655,35	618,90	458,23

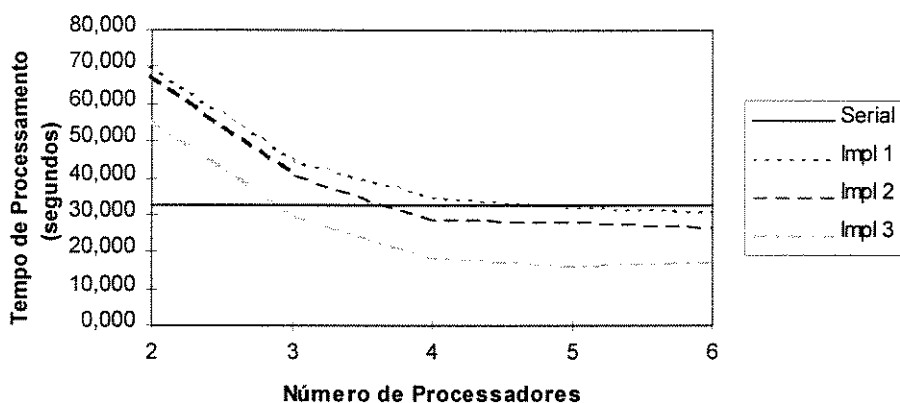
Figura 33. BAURU 9 Backtracking Informado



	Nós explorados	Tempo de execução(em segundos) em função do número de processadores (estações de trabalho)				
		2	3	4	5	6
Implementação 1	140	3,11	2,40	1,91	1,74	1,55
Implementação 2	134	3,77	2,80	2,48	2,19	1,91
Implementação 3	134	3,91	2,84	2,49	2,18	1,90

Figura 34. REDE 47 Backtracking Heurístico

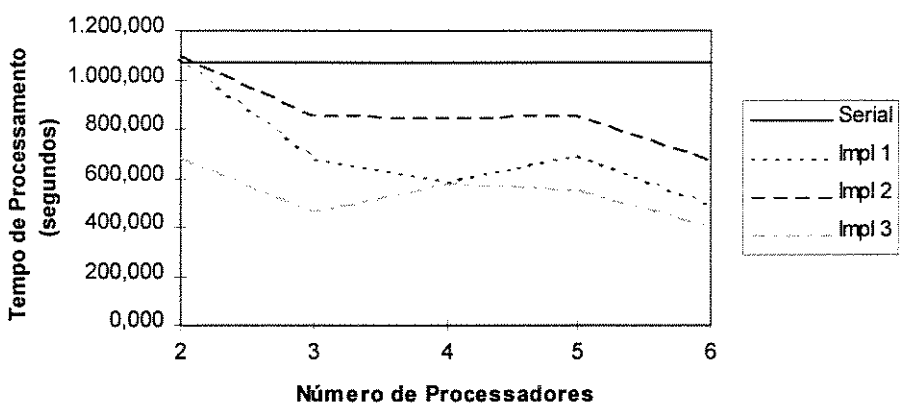
BWU01 - Backtracking Heurístico



	Nós explorados	Tempo de execução(em segundos) em função do número de processadores (estações de trabalho)				
		2	3	4	5	6
Implementação 1	4782	70,26	44,71	34,74	32,34	30,97
Implementação 2	2250	67,63	40,83	29,11	28,17	26,59
Implementação 3	2105	55,45	30,03	18,35	16,52	17,50

Figura 35. BWU01 Backtracking Heurístico

BAURU 9 - Backtracking Heurístico



	Nós explorados	Tempo de execução(em segundos) em função do número de processadores (estações de trabalho)				
		2	3	4	5	6
Implementação 1	20402	1085,03	678,63	584,80	693,38	490,94
Implementação 2	20008	1068,76	853,13	850,64	853,92	680,17
Implementação 3	10144	687,53	465,56	579,79	551,34	408,49

Figura 36. BAURU 9 Backtracking Heurístico

Algumas observações podem ser feitas sobre os resultados. Nota-se que o número de nós explorados nas implementações paralelas, em geral, são maiores do que na implementação serial. Isto acontece devido ao compartilhamento parcial das informações durante o processo de busca das subárvores independentes. A poda por dominância executada nos processos filhos depende do valor da solução factível armazenada. No início da busca, todos os processos filhos recebem o mesmo valor da solução factível do processo pai. Porém, quando um dos processos filhos encontra uma nova solução factível, melhor que a anterior, ela não é comunicada imediatamente aos demais processos filhos, podendo prejudicar a poda por dominância dos demais processos filhos – algumas configurações que seriam podadas, caso o valor da nova solução factível fosse conhecido, são explorados.

A implementação usando o terceiro critério de comunicação apresentou o melhor resultado. Este resultado é consequência de uma distribuição mais equilibrada do esforço computacional e do melhor sincronismo entre os processos. Em alguns casos o equilíbrio do esforço computacional é demonstrado pelo menor número de nós explorados, comparados com a implementação serial. O segundo critério de comunicação implementado apresentou resultados piores, devido a seu sincronismo inadequado.

A distribuição dos processos filhos é alterada quando acrescenta-se mais processadores. Se, por coincidência, processos mais demorados forem atribuídos ao mesmo processador, o tempo de processamento total pode aumentar quando utiliza-se um número menor de processadores. Esta condição pode ser observada no caso da rede BAURU9 para o algoritmo *backtracking informado*, quando foi utilizado quatro e cinco processadores em vez de três processadores.

Os resultados do algoritmo *backtracking heurístico* são sensíveis ao parâmetro δ (uma estimativa do acréscimo médio das perdas correspondentes a abertura de cada ciclo), utilizado na função de avaliação. Este parâmetro foi ajustado para cada um dos casos estudados, procurando-se obter o melhor compromisso entre o número de nós explorados e a perda da solução ótima do problema. A Tabela 15 apresenta

o número de nós explorados usando alguns valores de δ para a rede BAURU9 (Tão, 1997).

Tabela 15. Número de nós explorados em função do parâmetro δ

Valor do parâmetro δ	Número de nós explorados	Obtenção da solução ótimo
0,080	13577	SIM
0,085	13338	SIM
0,086	13332	SIM
0,087	13315	SIM
0,088	13280	SIM
0,089	13490	NÃO
0,090	13395	NÃO

Os valores de δ adotados para as implementações dos algoritmos paralelos foram 0,09; 0,89 e 0,088 para as redes 48, Wu e BAURU9, respectivamente. Esses valores são ajustados para a implementação serial. O ajuste não se mostrou eficiente para a REDE47, com implementação paralela.

6.5. ANÁLISE DE DESEMPENHO DE ALGORITMOS PARALELOS

Uma das métricas mais utilizadas na análise de desempenho de algoritmos paralelos é o *speedup* (Bertsekas e Tsitsiklis, 1989). *Speedup* (S_p) é a medida de ganho de tempo do algoritmo paralelo em relação a um dado algoritmo serial,

$$S_p = \frac{T^*}{T_p}$$

onde T^* é usualmente o tempo de execução do melhor algoritmo serial existente para a solução do problema e T_p é o tempo requerido pelo algoritmo paralelo utilizando p processadores. Para o propósito dos estudos de casos, T^* pode ser o tempo para resolver o problema com a implementação serial (um processador).

Em princípio o melhor valor para S_p é p (teoricamente, neste caso poderíamos obter um valor para S_p maior do que p , se a divulgação de informações obtidas por alguns dos processadores permitir podas mais eficientes do que na implementação serial). Na prática, mesmo a situação $S_p = p$ é dificilmente atingida, em consequência dos tempos de comunicação e sincronização dos processadores. As

Tabela 16. *Speedup* para o *backtracking informato*

a) REDE47					
	Número de processadores (estações de trabalho)				
	2	3	4	5	6
Implementação 1	0,77	1,04	1,28	1,35	1,61
Implementação 2	0,67	0,93	1,09	1,20	1,39
Implementação 3	0,75	1,03	1,17	1,34	1,54

(b) BWU01					
	Número de processadores (estações de trabalho)				
	2	3	4	5	6
Implementação 1	0,48	0,67	0,85	1,02	1,03
Implementação 2	0,70	1,11	1,35	1,33	1,49
Implementação 3	0,84	1,50	1,65	2,03	2,16

(c) BAURU9					
	Número de processadores (estações de trabalho)				
	2	3	4	5	6
Implementação 1	1,13	2,05	1,69	1,78	2,40
Implementação 2	1,02	1,41	1,26	1,25	1,62
Implementação 3	1,60	2,29	1,86	1,97	2,66

Tabela 17. *Speedup* para o *backtracking heurístico*

(a) REDE47					
	Número de processadores (estações de trabalho)				
	2	3	4	5	6
Implementação 1	0,50	0,64	0,80	0,88	0,99
Implementação 2	0,41	0,55	0,62	0,70	0,81
Implementação 3	0,39	0,54	0,62	0,71	0,81

(b) BWU01					
	Número de processadores (estações de trabalho)				
	2	3	4	5	6
Implementação 1	0,46	0,73	0,94	1,00	1,05
Implementação 2	0,48	0,80	1,12	1,15	1,22
Implementação 3	0,59	1,08	1,77	1,97	1,86

(c) BAURU9					
	Número de processadores (estações de trabalho)				
	2	3	4	5	6
Implementação 1	0,99	1,58	1,83	1,55	2,18
Implementação 2	1,00	1,26	1,26	1,25	1,58
Implementação 3	1,59	2,30	1,85	1,94	2,62

Tabelas 16(a), (b) e (c), e 17(a), (b) e (c) mostram os valores do *speedup* para o *backtracking informado* e *backtracking heurístico*, respectivamente.

O baixo *speedup* obtido em quase todos os casos é decorrente da ociosidade dos processadores. Experimentalmente pode-se observar que o tempo total de processamento é diretamente relacionado com o tempo de execução dos processos filhos mais demorados. Quando utiliza-se números maiores de processadores, alguns executam rapidamente seus processos e ficam aguardando os outros processadores terminarem os processos filhos mais demorados. Isso deixa alguns processadores ociosos, contribuindo para diminuir o *speedup* do processo paralelo.

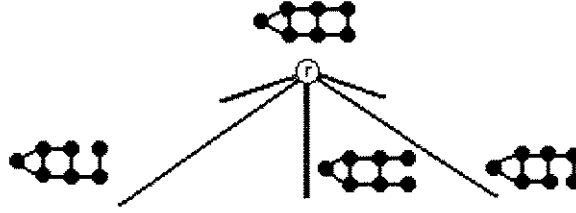


Figura 37. Árvore de busca parcial gerada pelo algoritmo A^*

A implementação paralela do *backtracking informado* apresentou melhor *speedup* do que o *backtracking heurístico*. Este resultado indica que o *backtracking heurístico* requer informações não compartilhadas totalmente pelas subárvores no procedimento paralelo de busca.

6.6. PARALELIZAÇÃO DO ALGORITMO A^*

Os resultados obtidos para o algoritmo A^* apresentados no Capítulo 5 são justificados, principalmente, pelo esforço computacional necessário para calcular o valor da função heurística, f_h . Pode-se notar, no entanto, que é possível calcular o valor da função heurística, independentemente, para cada filho durante a expansão do nó, conforme mostra a Figura 33. Esta observação permite desenhar um procedimento paralelo para o algoritmo A^* , descrito a seguir.

No procedimento descrito no Capítulo 5, o valor da função de avaliação heurística f_h é estimado para cada um dos nós filhos de n . Em seguida a lista ABERTO é reordenada para indicar o próximo nó a ser expandido. Esse processo continua até que um nó objetivo seja encontrado e a solução ótima do problema declarada.

Utilizando a estrutura de processamento paralelo *pai-filho*, pode-se designar, a cada um dos nós gerados, um processo filho para estimar o valor de f_h . O valor obtido é transmitido para um processo pai, que armazena-o na lista ABERTO. Após todos os processos filhos terem concluído suas tarefas, o processo pai reordena a lista ABERTO e verifica se o primeiro nó da lista é um nó objetivo. Em caso afirmativo encerra o processo, apresentando a solução ótima do problema. Caso não seja um nó objetivo, o procedimento de busca continua expandindo o nó, gerando todos os

seus nós filhos e repetindo o processo para estimar f_h . As etapas do *algoritmo A^* paralelo* são resumidas a seguir.

Algoritmo A^* Paralelo

Algoritmo do Processo Pai

- Etapa 1.* Inclua o nó inicial r na lista ABERTO. Calcule o valor da função $f_h(r)$ associado ao nó r - raiz da árvore de busca.
- Etapa 2.* Crie a lista FECHADO; inicialmente esta lista está vazia.
- Etapa 3.* Se a lista ABERTO está vazia, pare; não há solução para o problema. Caso contrário, continue na Etapa 4.
- Etapa 4.* Remova o nó n da lista ABERTO com o menor valor de $f_h(n)$; inclua-o na lista FECHADO. Se vários nós são qualificados, escolha um deles arbitrariamente.
- Etapa 5.* Expanda o nó n , gerando todos os seus sucessores e ligando-os com apontadores para o nó n .
- Etapa 6.* Se algum dos sucessores é um nó objetivo, pare e recupere a solução do obtida.
- Etapa 7.* Para cada um dos n' sucessores de n , designe um processo filho para estimar $f_h(n') = g^*(n') + h(n')$.
- Etapa 8.* Aguarde cada um dos processos filhos informar o valor de $f_h(n')$ e, então, inclua o nó n' na lista ABERTO, associando-o ao valor de $f_h(n')$ calculado.
- Etapa 9.* Após todos os processos filhos terminarem, classifique a lista ABERTO em ordem decrescente de f_h e continue na Etapa 3.

Algoritmo do Processo Filho

- Etapa 1.* Estime $h(n')$ e calcule o valor de $f_h(n') = g^*(n') + h(n')$.

Obtém-se o valor de f_h executando o problema de fluxo de custo mínimo não linear (*pfornl*) para cada uma das configurações de rede associada aos nós filhos do nó n . Portanto, o tempo de processamento do processo filho fica restrito à execução do algoritmo de *pfornl*, mais o tempo de comunicação ao processo pai da solução obtida.

A implementação do algoritmo A^* paralelo será tema da continuidade deste trabalho.

CAPÍTULO 7

COMENTÁRIOS FINAIS E CONCLUSÕES

Neste trabalho foi discutido o problema da minimização das perdas do sistema de distribuição de energia elétrica. Como o sistema de distribuição é operado em configuração radial, o problema pode ser interpretado como uma generalização do problema da árvore recobridora de custo mínimo. O objetivo foi encontrar a árvore recobridora do grafo representando a rede de distribuição, que minimize as perdas de energia elétrica no sistema, enquanto são satisfeitas as restrições de demanda e os limites para as quedas tensões e capacidades dos fluxos de correntes nas linhas do sistema.

A solução ótima desse problema combinatorial foi encontrada utilizando um procedimento de duas etapas, que usa técnicas de otimização de fluxos em redes não lineares e estratégias de busca da área de inteligência artificial. Na primeira etapa a restrição de operação radial da rede é relaxada, levando a uma solução otimista para o problema; as informações desta solução são usadas na segunda etapa, onde busca-se a solução factível de custo mínimo. Estas etapas são repetidas continuamente, até que a configuração de rede radial de mínimas perdas seja encontrada.

O procedimento recursivo usa conceitos da área de inteligência artificial para contornar a explosão exponencial do esforço computacional.

A configuração radial ótima é encontrada pela busca no espaço de estados, onde o estado (ou nó) é caracterizado pela configuração de rede com um padrão de fluxo ótimo. A busca se inicia no nó associado a configuração com malhas, solução do

problema radial. Os nós sucessores são gerados pela abertura de cada uma das malhas, com a solução de um novo problema relaxado. A busca termina quando for encontrado um nó objetivo, associado à configuração de rede radial de mínimas perdas.

Os procedimentos de busca investigados foram *busca irrevogável*, *backtracking informado*, *backtracking heurístico* e *algoritmo A^** . O procedimento busca irrevogável pertence à classe das buscas não informadas, ou cegas. Ele apresenta bom desempenho computacional, porém não garante a solução ótima do problema. É recomendável para melhorar uma solução radial conhecida.

Os procedimentos *backtracking informado*, *backtracking heurístico* e algoritmo A^* pertencem à classe das buscas informadas, ou guiadas. O *backtracking informado* foi concebido por meio da incorporação de conhecimento à estratégia de busca *backtracking padrão*. Dois tipos de conhecimento foram utilizados: a solução do problema relaxado foi usado para indicar o próximo nó a ser explorado e uma solução factível de referência foi adotada para permitir a poda de caminhos da árvore de busca. Este procedimento de busca garante a obtenção de uma solução ótima global para o problema de minimização das perdas, explorando apenas uma parcela do espaço de estados.

A estratégia de busca *backtracking heurístico* adota uma função de avaliação heurística para guiar o procedimento na exploração do espaço de estados. A função de avaliação é dividida em duas parcelas. A primeira parcela determina o aumento de perdas do nó inicial da árvore de busca, até o nó em exploração. A segunda parcela estima o aumento das perdas do nó em exploração até um nó objetivo. Esta abordagem reduz consideravelmente a porção do espaço de estados explorados, e encontra a solução ótima global, quando a estimativa de perdas é ajustada convenientemente.

Uma função heurística foi construída para guiar um procedimento A^* da classe busca em grafo, para encontrar a configuração de rede de mínimas perdas. A função heurística construída considerou a partição da rede em componentes associados à

rede radial e a rede com malhas. Foram provadas a admissibilidade e a monotonicidade (Nilson, 1980) do algoritmo para a função heurística construída.

Observa-se na bibliografia a utilização de pequenas redes para apresentar os resultados de métodos para a redução das perdas em sistemas de distribuição. No caso dos métodos exatos, estas redes são ainda menores. Este aspecto indica que o esforço computacional exigido por essa classe de problemas é ainda uma dificuldade a ser superada. Os algoritmos descritos neste trabalho são uma contribuição para contornar esta dificuldade. A estrutura desses algoritmos permite acrescentar formalmente o conhecimento heurístico relacionado ao problema, para guiar o processo de busca nas suas várias etapas. O conhecimento pode ser adicionado na etapa inicial, na seleção do próximo nó a ser explorado ou na geração dos nós filhos. Com o objetivo de buscar alternativas para melhorar o desempenho computacional, investigou-se também a paralelização dos algoritmos busca propostos (Tão, 1997; Tão, Cavellucci e Lyra, 1998).

A paralelização dos procedimentos *backtracking informado* e *backtracking heurístico* foi baseada na possibilidade do particionamento da árvore de busca em subárvores independentes. A implementação desses algoritmos utilizou a estrutura *pai-filho*. O processo pai resolve o problema relaxado associado ao nó inicial da árvore de busca, ativa todos os procedimentos filhos, estabelece uma estratégia de comunicação e coordena o procedimento de obtenção de uma solução para o problema de minimização de perdas. Cada processo filho executa um procedimento de busca no seu espaço de estados (subárvore) para encontrar a melhor solução. Resultados parciais, ou completos, são enviados ao processo pai, de acordo com o esquema de comunicação adotado. Uma proposta de paralelismo do algoritmo A^* foi também apresentada.

O trabalho abordou o problema de minimização das perdas do sistema de distribuição, considerando a redução das perdas decorrentes do fluxo de potência ativa. Os resultados obtidos são válidos para redes com um fator potência típico de sistemas de distribuição modernos, onde a potência reativa na rede é compensada localmente, para evitar as penalidades decorrentes da legislação vigente. Quando a

potência reativa apresenta valores significativos, pode-se utilizar o mesmo procedimento para obter uma configuração de rede que minimize as perdas da componente reativa. A configuração de rede encontrada pode ser combinada com a configuração de rede obtida para a potência ativa, indicando alternativas para a redução do total das perdas na rede.

Investigações futuras decorrentes deste trabalho envolverão os seguintes temas:

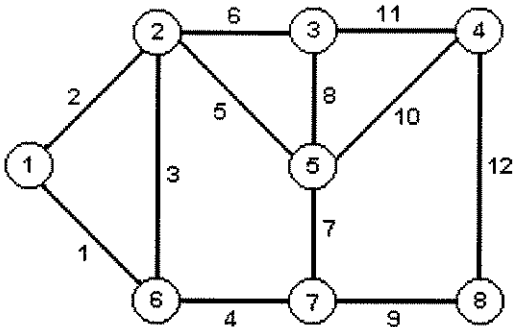
- estudo de técnicas para a obtenção de boas soluções aproximadas, para a solução do problema de mínimas perdas, como *grasp*, *redes neurais*, *algoritmo genético* e *sistemas classificatórios* ;
- Paralelismo do algoritmo A^* ;
- Investigação de melhores funções heurísticas de avaliação para o algoritmo A^* .
- Investigar a aplicação do conceito de *times* assíncronos.

APÊNDICE 1

OBTENÇÃO DOS CONJUNTOS M_{o_i} E M_{t_i}

A1.1. INTRODUÇÃO

As listas $p_o(i)$, $l_D(i)$ e $p_D(i)$, bem como os conjuntos M_{o_i} e M_{t_i} , são obtidos a



Arco(k)	1	2	3	4	5	6	7	8	9	10	11	12
o_k	1	1	2	6	2	2	5	3	7	4	3	4
t_k	6	2	6	7	5	3	7	5	8	5	4	8

Figura 38. Armazenamento na estrutura arco-orientada

partir das listas dos nós origens ($O = \{o_k\}$) e terminais ($T = \{t_k\}$) dos arcos. Para exemplificar a construção e armazenamento de M_{o_i} e M_{t_i} , utiliza-se o grafo apresentado no Capítulo 2, repetido na Figura 38.

A1.2. CONSTRUÇÃO DA LISTA P_o

Para construir a lista $p_o(i)$, renumera-se a lista de arcos em ordem crescente do nó origem. Assim,

se $o(k_i) < o(k_{i+1})$, então $k_i < k_{i+1}$

se $o(k_i) = o(k_{i+1})$, arbitra-se a ordem de k_i e k_{i+1}

A Tabela 18 mostra a lista de arcos da Figura 38, após a renumeração em ordem crescente do nó origem.

Para construir o conjunto M_{o_i} é necessário definir a lista auxiliar $P_o = \{p_o(i)\}$. Esta lista contém o menor índice do arco cuja origem esteja no nó i ; no caso de nenhum arco possuir sua origem no nó i , $p_o(i) = p_o(i+1)$. Assim,

$$p_o(1) = 1$$

$$p_o = \{k \text{ tal que } o(k) \geq i, o(k-1) < i\}, \text{ se } 1 < i \leq ne$$

$$p_o(n+1) = m+1$$

Tabela 18. Lista de arcos em ordem crescente do nó origem.

Arco(k)	1	2	3	4	5	6	7	8	9	10	11	12
o_k	1	1	2	2	2	3	3	4	4	5	6	7
t_k	6	2	6	5	3	5	4	5	8	7	7	8

Tabela 19. Lista P_o de apontadores para os arcos origens da lista de arcos

i	1	2	3	4	5	6	7	8
$p_o(i)$	1	3	6	8	10	11	12	13

onde n é o número de nós do grafo e m é o número de arcos. A Tabela 19 ilustra a lista de apontadores P_o para o grafo da Figura 38.

Com auxílio da lista P_o , pode-se obter o conjunto M_o

$$M_o = \{k \text{ tal que } p_o(i) \leq k \leq p_o(i+1)\}$$

A1.3. CONSTRUÇÃO DAS LISTAS L_T E P_T

A obtenção do conjunto M_i é realizada através das listas auxiliares $L_T = \{l_i(i)\}$, que armazena a lista de arcos em ordem crescente do nó terminal, e $P_T = \{p_i(i)\}$, cujos elementos apontam para a primeira entrada na lista L_T do arco que termina no nó i . Chamando-se de j' o índice do arco k em L_T , ordena-se os arcos na forma descrita a seguir.

$$\text{se } l(k_i) < l(k_{i+1}), \text{ então } j'_i < j'_{i+1}$$

$$\text{se } l(k_i) = l(k_{i+1}), \text{ arbitra-se a ordem de } j'_i \text{ e } j'_{i+1}$$

A Tabela 20 exemplifica a lista de apontadores L_T , renumerando os arcos da Figura 38 em ordem crescente dos nós terminais.

Tabela 20. Lista L_T para a ordenação dos arcos pelos nós terminais

k'	1	2	3	4	5	6	7	8	9	10	11	12
$l_T(k')$	2	5	7	4	6	8	1	3	10	11	9	12

Tabela 21. Lista P_T dos apontadores para os arcos terminais

i	1	2	3	4	5	6	7
$p_T(i)$	1	2	3	4	7	9	11

A definição da lista $P_T = \{p_T(i)\}$ é apresentada a seguir.

$$p_T(1) = 1$$

$$p_T = \{k \text{ tal que } t(l_T(j')) \geq i, t(l_T(j' - 1)) < i\}, \text{ se } 1 < i \leq ne$$

$$p_T(n+1) = m+1$$

onde n é o numero de nós do grafo e m é o número de arcos. A Tabela 21 ilustra a lista P_T de apontadores para os arcos terminais, correspondentes a Figura 38.

O conjunto M_{l_i} pode então ser obtido com auxílio das listas L_T e P_T ,

$$M_{l_i} \{l_T(j') \text{ tal que } p_T(i) \leq j' \leq p_T(i+1)\}.$$

APÊNDICE 2

FERRAMENTAS COMPUTACIONAIS PARA A OPERAÇÃO DA REDE

A2.1. INTRODUÇÃO

A supervisão e coordenação das atividades de operação do sistema de distribuição de energia elétrica são desenvolvidas no *Centro de Operação da Distribuição - COD*. Estas atividades podem ser agrupadas nas funções de *Atendimento*, *Operação* e *Engenharia Operacional*. A Figura 39 mostra o inter-relacionamento dessas funções e a ação resultante na rede de distribuição.

Na função *Atendimento* são registradas as reclamações dos consumidores de energia elétrica, tais como, falta de energia e nível de tensão inadequado. Após a triagem destas reclamações, o possível defeito e sua localização são definidos. Por exemplo, no caso de falta de energia as opções de defeito são: equipamento (transformador, chave, banco capacitor, etc.), instalação do consumidor, alimentador e subestação de distribuição. O defeito pode também ser externo ao sistema de distribuição, ou seja, no sistema de transmissão, que alimenta a subestação de distribuição.

A *Operação*, ao receber informações sobre o tipo de defeito e sua localização, inicia o procedimento de atendimento emergencial. A primeira providência é confirmar o

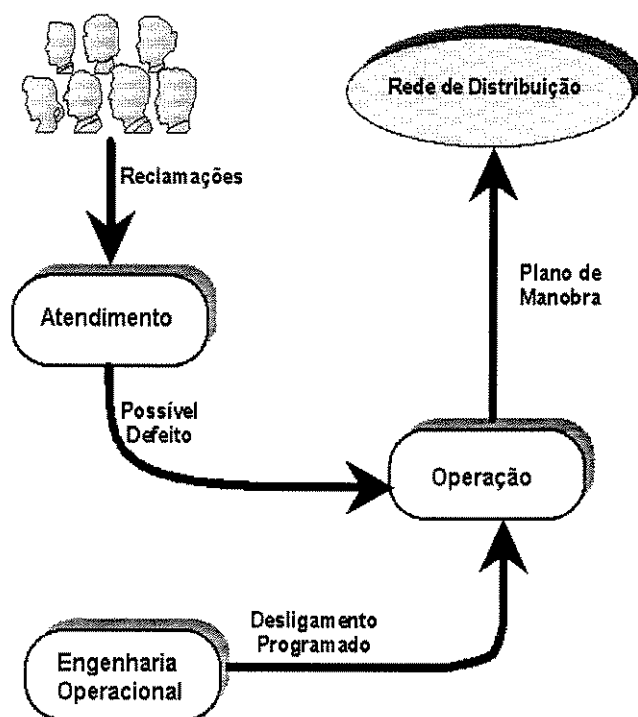


Figura 39. Diagrama funcional do COD

tipo de defeito ocorrido. O operador aciona a equipe de manutenção de campo mais próxima da localização do defeito, coordenando suas atividades via rádio. Após a confirmação do defeito pela equipe de manutenção, o operador elabora um plano de manobras para isolar o defeito e restaurar a área escura, ou utiliza um plano previamente elaborado pela engenharia operacional. Quando o plano é executado, os chaveamentos são confirmados um a um e as manutenções na rede, como troca de equipamento, são realizadas. No caso de redes de distribuição automatizadas, algumas das atividades discutidas (por exemplo, os chaveamentos) são executadas através de comandos enviados a *Unidades Terminais Remotas* - UTR, instaladas em vários pontos da rede.

A *Engenharia Operacional* é responsável pela elaboração dos planos de manobras programadas para transferência de carga, análise de contingência e coordenação das funções de atendimento e operação.

Este capítulo descreve três ferramentas computacionais: *Configurador de Rede*, *Simulador de Manobras* e *SBC-Manobra*. Estas ferramentas exemplificam o uso da

estrutura de dados descrita no capítulo anterior e métodos de simulação e inteligência artificial, em particular a técnica de sistemas especialistas, aplicados ao processo de operação de sistemas de distribuição de energia elétrica.

O configurador de rede apresenta, graficamente, o estado da rede de distribuição, auxiliando na identificação do estado das chaves (aberta ou fechada), na identificação do estado dos blocos de carga (energizado ou desenergizado), visualização da inversão do fluxo de corrente - importante em redes com reguladores de tensão e dispositivos de proteção, na localização de defeitos, na visualização da configuração atual de um circuito e na identificação dos componentes de um dado circuito.

O simulador de manobras verifica a possibilidade ou não do fechamento de uma chave, considerando os limites de queda de tensão e capacidade física dos componentes da rede. Esta ferramenta computacional auxilia a elaboração de planos de manobra, que compreendem a definição e seqüência de operação das chaves a serem manobradas para isolar um equipamento defeituoso, transferir carga de um circuito para outro, isolar parte da rede para manutenção ou restabelecer o fornecimento de energia elétrica na ocorrência de defeito numa subestação.

SBC-MANOBRA é um sistema baseado em conhecimento para auxiliar a operação da rede no estado de emergência, sugerindo planos de manobras para isolar o defeito e restabelecer a parte da rede que está desenergizada. Esta aplicação utiliza a técnica de sistemas especialistas para tratar o conhecimento empírico necessário à elaboração de planos de manobras emergenciais.

A seguir são descritas, sucintamente, cada uma destas ferramentas computacionais e apresentadas suas principais características.

A2.2. CONFIGURADOR DE REDE

O conhecimento da configuração da rede em operação é fundamental para a monitoração do sistema de distribuição de energia elétrica. O estado dos blocos de

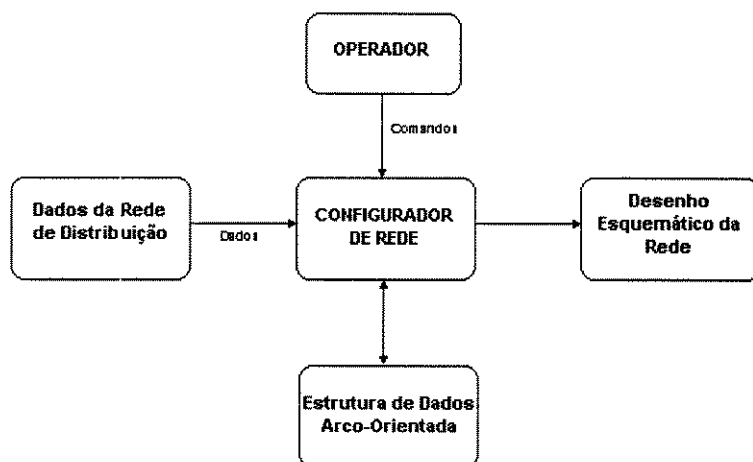


Figura 40. Principais componentes do *Configurador de Rede*

carga (energizados ou desenergizados) e das chaves (abertas ou fechadas) são requeridos constantemente pelos operadores de rede. O configurador de rede foi desenvolvido para auxiliar os operadores identificarem a configuração da rede em operação. Ele apresenta, dinamicamente, o estado da rede de distribuição, que é alterado pela abertura ou fechamento das chaves. A Figura 40 apresenta esquematicamente os principais componentes do configurador de rede.

A rede de distribuição é representada pelo modelo de grafo. Os dados da topologia da rede, e seu estado inicial, são lidos de um arquivo e armazenados na estrutura de dados arco-orientada. Comandos são digitados pelo operador para informar ao programa as ações a serem tratadas.

A rede de distribuição é apresentada de forma esquemática numa tela gráfica, mostrando os elementos da rede e seus respectivos estados, conforme ilustra a Figura 41. As subestações são representadas por quadrados, os blocos de carga por círculos e as chaves por linhas cheias ou pontilhadas, representando as chaves fechadas e abertas, respectivamente. No esquema é possível diferenciar os vários circuitos da rede e as alternativas de chaveamento para a mudança da configuração do sistema.

O operador de rede informa ao programa as chaves a serem abertas ou fechadas para alterar a configuração da rede. A ocorrência de um defeito em um elemento da

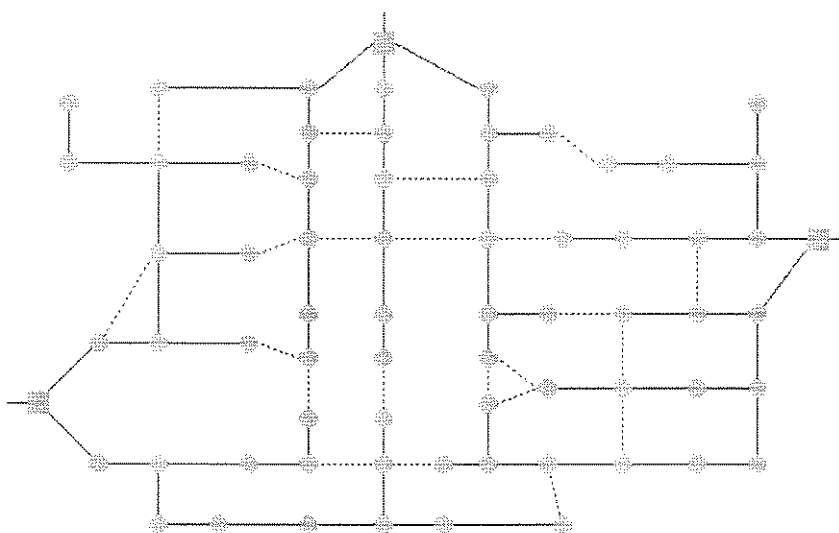


Figura 41. Diagrama esquemático da rede de distribuição em operação

rede pode também ser informada. O programa atualiza a estrutura de dados e em seguida apresenta as seguintes informações:

- *o estado dos blocos de carga e das chaves,*
- *o elemento de rede com defeito,*
- *as chaves candidatas para isolar os elementos de rede com defeito,*
- *as chaves candidatas para restaurar as áreas desenergizadas e*
- *possível inversão do fluxo de corrente.*

A Figura 42 exemplifica o desenho da rede após o operador informar a ocorrência de defeito na subestação à direita do diagrama esquemático. Os blocos de carga na cor preta são blocos desenergizados, definindo uma *área escura*. É possível, também, visualizar o conjunto de chaves abertas como alternativas de restabelecimento do fornecimento de energia à área escura. Escolhida uma destas chaves (e informando o programa), o configurador de rede mostra a área onde foi restabelecido o fornecimento de energia, conforme exemplifica a Figura 43. Observe na figura que apenas parte da área escura foi reenergizada. Procedimento semelhante, escolhendo-se outra alternativa de chaveamento, deve ser utilizado para o restabelecimento da área ainda sem energia.

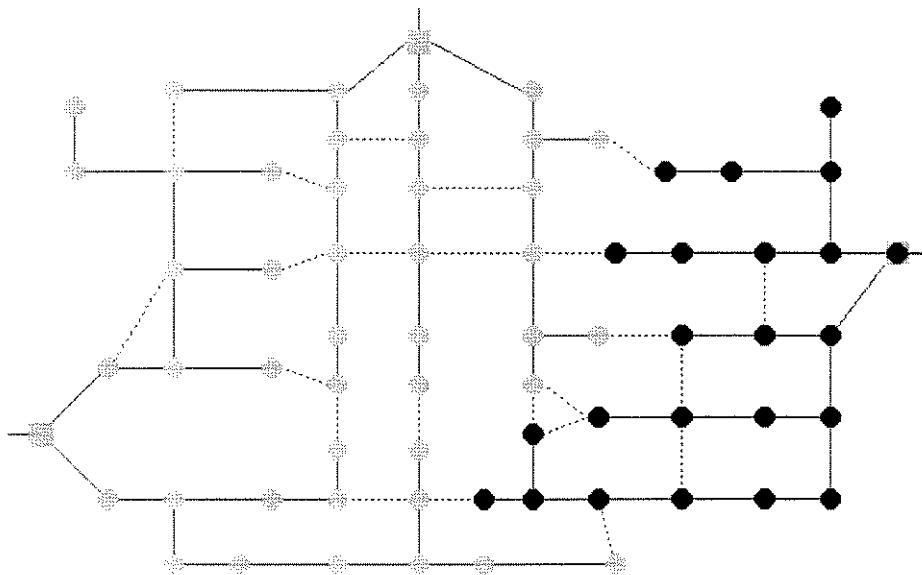


Figura 42. Defeito na subestação

O configurador de rede pode armazenar aproximadamente 5000 nós e 5000 arcos em um computador de 32 Mbytes de memória RAM. Seu desempenho num Pentium 100 MHz permite fornecer as informações necessárias instantaneamente.

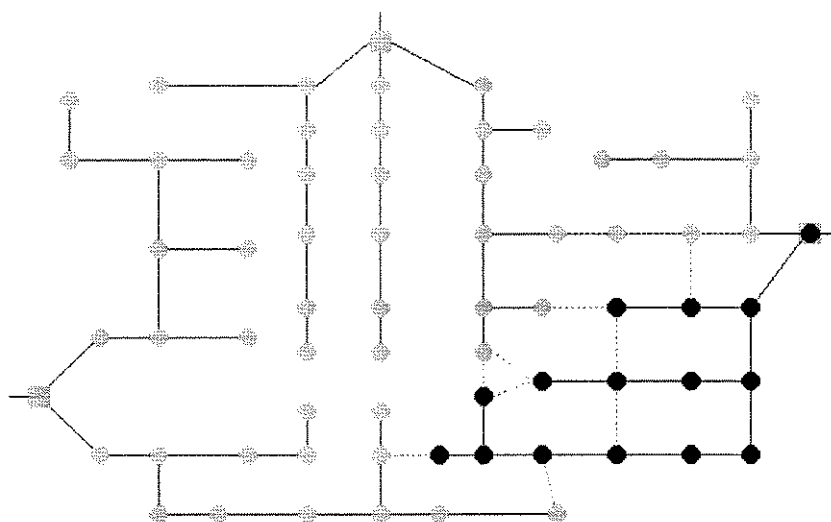


Figura 43. Restauração parcial da rede

A2.3. SIMULADOR DE MANOBRAS - SIMA

A ocorrência de interrupções, programadas ou não, no fornecimento de energia elétrica no sistema de distribuição, exige a execução de manobras para isolar o defeito e restabelecer a energia às áreas desenergizadas, consequência do isolamento do defeito. Estas manobras devem estar de acordo com as normas operativas vigentes na empresa distribuidora e com as restrições elétricas impostas pela rede.

Um plano de manobras propõe a seqüência de chaveamentos (abertura ou fechamento de chaves) para reduzir os efeitos da interrupção sobre os consumidores e permitir a execução segura dos serviços de manutenção emergencial ou programado. A seqüência de manobras é obtida através da análise do estado da rede, visando a busca de alternativas de chaveamento.

No processo de decisão sobre qual chave deve ser fechada para o restabelecimento da energia da área escura, é importante prever os efeitos desta operação antes de efetivá-la. O simulador denominado *SIMA* foi projetado para prever o comportamento elétrico da rede de distribuição, com o objetivo de auxiliar a análise de alternativas de chaveamento e a elaboração de planos de manobra. O *SIMA* avalia as restrições elétricas após o fechamento de uma chave, indicando se a manobra é aconselhável ou não. No caso de rejeição da manobra, o simulador apresenta o elemento de rede que teve suas restrições elétricas violadas ou o ponto da rede com queda de tensão abaixo dos limites estabelecidos. Quando a manobra é aconselhável, o simulador recalcula os valores das grandezas elétricas (corrente e tensão) da nova configuração da rede. A Figura 44 mostra os principais módulos do *SIMA*.

No diagrama da Figura 44 o bloco *Interface* trata os comandos e informações fornecidas pelo operador e apresenta no desenho unifilar da rede o resultado da simulação. O bloco *Configurador de Rede* trata dos aspectos da topologia da rede, conforme descrito anteriormente. Os cálculos de tensão e corrente são executados no bloco *Simulador de Manobras*. Dados como comprimento dos trechos de rede,

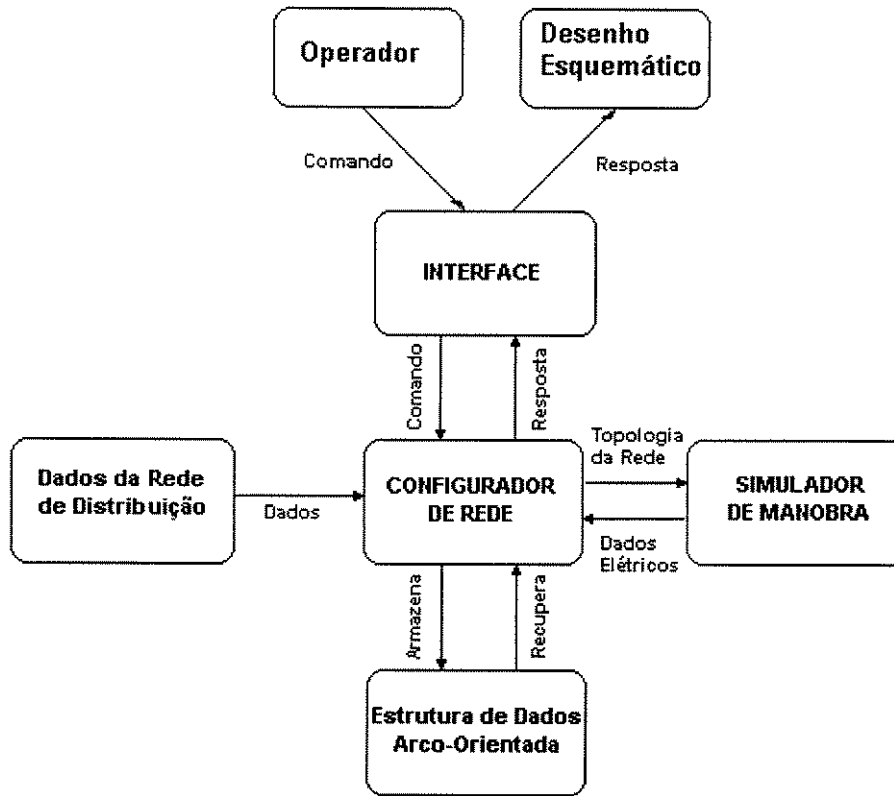


Figura 44. Módulos do *SIMA*

material e bitola dos cabos, e a demanda de corrente dos blocos de carga são armazenados utilizando a estrutura de dados arco-orientada, descrita no capítulo 2.

A avaliação dos requisitos elétricos usa o *Método dos Momentos* (Kasatkin e Perekalin, 1970). Os fluxos de corrente nas chaves e a queda de tensão nos blocos de carga são obtidos por procedimentos de cálculo e tratamento da rede semelhantes àqueles necessários à solução de problemas de fluxo de custo mínimo.

A Figura 45 ilustra o estado da rede de distribuição após o bloco com defeito ter sido isolado. Os blocos hachurados pertencem a rede energizada, o bloco com defeito, representado na cor preta, foi isolado por manobras anteriores e a área escura está demarcada por um retângulo. Para exemplificar, suponha que a chave k seja candidata a reenergizar a área escura. O simulador avalia o estado da rede após o fechamento da chave k na sequência de etapas relacionadas na Figura 46.

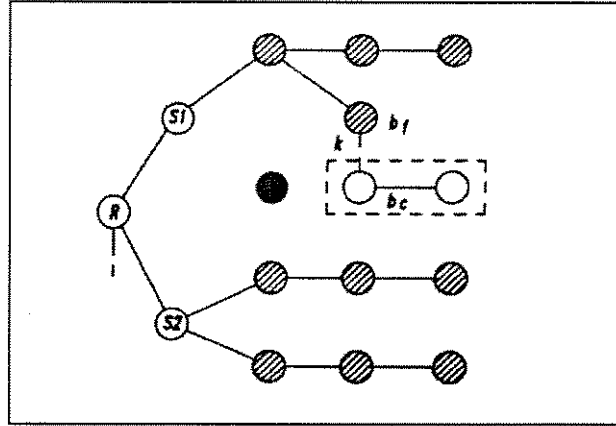


Figura 45. Rede após isolar o defeito na rede

Caso alguma das avaliações listadas na Figura 46 não seja satisfeita, o simulador informa o motivo pelo qual a chave k não deve ser fechada. Caso contrário, a chave k é fechada (i.e., incluída na árvore A) e os valores dos fluxos de potência e tensões são recalculados para esta nova configuração de rede.

O simulador *SIMA* foi desenvolvido na linguagem C++, em ambiente WINDOWS. A interação com o sistema é realizada através de menus, para facilitar

- E1. Avalia se a subestação comporta a carga a ser transferida*
- E2. Avalia se a capacidade da chave k permite a passagem carga a ser transferida*
- E3. Avalia se queda de tensão no bloco fonte, bf , e no bloco carga, bc , está dentro de limites preestabelecidos*
- E4. Avalia se queda de tensão nos blocos terminais da área escura está dentro de limites preestabelecidos*
- E5. Avalia se o alimentador do bloco fonte, bf , pode suportar a carga a ser transferida*
- E6. Avalia se a queda de tensão dos blocos terminais do alimentador do bloco fonte, bf , está dentro de limites estabelecidos em norma.*

Figura 46. Etapas para validação de uma manobra

Tabela 22. Funções e opções do menu

Função	Opção	Descrição
Arquivo	Abrir	permite a leitura do arquivo com os dados da rede
	Sair	finaliza o programa
Visual	Aproxima	visualiza uma área numa escala definida
	Afasta	ampliação a escala de visualização
	Panorâmica	visualização da rede em estudo
Relatórios		imprime os dados do estado atual da rede
Plano		grava um esboço do plano de manobras
Sobre		apresenta dados sobre o programa

a segurança e rapidez dos diálogos. A tela principal do simulador é apresentada na Figura 47 e um resumo das funções e opções são listadas na Tabela 22.

A leitura dos dados da rede de distribuição é realizada através da escolha da opção *Abrir* da função *Arquivo*. Após a leitura do arquivo de dados da rede de

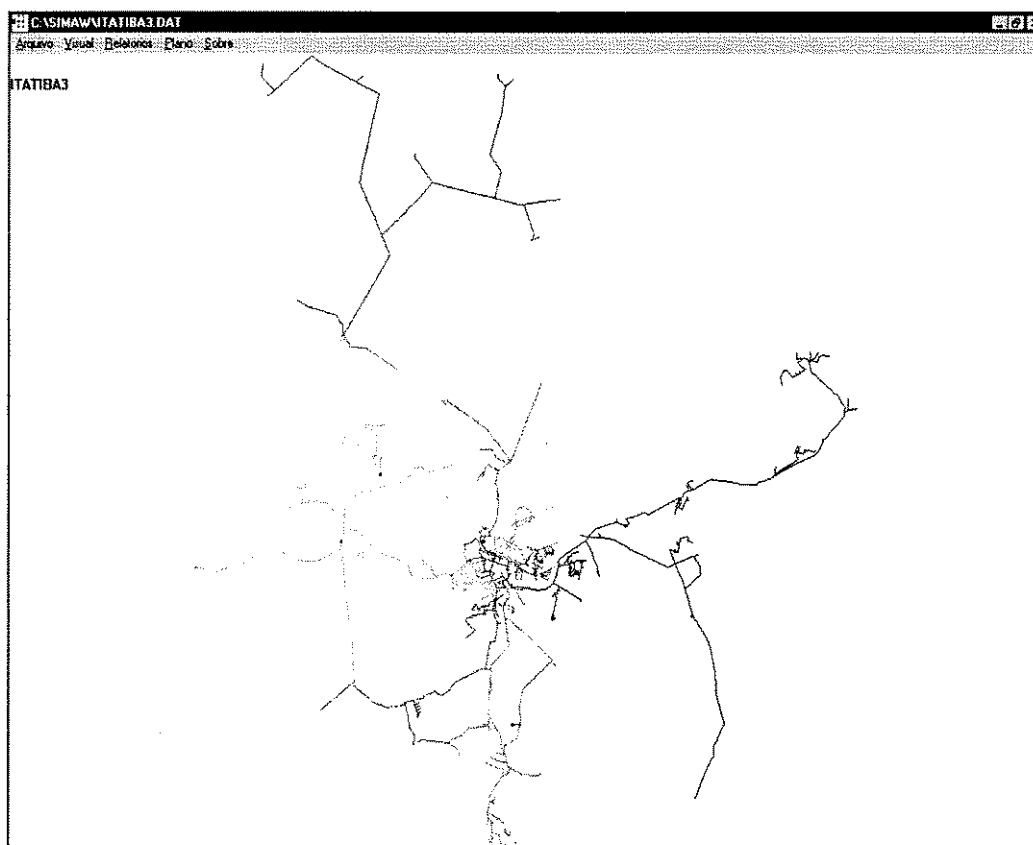


Figura 47. Desenho do unifilar da rede de distribuição

Tabela 23. Simbologia adotada para as entidades da rede

Entidade	Símbolo	Cor
subestação	quadrado	azul claro
ponto de rede	círculo	azul escuro = energizado vermelho = desenergizado
chaves/disjuntores	quadrado	vermelho = fechado verde = aberto
trechos	reta	do alimentador = energizado cinza = desenergizado

distribuição, o SIMA desenha o diagrama unifilar da rede. A Figura 47 mostra o desenho do diagrama unifilar para a cidade de Itatiba (região de Campinas - SP).

No diagrama unifilar, os alimentadores são desenhados em cores diferentes, para melhor visualização. Seu traçado é baseado em coordenadas georeferenciadas para localização dos elementos da rede nos mapas da empresa. Os símbolos e cores adotadas para representar as entidades da rede de distribuição estão listados na Tabela 23.

Com as opções de visualização da função *Visual* é possível ampliar, reduzir e ter uma visão geral do diagrama unifilar. Selecionada uma região no diagrama, a opção

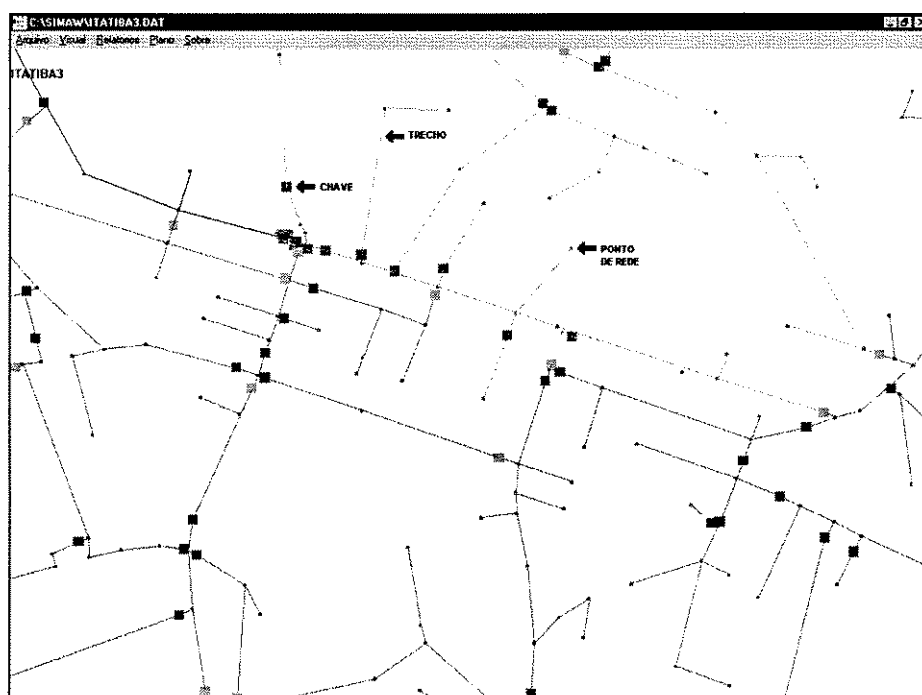


Figura 48. Detalhamento da rede de distribuição

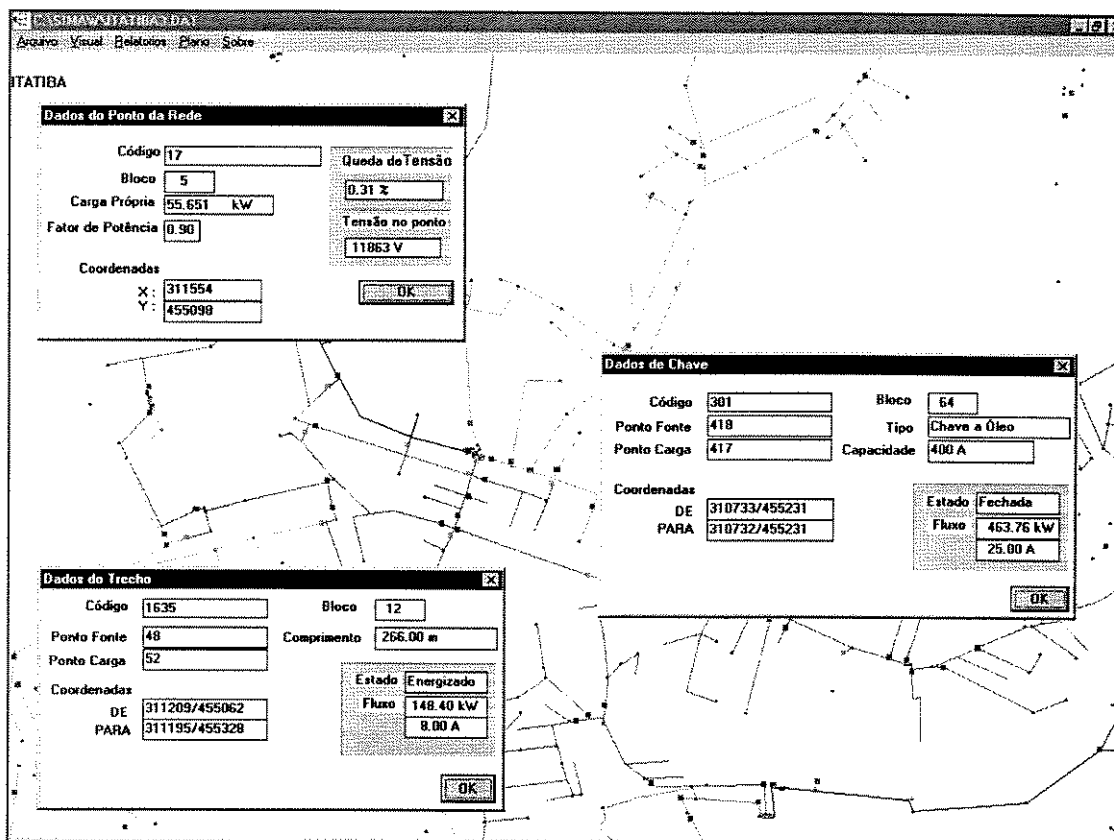


Figura 49. Dados dos elementos da rede de distribuição

Aproxima amplia esta região na escala apropriada. A Figura 48 mostra a visualização detalhada da rede de distribuição de Itatiba; pode-se identificar na figura as chaves, trechos e pontos de rede.

Inversamente, com a opção *Afasta* é possível reduzir o diagrama a escalas apropriadas para análises das manobras desejadas. A opção *Panorâmica* retoma a visão geral apresentada na Figura 47.

As informações dos elementos de rede (chave, trecho e pontos de rede) são apresentadas pelo SIMA em janelas específicas. A Figura 49 mostra exemplos de dados para ponto de rede, chave e trecho.

A simulação de abertura ou fechamento de chaves é realizada com o acionamento do botão esquerdo do “mouse” sobre a chave selecionada. A chave escolhida muda de cor, indicando seu novo estado; de vermelho para verde, se a chave foi aberta e inversamente, no fechamento da chave (de verde para vermelho). Na abertura da

chave, os trechos desenergizados são redesenhados na cor cinza e os pontos de rede na cor vermelha, delimitando a área atingida pela manobra. No fechamento de chave, os trechos são redesenhados na mesma cor do circuito fonte e os pontos de rede na cor azul, indicando a reenergização da área. Em qualquer um dos casos, abertura ou fechamento de chave, o SIMA recalcula o fluxo de corrente e a queda de tensão para a nova configuração da rede; verifica também a validade da manobra, considerando as etapas listadas na Figura 46. No caso da manobra levar ao surgimento de um anel (ciclo) na rede, ou não for validada, uma mensagem é apresentada, indicando a causa da rejeição.

Todas as manobras válidas são armazenadas numa lista, mantendo a seqüência da operação. Esta lista de chaveamentos define uma opção de plano de manobras para a hipótese de contingência ou transferência de carga. A função *PLANO* grava um esboço da opção do plano de manobras, conforme mostra a Figura 50.

No esboço do plano de manobras estão listadas a seqüência de chaveamentos e as

Hipótese:		
Horário :		
Carga :		
Subestações		
	Demanda Máxima	
Código	Atual(A)	Final(A)
COL	780.000	454.000
CRO	794.000	954.000
IBA	488.000	488.000
USG	0.000	0.000
Função	Tipo	Código
ABRIR	Chave	475
ABRIR	Disjuntor	COL05
FECHAR	Chave a Óleo	453
ABRIR	Chave a Óleo	373
FECHAR	Chave a Óleo	367
Obs: Chaves-Faca deverão ser abertas com ferramenta Load-Booster.		

Figura 50. Esboço do plano de manobras

demandas, atual e final, de cada subestação da rede ou seja, define-se a distribuição da carga antes e depois da sequência de manobras. Completando os dados *hipótese* (objetivo do plano de manobras), *horário* (de aplicação do plano), *carga* (no caso de transferência de carga) e outras observações necessárias ao esclarecimento e execução, obtém-se o plano de manobras final, com a sequência de manobras adequadas para utilização na operação da rede de distribuição.

A2.4. SISTEMA BASEADO EM CONHECIMENTO SBC-MANOBRA

O sistema baseado em conhecimento *SBC-MANOBRA* (Cavellucci e Lyra, 1993) foi projetado para sugerir alternativas de planos de manobras aos operadores de rede. O plano de manobra compreende os chaveamentos para isolar o defeito e aqueles para restabelecer o fornecimento de energia. A decisão sobre um plano de manobra requer conhecimento sobre a topologia, o estado (tensão e corrente) e as restrições de operação da rede; devem também observar as normas da empresa e o conhecimento adquirido pelos operadores no exercício de suas funções. Estes requisitos são representados por módulos distintos no *SBC-MANOBRA*. A Figura 51. ilustra esquematicamente os principais componentes do *SBC-MANOBRA*.

O conhecimento adquirido por experiência anterior ou treinamento e as normas da empresa são representados na *base de conhecimento*. A topologia da rede é armazenada na *base de dados da rede* e manuseada pelo *configurador de rede*. O *simulador de manobras* (descrito anteriormente) avalia a queda de tensão e os limites de corrente, indicando se a manobra atende ou não as restrições de operação. A *máquina de inferência* guia a busca de uma alternativa de plano de manobra, simulando o processo de raciocínio do operador de rede.

A base de dados armazena *fatos* e *regras de produção* (Nilson, 1982), representando o conhecimento sobre operação. Os fatos são representados por triplas objeto-atributo-valor (Winston, 1984). Os *objetos* podem representar entidades físicas, tais como uma chave, ou conceituais como um alimentador. Os *atributos* são geralmente

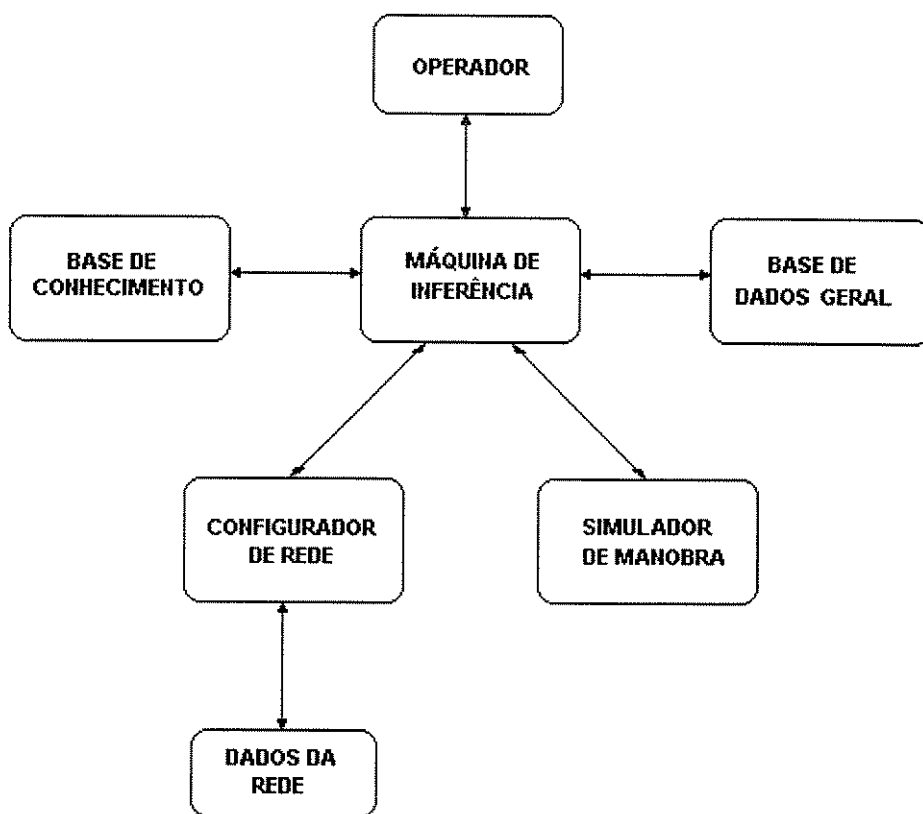


Figura 51. Diagrama de blocos do *SBC-MANOBRA*

características ou propriedades associadas ao objeto. Por exemplo, o código e o tipo são atributos do objeto chave. Finalmente, o *valor* especifica a natureza de um atributo em uma situação particular. O valor para o atributo tipo do objeto chave pode ser, por exemplo, disjuntor em uma dada instância.

O conjunto de *regras de produção* é uma das representações do conhecimento, usada para descrever o relacionamento entres os fatos. Elas têm o formato *se-cláusula/então-cláusula*. A *se-cláusula* é chamada premissa e a *então-cláusula* conclusão. Ambas, premissa e conclusão, podem ter mais de uma cláusula, conectadas pelo operador lógico *E*.

A máquina de inferência guia o processo de raciocínio de *SBC-MANOBRA*. Suas principais tarefas são: avaliar os fatos e regras existentes, para adicionar novos fatos à base de conhecimento (isto é, inferir novos conhecimentos) e decidir em que ordem as regras devem ser avaliadas (isto é, controlar o processo de raciocínio). A

estratégia de controle adotada para a máquina de inferência foi do tipo *backward chaining*, eficiente quando os possíveis resultados são conhecidos e em número razoavelmente pequeno, características predominantes do problema de restauração da rede de distribuição.

Na implementação do *SBC-MANOBRA* foi adotada uma abordagem híbrida. Os módulos básicos de um sistema baseado em conhecimento (a base de conhecimento, máquina de inferência e base de dados geral) foram codificados na linguagem PROLOG. O configurador de rede e o simulador de manobras (módulos numéricos) foram codificados na linguagem C++.

A2.5. COMENTÁRIOS

As ferramentas como o simulador de manobras e o *SBC-MANOBRA* têm como objetivo buscar, ou auxiliar na busca, de soluções que satisfaça restrições previamente estabelecidas, não havendo necessidade de nenhum esforço computacional adicional pela busca da solução ótima. Entretanto, há situações onde é necessário soluções de melhor qualidade, ou mesmo a solução ótima. O corpo deste trabalho de tese foi dedicado ao problema da reconfiguração da rede para mínimas perdas.

APÊNDICE 3

PUBLICAÇÕES VINCULADAS AO TRABALHO

Cavellucci, C.; Lyra, C. (1993). A graph oriented knowledge-based system to assist restoration of distribution system. Proceedings of IFAC 1993 World Congress (G.C. Goodwin and R.J. Evans, Eds). Vol 8. The Institute of Engineers (Australia). pp. 263-289.

Cavellucci, C; Lyra, C. (1994). Estratégia de busca e técnicas de grafo para minimização de perdas em rede de distribuição de energia elétrica. *XXVI Simpósio Brasileiro de Pesquisa Operacional*, Nov-Dez, 1994, pp. 168-173.

Cavellucci, C.; Lyra, C. (1995). Minimization of energy losses in electric power distribution system by intelligent search strategies. *7th IFAC/IFORS/IMACS Symposium on Large Scale Systems*, London, UK, July, 1995, vol 2, pp. 641-646.

Cavellucci, C.; Lyra Filho, C. (1995) Conceitos de grafo para a avaliação de manobras em sistemas de distribuição de energia elétrica. *VIII CLAIO/XXVIII SPBO*, Rio de Janeiro, pp. 363-368.

- Cavellucci, C.; Lyra, C. (1996). Sistema baseado em conhecimento orientado por grafo para elaboração de planos de manobras em rede de distribuição. *CIREC Congresso Internacional de Redes Electricas de Distribuicion*, Buenos Aires, Argentina, Diciembre, 1996, trabalho 3-19, Vol. 3.
- Cavellucci, C.; Lyra, C. (1997). Minimization of Energy Losses in Electric Power Distribution Systems by Intelligent Search Strategies. *International Transactions in Operational Research*, Vol. 4, No. 1, 1997, pp. 23-33.
- Tão, W.; Cavellucci, C.; Lyra Filho, C. (1995). Processamento paralelo e inteligência artificial para minimização de perdas na distribuição de energia elétrica. *XXVII – Simpósio Brasileiro de Pesquisa operacional – SBPO*, Novembro 1995, pp. XXXX
- Tão, W; Cavellucci, C.; Lyra Filho, C. (1996). Minimização de perdas na distribuição de energia elétrica por busca informada com processamento paralelo. *XI Congresso Brasileiro de Automática – SBA*, Setembro 1996, p.p. 1769 a 1774.
- Tão, W; Cavellucci C.; Lyra, C. (1998). Parallel Intelligent Search for Loss Minimization in Distribution System. Aceito para apresentação e publicação nos anais do IEEE Transmission and Distribution Conference, New Orleans, USA, April 1999.

BIBLIOGRAFIA

- Augugliaro, A.; Dusonchet, L. & Mangione S. (1995). An efficient greedy approach for minimum loss reconfiguration of distribution networks. *Electrical Power Systems Research*, v. 35, p. 167-176.
- Ahuja, R.; Magnanti, T. & Orlin, J. (1993). *Network Flows: theory, algorithms, and applications*. Englewood Cliffs: Prentice Hall.
- Aoki, R.; Tchimori, T. & Kanezashi, M. (1987). Normal state optimal load allocation in distribution systems. *IEEE Transactions on Power Delivery*, v. 2, n. 1, p. 147-155.
- Augugliaro, A.; Dusonchet, L.; Mangione, S. (1995). An efficient greedy approach for minimum loss reconfiguration of distribution networks. *Electric Power System Research*, n. 35, p. 167-176.
- Baran, M. & Wu, F. (1989a). Optimal sizing of capacitors placed on a radial distribution system. *IEEE Transactions on Power Delivery*, v. 4, n. 1, p. 735-743.
- _____. (1989b). Network reconfiguration in distribution systems for loss reduction and load balancing. *IEEE Transactions on Power Delivery*, v. 4, n. 2, p. 1401-1407.
- Bertsekas, D. P. & Tsitsiklis, J. N. (1989) *Parallel and Distribution Computation: numerical methods*. Prentice Hall.
- Borožan, V.; Rajčić, D.; Ackovski, R. (1995). Improved method for loss minimization in distribution networks. *IEEE Transactions on Power Systems*, v. 10, n. 3, p. 1420-1425.
- Broadwater, R.; Khan, A.; Shalaan, H.; Lee, R. (1993). Time varying load analysis to reduce distribution losses through reconfiguration. *IEEE Transactions on Power Delivery*, v. 8, n. 1, p. 294-300.
- Buch, J.; Miller, R.; Wheeler, J. (1982). Distribution system integrated voltage and reactive power control. *IEEE Transactions on Power Apparatus and Systems*, v. 101, n. 2, p. 284-289.

- Castro, C. & Watanabe, A. (1990). An efficient reconfiguration algorithm for loss reduction of distribution systems. *Electric Power Systems Research*, n. 19, p. 137-144.
- Cavellucci, C. (1989). *Sistema especialista em manobras de redes de distribuição de energia elétrica*. Campinas: Unicamp. (Dissertação de Mestrado apresentada à Faculdade Engenharia Elétrica da Universidade Estadual de Campinas).
- Cavellucci, C. & Lyra, C. (1993). A graph oriented knowledge-based system to assist restoration of distribution system. In: IFAC World Congress, Sidney. Vol 8, p. 263-289.
- _____. (1995). Minimization of energy losses in electric power distribution systems by intelligent search strategies. 7th International Federation of Automatic Symposium on Large Scale Systems, London, UK, July. Vol. 2, p. 641-646.
- _____. (1996). Conceitos de grafo para a avaliação de manobras em sistemas de distribuição de energia elétrica. In: Congresso Latino Americano de Investigacion Operacional, 8º Encontro da Sociedade Brasileira de Pesquisa Operacional, 28, Rio de Janeiro, 1996. p. 363-368.
- Chen, C. S. & Cho, M.Y. (1992). Determination of critical switches in distribution systems. *IEEE Transactions on Power Delivery*, v. 7, n. 3, p. 1443-1449.
- _____. (1993). Energy loss reduction by critical switches. *IEEE Transactions on Power Delivery*, v. 8, n. 3, p. 1246-1253.
- Cherkaoui, R.; Bart A.; Germond, A. (1993). Optimal configuration of electrical distribution networks using heuristic methods. In: *Power System Computation Conference (PSCC)*, 11, Avignon, France, 1993. p. 147-154.
- Chiang, H. & Jumeau, R. (1990a). Optimal network reconfiguration in distribution systems: Part1: a new formulation and a solution methodology. *IEEE Transactions on Power Delivery*, v. 5, n. 4, p. 1902-1909.
- Chiang, H. & Jumeau, R. (1990b). Optimal network reconfiguration in distribution systems: Part2: solution algorithms and numerical results. *IEEE Transactions on Power Delivery*, v. 5, n. 3, p. 1568-1574.
- Choi, D-S.; Kim, C-S.; Hasegawa, J. (1995). An application of genetic algorithms to the network reconfiguration in distribution for loss minimization and load balancing problem (part2). In: International Conference on Energy Management and Power Delivery (EMPD'95). p. 376-381.
- Cipoli, J.A. (1993). *Engenharia de Distribuição*. Rio de Janeiro: Qualitymark.

- Cinvalar, S.; Grainger, J. J.; Yin, H.; Lee, S. S. H. (1988). Distribution feeder reconfiguration for loss reduction. *IEEE Transactions on Power Apparatus and Systems*, v. 3, n. 3, p. 1217-1223.
- CODI. (1994). *Indicadores Gerenciais*. Rio de Janeiro: ABRADÉE.
- _____. (1997). *Método para determinação, análise e otimização das perdas técnicas em sistemas de distribuição*. Rio de Janeiro: ABRADÉE. (Relatório CODI 19.34.0)
- Fortunato, L.; Araripe Neto, T.; Albuquerque, J.; Pereira, M. (1990). *Introdução ao Planejamento da Expansão e Operação de Sistemas de Produção de Energia Elétrica*. Rio de Janeiro: Universidade Federal Fluminense.
- Garfinkel, R. & Nemhauser, G. (1972). *Integer Programming*. New York: John Wiley .
- Geist, A.; Beguelim, A.; Dongarva, J.; Jiang, W.; Manchek, R. & Sunderam, V. (1994). *PVM 3 User's Guide and Reference Manual*.
- Glamocanin, V. (1990). Optimal loss reduction of distribution networks. *IEEE Transactions on Power Systems*, v. 5, n. 3, p. 774-781.
- Glover, F. (1988). *Tabu search*, CAAI Report 88-3, Center For Applied Artificial Intelligence, Boulder, Colorado.
- Goswami, S. & Basu, S. (1992). A new algorithm for the reconfiguration of distribution feeders. *IEEE Transactions on Power Systems*, v. 7, n. 3, p. 1484-1491.
- Jensen, P. & Barnes, J. (1980). *Network flow programming*. New York: John Wiley.
- Jasmon, G.; Callistus, L.; Lee, C. (1991). A modified technique for minimization of distribution system losses. *Electric Power Systems Research*, 20, p. 81-88.
- Hart, P. E.; Nilson, Nils J. & Raphael, B. (1968). A formal basis for heuristic determination of minimum cost path. *IEEE Transactions on Systems Science and Cybernetics*, v. 4, n. 2, p. 100-107.
- Hadley, S.; Mark B. ; Vannelli, A. (1992). An efficient eigenvector approach for finding netlist partitions. *IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems*, v. 11, n. 7, p. 885-892.
- Huddleston, C.; Broadwater, R.; Chandrasekaran, A. (1990). Reconfiguration algorithm for minimizing losses in radial electric distribution system. *Electric Power Systems Research*, 18, p. 57-66.
- Kasatkin, A. & Perekalin, N. (1970). *Basic Electrical Engineering*. Moscou: Mir.
- Kennington, A. & Helgason, R. (1980). *Algorithms for Network Programming*. New York: John Wiley.

- Kim, H.; Ko, Y.; Jung, K.H. (1993). Artificial neural-network based feeder reconfiguration for loss reduction in distribution systems. *IEEE Transactions on Power Delivery*, v. 8, n. 3, p. 1356-1366.
- Lippmann, R. P. (1987). An introduction to computing with neural nets. *IEEE ASSP MAGAZINE*, April, p.p. 4-22
- Lee, T. E.; Cho, M. Y.; Chen, C. S. (1994). Distribution system reconfiguration to reduce resistive losses. *Electric Power System Research*, 30, p. 25-33.
- Liu, C.C.; Lee, S.J.; Vu, K. (1989). Loss minimization feeders: optimality and algorithms. *IEEE Transactions on Power Delivery*, v. 4, n. 2, p. 1281-1289.
- Luenberger, D. (1984). *Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley.
- Lyra, C. (1984). *Contribuição ao planejamento da produção de energia em sistemas de potência*. Campinas: Unicamp. (Tese de Doutorado apresentada a Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas - Publicação FEC 040/84).
- Lyra, C. & Tavares, H. (1988). A contribution to the midterm scheduling of large scale hydrothermal power systems. *IEEE Transactions on Power Systems*, v. 3, n. 3, p. 852-857.
- Merlin, A., Back, H. (1975). Search for minimal-loss operating spanning tree configuration in an urban power distribution system. **In:** Power System Computation Conference (PSCC), 5, Cambridge, UK, 1975. *Proceedings...* p. 1-18.
- Murphy, L. & Wu, F. (1990). *A comprehensive analysis of distribution automation systems*. Berkeley, CA: University of California. (Memorandum No. UCB/ERL M90/72. Electronics Research Laboratory)
- Nilsson, N. (1980). *Principles of Artificial Intelligence*. Palo Alto: Tioga.
- Nara, K.; Shiose, A.; Kitagawa, M.; Ishihara, T. (1992). Implementation of genetic algorithm for distribution systems loss minimum re-configuration. *IEEE Transactions on Power Systems*, v. 7, n. 3, p. 1044-1051.
- Pansini, A. T. (1983). *Electrical distribution engineering*. New York: McGraw-Hill.
- Pearl, J & Kim, J. H. (1982). Studies in semi-admissible heuristics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 4, p. 392-399.
- Pearl, J. (1984). *Heuristics: intelligent search strategies for computer problem solving*. Reading, MA: Addison-Wesley.

- Peponis, G.; Papadopoulos, M.; Hatziargyriou, N. (1995). Distribution network reconfiguration to minimize resistive line losses. *IEEE Transactions on Power Delivery*, v. 10, n. 3, p. 1338-1342.
- Rich, E. (1983). *Artificial Intelligence*. New York: McGraw-Hill.
- Ross, D.; Patton, J.; Cohen, A.; Carson, M. (1981). New methods for evaluating distribution automation and control (dac) systems benefits. *IEEE Transactions on Power Apparatus and Systems*, v. 100, n. 6, p. 2978-2986.
- Sárfi, R.; Salama, M.; Chikhani, A. (1996). Distribution system reconfiguration for loss reduction: an algorithm based on network partitioning theory. *IEEE Transactions on Power System*, v. 11, n. 1, p. 504-510.
- Sarma, N. & Rao, K. (1995). A new 0-1 integer programming method of feeder reconfiguration for loss minimization in distribution systems. *Electric Power System Research*, 33, p. 125-131.
- Shirmohammadi, D. & Hong, H. (1989). Reconfiguration of electric distribution networks for resistive line losses reduction. *IEEE Transactions on Power Delivery*, v. 4, n. 2, p. 1492-1498.
- Silva, M.C. (1990). *Planejamento a longo prazo em sistemas de distribuição de energia elétrica*. Campinas: Unicamp. (Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas).
- SIESEE - SISTEMA DE INFORMAÇÕES EMPRESARIAIS DO SETOR DE ENERGIA ELÉTRICA (1985). *Relatório estatístico do setor de energia elétrica para o período 1970/1984*. Rio de Janeiro: Eletrobrás.
- Tão, W.; Cavellucci, C.; Lyra Filho, C. (1996). Minimização de perdas na distribuição de energia elétrica por busca informada com processamento paralelo. In: Congresso Brasileiro de Automática, 11, São Paulo, 1996. *Anais...*
- Tão, W. (1997) *Minimização de perdas em redes de distribuição de energia elétrica através de métodos de busca inteligentes com processamento paralelo*. Campinas: Unicamp. (Tese de mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas).
- Tão, W.; Cavellucci C. & Lyra, C. (1998). *Parallel Intelligent Search for Loss Minimization in Distribution System*. Prosedings of the 1999 IEEE Transmission and Distribution Conference(New Orleans, April 1999), Vol. 1, pp. 218-222.
- Taylor, T. & Lubkeman, D. (1990). Implementation of heuristic search strategies for distribution feeder reconfiguration. *IEEE Transactions on Power Delivery*, v. 5, n. 1, p. 239-246.

- Wagner, T.; Chikhani, A.; Hackam, R. (1991). Feeder reconfiguration for loss reduction: an application of distribution automation. *IEEE Transactions on Power Delivery*, v. 6, n. 4, p. 1922-1933.
- Wang, J-C.; Chiang, H-D; Darling, G. R. (1996). An efficient algorithm for real-time network reconfiguration in large scale unbalanced distribution systems. *IEEE Transactions on Power Systems*, v. 11, n. 1, p. 511-517.
- Winston, P. H. (1984). *Artificial Intelligence*, 2^a ed. Reading, MA: Addison-Wesley.
- Wu, J.S.; Tomsovic, K.L.; Chen, C.S. (1991). A heuristic search approach to feeder switching operations. *IEEE Transactions on Power Delivery*, v. 6, n. 4, p. 1579-1585.