

**Eduardo Zambon**

*Otimização de índices de confiabilidade em redes de  
distribuição de energia elétrica*

Vitória - ES, Brasil

04 de dezembro de 2006

**Eduardo Zambon**

***Otimização de índices de confiabilidade em redes de  
distribuição de energia elétrica***

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo para obtenção do título de Mestre em Informática.

Orientador:

Berilhes Borges Garcia

Co-orientador:

Sérgio Antônio Andrade de Freitas

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA  
DEPARTAMENTO DE INFORMÁTICA  
CENTRO TECNOLÓGICO  
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Vitória - ES, Brasil

04 de dezembro de 2006

Dados Internacionais de Catalogação-na-publicação (CIP)  
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

---

Zambon, Eduardo, 1979-  
Z24o Otimização de índices de confiabilidade em redes de distribuição de energia elétrica / Eduardo Zambon. – 2006.  
79 f. : il.

Orientador: Berilhes Borges Garcia.  
Co-Orientador: Sérgio Antonio Andrade de Freitas.  
Dissertação (mestrado) – Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Otimização matemática. 2. Programação linear. 3. Programação inteira. 4. Pesquisa operacional. I. Garcia, Berilhes Borges. II. Freitas, Sérgio Antonio Andrade de. III. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 004

---

# *Resumo*

As empresas responsáveis pelo fornecimento de energia elétrica (concessionárias) devem instalar equipamentos de proteção (religadores e fusíveis) e de seccionamento (chaves) em locais adequados da rede de distribuição para prestar um serviço de boa qualidade. Órgãos reguladores estabelecem métricas (índices de continuidade) para quantificar e analisar o desempenho das concessionárias.

O problema abordado neste trabalho pode ser dividido em duas partes. A primeira trata de otimizar a alocação de religadores em uma rede de distribuição, buscando melhorar os índices de continuidade. A meta-heurística *Simulated Annealing* foi empregada e os resultados obtidos nos testes realizados ficaram muito próximos dos valores ótimos. Na segunda parte, tenta-se determinar a melhor alocação de religadores, fusíveis e chaves de forma simultânea, novamente para melhorar os índices de uma rede de distribuição. Um modelo de Programação Linear Binária proposto na literatura foi implementado e testado. As deficiências identificadas neste modelo motivaram o desenvolvimento de uma nova formulação de Programação Não-linear Binária, mais abrangente, e um algoritmo de *branch-and-bound* específico para resolver as formulações obtidas com o novo modelo.

As soluções propostas neste trabalho permitem que as concessionárias projetem ou reestruturem a proteção das redes de distribuição de energia elétrica de forma a melhorar o serviço prestado aos consumidores. Com isto, as empresas podem diminuir o custo dos investimentos e ao mesmo tempo garantir que os seus clientes serão melhor atendidos, o que gera benefícios econômicos a ambos.

# *Abstract*

An electric utility must install protection (reclosers and fuses) and sectionalizing (switches) devices in key points of its distribution network to serve its customers with a reliable power supply. Utility boards or similar commissions define measures (reliability indices) to quantify and analyze the electric utility services.

The problem addressed in this work is twofold. First, we want to optimize the allocation of reclosers in a distribution network to improve its reliability indices. The Simulated Annealing meta-heuristic was employed and its results were close to the optimal values in the tests performed. Second, we want to improve the effectiveness of a distribution protective design by identifying the type (recloser, fuse or switch) and location of devices to be installed. A Binary Linear Programming model found in the literature was implemented and tested. Its deficiencies led to the development of a more complete Binary Nonlinear Programming model and a specific branch-and-bound algorithm to solve it.

The solutions proposed in this work allow an electric utility to project and restructure the protection design of its distribution networks, allowing it to improve the service provided to its consumers. Hence, the company can cut investments costs and still ensure a better quality of service to its clients, generating economic benefits to both parties.

# *Dedicatória*

Ao meu pai (*in memoriam*).

# *Agradecimentos*

Ao Prof. Berilhes por me permitir desenvolver este trabalho, pelos vários conselhos e incentivos, pela amizade e por ser um ótimo chefe.

Ao Prof. Sérgio pelas várias sugestões de melhorias, pela amizade e consideração e por ser um modelo de pessoa responsável e esforçada.

Ao Prof. Flávio pelas ajudas e conselhos, pelo companherismo e pelos lanches na cantina.

Ao Daniel, à Débora e à Veruska por formarem a melhor equipe de desenvolvimento que alguém pode querer. Extremamente competentes, dedicados e responsáveis, são pessoas com quem sempre se pode contar e companhias agradabilíssimas.

Um agradecimento especial ao Prof. Raul. Infelizmente não foi possível terminarmos o trabalho começado mas ele sempre será a pessoa com quem eu mais aprendi nesses últimos 5 anos de convivência. Agradeço pelos conselhos, pelas broncas e pelas inúmeras vezes em que ele me ajudou. Espero sempre corresponder às suas expectativas.

Ao professores do Departamento de Informática, pelas aulas, pela simpatia, pelo coleguismo e pelos vários almoços.

Ao amigo Idílio, pelas várias caronas, pelos rocks e pelas discussões altamente construtivas sobre os mais variados assuntos, principalmente nossos projetos.

Aos amigos, sempre presentes nos bons e maus momentos.

À minha família, por toda a ajuda me foi dada até hoje e pelo incentivo para seguir sempre em frente.

Ao Mestre Leonardo Guimarães, por me ensinar sobre o Tao.

À Ve por me fazer crescer. Hoje eu me sinto muito mais preparado para encarar quaisquer desafios que venham a surgir graças à ela.

# *Sumário*

## **Lista de Figuras**

## **Lista de Tabelas**

<b>1</b>	<b>Introdução</b>	p. 13
1.1	Motivação . . . . .	p. 13
1.2	Descrição do problema . . . . .	p. 14
1.3	Revisão bibliográfica . . . . .	p. 15
1.4	Estrutura da dissertação . . . . .	p. 16
<b>2</b>	<b>Caracterização do problema</b>	p. 17
2.1	Características da rede de distribuição . . . . .	p. 17
2.2	Classificação dos tipos de falha e interrupções . . . . .	p. 18
2.3	Equipamentos de proteção / seccionamento . . . . .	p. 19
2.4	Estrutura de um alimentador . . . . .	p. 19
2.5	Índices de continuidade da rede de distribuição . . . . .	p. 21
2.5.1	Definição da ANEEL . . . . .	p. 21
2.5.2	Parâmetros necessários para a estimativa de índices . . . . .	p. 22
2.5.3	Redefinição da forma de cálculo dos índices . . . . .	p. 23
2.6	Estrutura e tratamento dos dados . . . . .	p. 24
2.6.1	Descrição da massa de teste: subestações . . . . .	p. 24
2.6.2	Informações sobre a topologia dos alimentadores . . . . .	p. 24
2.6.3	Histórico de interrupções . . . . .	p. 24

2.6.4	Extração dos parâmetros de estimativa . . . . .	p. 26
2.6.4.1	Cálculo das taxas de falhas . . . . .	p. 26
2.6.4.2	Cálculo do MTTR e MTTS . . . . .	p. 27
<b>3</b>	<b>Otimização da alocação de religadores em um alimentador</b>	<b>p. 28</b>
3.1	Ambiente de desenvolvimento e testes . . . . .	p. 28
3.2	Simulação Analítica . . . . .	p. 29
3.2.1	Algoritmo para estimativa de índices de confiabilidade . . . . .	p. 29
3.2.2	Análise de complexidade da Simulação Analítica . . . . .	p. 31
3.2.3	Comparação entre índices históricos e estimados . . . . .	p. 32
3.2.4	Restauração A Montante . . . . .	p. 33
3.3	Busca Exaustiva . . . . .	p. 35
3.3.1	Descrição do algoritmo . . . . .	p. 35
3.3.2	Análise de complexidade da Busca Exaustiva . . . . .	p. 37
3.3.3	Resultados experimentais . . . . .	p. 37
3.4	<i>Simulated Annealing</i> . . . . .	p. 38
3.4.1	Idéia geral do algoritmo . . . . .	p. 38
3.4.2	As diferentes formas de vizinhança . . . . .	p. 40
3.4.2.1	Avaliação do desempenho das vizinhanças . . . . .	p. 41
3.4.3	A versão final do algoritmo . . . . .	p. 42
3.4.3.1	Resultados . . . . .	p. 43
<b>4</b>	<b>Otimização da alocação de dispositivos em um alimentador</b>	<b>p. 45</b>
4.1	A modelagem de PLB de Soudi e Tomsovic . . . . .	p. 45
4.1.1	Formulação . . . . .	p. 46
4.1.1.1	Restrições . . . . .	p. 49
4.1.2	Um exemplo . . . . .	p. 49

4.1.3	Resultados . . . . .	p. 51
4.2	Uma nova modelagem PNLB para o problema . . . . .	p. 54
4.2.1	Apresentação . . . . .	p. 54
4.2.1.1	Restrições . . . . .	p. 57
4.2.2	Um exemplo . . . . .	p. 58
4.2.3	Resultados . . . . .	p. 60
4.3	Um algoritmo <i>branch-and-bound</i> para resolver os problemas de PNLB . . .	p. 61
4.3.1	Descrição . . . . .	p. 61
4.3.2	Resultados . . . . .	p. 67
<b>5</b>	<b>Conclusão</b>	p. 69
5.1	Trabalhos futuros . . . . .	p. 70
	<b>Referências</b>	p. 72
	<b>Apêndice A – Testes do <i>Simulated Annealing</i></b>	p. 74

## *Lista de Figuras*

2.1	Topologia do alimentador $C7$ . . . . .	p. 20
2.2	Árvore do alimentador $C7$ . . . . .	p. 21
3.1	Restauração A Montante . . . . .	p. 33
4.1	Um tronco principal ou ramal lateral com $n$ blocos [ST98] . . . . .	p. 47
4.2	Um alimentador radial simples com 7 blocos [ST98] . . . . .	p. 50
4.3	Uma árvore de um alimentador fictício . . . . .	p. 52
4.4	Árvore de busca para $FO$ . . . . .	p. 62
4.5	Árvore de busca para $FO$ com poda . . . . .	p. 63

# *Lista de Tabelas*

2.1	Características das subestações de teste . . . . .	p. 25
3.1	Simulação Analítica sem Restauração A Montante × Histórico . . . . .	p. 32
3.2	Simulação Analítica com Restauração A Montante × Histórico . . . . .	p. 34
3.3	Tempo de execução da Busca Exaustiva . . . . .	p. 37
3.4	Comparação entre as formas de vizinhança . . . . .	p. 42
3.5	<i>Simulated Annealing</i> × Busca Exaustiva: A4, D2 e F6 . . . . .	p. 44
4.1	Dados do alimentador da Figura 4.2 [ST98] . . . . .	p. 51
4.2	Soudi e Tomsovic × Busca Exaustiva: alimentadores A2, C3 e E4 . . . . .	p. 53
4.3	Consumidores acumulados para cada bloco do caso de teste . . . . .	p. 58
4.4	Nova formulação × Soudi e Tomsovic . . . . .	p. 60
4.5	Resultados da nova formulação resolvida pelo Algoritmo 4.1 . . . . .	p. 67
A.1	<i>Simulated Annealing</i> × Busca Exaustiva: subestação A . . . . .	p. 74
A.2	<i>Simulated Annealing</i> × Busca Exaustiva: subestação B . . . . .	p. 75
A.3	<i>Simulated Annealing</i> × Busca Exaustiva: subestação D . . . . .	p. 75
A.4	<i>Simulated Annealing</i> × Busca Exaustiva: subestação C . . . . .	p. 76
A.5	<i>Simulated Annealing</i> × Busca Exaustiva: subestação E . . . . .	p. 77
A.6	<i>Simulated Annealing</i> × Busca Exaustiva: subestação F . . . . .	p. 78

# *Lista de Algoritmos*

3.1	Simulação Analítica . . . . .	p. 30
3.2	Procedimento de Restauração A Montante . . . . .	p. 34
3.3	Busca Exaustiva . . . . .	p. 35
3.4	Forma geral do <i>Simulated Annealing</i> . . . . .	p. 39
3.5	<i>Simulated Annealing</i> adaptado ao problema . . . . .	p. 42
4.1	<i>Branch-and-bound</i> para os problemas de PNLB . . . . .	p. 64

# 1 *Introdução*

Este trabalho, pertencente à área de Pesquisa Operacional, descreve a aplicação de um conjunto de métodos de Otimização a um problema real: determinar os melhores pontos de alocação de equipamentos de proteção em uma rede de distribuição de energia elétrica.

Tal problema é o objeto de investigação do projeto de pesquisa intitulado “Otimização de dispositivos de proteção instalados na rede de 15 kV”, desenvolvido pelo Núcleo de Inferência e Algoritmos (NINFA) do Departamento de Informática (DI) da Universidade Federal do Espírito Santos (UFES) e financiado pela ESCELSA (Espírito Santo Centrais Elétricas S.A.).

## 1.1 *Motivação*

O objetivo principal de uma empresa de distribuição de energia elétrica (concessionária) é atender os seus consumidores de uma forma confiável e a um baixo custo. Além, é claro, de obter lucro com esta atividade. Com a privatização das empresas do setor na década de 1990 pelo governo brasileiro, criou-se um órgão responsável por acompanhar, regular e fiscalizar as atividades das concessionárias elétricas: a Agência Nacional de Energia Elétrica (ANEEL).

Para poder quantificar a qualidade do serviço prestado pelas concessionárias, a ANEEL define os *indicadores de continuidade* [ANE00] de uma rede de distribuição, dentre os quais destacam-se o DEC (Duração Equivalente de Interrupção por Unidade Consumidora) e o FEC (Frequência Equivalente de Interrupção por Unidade Consumidora). Este indicadores de continuidade também são chamados de *índices de confiabilidade* na literatura.

A ANEEL estabelece metas anuais para esse indicadores. Concessionárias que não as atingem ficam sujeitas a sanções, tais como multas. A ESCELSA, até o momento, vem conseguindo respeitar as metas estabelecidas pela ANEEL. No entanto, a cada ano estes valores ficam mais exigentes, obrigando a empresa a um investimento cada vez maior na proteção da rede de distribuição de energia elétrica. Uma vez que o atual planejamento da proteção baseia-se exclusivamente no conhecimento *ad-hoc* adquirido pelo Engenheiro de Proteção, as metas futuras

tendem a se tornar mais difíceis de serem atingidas, geralmente a um custo sempre crescente.

Desta forma, o estudo do planejamento da proteção deve assumir uma postura mais sistemática e científica, permitindo assim o desenvolvimento de ferramentas que auxiliem o Engenheiro de Proteção a decidir onde os dispositivos de proteção devem ser instalados.

## 1.2 Descrição do problema

Uma rede de distribuição de energia elétrica típica possui uma estrutura em árvore, onde o nó raiz da árvore é a fonte de onde parte o fluxo da energia em direção aos nós folhas. A cada nó associa-se uma dada quantidade de consumidores que estão instalados na área geográfica correspondente àquele nó. Uma aresta entre dois nós indica uma ligação (cabearno elétrico) entre eles.

Uma falha em um trecho da rede de distribuição pode levar a uma desconexão (remoção de uma aresta) de um nó, criando uma subárvore formada pelo nó desconectado e todos os seus filhos. Enquanto a aresta removida não for reinsertada (i.e., a falha não for consertada), todos os consumidores associados aos nós pertencentes à subárvore ficarão sem energia elétrica.

Os índices de continuidade definidos pela ANEEL indicam o número médio de interrupções sofridas pelos consumidores (FEC) e a duração média destas ocorrências (DEC). A confiabilidade do fornecimento de energia elétrica é inversamente proporcional aos índices de DEC e FEC. Assim, para melhorar (maximizar) a confiabilidade de uma rede deve-se buscar diminuir (minimizar) os seus indicadores.

Os equipamentos de proteção (religadores e fusíveis) e seccionamento (chaves) instalados em uma rede de distribuição têm as seguintes funções: (1) isolar uma falha e (2) tentar tratá-la, de forma que ela atinja o menor número possível de consumidores e que esses clientes atingidos fiquem o menor tempo possível sem energia. Em cada um dos nós da árvore de distribuição deve haver um equipamento de proteção ou seccionamento instalado. Uma boa alocação destes dispositivos em uma rede tende a diminuir os seus indicadores de continuidade.

Isto posto, pode-se enunciar os objetivos deste trabalho:

1. Minimizar os índices de DEC e FEC de uma rede de distribuição, partindo de uma configuração de equipamentos de proteção já existente e determinando os melhores locais para a alocação e/ou realocação de religadores.
2. Minimizar os índices de DEC e FEC de uma rede de distribuição, sugerindo uma confi-

guração ótima para a alocação de religadores, fusíveis e chaves.

O capítulo 3 mostra como o objetivo 1 foi alcançado. O segundo objetivo é atacado no capítulo 4.

### 1.3 Revisão bibliográfica

No início do projeto, realizou-se uma pesquisa bibliográfica e constatou-se a existência de grande material para estudo: livros, artigos e dissertações de mestrado. Esta seção apresenta breves comentários sobre os textos considerados mais relevantes.

Dias, em sua dissertação de mestrado [Dia02], apresenta uma metodologia capaz de quantificar os impactos de vários tipos de ações operacionais nos indicadores de continuidade. Desta forma, é possível prever como cada uma destas ações afetará a confiabilidade do fornecimento de energia elétrica. Embora o autor apresente a sua metodologia de forma clara e detalhada, nenhuma referência a algoritmos computacionais é feita.

Em [ST97] e [ST98], Soudi e Tomsovic apresentam uma formulação baseada em Programação Linear Binária (PLB) para identificar uma configuração para o tipo e a localização de dispositivos de proteção em uma rede de distribuição, buscando minimizar os índices SAIFI (*System Average Interruption Frequency Index*) ou SAIDI (*System Average Interruption Duration Index*). Estes indicadores são utilizados por concessionárias norte-americanas e correspondem ao FEC e ao DEC, respectivamente. O modelo proposto utiliza heurísticas da Engenharia de Proteção para simplificar o problema e diminuir o tempo necessário para a sua solução. Um artigo subsequente dos mesmos autores [ST99] trata da análise da qualidade das soluções obtidas e da complexidade do algoritmo. Finalmente, em [ST01], Soudi e Tomsovic apresentam uma modelagem que utiliza Programação Orientada a Objetivos (*Goal Programming*) na tentativa de buscar um equilíbrio na otimização simultânea dos índices SAIFI e SAIDI.

Baseando-se no trabalho de Soudi e Tomsovic [ST98], Silva desenvolve um modelo de Programação Não-Linear Binária (PNLB) para o problema e aplica a meta-heurística de Algoritmos Genéticos para tentar encontrar soluções viáveis para o modelo [dS02]. Os resultados obtidos são resumidos em [dSPM04]. Convém ressaltar que a formulação apresentada é bastante complexa e incorpora as mesmas heurísticas aplicadas na original.

Um problema similar, relacionando os diferentes tipos de cargas, configurações da rede de distribuição e tipos de consumidores pode ser vista em [BJ96]. Neste artigo, Billinton e Jonnavithula utilizam a meta-heurística de *Simulated Annealing* para buscar soluções.

Finalmente, em seu livro [Bro02], Brown apresenta técnicas para a otimização da confiabilidade da distribuição de forma estruturada e sistemática. Alguns dos algoritmos mostrados nesse livro foram empregados neste trabalho.

## 1.4 Estrutura da dissertação

O restante deste trabalho está dividido em 4 capítulos:

- O capítulo 2 descreve o problema, apresentando os conceitos, termos e definições utilizadas no restante do documento. A estrutura da massa de dados de teste disponível é mostrada, bem como o tratamento que lhe foi aplicado para se extrair as informações necessárias aos algoritmos descritos nos capítulos seguintes.
- O capítulo 3 trata da otimização da alocação de religadores em um alimentador. Três algoritmos são apresentados: a Simulação Analítica, a Busca Exaustiva e o *Simulated Annealing*. A Simulação Analítica é um método de estimativa de índices (SAIFI e SAIDI) apresentado por Brown [Bro02]. Uma versão adaptada do algoritmo é mostrada na seção 3.2. Esta simulação serve como base para os algoritmos das seções 3.3 e 3.4. A solução força bruta (Busca Exaustiva) aparece na seção 3.3. Uma análise da complexidade do algoritmo e resultados experimentais compatíveis buscam mostrar a dificuldade da sua aplicação devido ao seu grande tempo de execução. A seção 3.4 mostra a adaptação da meta-heurística de *Simulated Annealing* para o problema e faz uma comparação com os resultados obtidos com a Busca Exaustiva.
- O capítulo 4 apresenta modelos de programação matemática para se determinar a configuração ótima para a alocação de religadores, fusíveis e chaves em um alimentador. Na seção 4.1, a modelagem de PLB de Soudi e Tomsovic [ST98] e sua adaptação são mostrados. Os problemas encontrados na utilização deste modelo levaram à criação de uma nova modelagem, mais geral que a de Soudi e Tomsovic e mais simples que a de Silva e Mantovani [dSPM04], como pode ser visto na seção 4.2. O esforço computacional necessário para se resolver os problemas obtidos com o novo modelo de PNLB criado motivou a busca por um algoritmo de *branch-and-bound* específico para as suas características. Este novo algoritmo se encontra na seção 4.3.
- Finalmente, o capítulo 5 apresenta as considerações finais, conclusões e trabalhos futuros.

## 2 *Caracterização do problema*

Este capítulo apresenta os principais conceitos e definições necessárias para o entendimento do problema.

### 2.1 **Características da rede de distribuição**

A energia elétrica produzida em usinas hidroelétricas, termoelétricas e outras fontes passa por um longo caminho até chegar aos consumidores. É papel da rede de transmissão / distribuição garantir que essa energia atinja o seu destino com o mínimo de perdas. Inicialmente, a transmissão é realizada em altíssima tensão ( $> 100$  kV) até a subestação.

**Subestação** é uma estação subsidiária de um sistema de geração, transmissão e distribuição de energia elétrica onde a tensão é abaixada (de  $\sim 100$  kV para  $\sim 15$  kV) por meio de transformadores. Para efeitos do problema aqui tratado, considera-se que a subestação é o ponto inicial de fornecimento da energia.

O entorno geográfico da subestação está coberto por uma rede de distribuição que é formada por um ou mais alimentadores.

**Alimentador** é um sistema de distribuição que leva a energia até os consumidores presentes na região por ele coberta. Ao longo desta rede estão instalados transformadores, responsáveis por abaixar a tensão de distribuição (15 kV) até a tensão esperada pelo cliente (geralmente 110 V para consumidores residenciais).

Este trabalho trata somente de alimentadores radiais, isto é, que possuem uma única fonte de energia (no caso, a subestação). Existem chaves ditas “normalmente abertas” que podem interligar dois pontos de alimentadores distintos e ser utilizadas para realizar *manobras* que redirecionam o fluxo da energia na malha em caso de falhas. Uma vez que a determinação

dos melhores pontos para a instalação destas chaves constitui por si só um problema bastante complexo, neste estudo as chaves normalmente abertas são desconsideradas: assume-se que todos os alimentadores são completamente isolados.

## 2.2 Classificação dos tipos de falha e interrupções

As interrupções e falhas que ocorrem em um sistema de distribuição podem ser separadas em tipos distintos. Essa classificação é mostrada a seguir, juntamente com outras definições importantes.

**Estado Normal de Operação** o estado do sistema de distribuição em que todas as chaves se encontram na sua posição normal, nenhum equipamento de proteção foi acionado e todos os componentes estão operando adequadamente.

**Contingência** um evento imprevisível (tal como uma falha) que leva o sistema de distribuição a deixar o seu estado normal de operação.

**Interrupção Permanente** é uma interrupção no fornecimento de energia elétrica aos consumidores de um alimentador cuja duração seja igual ou superior a 3 minutos.

**Interrupção Temporária** corresponde às interrupções inferiores a 3 minutos.

**Falha de Curto-circuito Permanente** este tipo de contingência gera um fluxo de corrente de falha, levando o sistema de proteção a operar. Requer que uma equipe de reparo seja despachada para realizar o conserto e retornar o sistema ao estado normal de operação. Utiliza-se Falha Permanente para um termo mais curto. Este tipo de falha sempre leva à uma interrupção permanente.

**Falha de Curto-circuito Temporária** este tipo de contingência gera um fluxo de corrente de falha, mas desaparece quando o circuito é desenergizado. Utiliza-se Falha Temporária para um termo mais curto. Este tipo de falha pode causar uma interrupção temporária ou permanente, dependendo do equipamento de proteção que ela tenha acionado.

## 2.3 Equipamentos de proteção / seccionamento

Os equipamentos de proteção e seccionamento estão instalados na subestação e ao longo dos alimentadores. Os dispositivos existentes são divididos em três grandes grupos: religadores, fusíveis e chaves.

**Religador** é um equipamento de proteção automático que desenergiza um trecho do alimentador atingido por uma falha. Desta forma, uma falha temporária é “limpa” pelo religador, causando somente uma interrupção temporária. Após uma certa quantidade de tentativas de reenergização, o religador “trava” (*locks-out*) e deve ser reativado manualmente (ou remotamente caso possua esta funcionalidade). Esta situação surge nas ocorrências de falhas permanentes.

Um religador é um equipamento com um custo elevado, logo os pontos para sua alocação devem ser bem escolhidos. Nos locais onde a instalação de um religador não é viável ou economicamente interessante, pode-se utilizar fusíveis.

**Fusível** é um equipamento de proteção simples e barato. É projetado de forma a “estourar” quando há um fluxo de corrente elétrica superior à sua corrente nominal, interrompendo o fornecimento e obrigando a sua troca. Tanto falhas permanentes quanto temporárias causam uma interrupção permanente ao atingirem um fusível.

**Chave** uma chave não é um dispositivo de proteção e sim de seccionamento. Embora não respondam ativamente a uma contingência, as chaves podem ser utilizadas para isolar trechos com falha de um alimentador. Isso permite que alguns consumidores sejam restaurados mais rapidamente que outros, diminuindo o impacto das falhas no DEC.

## 2.4 Estrutura de um alimentador

Um alimentador é formado por *blocos* que são delimitados por dispositivos de proteção ou seccionamento.

**Bloco** é a unidade de divisão do alimentador, composta pelo cabeamento de distribuição, capacitores e transformadores. Um bloco é identificado pelo número do equipamento que o inicia.

O número de consumidores de um bloco corresponde à quantidade de clientes conectados aos seus transformadores e cabeamento. Os blocos são a menor unidade de interrupções:

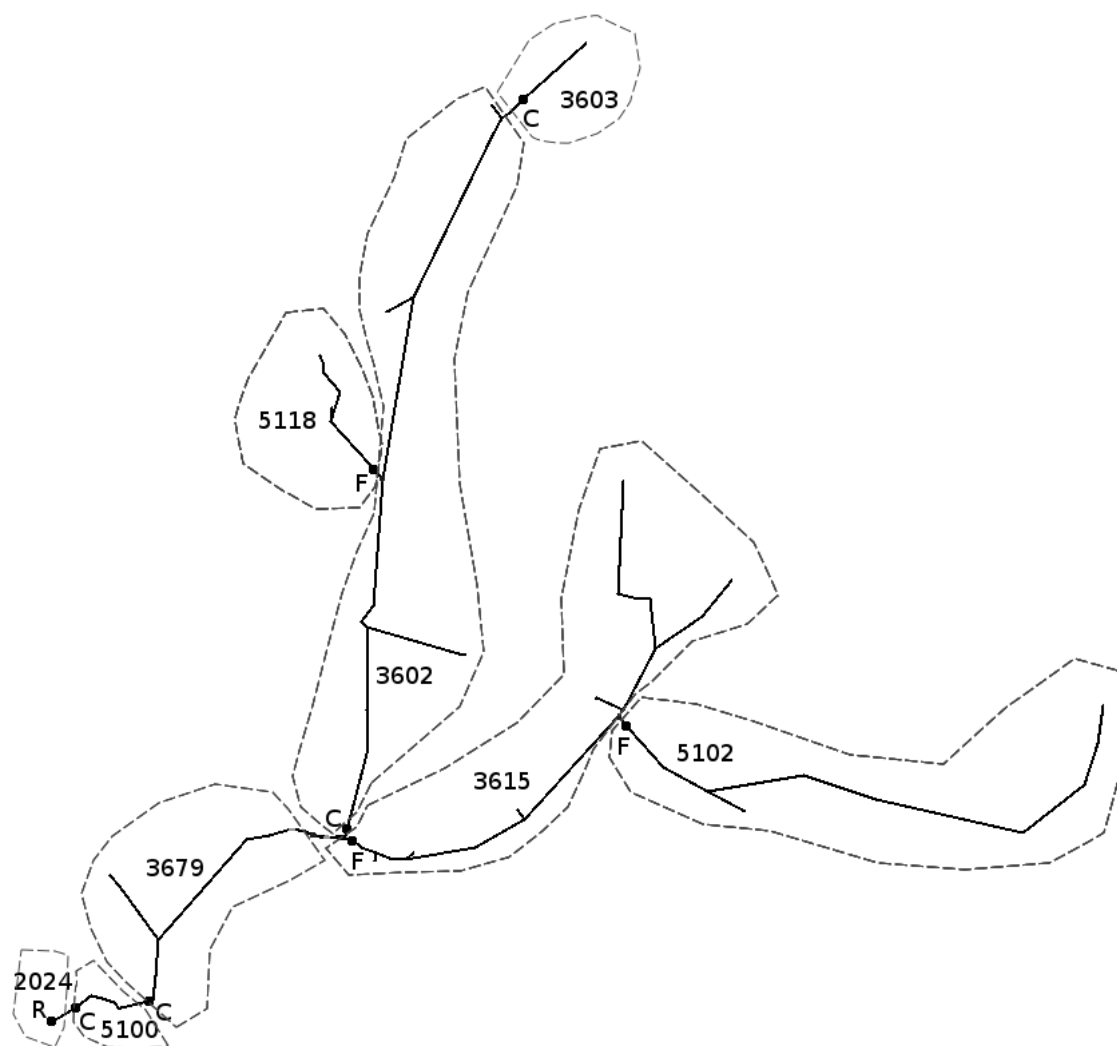


Figura 2.1: Topologia do alimentador C7

assume-se que todos os seus consumidores são desligados quando um bloco sofre uma interrupção. A Figura 2.1 apresenta a topologia de um alimentador real e as áreas correspondentes aos seus oito blocos. As letras R, F e C indicam qual o tipo do equipamento delimitador do bloco: religador, fusível ou chave.

Analizando a estrutura de um alimentador, verifica-se de imediato a existência de uma hierarquia de distribuição em que um bloco possui um pai (exceto o primeiro) de onde chega a energia e zero ou mais filhos para onde ela flui. Tomando como exemplo o bloco 3602 do alimentador C7, vemos que o abastecimento dos seus consumidores está condicionado à não interrupção dos blocos 2024, 5100 e 3679.

Esta forma de estruturação dos alimentadores permite a sua representação como uma árvore, onde cada nó corresponde a um bloco. A Figura 2.2 mostra a árvore do alimentador C7. Círculos, quadrados e losangos representam religadores, fusíveis e chaves, respectivamente.

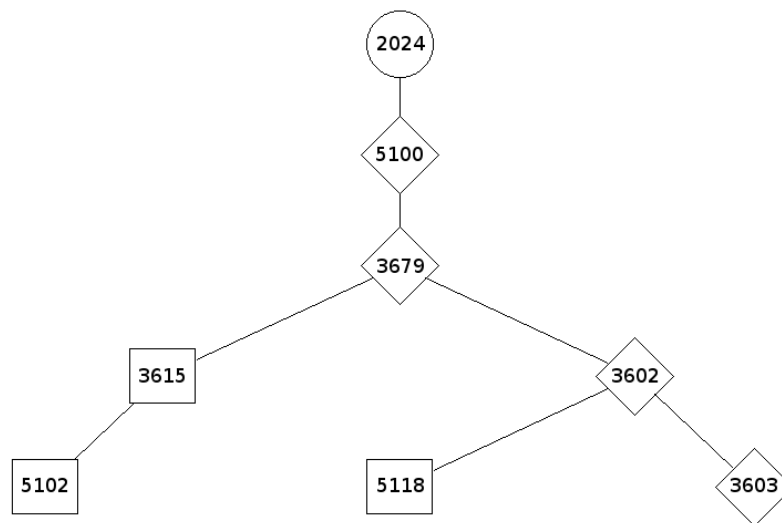


Figura 2.2: Árvore do alimentador C7

*O primeiro bloco de um alimentador sempre possui um religador automático de subestação instalado. Isto garante o isolamento entre os alimentadores de uma subestação.*

## 2.5 Índices de continuidade da rede de distribuição

Os índices de continuidade fornecem uma indicação da qualidade do serviço de fornecimento de energia elétrica prestado pelo concessionária na área onde eles foram apurados.

### 2.5.1 Definição da ANEEL

Na resolução publicada em janeiro de 2000 [ANE00], a ANEEL define os seguintes índices de continuidade:

- **DEC:** Duração Equivalente de Interrupção por Unidade Consumidora.
- **FEC:** Frequência Equivalente de Interrupção por Unidade Consumidora.
- **DIC:** Duração de Interrupção Individual por Unidade Consumidora.
- **DMIC:** Duração Máxima de Interrupção Contínua por Unidade Consumidora.
- **FIC:** Frequência de Interrupção Individual por Unidade Consumidora.

Este trabalho trata dos índices DEC e FEC. Os demais indicadores não foram estudados e portanto não serão mais citados.

A ANEEL define o DEC como

$$DEC = \frac{\sum_{i=1}^k C_a(i) \times t(i)}{C_c} \quad [\text{horas / per. deapur.}] \quad (2.1)$$

e o FEC como

$$FEC = \frac{\sum_{i=1}^k C_a(i)}{C_c} \quad [\text{interrup. / per. deapur.}] \quad (2.2)$$

Onde:

$C_a(i)$  número de consumidores afetados pela interrupção  $i$ , no período de apuração.

$t(i)$  duração da interrupção  $i$ , no período de apuração.

$k$  total de interrupções no período considerado.

$C_c$  número total de consumidores da região considerada.

O período de apuração usual para estes índices é de um ano. Assim, geralmente o DEC é medido em [horas / ano] e o FEC em [interrupções / ano].

### 2.5.2 Parâmetros necessários para a estimativa de índices

As fórmulas para o cálculo de DEC e FEC definidas pela ANEEL são aplicáveis somente ao histórico de falhas da concessionária. Uma vez que neste trabalho busca-se estimar como mudanças nos equipamentos de proteção afetam estes índices, torna-se necessária a utilização de taxas que indiquem uma expectativa para o número de falhas de cada bloco de um alimentador.

**Taxa de Falha de Curto-circuito Permanente ( $\lambda$ )** descreve uma expectativa para o número de vezes por ano (ou por hora) que um bloco venha a experimentar uma falha de curto-circuito permanente.

**Taxa de Falha de Curto-circuito Temporária ( $\gamma$ )** descreve uma expectativa para o número de vezes por ano (ou por hora) que um bloco venha a experimentar uma falha de curto-circuito temporário.

Além do  $\lambda$  e do  $\gamma$ , outros dois parâmetros podem influenciar na estimativa do DEC: MTTR e MTTT.

**Tempo Médio de Reparo - Mean Time to Repair (MTTR)** representa o tempo estimado em horas para se realizar o reparo de uma falha (medido a partir do momento que a falha ocorrer).

Os algoritmos apresentados por Brown em [Bro02] utilizam um único valor de MTTR para todos os blocos de um alimentador. Tal simplificação geralmente é necessária pois não há um histórico de falhas de onde informações mais detalhadas possam ser extraídas. Quando há dados históricos suficientes é possível calcular os valores de MTTR para cada bloco.

**Tempo Médio de Chaveamento - Mean Time to Switch (MTTS)** representa o tempo esperado para a operação de uma chave seccionalizadora depois da ocorrência de uma falha no sistema. Para chaves manuais, este é o tempo necessário para o despacho e deslocamento de uma equipe de manutenção até o local onde a chave se encontra. Para chaves automatizadas este tempo é muito menor, geralmente desprezível.

A qualidade da estimativa dos índices é dependente da precisão destes parâmetros. Em [BO98], Brown e Ochoa apresentam formas de se obter, estimar e validar estes parâmetros, mesmo quando não se possui um histórico de falhas significativo. A seção 2.6.4 detalha como estas informações foram extraídas a partir dos dados disponíveis.

### 2.5.3 Redefinição da forma de cálculo dos índices

A partir dos parâmetros definidos na seção anterior, é definida uma nova forma para o cálculo da estimativa dos índices DEC e FEC.

O DEC de um alimentador pode ser estimado por

$$DEC = \frac{\sum_{i \in B} D_i N_i}{N_T} \quad [\text{horas / ano}] \quad (2.3)$$

e o FEC por

$$FEC = \frac{\sum_{i \in B} F_i N_i}{N_T} \quad [\text{interrup. / ano}] \quad (2.4)$$

Onde:

$B$  o conjunto formado por todos os blocos de um alimentador.

$N_i$  número de consumidores do bloco  $i$ .

$N_T$  número total de consumidores do alimentador.

$D_i$  estimativa do número de horas por ano que o bloco  $i$  fica interrompido.

$F_i$  estimativa do número de interrupções por ano sofridas pelo bloco  $i$ .

Os valores  $D_i$  e  $F_i$  podem ser calculados de diferentes formas a partir dos parâmetros  $\lambda$ ,  $\gamma$ , MTTR e MTTS de cada bloco do alimentador. A seção 3.2 mostra como este cálculo é realizado pela Simulação Analítica.

## 2.6 Estrutura e tratamento dos dados

Esta seção apresenta os dados utilizados nos testes e explica como as informações necessárias aos algoritmos empregados foram extraídas.

### 2.6.1 Descrição da massa de teste: subestações

A Tabela 2.1 mostra as principais características das subestações de teste, listando para cada alimentador que as compõem: a quantidade de blocos, o total de consumidores e o número de religadores já instalados (sem contar o religador do nó raiz).

### 2.6.2 Informações sobre a topologia dos alimentadores

Os dados sobre a topologia dos alimentadores normalmente são armazenados pelas concessionárias em sistemas de geoprocessamento. Embora o formato de armazenamento empregado possa mudar, os dados sobre as coordenadas geográficas dos componentes, comprimentos e tipos de cabeamento e pontos de conexão dos equipamentos, geralmente estão disponíveis. Analisando-se estes dados, é possível construir a representação em árvore de um alimentador, como o exemplo mostrado na Figura 2.2.

Além disso, utilizando os dados sobre os transformadores na rede que indicam a qual bloco um transformador pertence e a quantidade de clientes a ele conectada, é possível se obter o número de consumidores de cada bloco de um alimentador.

### 2.6.3 Histórico de interrupções

A ANEEL determina que toda concessionária deve manter um histórico das interrupções permanentes ocorridas em suas redes de distribuição. Utilizando-se estes registros é possível extrair os parâmetros necessários para a estimativa dos índices definidos na seção 2.5.2.

Obteve-se um histórico de interrupções de cada um dos alimentadores listados na Tabela 2.1, com os seguintes dados relevantes:

Subestação	Alimentador	Blocos	Consumidores	Religadores
A	A1	43	5391	0
	A2	46	1746	0
	A3	133	2581	2
	A4	167	3223	2
B	B1	118	2827	3
	B2	41	371	0
	B3	34	793	0
	B4	59	566	0
C	C1	147	1356	1
	C2	60	670	0
	C3	95	1346	2
	C4	124	1968	1
	C5	107	2198	1
	C6	16	1132	0
	C7	8	47	0
	C8	63	973	1
D	D1	246	4697	2
	D2	136	1232	3
	D3	60	4881	0
	D4	38	495	0
E	E1	37	5895	0
	E2	88	4793	1
	E3	42	4801	2
	E4	70	7770	1
	E5	104	7329	1
	E6	48	5593	1
	E7	143	4641	1
	E8	142	4803	4
	E9	23	3490	0
F	F1	68	1357	0
	F2	98	3250	1
	F3	100	6806	1
	F4	38	5407	0
	F5	28	2681	0
	F6	56	2945	0
	F7	100	4931	2

Tabela 2.1: Características das subestações de teste

- Ano de ocorrência da interrupção.
- Número do bloco onde ocorreu a falha que gerou a interrupção.
- Número de consumidores afetados pela interrupção.
- Duração da interrupção.

Estes dados foram apurados entre os anos de 2000 e 2005 (inclusive).

## 2.6.4 Extração dos parâmetros de estimativa

Esta seção mostra como é feita a extração dos parâmetros de estimativa a partir do histórico de interrupções dos alimentadores.

### 2.6.4.1 Cálculo das taxas de falhas

Silva [dS02] apresenta uma análise estatística afirmando que cerca de 80% das falhas que surgem nas redes de distribuição são de caráter temporário. Esta consideração foi utilizada aqui, conforme detalhado a seguir.

Primeiramente calcula-se a taxa de falhas ( $TF$ ) de cada bloco do alimentador, dividindo-se o número de ocorrências de falhas do bloco no histórico pelo número de anos apurados. Aos blocos sem registro de contingências é atribuída uma  $TF$  igual a 10% da menor taxa de falhas calculada para todo o alimentador. Isto é necessário pois mesmo que não haja nenhum registro de falhas para um bloco no período considerado, não significa que a probabilidade de ocorrência de falhas neste bloco seja zero.

Em seguida, baseando-se no tipo de equipamento instalado no bloco, são calculadas as taxas de falhas permanentes ( $\lambda$ ) e temporárias ( $\gamma$ ), da seguinte forma:

- **Fusível:** em um bloco delimitado por um fusível, tanto falhas permanentes quanto temporárias se transformam em interrupções permanentes. Desta forma, a taxa de falhas obtida do histórico é dividida da forma convencional:

$$\begin{aligned}\lambda &= 20\% \times TF \\ \gamma &= 80\% \times TF\end{aligned}\tag{2.5}$$

- **Religador:** em um bloco delimitado por um religador, somente as falhas permanentes causam interrupções permanentes. Assim, a taxa de falhas calculada a partir do histórico corresponde diretamente ao valor de  $\lambda$ , já que falhas temporárias são limpas pelo religador e não aparecem no histórico. Os valores de  $\lambda$  e  $\gamma$  são calculados desta forma:

$$\begin{aligned}\lambda &= TF \\ \gamma &= 4 \times TF\end{aligned}\tag{2.6}$$

- **Chave:** quando um bloco é delimitado por uma chave, busca-se qual equipamento acima dela atua em caso de falha e realiza-se o cálculo conforme o tipo do equipamento encontrado.

#### 2.6.4.2 Cálculo do MTTR e MTTS

O cálculo do MTTR é similar ao realizado para as taxas de falhas dos blocos. Para cada ano do histórico, calcula-se o MTTR anual somando-se a duração de todas as interrupções de um bloco e dividindo-se pelo número de ocorrências. A seguir, faz-se uma média aritmética simples entre os valores de cada ano. Foi verificada a necessidade de se normalizar alguns dos valores obtidos, para se evitar o surgimento de blocos com MTTRs extremos (altos ou baixos demais), causados por dados incompletos ou por uma exceção de um conserto muito demorado. Esta normalização é feita a partir do cálculo da média dos MTTRs dos blocos e do seu desvio padrão, determinando um valor mínimo e máximo aceitável.

O valor do MTTS não é calculado. Utiliza-se o valor padrão da literatura que é 1 hora para chaves manuais e zero para os demais dispositivos [Bro02].

## 3 *Otimização da alocação de religadores em um alimentador*

Este capítulo apresenta os algoritmos empregados para otimizar a alocação de religadores em um alimentador com uma configuração de equipamentos de proteção definida. O objetivo é minimizar os índices de DEC e FEC de um alimentador, identificando os melhores blocos para a alocação de novos religadores e realocação dos já existentes.

### 3.1 Ambiente de desenvolvimento e testes

Os algoritmos apresentados nesta dissertação foram implementados em uma das seguintes linguagens de programação: Python e C. A primeira foi escolhida pela sua facilidade de uso, a segunda pela sua eficiência. Os testes apresentados a partir deste capítulo foram realizados no seguinte ambiente:

- **Hardware**

- Processador Intel Pentium 4 2.66GHz. FSB 533 MHz. Cache L2 de 512 KB.
- Placa-mãe IBM com chipset Intel 865G.
- 1 GB de memória RAM DDR 333.
- HD 120GB IDE ATA-133 7200 RPM.

- **Software**

- Sistema operacional Debian GNU/Linux versão Etch.
- Kernel Linux versão 2.6.12.5
- GCC versão 4.0.3
- Python versão 2.3.5

## 3.2 Simulação Analítica

A Simulação Analítica é um método para estimar os índices de confiabilidade de um alimentador. Esta simulação assume que a rede de distribuição permanece em seu estado normal de operação a maior parte do tempo e que as contingências são independentes e mutuamente exclusivas. Esta técnica é capaz de modelar características físicas e operacionais detalhadas [Bro02].

Esta forma de simulação analisa as respostas do sistema de proteção às contingências, permitindo determinar o impacto que uma falha exerce sobre cada componente. Este impacto é ponderado pela sua probabilidade de ocorrência ( $\lambda$  ou  $\gamma$ ), resultando em uma estimativa do efeito de uma falha em cada componente. Os valores de  $D$  e  $F$  (seção 2.5.3) são obtidos acumulando-se as contribuições individuais de cada contingência.

O restante desta seção apresenta um algoritmo de Simulação Analítica para redes de distribuição radiais proposto por Brown [Bro02].

### 3.2.1 Algoritmo para estimativa de índices de confiabilidade

Para se realizar uma estimativa dos índices DEC e FEC é necessário considerar a estrutura do alimentador. Assim, inicialmente, deve-se construir a árvore do alimentador, na qual cada nó da árvore corresponde a um bloco e contém as seguintes informações:

- Tipo e identificação do equipamento instalado no bloco.
- Número de consumidores do bloco ( $N$ ).
- Taxa de falhas permanentes ( $\lambda$ ).
- Taxa de falhas temporárias ( $\gamma$ ).
- Tempo médio de reparo (MTTR).
- Tempo médio de chaveamento (MTTS).

Esta representação em árvore é a entrada do algoritmo de Simulação Analítica, adaptado de Brown [Bro02] e mostrado no Algoritmo 3.1. Os índices DEC e FEC obtidos como saída do algoritmo têm um período de apuração correspondente à unidade das taxas  $\lambda$  e  $\gamma$ . Assim, se estas taxas estão indicadas em [falhas / ano], os índices de DEC e FEC calculados correspondem a [horas / ano] e [interrupções / ano], respectivamente.

## Algoritmo 3.1: Simulação Analítica

```

1  {Entrada: a árvore de um alimentador com os parâmetros  $N$ ,  $\lambda$ ,  $\gamma$ ,  $MTTR$  e  $MTTS$ 
2     definidos para todos os blocos}
3
4  {Variáveis:
5     -  $D_i$  = estimativa do número de horas por ano que o bloco  $i$  fica interrompido
6     -  $F_i$  = estimativa do número de interrupções por ano sofridas pelo bloco  $i$ 
7     -  $t_b$  = taxa de falha (permanente ou temporária) do bloco  $b$ }
8
9  {Saída: estimativa dos índices DEC e FEC para a configuração do alimentador
10     de entrada}
11
12 { $B$  é o conjunto formado por todos os blocos do alimentador}
13  $\forall i \in B: F_i \leftarrow 0$ 
14  $\forall i \in B: D_i \leftarrow 0$ 
15 Para todo  $b \in B$  faça
16     Gere uma falha temporária e uma permanente em  $b$ .
17     Seja  $p$  o equipamento de proteção a montante mais próximo da falha.
18     Se a falha é permanente  $\vee$  (a falha é temporária  $\wedge p$  é um fusível) então
19         {Ocorreu uma interrupção permanente}
20         Se a falha é permanente então
21              $t_b \leftarrow \lambda_b$ 
22         Senão
23              $t_b \leftarrow \gamma_b$ 
24         Fim se
25         Para todo bloco  $i$  a jusante do bloco definido por  $p$  (inclusive) faça
26              $F_i \leftarrow F_i + t_b$ 
27              $D_i \leftarrow D_i + t_b \times MTTR_b$ 
28         Fim faça
29         Execute o procedimento de Restauração A Montante {se for o caso}
30     Fim se
31 Fim faça
32 Calcule os valores de DEC e FEC utilizando as equações (2.3) e (2.4)

```

A seguir duas definições importantes para o entendimento do algoritmo são apresentadas.

**A montante** Em direção à fonte de energia (raiz da árvore). Uma busca a montante parte de um nó e segue pelos pais subsequentes até que um critério de parada é atingido ou a busca chegue

à raiz.

**A jusante** Se afastando da fonte de energia. O conjunto de nós a jusante do bloco  $b$  pode ser identificado realizando-se uma busca em largura ou em profundidade na árvore, a partir de  $b$ .

O algoritmo simula uma falha permanente (ponderada pelo  $\lambda$ ) e uma temporária (pelo  $\gamma$ ) para cada bloco do alimentador. Seja  $b$  o bloco onde uma contingência está sendo simulada. Para se determinar o equipamento de proteção  $p$  que atua sobre a falha, basta realizar uma busca a montante de  $b$ . Quando  $b$  é definido por um religador ou um fusível,  $p$  e  $b$  correspondem ao mesmo nó.

A Restauração A Montante (seção 3.2.4), quando empregada, simula a utilização de chaves para isolar falhas e restaurar o maior número possível de consumidores interrompidos a montante do bloco faltoso. Sua aplicação tende a diminuir o DEC do alimentador mas não afeta o FEC.

O Algoritmo 3.1 pode ser facilmente modificado para contabilizar outros aspectos que venham a ser considerados relevantes. Por exemplo, caso se tenha interesse em investigar o número de interrupções temporárias ocorridas em um alimentador basta incluir um procedimento que contabiliza as falhas temporárias limpas por religadores.

### 3.2.2 Análise de complexidade da Simulação Analítica

Uma vez que a Busca Exaustiva utiliza a Simulação Analítica, para se calcular a complexidade da primeira precisa-se da complexidade da segunda. Esta seção apresenta a análise de pior caso para a simulação.

Seja  $|B|$  a cardinalidade do conjunto  $B$ . Para um alimentador com  $n$  blocos tem-se então  $|B| = n$ . O laço definido na linha 15 é executado  $2n$  vezes, já que são geradas 2 duas falhas para cada bloco (linha 16). O pior caso do algoritmo ocorre quando um alimentador for uma árvore degenerada de altura  $n$  com um único fusível na raiz e chaves em todos os outros blocos. Nesta situação,  $p$  será sempre a raiz e o loop definido na linha 25 sempre será executado  $n$  vezes. Tomando  $T_{SA}$  como o tempo de execução da Simulação Analítica no pior caso, tem-se:

$$T_{SA}(n) = 2n[(0 + 1 + 2 + 3 + \dots + n) + n]. \quad (3.1)$$

A progressão aritmética entre parênteses corresponde ao número de nós visitados pela busca a montante realizada para localizar o equipamento  $p$  (linha 17). Simplificando a equação (3.1),

vem

$$T_{SA}(n) = 2n \left[ \left( \frac{n(n+1)}{2} \right) + n \right] \quad (3.2)$$

e finalmente

$$T_{SA}(n) = n^3 + 3n^2 \in O(n^3). \quad (3.3)$$

A situação em que um fusível é instalado na raiz do alimentador não ocorre na prática mas é suficiente para a análise de pior caso pois fornece a estimativa mais pessimista possível para o tempo de execução do algoritmo. A Restauração A Montante não foi considerada na análise de pior caso mas tende a aumentar o tempo de execução do algoritmo pois exige que alguns blocos da árvore sejam visitados mais vezes.

### 3.2.3 Comparação entre índices históricos e estimados

Para verificar se os valores estimados pela Simulação Analítica são coerentes, realizou-se uma comparação entre os índices DEC e FEC históricos e simulados para todos os alimentadores da massa de teste.

A Tabela 3.1 mostra esta comparação para a subestação E. A primeira coluna da tabela informa o nome do alimentador. As colunas  $DEC_H$  e  $FEC_H$  correspondem aos índices calculados a partir do histórico de falhas do alimentador, utilizando-se as equações (2.1) e (2.2). Os valores de  $DEC_S$  e  $FEC_S$  foram obtidos com a Simulação Analítica sem Restauração A Montante. Idealmente, busca-se uma razão entre os valores simulados e históricos igual a 1.0. No entanto, uma vez que a Simulação Analítica não contempla todas as possíveis situações do mundo real, as diferenças vistas nas comparações são compreensíveis e foram consideradas aceitáveis. Nos testes realizados para as outras subestações obteve-se resultados semelhantes.

É interessante notar que na maioria dos casos os valores simulados são maiores que os

Alim.	$DEC_H$	$DEC_S$	Razão (S/H)	$FEC_H$	$FEC_S$	Razão (S/H)
E1	4.20	8.94	2.128	1.52	3.65	2.391
E2	18.68	28.22	1.510	10.05	14.12	1.404
E3	1.98	6.39	3.231	1.48	3.46	2.338
E4	3.65	10.16	2.779	2.21	5.44	2.462
E5	13.13	12.57	0.957	3.87	6.35	1.639
E6	4.26	6.36	1.492	2.45	2.96	1.210
E7	16.19	22.71	1.402	5.62	10.04	1.784
E8	21.04	19.39	0.921	5.23	6.29	1.202
E9	3.82	6.86	1.796	1.81	2.78	1.532

Tabela 3.1: Simulação Analítica sem Restauração A Montante  $\times$  Histórico

históricos. A razão (S/H) média entre DEC e FEC para os alimentadores da subestação E é 1.80 com extremos variando entre 0.92 e 3.23. Neste problema, superestimar um índice é menos grave que subestimá-lo. A primeira situação pode forçar a utilização de um número de equipamentos de proteção maior do que realmente necessário. Já a segunda pode levar ao não cumprimento das metas da ANEEL.

### 3.2.4 Restauração A Montante

Para tentar melhorar (aproximar para o valor 1.0) a razão entre o DEC simulado e histórico, a Restauração A Montante pode ser utilizada. Quando uma falha permanente ocorre em um bloco  $b$  definido por uma chave e leva um religador  $p$  a montante a travar, todos os consumidores a jusante de  $p$  sofrem uma interrupção permanente (veja a Figura 3.1a). Nestes casos, pode ser interessante isolar a falha abrindo a chave  $b$ , permitindo que o religador  $p$  seja fechado e o abastecimento aos clientes entre  $p$  e  $b$  seja reestabelecido mais rapidamente, como ilustra a Figura 3.1b.

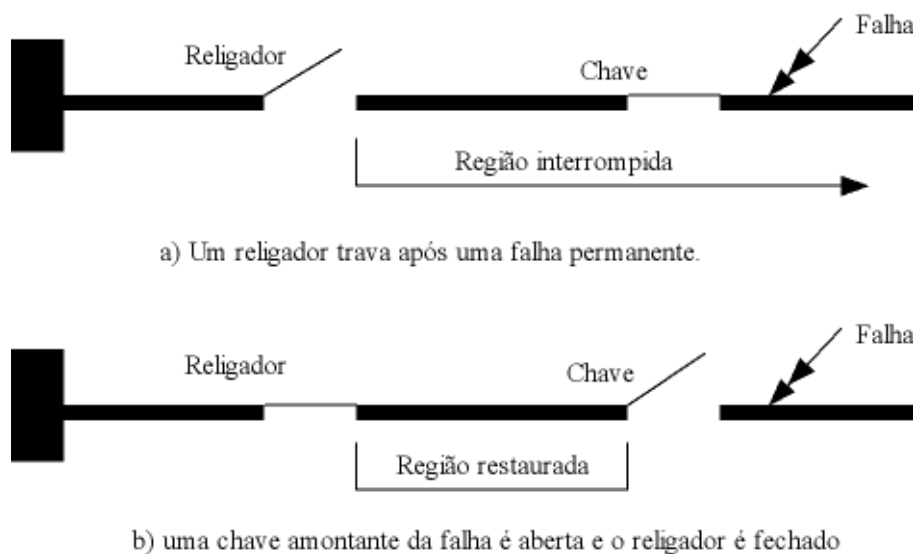


Figura 3.1: Restauração A Montante

O procedimento de Restauração A Montante pode ser visto no Algoritmo 3.2. Quando o tempo estimado para se abrir a chave do bloco  $b$  e fechar o religador  $p$  ( $MTTS_b$ ) for maior que a expectativa de duração do conserto ( $MTTR_b$ ) não faz sentido realizar a Restauração A Montante, como mostram as linhas 13 – 15. Quando o chaveamento é de fato realizado, os consumidores da região restaurada passam a experimentar uma interrupção cuja duração corresponde somente ao tempo  $MTTS_b$ . Como a Simulação Analítica já incrementou o valor  $D$  de  $MTTR_b$  para todos os blocos afetados pela falha, a Restauração A Montante deve subtrair a

diferença ( $MTTR_b - MTTS_b$ ) dos consumidores restaurados (*loop* definido na linha 17).

Algoritmo 3.2: Procedimento de Restauração A Montante

```

1  {Entrada :
2  - a árvore de um alimentador
3  - b = bloco onde ocorreu a falha
4  - p = equipamento de proteção que operou }
5
6  {Saída: os campos D dos blocos restaurados são modificados}
7
8  Se b = p então
9      {A falha ocorreu no mesmo bloco do equipamento de proteção , nada a fazer}
10     retorne
11 Senão
12     {b é definido por uma chave}
13     Se  $MTTS_b > MTTR_b$  então
14         {É mais rápido consertar a falha do que chavear}
15         retorne
16     Senão
17         Para todo bloco i entre p (inclusive) e b (exclusive) faça
18              $D_i \leftarrow D_i - \lambda_b(MTTR_b - MTTS_b)$ 
19         Fim faça
20     Fim se
21 Fim se

```

A mesma comparação da seção 3.2.3 pode ser vista na Tabela 3.2, mas com o procedimento de Restauração A Montante incorporado à Simulação Analítica. Analisando-se as colunas  $DEC_S$  das duas tabelas verifica-se que em 7 dos 9 alimentadores testados a razão entre os

Alim.	$DEC_H$	$DEC_S$	Razão (S/H)	$FEC_H$	$FEC_S$	Razão (S/H)
E1	4.20	6.75	1.606	1.52	3.65	2.391
E2	18.68	24.80	1.327	10.05	14.12	1.404
E3	1.98	4.19	2.118	1.48	3.46	2.338
E4	3.65	6.77	1.852	2.21	5.44	2.462
E5	13.13	8.85	0.674	3.87	6.35	1.639
E6	4.26	5.86	1.376	2.45	2.96	1.210
E7	16.19	15.41	0.951	5.62	10.04	1.784
E8	21.04	10.84	0.515	5.23	6.29	1.202
E9	3.82	5.53	1.449	1.81	2.78	1.532

Tabela 3.2: Simulação Analítica com Restauração A Montante × Histórico

índices simulados e históricos ficou mais próxima a 1.0 (Tabela 3.2). No entanto, os alimentadores E5 e E8 que apresentavam uma boa razão na Tabela 3.1 passaram a ter um valor de DEC simulado muito menor que o histórico com a utilização da Restauração A Montante. Com isto, verifica-se que nem sempre este procedimento melhora a estimativa do DEC, devendo ser empregado somente quando contribui positivamente na simulação. Foram realizados testes nas outras subestações e obteve-se resultados semelhantes.

## 3.3 Busca Exaustiva

Esta seção apresenta o primeiro algoritmo utilizado para se chegar ao objetivo deste capítulo: a Busca Exaustiva.

### 3.3.1 Descrição do algoritmo

Por se tratar de um método de solução força bruta, a idéia geral do algoritmo da Busca Exaustiva é bastante simples, como pode ser visto no Algoritmo 3.3. A árvore de entrada é a mesma definida para a Simulação Analítica e o valor de  $r$  corresponde ao número de religadores que se deseja alocar no alimentador.

Para se explorar todas as configurações possíveis para a alocação dos  $r$  religadores nos blocos do alimentador (definidos pelo conjunto  $B$  – linha 21 do Algoritmo 3.3), deve-se enumerar o conjunto  $C$  (linha 22), formado por todas as combinações  $r$  a  $r$  dos blocos de  $B$ .

A cada passo do *loop* da linha 23 uma Simulação Analítica é realizada sobre a nova árvore do alimentador (com os  $r$  religadores alocados nos blocos indicados por  $c$ ), para estimar os índices desta configuração. Ao final do algoritmo, as variáveis  $DEC^*$  e  $FEC^*$  correspondem ao DEC e FEC ótimos para o alimentador. É importante notar que os conjuntos dos pontos de alocação de religadores para se obter estes valores ótimos ( $R_{DEC^*}$  e  $R_{FEC^*}$ ) podem ser diferentes. Isto ocorre porque as otimizações do DEC e do FEC são objetivos potencialmente conflitantes: os melhores pontos de alocação para o FEC não necessariamente correspondem aos melhores pontos para o DEC (e vice-versa). Nestas situações, pode-se buscar minimizar os índices simultaneamente de forma ponderada, como mostrado na seção 3.4.1.

Quando um alimentador já possui um certo número  $r_i$  de religadores instalados, estes podem ser realocados retirando-os da árvore e realizando uma Busca Exaustiva com  $r = r_i$ . Combinações de realocação e alocações de novos dispositivos também são possíveis.

## Algoritmo 3.3: Busca Exaustiva

```

1  {Entrada :
2  - a árvore de um alimentador com os parâmetros  $N$ ,  $\lambda$ ,  $\gamma$ ,  $MTTR$  e  $MTTS$ 
3    definidos para todos os blocos
4  -  $r$  = o número de religadores a serem instalados
5
6  {Variáveis :
7  -  $DEC^*$  = o melhor  $DEC$  obtido até o momento
8  -  $R_{DEC^*}$  = conjunto dos blocos onde os religadores devem ser instalados para
9    se obter o  $DEC^*$ 
10 -  $FEC^*$  = o melhor  $FEC$  obtido até o momento
11 -  $R_{FEC^*}$  = conjunto dos blocos onde os religadores devem ser instalados para
12   se obter o  $FEC^*$ }
13
14 {Saída: os valores ótimos de  $DEC$  e  $FEC$  e os blocos onde os religadores devem
15   ser instalados}
16
17  $DEC^* \leftarrow \infty$ 
18  $R_{DEC^*} \leftarrow \emptyset$ 
19  $FEC^* \leftarrow \infty$ 
20  $R_{FEC^*} \leftarrow \emptyset$ 
21 { $B$  é o conjunto formado por todos os blocos do alimentador}
22 { $C$  é o conjunto de todas as combinações  $r$  a  $r$  dos blocos de  $B$ }
23 Para todo  $c \in C$  faça
24     Substitua os equipamentos dos  $r$  blocos de  $c$  por religadores
25     Execute uma Simulação Analítica nesta nova configuração da árvore
26     { $DEC_S$  e  $FEC_S$  são os resultados da simulação}
27     Se  $DEC_S < DEC^*$  então
28          $DEC^* \leftarrow DEC_S$ 
29          $R_{DEC^*} \leftarrow c$ 
30     Fim se
31     Se  $FEC_S < FEC^*$  então
32          $FEC^* \leftarrow FEC_S$ 
33          $R_{FEC^*} \leftarrow c$ 
34     Fim se
35     Restaure a configuração original do alimentador
36 Fim faça

```

### 3.3.2 Análise de complexidade da Busca Exaustiva

O algoritmo da Busca Exaustiva é exponencial pois para  $|B| = n$ , temos  $|C| = \binom{n}{r}$ , correspondendo ao número de vezes que o laço definido na linha 23 do algoritmo é executado. Para alocar os  $r$  religadores na árvore (linha 24), pode ser necessário percorrer todo o alimentador, o que implica visitar  $n$  nós. O mesmo vale para o processo inverso da linha 35. Assim, tomando  $T_{BE}(n, r)$  como o tempo de execução da Busca Exaustiva, tem-se:

$$T_{BE}(n, r) = \binom{n}{r} (n + T_{SA}(n) + n). \quad (3.4)$$

Onde  $T_{SA}(n)$  é o tempo de execução de uma Simulação Analítica, apresentado na equação (3.3). Expandindo o binômio, substituindo  $T_{SA}(n)$  e rearrumando os termos, obtém-se

$$T_{BE}(n, r) = \left( \frac{n!}{r!(n-r)!} \right) (n^3 + 3n^2 + 2n) \in \Theta(n!) \quad (3.5)$$

logo,

$$T_{BE}(n, r) \in \omega(2^n). \quad (3.6)$$

Verifica-se, como esperado, que a Busca Exaustiva só possui um tempo de execução aceitável quando aplicada a alimentadores com poucos blocos ou quando  $r$  é muito pequeno.

### 3.3.3 Resultados experimentais

Foi realizada uma Busca Exaustiva em todos os alimentadores da massa de teste, com  $r$  variando de 1 a 4. A Tabela 3.3 mostra os tempos de execução em segundos do algoritmo para os alimentadores C5 e F2. O restante dos testes com os outros alimentadores apresentam a mesma tendência de crescimento exponencial.

Alimentador	$r$				
	1	2	3	4	5
C5 $n = 107$	0.027	1.318	45.516	1167.017	25542.431
F2 $n = 98$	0.028	1.311	40.651	1020.111	17308.252

Tabela 3.3: Tempo de execução da Busca Exaustiva

Apesar do tempo de execução da Busca Exaustiva ser alto na maioria dos casos, os resultados obtidos com este algoritmo são importantes, pois permitem que a qualidade das soluções obtidas com outros métodos seja comparada com os valores ótimos. A seção seguinte apresenta a aplicação da meta-heurística *Simulated Annealing* ao problema, buscando diminuir o esforço computacional necessário para se obter uma solução.

## 3.4 *Simulated Annealing*

Acredita-se que o problema aqui tratado pertença à classe de complexidade NP-hard, embora isto não seja provado nesta dissertação. Esta expectativa e o fato da Busca Exaustiva ser um algoritmo exponencial, motivaram a decisão de se tentar utilizar meta-heurísticas como forma de solução.

Dentre as meta-heurísticas mais comuns foi escolhida a de *Simulated Annealing* por se tratar de um algoritmo de simples implementação, empregado em variados problemas de otimização e que normalmente apresenta bons resultados. Além disso, esta meta-heurística já foi utilizada com sucesso em um problema similar ao tratado neste trabalho [BJ96]. A princípio, planejava-se realizar uma comparação com outras meta-heurísticas tais como Algoritmos Genéticos e Busca Tabu. No entanto, como os resultados obtidos com o *Simulated Annealing* ficaram muito próximos aos da Busca Exaustiva, esta comparação foi considerada desnecessária.

### 3.4.1 Idéia geral do algoritmo

O *Simulated Annealing* é uma meta-heurística proposta por Kirkpatrick, Gellat e Vecchi em 1983 [KGV83]. Ela foi inicialmente utilizada para encontrar soluções para o problema do caixeiro viajante (*Traveling Salesman Problem* – TSP) e depois empregada em muitos outros problemas. O *Simulated Annealing* é um algoritmo de busca de soluções capaz de escapar de ótimos locais. Seu nome vem de uma analogia com o processo de resfriamento de sólidos, em que um material é aquecido e então deixado esfriar lentamente até atingir uma certa temperatura final. A meta-heurística baseia-se nesta idéia para direcionar a busca pelo ótimo global em problemas de otimização. Mais detalhes sobre a história e desenvolvimento do *Simulated Annealing*, bem como resultados sobre a sua convergência pode ser vistos em [GK02].

O Algoritmo 3.4 apresenta a forma geral do *Simulated Annealing* adaptada de [Ata98]. A cada iteração do algoritmo, uma nova solução é gerada e comparada com a solução atual. Em caso de melhoria, o novo estado é aceito. Já as soluções piores são aceitas com uma probabilidade que é inversamente proporcional à temperatura atual do sistema ( $T$ ), o que pode levar a busca a escapar de ótimos locais. Uma vez que  $T$  é não-crescente, os piores estados são aceitos com menos frequência à medida que o sistema esfria, até o seu “congelamento”: o momento de parada do algoritmo.

No contexto deste trabalho, o sistema tratado no algoritmo corresponde a um alimentador e um estado ou solução equivale aos pontos de alocação dos  $r$  religadores. A temperatura inicial  $T_0$  e a taxa de resfriamento  $AR$  são parâmetros do algoritmo e devem ser calibradas

adequadamente (seção 3.4.2.1). O estado inicial  $S_0$  é gerado escolhendo-se aleatoriamente  $r$  blocos do alimentador como os locais para alocação dos  $r$  religadores.

Algoritmo 3.4: Forma geral do *Simulated Annealing*

```

1  {Entrada :
2  -  $AR = a$  taxa de resfriamento (Annealing rate).  $AR \in (0,1)$ 
3  -  $T_0 = a$  temperatura inicial.  $T_0 \in (0,1)$ 
4  -  $S_0 = o$  estado inicial do sistema}
5
6  {Variáveis :
7  -  $T =$  temperatura atual
8  -  $S = o$  estado atual do sistema}
9
10 {Saída:  $S$  corresponde à melhor solução encontrada}
11
12  $T \leftarrow T_0$ 
13  $S \leftarrow S_0$ 
14 Enquanto (critério de parada não for satisfeito) faça
15     Enquanto (o número mínimo de estados não for gerado) faça
16         Gere um novo estado  $S'$  na vizinhança de  $S$ 
17         { $E$  é a função de avaliação de um estado}
18          $\Delta E = E(S') - E(S)$ 
19         Se  $\Delta E < 0$  então { $S'$  é um estado melhor, aceite-o}
20              $S \leftarrow S'$ 
21         Senão { $S'$  é um estado pior}
22             Gere uma variável aleatória  $\alpha$ ,  $0 \leq \alpha \leq 1$ 
23             Se  $\alpha \leq e^{(-\Delta E)/T}$  então
24                  $S \leftarrow S'$  {aceite  $S'$  para tentar escapar de um ótimo local}
25             Fim se
26         Fim se
27     Fim faça
28      $T \leftarrow T \times AR$  {a temperatura do sistema diminui}
29 Fim faça

```

O critério de parada do *loop* da linha 14 deve ser escolhido de forma a levar o algoritmo a terminar quando a solução estiver congelada. Este critério foi determinado em conjunto com os testes realizados para a calibração dos parâmetros  $T_0$  e  $AR$ , buscando-se estabelecer um compromisso entre o tempo de execução do algoritmo e a qualidade das soluções encontradas.

Nestes testes, verificou-se que parar após 15 iterações sem melhoria da solução corrente  $S$  era um critério adequado.

A cada temperatura  $T$ , um certo número de soluções deve ser gerado, como definido no laço da linha 15. Para este problema, convencionou-se que o número mínimo de soluções corresponde a  $r$  configurações distintas para a alocação dos  $r$  religadores. Um novo estado  $S'$  pode ser gerado a partir de  $S$  (linha 16) de variadas formas, como mostra a seção 3.4.2.

Dada uma solução proposta pelo *Simulated Annealing*, é preciso avaliá-la e compará-la com a solução atual. Como já dito na seção anterior, as buscas para minimizar o DEC e o FEC podem ser conflitantes. Assim, definiu-se a função de avaliação  $E$  (linha 17) como [Bro02]:

$$E(DEC, FEC) = W_{DEC} \left( \frac{DEC}{DEC_0} \right) + W_{FEC} \left( \frac{FEC}{FEC_0} \right). \quad (3.7)$$

Os pesos  $W_{DEC}$  e  $W_{FEC}$  da equação (3.7) podem ser ajustados de acordo com a importância relativa da redução de cada índice. Logo, para que o algoritmo busque soluções que melhorem os dois índices simultaneamente, quando estes têm a mesma importância, utiliza-se  $W_{DEC} = 0.5$  e  $W_{FEC} = 0.5$ . Para os casos em que se deseja otimizar somente um dos índices, basta tomar o seu peso como 1 e fazer o outro igual a zero. Os valores  $DEC_0$  e  $FEC_0$  são utilizados para normalizar os índices e impedir que a função  $E$  seja dominada por um DEC ou FEC muito grande. Eles são obtidos realizando-se uma Simulação Analítica sobre a configuração inicial do alimentador, após todos os religadores já instalados terem sido retirados. A Simulação Analítica continua sendo utilizada para se estimar os índices de qualquer solução que venha a ser gerada pelo *Simulated Annealing*.

O teste da linha 19 do Algoritmo 3.4 aceita as soluções que melhoram o valor de  $E$  e as linhas 22-25 indicam que a probabilidade de se aceitar uma configuração que piora a função  $E$  é dada por  $e^{(-\Delta E)/T}$ .

### 3.4.2 As diferentes formas de vizinhança

Para se gerar uma nova configuração para alocação de religadores a partir de um estado atual  $S$  deve-se explorar o espaço de soluções em uma dada vizinhança de  $S$ , que pode ser definida de diferentes formas. As quatro formas empregadas neste trabalho são:

- **Randômica:** neste caso, um novo estado é gerado movendo-se um dos  $r$  religadores instalados na árvore para uma nova posição determinada de forma aleatória. Esta forma de vizinhança permite que o algoritmo escape facilmente de mínimos locais mas exige o

uso de valores mais altos para  $T_0$  e  $AR$  para a obtenção de boas soluções.

- **Blocos ordenados por  $\lambda$ :** analisando-se os resultados obtidos com a Busca Exaustiva, verificou-se que geralmente as soluções ótimas indicavam a alocação de religadores nos blocos com as maiores taxas de falhas. Assim, este caso toma como vizinhança de  $S$  os blocos com os maiores valores de  $\lambda$  que ainda não foram visitados.
- **Chaves ordenadas por  $\lambda$ :** este caso é similar ao anterior e baseia-se na tendência de alocação de religadores em blocos onde há chaves instaladas.
- **Vizinhança na árvore:** um religador escolhido arbitrariamente é movido para os seus blocos vizinhos na árvore (pai e filhos) e o melhor resultado obtido passa a ser o estado  $S'$ .

As três primeiras formas de definição de vizinhanças podem levar o algoritmo a dar grandes saltos no espaço de soluções. Desta forma, pode ser necessário realizar uma busca gulosa a partir de  $S$  no final do algoritmo para se tentar chegar a um ótimo local. Esta busca é similar à vizinhança na árvore mas somente aceita novas configurações que melhoram o valor da função de avaliação  $E$ .

### 3.4.2.1 Avaliação do desempenho das vizinhanças

Foram realizados testes em todos os alimentadores com as quatro formas de vizinhança aqui definidas. Nestes testes determinou-se os valores da temperatura inicial  $T_0$  e da taxa de resfriamento  $AR$  que levam às melhores soluções para cada forma de vizinhança empregada. São eles:  $T_0 = 0.6$  e  $AR = 0.996$  para a vizinhança randômica e  $T_0 = 0.4$  e  $AR = 0.99$  para as demais.

Parte dos resultados obtidos pode ser vista na Tabela 3.4. As colunas  $DEC_{SA}$  e  $FEC_{SA}$  correspondem aos melhores índices obtidos pelo *Simulated Annealing* com a vizinhança indicada. Para se chegar aos melhores DEC e FEC, o algoritmo foi sempre executado duas vezes, com os pesos da função  $E$  ajustados para considerar cada índice separadamente. Os campos  $SA/BE$  indicam a razão entre os resultados da heurística e da Busca Exaustiva. Analizando estas razões, verifica-se que não há uma forma de vizinhança que se destaca das demais em todos os casos. Na verdade, conforme o alimentador e o valor de  $r$  testado, pode-se notar grandes diferenças na qualidade das soluções encontradas. O restante dos resultados obtidos são similares aos mostrados na Tabela 3.4.

Alimentador	Vizinhança	$DEC_{SA}$	$SA/BE$	$FEC_{SA}$	$SA/BE$
F3 $r = 4$	Randômica	14.11	1.030	7.74	1.025
	Blocos ordenados por $\lambda$	14.08	1.028	7.88	1.043
	Chaves ordenadas por $\lambda$	13.69	1.000	7.65	1.013
	Vizinhança na árvore	14.73	1.076	8.01	1.061
F7 $r = 1$	Randômica	28.37	1.000	16.03	1.000
	Blocos ordenados por $\lambda$	28.37	1.000	16.03	1.000
	Chaves ordenadas por $\lambda$	28.37	1.000	16.03	1.000
	Vizinhança na árvore	31.60	1.113	19.27	1.202
C8 $r = 2$	Randômica	22.31	1.000	5.18	1.000
	Blocos ordenados por $\lambda$	24.48	1.097	7.55	1.457
	Chaves ordenadas por $\lambda$	23.08	1.034	6.72	1.296
	Vizinhança na árvore	22.61	1.013	5.18	1.000

Tabela 3.4: Comparação entre as formas de vizinhança

### 3.4.3 A versão final do algoritmo

Uma vez que não foi possível identificar as situações que levam ao bom desempenho de uma forma de vizinhança e geram soluções ruins em outras, buscou-se combinar as quatro formas em um único algoritmo, na expectativa de que um bom resultado obtido por um tipo de vizinhança fosse mantido e talvez melhorado pelas demais. Com isso, chega-se à versão final do *Simulated Annealing* adaptado ao problema, mostrada no Algoritmo 3.5.

Foi colocado um laço mais externo (linha 12) que define qual forma de vizinhança deve ser utilizada em cada uma das suas 4 iterações. Inicialmente,  $S$  começa com a solução  $S_0$  e depois se propaga pelas execuções com diferentes vizinhanças.

Esta alteração aumenta o esforço de processamento do algoritmo. No entanto, o tempo de execução do *Simulated Annealing* permaneceu baixo (menos de 1 minuto), mesmo para alimentadores com mais de 100 blocos.

Algoritmo 3.5: *Simulated Annealing* adaptado ao problema

1	{Entrada:
2	– $r =$ o número de religadores a serem instalados
3	– $S_0 =$ estado inicial do alimentador (alocação aleatória dos $r$ religadores)}
4	
5	{Variáveis:
6	– $T =$ temperatura atual
7	– $S =$ configuração atual da árvore}
8	
9	{Saída: a melhor solução obtida a partir do estado $S$ }

```

10
11  $S \leftarrow S_0$ 
12 Para cada forma de vizinhança faça
13     Ajuste  $T$  e  $AR$  conforme a vizinhança utilizada
14     Enquanto não passaram 15 iterações sem melhoria de  $S$  faça
15         Enquanto não forem geradas  $r$  formas de alocação distintas faça
16             Gere um novo estado  $S'$  conforme definição de vizinhança corrente
17              $\{E \text{ é definida na equação (3.7)}\}$ 
18              $\Delta E = E(S') - E(S)$ 
19             Se  $\Delta E < 0$  então
20                  $S \leftarrow S'$ 
21             Senão
22                 Gere uma variável aleatória  $\alpha$ ,  $0 \leq \alpha \leq 1$ 
23                 Se  $\alpha \leq e^{(-\Delta E)/T}$  então
24                      $S \leftarrow S'$ 
25                 Fim se
26             Fim se
27         Fim faça
28          $T \leftarrow T \times AR$ 
29     Fim faça
30     Realize uma busca gulosa a partir de  $S$  para tentar melhorar a solução
31 Fim faça

```

### 3.4.3.1 Resultados

Alguns dos resultados obtidos com o Algoritmo 3.5 podem ser vistos na Tabela 3.5. O restante dos testes realizados está apresentado no Apêndice A. As colunas  $T_{SA}$  e  $T_{BE}$  correspondem ao tempo de execução em segundos do *Simulated Annealing* e da Busca Exaustiva, respectivamente. Analizando-se estes dados verifica-se que a heurística obteve resultados excelentes pois os valores encontrados para os índices DEC e FEC estão sempre muito próximos aos da Busca Exaustiva (que são ótimos) e o tempo  $T_{SA}$  é várias ordens de grandeza menor que  $T_{BE}$ .

Alimentador		$DEC_{SA}$	$SA/BE$	$FEC_{SA}$	$SA/BE$	$T_{SA}$	$T_{BE}$
A4 $n = 167$	$r = 1$	37.01	1.000	14.66	1.000	1.57	0.16
	$r = 2$	35.08	1.000	12.71	1.000	5.72	9.74
	$r = 3$	33.61	1.000	11.53	1.000	19.09	526.12
	$r = 4$	32.61	1.000	10.22	1.025	36.45	22961.21
D2 $n = 136$	$r = 1$	28.61	1.000	13.51	1.000	0.96	0.11
	$r = 2$	25.07	1.000	9.97	1.000	1.66	4.71
	$r = 3$	23.24	1.000	8.14	1.000	2.92	203.52
	$r = 4$	22.35	1.000	7.61	1.050	4.79	6514.49
F6 $n = 56$	$r = 1$	14.13	1.000	6.20	1.000	0.25	0.01
	$r = 2$	13.21	1.000	5.28	1.000	0.71	0.26
	$r = 3$	12.37	1.000	4.63	1.000	0.99	3.79
	$r = 4$	11.71	1.000	4.13	1.043	1.10	46.87

Tabela 3.5: *Simulated Annealing* × Busca Exaustiva: A4, D2 e F6

Com estes resultados, o objetivo de otimizar a alocação de religadores em um alimentador foi alcançado. Resta agora buscar soluções que indiquem configurações para a alocação de todos os equipamentos (religadores, fusíveis e chaves) de um alimentador. Talvez o *Simulated Annealing* também pudesse ter sido empregado nesta tarefa, mas optou-se por métodos de programação matemática, como pode ser visto no próximo capítulo.

## 4 *Otimização da alocação de dispositivos em um alimentador*

Uma vez que o primeiro objetivo deste trabalho foi atingido no capítulo anterior, resta agora buscar um método de solução que encontre uma configuração ótima para a alocação não só de religadores, mas também de fusíveis e chaves. O método investigado utiliza modelos de programação matemática para tal.

Soudi e Tomsovic [ST98] apresentam um modelo de Programação Linear Binária (PLB) para o problema. Esta formulação foi implementada e adaptada nesta pesquisa, como mostra a próxima seção.

### 4.1 A modelagem de PLB de Soudi e Tomsovic

O conteúdo desta seção é um resumo do artigo de Soudi e Tomsovic [ST98]. Estas informações estão aqui apresentadas para auxiliar na contextualização e compreensão das seções subsequentes.

A Programação Linear Inteira busca maximizar ou minimizar uma função objetivo linear sujeita à restrições também lineares, onde todas as variáveis de decisão que as compõem devem possuir valores inteiros. A Programação Linear Binária é um caso particular da Programação Linear Inteira, onde as variáveis de decisão podem assumir apenas os valores 0 ou 1. Os modelos de PLB possuem a seguinte forma geral:

$$\min/\max \sum_{i=1}^m c_i x_i$$

sujeito à

$$\sum_{i=1}^m a_i x_i \leq b_j, \quad \forall j \in \{1, \dots, m\} \quad \text{e} \quad x_i \in \{0, 1\} \quad (4.1)$$

onde  $x_i$  são as  $nm$  variáveis de decisão,  $c_i$  são os coeficientes de custo,  $a_i$  e  $b_j$  são parâmetros que descrevem as restrições e  $m$  é o número de restrições.

Problemas de Programação Não-Linear Binária (PNLB) podem ser reduzidos a um problema de PLB, substituindo-se os termos produto da função objetivo por uma variável  $x_{n+1}$  e adicionando-se as seguintes  $p + 1$  restrições:

$$\begin{aligned} \sum_{i \in P} x_i - x_{n+1} &\leq p - 1 \\ x_{n+1} - x_i &\leq 0 \quad \forall i \in P \\ x_{n+1} &\in \{0, 1\} \end{aligned} \quad (4.2)$$

onde  $P$  é o conjunto dos índices das variáveis de decisão que compõem o termo produto e  $|P| = p$ . Por exemplo, dado o termo  $x_1 x_3 x_5$ , então  $P = \{1, 3, 5\}$  e  $p = 3$ .

### 4.1.1 Formulação

O método proposto por Soudi e Tomsovic usa uma formulação de PLB para identificar o tipo e localização de dispositivos de proteção em um alimentador, buscando minimizar o índice SAIFI (*System Average Interruption Frequency Index*) ou SAIDI (*System Average Interruption Duration Index*). Estes índices correspondem ao FEC e ao DEC, respectivamente.

Os autores consideram que um alimentador é formado por um tronco principal e ramais laterais, divisão comum em concessionárias. Os ramais podem ser classificados em três categorias:

- **Categoria 1:** um ramal lateral curto (geralmente com menos de 3 blocos), onde não é instalado nenhum equipamento de proteção.
- **Categoria 2:** um ramal de extensão moderada com uma demanda de fornecimento pequena, onde somente são instalados fusíveis.
- **Categoria 3:** um ramal lateral que atende a muitos consumidores e/ou possui uma extensão muito grande, onde qualquer tipo de dispositivo de proteção pode ser instalado.

No cálculo do SAIFI e SAIDI, as taxas de falhas e os consumidores dos ramais laterais de categoria 1 podem ser incorporados ao tronco principal. Já o efeito das laterais de categoria 2 nos índices é constante. Além disso, assume-se os seguintes fatos:

- O alimentador é operado de forma radial.
- O número de consumidores é conhecido para cada bloco.

- Não há sobreposição de falhas e todas as interrupções são reparadas antes da próxima falha ocorrer.
- O tronco principal e os ramais laterais do alimentador foram identificados e adequadamente classificados.
- Há um religador de subestação instalado no primeiro bloco do alimentador e não é permitida a instalação de fusíveis no tronco principal.
- As laterais de categoria 2 possuem somente um fusível instalado no ponto de ramificação do tronco principal (*tap*) e nenhum outro dispositivo de proteção nos outros blocos.
- As laterais de categoria 3 possuem um religador ou fusível instalado no *tap* e podem ter outros equipamentos instalados no restante dos blocos que as compõem.

A Figura 4.1 mostra uma representação para o tronco principal ou um ramal de categoria 3 com  $n$  locais possíveis para a instalação de dispositivos de proteção (blocos). Para o tronco principal, um religador de subestação é instalado no bloco 1.

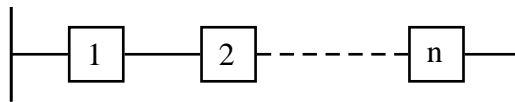


Figura 4.1: Um tronco principal ou ramal lateral com  $n$  blocos [ST98]

O índice SAIFI de um alimentador é definido como

$$SAIFI = \frac{\sum \phi_i N_i}{N_T} \quad (4.3)$$

onde  $\phi_i$  é a taxa de falha do bloco  $i$ ,  $N_i$  é o número de consumidores do bloco  $i$  e  $N_T$  corresponde ao total de consumidores do alimentador. O numerador de (4.3) pode ser dividido nas contribuições do tronco principal e de cada ramal lateral como

$$\sum \phi_i N_i = \sum_{q=1}^{\alpha+\beta+1} A_q \quad (4.4)$$

onde  $\alpha$  é o número de ramais laterais de categoria 3,  $\beta$  é o número de laterais de categoria 2 e o primeiro termo ( $q = 1$ ) é a contribuição do tronco principal. O cálculo que deve ser feito para

cada seção  $q$  é

$$\begin{aligned}
A_q = & \sum_{i=1}^{q_n} (\lambda_{qi} + \gamma_{qi}) \sum_{j=i}^{q_n} N_{qj} - \sum_{i=1}^{q_n} \gamma_{qi} x_{qi2} \sum_{j=i}^{q_n} N_{qj} \\
& + \sum_{i=2}^{q_n} \lambda_{qi} \sum_{j=1}^{i-1} N_{qj} \prod_{k=j+1}^i x_{qk1} x_{qk2} + \sum_{i=2}^{q_n} \gamma_{qi} \sum_{j=1}^{i-1} (1 - x_{qj2}) \sum_{k=j}^{q_n} N_{qk} \prod_{l=j+1}^i x_{ql1} x_{ql2} \quad (4.5)
\end{aligned}$$

onde  $q_n$  é o número de blocos da seção  $q$ ,  $\lambda_{qi}$  e  $\gamma_{qi}$  correspondem às taxas de falhas permanente e temporárias do bloco  $i$  de  $q$ , respectivamente, e  $N_{qj}$  é o número de consumidores do bloco  $j$  de  $q$ , incluindo todas as laterais conectadas à  $j$ . É importante notar que se há um religador no ponto  $qk$ , então a variável de decisão  $x_{qk1} = 0$  e caso contrário  $x_{qk1} = 1$ . Aqui, o subscrito 1 é utilizado para representar um religador e o subscrito 2, um fusível. Estas definições contra intuitivas para as variáveis  $x$  são utilizadas para simplificar a equação (4.5).

A primeira parcela da equação (4.5) representa o impacto que as falhas permanentes e temporárias de um bloco  $qi$  causam nos consumidores a jusante de e em  $qi$  quando não há nenhum dispositivo de proteção instalado. Quando não há um fusível alocado em  $qi$  ( $x_{qi2} = 1$ ) e ocorre uma falha temporária neste bloco, a corrente de curto-circuito se propaga até encontrar o primeiro dispositivo de proteção a montante. Se este equipamento for um religador, a falha é “limpa” e não há interrupção. Deve-se, portanto, subtrair o impacto das falhas temporárias calculado na primeira parcela da equação, o que é feito pelo segundo termo. A terceira parcela contabiliza o efeito de falhas permanentes originadas em  $qi$  nos blocos a montante. O último termo faz o mesmo para falhas temporárias.

Uma vez que no tronco principal um religador deve ser instalado na subestação e não são permitidos fusíveis, a equação (4.5) se reduz a

$$A_q = \sum_{i=1}^{q_n} \lambda_{qi} \sum_{j=i}^{q_n} N_{qj} + \sum_{i=2}^{q_n} \lambda_{qi} \sum_{j=1}^{i-1} N_{qj} \prod_{k=j+1}^i x_{qk1}, \quad q = 1. \quad (4.6)$$

O cálculo para os ramais laterais de categoria 2 também pode ser bastante simplificado pois um fusível é instalado no *tap* e nenhum outro dispositivo de proteção é instalado na lateral. Logo,

$$A_q = \sum_{i=1}^{q_n} (\lambda_{qi} + \gamma_{qi}) \sum_{j=i}^{q_n} N_{qj}, \quad q \in \alpha + 2, \dots, \alpha + \beta + 1. \quad (4.7)$$

Deve-se notar que (4.7) é constante, pois não há variáveis  $x$  em sua formulação. Desta forma, minimizar

$$z = \sum_{q=1}^{\alpha+1} A_q \quad (4.8)$$

é equivalente a minimizar o índice SAIFI.

#### 4.1.1.1 Restrições

As restrições deste problema tratam das limitações de projeto, operacionais e de custo, além das que são introduzidas quando se elimina os termos não-lineares da função objetivo.

Para laterais de categoria 3, um religador ou fusível deve ser instalado no *tap*, então

$$x_{q11} + x_{q12} = 1, \quad q \in \{2, \dots, \alpha + 1\}. \quad (4.9)$$

Para quaisquer outros blocos, somente um dispositivo de proteção pode ser alocado, logo

$$x_{qi1} + x_{qi2} \geq 1, \quad i \in \{2, \dots, q_n\} \quad \text{e} \quad q \in \{2, \dots, \alpha + 1\}. \quad (4.10)$$

Se no bloco  $qi$ , devido a alguma limitação, somente um religador é permitido, então

$$x_{qi2} = 1 \quad (4.11)$$

ou se apenas fusíveis devem ser utilizados, então

$$x_{qi1} = 1 \quad (4.12)$$

deve ser adicionada como uma restrição. Se um fusível é instalado no bloco  $qi$ , os autores assumem que um religador não pode ser alocado em nenhum ponto a jusante do fusível, logo

$$x_{qi2} + x_{q(j+1)1} \geq 1, \quad j \in \{i, \dots, q_n - 1\} \quad \text{e} \quad q \in \{2, \dots, \alpha + 1\}. \quad (4.13)$$

Há um número limitado de religadores que podem ser utilizados, portanto a seguinte restrição também é necessária:

$$\sum_{q=1}^{\alpha+1} \sum_{i=1}^{q_n} x_{qi1} \geq \sum_{q=1}^{\alpha+1} q_n - g_1, \quad (4.14)$$

onde  $g_1$  é o número de religadores disponíveis para alocação no alimentador, incluindo o da subestação.

#### 4.1.2 Um exemplo

Soudi e Tomsovic apresentam um caso de teste para ilustrar a aplicação do modelo por eles desenvolvido. A Figura 4.2 mostra um pequeno alimentador fictício com 7 blocos. As taxas de

falhas permanentes e temporárias e o número de consumidores de cada bloco podem ser vistos na Tabela 4.1.

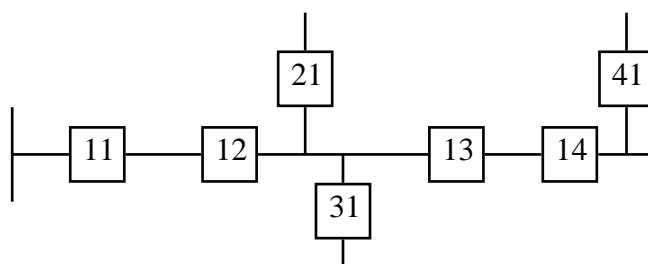


Figura 4.2: Um alimentador radial simples com 7 blocos [ST98]

Neste exemplo, o objetivo é minimizar o índice SAIFI, identificando o tipo e localização dos equipamentos de proteção, seguindo as seguintes restrições:

- Há somente dois religadores disponíveis. De (4.14), vem

$$x_{111} + x_{121} + x_{131} + x_{141} + x_{211} + x_{311} + x_{411} \geq 4. \quad (4.15)$$

- Não é permitido instalar fusíveis no tronco principal. Logo, de (4.11), tem-se 4 restrições:

$$x_{112} = 1 \quad x_{122} = 1 \quad x_{132} = 1 \quad x_{142} = 1. \quad (4.16)$$

- Um religador ou fusível deve ser instalado em cada *tap*. De (4.9), vem

$$x_{211} + x_{212} = 1 \quad x_{311} + x_{312} = 1 \quad x_{411} + x_{412} = 1 \quad (4.17)$$

e para os restante dos blocos, de (4.10), tem-se:

$$x_{111} + x_{112} \geq 1 \quad x_{121} + x_{122} \geq 1 \quad x_{131} + x_{132} \geq 1 \quad x_{141} + x_{142} \geq 1. \quad (4.18)$$

- Um religador de subestação deve ser instalado no bloco 11. Então:

$$x_{111} = 0. \quad (4.19)$$

Considerando que todos os ramais laterais são de categoria 3, a partir das equações (4.5) e (4.6), obtém-se a função objetivo a ser minimizada:

$$225x_{121} + 562.5x_{131} + 250x_{141} + 675x_{121}x_{131} + 500x_{131}x_{141} + 600x_{121}x_{131}x_{141} - 15x_{212} - 100x_{312} - 25x_{412} + 2637.5. \quad (4.20)$$

Bloco	$\lambda_{qi}$ [falhas/ano]	$\gamma_{qi}$ [falhas/ano]	$N_{qi}$ [cons.]
11	1.00	2.00	300
12	0.75	1.75	250
13	2.25	5.50	125
14	2.00	4.75	200
21	0.25	0.75	20
31	1.00	2.00	50
41	0.50	2.50	10

Tabela 4.1: Dados do alimentador da Figura 4.2 [ST98]

Como o alimentador possui 7 blocos, são necessárias 14 variáveis de decisão. Incluindo as variáveis auxiliares  $x_{15}$ ,  $x_{16}$  e  $x_{17}$  para se eliminar os termos não-lineares da função objetivo, chega-se à formulação final de PLB:

$$\begin{aligned} \min \text{SAIFI} = & (225x_{121} + 562.5x_{131} + 250x_{141} + 675x_{15} + 500x_{16} + \\ & 600x_{17} - 15x_{212} - 100x_{312} - 25x_{412} + 2637.5)/875 \end{aligned} \quad (4.21)$$

sujeito à

$$\begin{aligned} x_{111} + x_{121} + x_{131} + x_{141} + x_{211} + x_{311} + x_{411} &\geq 4 \\ x_{111} &= 0 \\ x_{112} = 1 \quad x_{122} = 1 \quad x_{132} = 1 \quad x_{142} = 1 \\ x_{211} + x_{212} = 1 \quad x_{311} + x_{312} = 1 \quad x_{411} + x_{412} = 1 \\ x_{111} + x_{112} \geq 1 \quad x_{121} + x_{122} \geq 1 \quad x_{131} + x_{132} \geq 1 \quad x_{141} + x_{142} \geq 1 \\ x_{15} - x_{121} \leq 0 \quad x_{15} - x_{131} \leq 0 \quad x_{121} + x_{131} - x_{15} \leq 1 \\ x_{16} - x_{131} \leq 0 \quad x_{16} - x_{141} \leq 0 \quad x_{131} + x_{141} - x_{16} \leq 1 \\ x_{17} - x_{121} \leq 0 \quad x_{17} - x_{131} \leq 0 \quad x_{17} - x_{141} \leq 0 \quad x_{121} + x_{131} + x_{141} - x_{17} \leq 2. \end{aligned}$$

A solução ótima para este problema corresponde a um valor de 3.27 interrupções por ano para o índice SAIFI, onde os dois religadores são alocados nos blocos 13 e 14, os *taps* 21, 31, 41 recebem fusíveis, o religador da subestação está localizado no ponto 11 e a posição 12 não recebe nenhum dispositivo de proteção. Nestes blocos onde  $x_{qi1} = 1$  e  $x_{qi2} = 1$  podem ser instaladas chaves.

### 4.1.3 Resultados

Para se aplicar a modelagem apresentada na seção anterior, um alimentador deve ser dividido em um troco principal e ramais laterais de categoria 1, 2 ou 3. No entanto, nem todas as concessionárias utilizam essa classificação e, nesta pesquisa, os alimentadores de teste dispo-

níveis não estão divididos desta forma. Como não há parâmetros quantitativos bem definidos para se realizar esta classificação, torna-se necessário determinar um critério de escolha. Neste trabalho, tomou-se como o tronco principal de um alimentador o caminho entre a raiz da árvore e um nó folha que possui o maior número de chaves. Caso haja mais de um caminho com a mesma quantidade de chaves, o que for mais longo é escolhido como o tronco principal. As sub-árvores que estão conectadas a esse tronco passam a ser os ramos laterais. Todos estes foram classificados como categoria 3 para um maior grau de liberdade na alocação dos dispositivos.

O modelo de Soudi e Tomsovic assume que nenhuma lateral possui ramificação, no entanto este pode não ser o caso das sub-árvores obtidas após a determinação do tronco principal. Nestas situações, cada uma destas laterais é novamente dividida em um sub-tronco e sub-ramais, pelo mesmo critério já apresentado. Este processo é repetido recursivamente nos sub-ramais que ainda possuem ramificações, até que todas as laterais fiquem na forma adequada. Tomando o alimentador da Figura 4.3 como exemplo verifica-se que o tronco principal é formado pelos nós 11, 12, 13, 14 e 15, pois esse é o caminho com o maior número de chaves. Os ramos são as subárvores formadas a partir dos nós 21, 31 e 41. As duas últimas laterais não possuem ramificações, mas a primeira ainda deve ser subdividida. Utilizando o mesmo critério, o sub-tronco principal desta lateral corresponde ao caminho formado pelos nós 21, 22 e 23. O nó 24 é uma sub-lateral do ramo 21.

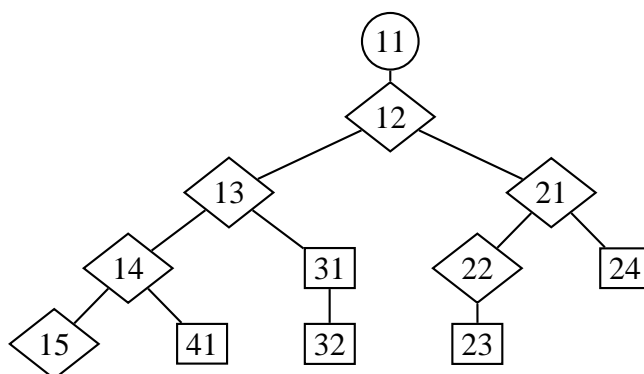


Figura 4.3: Uma árvore de um alimentador fictício

O cálculo para o índice FEC é o mesmo do índice SAIFI apresentado na seção anterior. Já para o DEC, o termo  $A_q$  da equação (4.5) passa a ser definido como

$$\begin{aligned}
 A_q = & \sum_{i=1}^{q_n} (\lambda_{qi} + \gamma_{qi}) t_{qi} \sum_{j=i}^{q_n} N_{qj} - \sum_{i=1}^{q_n} \gamma_{qi} t_{qi} x_{qi2} \sum_{j=i}^{q_n} N_{qj} \\
 & + \sum_{i=2}^{q_n} \lambda_{qi} t_{qi} \sum_{j=1}^{i-1} N_{qj} \prod_{k=j+1}^i x_{qk1} x_{qk2} + \sum_{i=2}^{q_n} \gamma_{qi} t_{qi} \sum_{j=1}^{i-1} (1 - x_{qj2}) \sum_{k=j}^{q_n} N_{qk} \prod_{l=j+1}^i x_{ql1} x_{ql2}
 \end{aligned} \quad (4.22)$$

que é igual ao termo  $A_q$  original, à exceção do valor  $t_{qi}$ , equivalente ao tempo médio de reparo

(MTTR) das falhas do bloco  $qi$ . É interessante notar que as equações (4.5) e (4.22) correspondem diretamente aos cálculos para estimar o DEC e FEC realizados pela Simulação Analítica sem a Restauração A Montante.

Todos os alimentadores da massa de teste foram modelados com este método e os problemas de PLB resolvidos com o pacote GLPK <sup>1</sup> (*GNU Linear Programming Kit*), um software livre com rotinas para a solução de problemas de Programação Linear e Inteira. A Tabela 4.2 apresenta os resultados obtidos para 3 alimentadores e faz uma comparação com a Busca Exaustiva. Os campos  $DEC_{ST}$  e  $FEC_{ST}$  são os índices calculados pela modelagem de Soudi e Tomsovic e os indicadores com subscrito  $BE$  correspondem aos valores da Busca Exaustiva sem Restauração A Montante. Os valores  $ST/BE$  correspondem às razões entre os valores obtidos pelo modelo de programação matemática e a Busca Exaustiva. Os tempos de execução em segundos de ambos os métodos de solução estão apresentados nas duas últimas colunas da tabela. O tempo  $T_{ST}$  é a média aritmética simples das duas execuções realizadas para se otimizar cada um dos índices DEC e FEC pelo modelo de PLB. A variável  $r$  indica o número de religadores disponíveis para alocação.

Alimentador		$DEC_{ST}$	$DEC_{BE}$	$ST/BE$	$FEC_{ST}$	$FEC_{BE}$	$ST/BE$	$T_{ST}$	$T_{BE}$
A2 $n = 46$	$r = 1$	11.73	8.86	1.324	5.20	3.37	1.540	0.07	0.01
	$r = 2$	9.48	6.61	1.434	3.85	2.74	1.408	0.09	0.06
	$r = 3$	7.63	6.15	1.239	2.89	2.41	1.202	0.13	0.92
	$r = 4$	5.99	5.71	1.049	2.25	2.14	1.055	0.14	9.83
C3 $n = 95$	$r = 1$	43.10	37.85	1.138	8.58	8.09	1.061	1.05	0.02
	$r = 2$	29.57	29.67	0.996	6.09	6.12	0.995	1.64	1.01
	$r = 3$	26.42	27.26	0.969	5.58	5.74	0.970	1.37	30.44
	$r = 4$	25.62	26.00	0.985	5.32	5.42	0.981	8.12	683.53
E4 $n = 70$	$r = 1$	6.03	10.12	0.597	3.25	5.43	0.596	3.19	0.02
	$r = 2$	4.88	7.28	0.670	2.75	4.01	0.686	6.19	0.60
	$r = 3$	4.32	4.83	0.893	2.44	2.74	0.890	20.53	12.89
	$r = 4$	4.14	4.34	0.950	2.33	2.48	0.936	25.78	204.74

Tabela 4.2: Soudi e Tomsovic  $\times$  Busca Exaustiva: alimentadores A2, C3 e E4

Analisando-se os dados da tabela verifica-se que há casos em que a razão  $ST/BE$  é menor que 1. Isto se deve ao fato da Busca Exaustiva indicar somente os melhores pontos para a alocação de religadores, mantendo a configuração original do alimentador para fusíveis e chaves, enquanto o modelo de PLB otimiza a utilização de todos estes dispositivos conjuntamente. O melhor ganho relativo  $ST/BE$  foi alcançado no alimentador E4 e o pior resultado em A2. Os resultados do alimentador C3 representam as razões obtidas na maioria dos testes.

É interessante notar que a solução dos problemas de PLB obtidos pela formulação apresen-

<sup>1</sup><http://www.gnu.org/software/glpk/>

tada neste capítulo requer um tempo de execução muito menor que a Busca Exaustiva quando  $r$  cresce. Embora o *Simulated Annealing* também apresente esta característica, somente a otimização da alocação de todos os dispositivos simultaneamente pode obter índices melhores que a Busca Exaustiva. Esta comparação entre os dois métodos de solução é interessante pois permite avaliar o quanto os resultados dos algoritmos que tratam somente do posicionamento de religadores ainda podem ser melhorados.

Os resultados obtidos com os testes desta seção mostram que a formulação de Soudi e Tomsovic pode gerar soluções próximas ou melhores àquelas alcançadas com os algoritmos de Busca Exaustiva e *Simulated Annealing*, com um tempo de execução inferior a 1 minuto, mesmo para alimentadores com mais de 100 blocos. No entanto, há casos em que as soluções ficam aquém do esperado. Isto indica que a modelagem de Soudi e Tomsovic e a sua adaptação não são adequadas para todas as situações. Tal fato motivou o desenvolvimento de um novo modelo de programação matemática, apresentado na próxima seção.

## 4.2 Uma nova modelagem PNLB para o problema

Como visto na seção anterior, o modelo de PLB desenvolvido por Soudi e Tomsovic apresenta resultados bons para alguns alimentadores e ruins para outros. A suspeita de que certas heurísticas adotadas pelos autores (tal como a obrigatoriedade da instalação equipamentos de proteção em todos os *taps*) não são adequadas a todos os casos estudados motivou o desenvolvimento de uma nova modelagem. Esta formulação, mais geral, analisa o alimentador como um todo e permite diminuir ou até mesmo evitar o uso destas heurísticas.

### 4.2.1 Apresentação

Em [dS02], Silva apresenta um modelo de PNLB baseado na formulação original de Soudi e Tomsovic que contabiliza o impacto de falhas dos ramais laterais no tronco principal. Isto permite eliminar a restrição que exige a alocação de dispositivos de proteção no *tap*. No entanto, a modelagem de Silva é bastante complexa e ainda requer que o alimentador esteja dividido em tronco principal e ramais laterais. Por isso, buscou-se criar um novo modelo, mostrado a seguir.

Para simplificar o desenvolvimento e facilitar a compreensão da modelagem, utiliza-se as variáveis “fantasmas”  $X$ ,  $Y$  e  $Z$ . Elas são chamadas desta forma porque são substituídas adiante pelas variáveis de decisão  $x$  e  $y$ , não aparecendo na formulação final. A variável binária  $X_j$

indica se há ou não um religador instalado no bloco  $j$  e é definida da seguinte forma:

$$X_j = \begin{cases} 0 & \text{se não há um religador alocado no bloco } j, \\ 1 & \text{caso contrário.} \end{cases} \quad (4.23)$$

De forma similar, a variável  $Y_j$  indica a alocação de um fusível no bloco  $j$ :

$$Y_j = \begin{cases} 0 & \text{se não há um fusível alocado no bloco } j, \\ 1 & \text{caso contrário.} \end{cases} \quad (4.24)$$

Considera-se também as variáveis complementares  $\bar{X}_j$  e  $\bar{Y}_j$ , que sempre assumem o valor inverso de  $X_j$  e  $Y_j$ . Além disso, também é necessária uma outra variável binária  $Z_j$  que assume o valor 1 quando qualquer equipamento de proteção estiver instalado em  $j$ . Uma vez que não é possível colocar simultaneamente um religador e um fusível em um mesmo bloco,  $Z_j$  e seu complemento  $\bar{Z}_j$  podem ser definidos como:

$$Z_j = X_j + Y_j \quad \bar{Z}_j = \bar{X}_j \bar{Y}_j. \quad (4.25)$$

A equação (4.25) obriga a existência de uma restrição impedindo que  $X_j$  e  $Y_j$  sejam iguais a 1 para um mesmo bloco  $j$ .

Antes de se empregar o novo modelo, deve-se especificar qual é o primeiro dispositivo de proteção  $d_i$  a montante de cada bloco  $i$  do alimentador. Este valor  $d_i$  não é uma variável de decisão, sendo utilizado apenas como um separador de sub-árvores de um alimentador. Isto permite que a árvore seja dividida de forma arbitrária quando se sabe que um ou mais blocos terão um equipamento de proteção instalado. Para se considerar o alimentador por inteiro,  $d_i$  deve ser igual à raiz da árvore, para todo  $i \in B$ . Assume-se que sempre há um religador de subestação instalado na raiz.

Partindo-se da equação (2.4) para a estimativa do FEC, o numerador pode ser reescrito como

$$\sum_{i \in B} \lambda_i \left[ \sum_{j \in U_i} T_j Z_j \left( \prod_{k \in C_i^j} \bar{Z}_k \right) \right] + \sum_{i \in B} \gamma_i \left[ \sum_{j \in U_i} T_j Y_j \left( \prod_{k \in C_i^j} \bar{Z}_k \right) \right] \quad (4.26)$$

onde:

$B$  o conjunto formado por todos os blocos de um alimentador.

$\lambda_i$  taxa de falha permanente do bloco  $i$ .

$\gamma_i$  taxa de falha temporária do bloco  $i$ .

$U_i$  o conjunto de blocos pertencentes ao caminho entre o nó  $i$  e o primeiro dispositivo de

proteção  $d_i$  a montante de  $i$ .

$C_i^j = U_i - \{j\}$ . Conjunto dos blocos entre  $i$  (inclusive) e  $j$  (exclusive) onde não há nenhum equipamento de proteção instalado.

$T_j$  total de consumidores a jusante do bloco  $j$  (inclusive).

Como o número de consumidores de cada bloco é conhecido, os valores de  $T_j$  podem ser calculados *a priori* para todos os blocos do alimentador. O primeiro termo da equação (4.26) calcula o impacto das falhas permanentes de um bloco  $i$  quando o dispositivo de proteção do bloco  $j$  atua, podendo ser interpretado da seguinte forma: contabilize uma interrupção permanente para todos os consumidores a jusante e em  $j$  ( $T_j$ ), ponderada pela taxa de falha permanente do bloco onde ocorreu a falha ( $\lambda_i$ ), se um dispositivo de proteção estiver instalado em  $j$  ( $Z_j$ ) e não houver nenhum equipamento de proteção entre  $i$  e  $j$  ( $\prod_{k \in C_i^j} \bar{Z}_k$ ). O segundo termo da equação (4.26) contabiliza o efeito das falhas temporárias de um bloco  $i$  quando o fusível do bloco  $j$  estoura, causando uma interrupção permanente. Este termo pode ser lido como: contabilize uma interrupção permanente para todos os consumidores a jusante e em  $j$  ( $T_j$ ), ponderada pela taxa de falha temporária do bloco onde ocorreu a falha ( $\gamma_i$ ), se um fusível estiver instalado em  $j$  ( $Y_j$ ) e não houver nenhum equipamento de proteção entre  $i$  e  $j$  ( $\prod_{k \in C_i^j} \bar{Z}_k$ ).

Agrupando os termos da equação (4.26), vem

$$\sum_{i \in B} \left[ \sum_{j \in U_i} T_j (\lambda_i Z_j + \gamma_i Y_j) \left( \prod_{k \in C_i^j} \bar{Z}_k \right) \right]. \quad (4.27)$$

Agora, as variáveis “fantasmas” devem ser substituídas pelas variáveis de decisão  $x$  e  $y$ . A variável  $x_j$  é definida como

$$x_j = \begin{cases} 0 & \text{se um religador deve ser alocado no bloco } j, \\ 1 & \text{caso contrário} \end{cases} \quad (4.28)$$

e  $y_j$  por

$$y_j = \begin{cases} 0 & \text{se um fusível deve ser alocado no bloco } j, \\ 1 & \text{caso contrário} \end{cases} \quad (4.29)$$

Estas duas definições são as mesmas feitas por Soudi e Tomsovic [ST98], onde  $x_j$  é equivalente à  $x_{qj1}$  e  $y_j$  corresponde à  $x_{qj2}$ . As modificações foram feitas somente para diminuir o uso de subscritos e aumentar a clareza. A partir de (4.28) e (4.29) as variáveis  $X_j$  e  $Y_j$  podem ser

redefinidas como

$$X_j = 1 - x_j \quad \bar{X}_j = x_j \quad (4.30)$$

$$Y_j = 1 - y_j \quad \bar{Y}_j = y_j. \quad (4.31)$$

Substituindo (4.30) e (4.31) em (4.25), tem-se:

$$Z_j = 2 - x_j - y_j \quad \bar{Z}_j = x_j y_j. \quad (4.32)$$

Finalmente, rescrevendo (4.27) utilizando-se (4.31) e (4.32) e rearrumando os coeficientes de  $x_j$  e  $y_j$ , chega-se a:

$$\sum_{i \in B} \left\{ \sum_{j \in U_i} T_j [2\lambda_i + \gamma_i - \lambda_i x_j - (\lambda_i + \gamma_i) y_j] \left( \prod_{k \in C_i^j} x_k y_k \right) \right\}. \quad (4.33)$$

É possível utilizar uma definição inversa para as variáveis de decisão  $x$  e  $y$ . Isto não foi feito porque, neste caso,  $\bar{Z}_j = 2 - x_j - y_j$  e como  $\bar{Z}$  aparece dentro de um produtório, o número de termos produto na equação final da função objetivo é maior.

É importante notar que é possível se obter uma formulação equivalente à de Soudi e Tomsovic com a modelagem apresentada. Para isto, o alimentador deve ser dividido em um troco principal e ramais laterais de categoria 3, e as variáveis  $d_i$  devem ser definidas como:

- para todo  $i$  no tronco principal,  $d_i$  é igual à raiz da árvore.
- para cada lateral  $l$ , para todo  $i$  em  $l$ ,  $d_i$  é igual ao *tap* de  $l$ .

Considera-se uma “formulação equivalente” um problema de PNLB diferente do obtido pelo modelo de Soudi e Tomsovic mas que leva à mesma solução ótima.

Para se calcular o DEC do alimentador, a equação (4.33) deve ser substituída por

$$\sum_{i \in B} \left\{ \sum_{j \in U_i} T_j t_i [2\lambda_i + \gamma_i - \lambda_i x_j - (\lambda_i + \gamma_i) y_j] \left( \prod_{k \in C_i^j} x_k y_k \right) \right\} \quad (4.34)$$

onde  $t_i$  é o tempo médio de reparo (MTTR) das falhas do bloco  $i$ .

#### 4.2.1.1 Restrições

As restrições apresentadas nesta seção são as mesmas utilizadas por Soudi e Tomsovic. Embora somente as obrigatórias estão listadas, nada impede que certas restrições operacionais (tal como não se alocar religadores depois de fusíveis) também sejam utilizadas.

Não é possível instalar mais de um equipamento de proteção no mesmo bloco, logo

$$x_i + y_i \geq 1, \quad i \in B. \quad (4.35)$$

Além disso, para todos os blocos onde se sabe que haverá um dispositivo instalado deve-se fazer

$$x_j + y_j = 1, \quad j \in \{d_i\}, \quad i \in B. \quad (4.36)$$

Como sempre há um religador de subestação instalado na raiz  $b$  do alimentador, tem-se

$$x_b = 0. \quad (4.37)$$

Com relação ao custo, a seguinte restrição deve ser utilizada:

$$\sum_{i \in B} x_i \geq |B| - (r + 1), \quad (4.38)$$

onde  $r$  é o número de religadores disponíveis para alocação, sem contar o religador da subestação.

### 4.2.2 Um exemplo

O caso de teste apresentado por Soudi e Tomsovic (o alimentador da Figura 4.2) pode ser utilizado para ilustrar a aplicação do novo modelo. Para se considerar o alimentador por inteiro, sem divisões de tronco principal e ramos laterais, basta fazer  $d_i = 11$  para todos os blocos ( $i \in \{11, 12, 13, 14, 21, 31, 41\}$ ). É necessário também calcular os valores  $T_j$ , mostrados na Tabela 4.3.

$j$	11	12	13	14	21	31	41
$T_j$	875	575	325	200	20	50	10

Tabela 4.3: Consumidores acumulados para cada bloco do caso de teste

Utilizando a equação (4.33), chega-se à função objetivo da equação (4.39). Nota-se imediatamente que número de termos produto da função objetivo é muito maior que da formulação original. Isto já era esperado pois a separação do alimentador em tronco principal e ramais laterais, juntamente com as restrições que obrigam a instalação de dispositivos nos *taps* isolam cada uma das laterais, simplificando a função objetivo de Soudi e Tomsovic. Aqui, nenhuma parte do alimentador é considerada em separado e portanto o número de termos é maior. É possível dividir o alimentador como na formulação original fazendo  $d_{21} = 21$ ,  $d_{31} = 31$  e  $d_{41} = 41$ , formando as laterais e  $d_i = 11$  com  $i \in \{11, 12, 13, 14\}$ , definido o tronco principal. Neste caso,

chega-se a uma função objetivo equivalente à de Soudi e Tomsovic.

$$\begin{aligned}
i = 11 : \quad j = 11 : & [(3500 - 875x_{11} - 2625y_{11}) + & (4.39) \\
i = 12 : \quad j = 12 : & (1868.75 - 431.25x_{12} - 1437.5y_{12}) + \\
& j = 11 : (2843.75 - 656.25x_{11} - 2187.5y_{11})x_{12}y_{12} + \\
i = 13 : \quad j = 13 : & (3250 - 731.75x_{13} - 2518.25y_{13}) + \\
& j = 12 : (5750 - 1293.75x_{12} - 4456.25y_{12})x_{13}y_{13} + \\
& j = 11 : (8750 - 1968.75x_{11} - 6781.25y_{11})x_{13}y_{13}x_{12}y_{12} + \\
i = 14 : \quad j = 14 : & (1750 - 400x_{14} - 1350y_{14}) + \\
& j = 13 : (2843.75 - 650x_{13} - 2193.75y_{13})x_{14}y_{14} + \\
& j = 12 : (5031.25 - 1150x_{12} - 3881.25y_{12})x_{14}y_{14}x_{13}y_{13} + \\
& j = 11 : (7656.25 - 1750x_{11} - 5906.25y_{11})x_{14}y_{14}x_{13}y_{13}x_{12}y_{12} + \\
i = 21 : \quad j = 21 : & (25 - 5x_{21} - 20y_{21}) + \\
& j = 12 : (718.75 - 143.75x_{12} - 575y_{12})x_{21}y_{21} + \\
& j = 11 : (1093.75 - 218.75x_{11} - 875y_{11})x_{21}y_{21}x_{12}y_{12} + \\
i = 31 : \quad j = 31 : & (200 - 50x_{31} - 150y_{31}) + \\
& j = 12 : (2300 - 575x_{12} - 1725y_{12})x_{31}y_{31} + \\
& j = 11 : (3500 - 875x_{11} - 2625y_{11})x_{31}y_{31}x_{12}y_{12} + \\
i = 41 : \quad j = 41 : & (35 - 5x_{41} - 30y_{41}) + \\
& j = 14 : (700 - 100x_{14} - 600y_{14})x_{41}y_{41} + \\
& j = 13 : (1137.5 - 162.5x_{13} - 975y_{13})x_{41}y_{41}x_{14}y_{14} + \\
& j = 12 : (2012.5 - 287.5x_{12} - 1725y_{12})x_{41}y_{41}x_{14}y_{14}x_{13}y_{13} + \\
& j = 11 : (3062.5 - 437.5x_{11} - 2625y_{11})x_{41}y_{41}x_{14}y_{14}x_{13}y_{13}x_{12}y_{12}]/875.
\end{aligned}$$

Considerando que há dois religadores disponíveis para alocação ( $r = 2$ ), as restrições se resumem a:

$$\begin{aligned}
x_{11} + x_{12} + x_{13} + x_{14} + x_{21} + x_{31} + x_{41} & \geq 4 \\
x_{11} & = 0 \\
x_{11} + y_{11} = 1 \quad x_{12} + y_{12} & \geq 1 \quad x_{13} + y_{13} & \geq 1 \quad x_{14} + y_{14} & \geq 1 \\
x_{21} + y_{21} & \geq 1 \quad x_{31} + y_{31} & \geq 1 \quad x_{41} + y_{41} & \geq 1.
\end{aligned}$$

A solução ótima para este problema de PNLB é o mesmo da formulação original, com um

FEC de 3.27 interrupções por ano.

### 4.2.3 Resultados

Embora o resultado obtido no caso de teste de Soudi e Tomsovic tenha sido igual ao da formulação original, buscou-se verificar se a nova modelagem resulta em melhores resultados para os alimentadores da massa de teste. Como o GLPK não possui rotinas para solução de problemas de PNLB, é necessário transformar o modelo obtido com a nova formulação em um problema de PLB que possa ser resolvido pelo pacote. Esta transformação é a mesma realizada para a modelagem original.

A Tabela 4.4 apresenta os resultados dos testes realizados. Em todos os casos o alimentador foi analisado por inteiro ( $d_i = b, \forall i \in B$ ). Os campos NF mostram os valores ótimos de DEC e FEC obtidos com a nova formulação. As colunas ST correspondem aos índices do modelo de Soudi e Tomsovic e BE aos da Busca Exaustiva. O tempo médio em segundos das duas execuções da nova formulação pode ser visto na coluna  $T_{NF}$ . Todos os testes foram feitos com  $r = 1$ .

Alimentador	DEC			FEC			$T_{NF}$
	NF	ST	BE	NF	ST	BE	
C6 $n = 16$	4.23	5.77	5.46	1.36	1.85	1.73	5.7
C7 $n = 8$	1.08	3.53	3.53	0.41	1.21	1.21	1.8
E9 $n = 23$	5.09	5.19	5.41	1.77	1.86	2.23	23621.9
F5 $n = 28$	5.62	5.75	6.02	2.51	2.62	2.81	103159.9

Tabela 4.4: Nova formulação  $\times$  Soudi e Tomsovic

Nota-se que todos os índices obtidos com a nova formulação são melhores que os alcançados com as outras formas de solução. No entanto,  $T_{NF}$  mostra que o esforço computacional necessário para se obter a solução ótima do novo modelo é alto, mesmo para alimentadores com menos de 30 blocos. Isso se deve principalmente ao grande número de termos produtos na função objetivo, o que leva ao aparecimento de muitas novas variáveis no problema linearizado. Tomando como exemplo o alimentador F5, com 28 blocos: o problema de PLNB possui 56 variáveis de decisão e o de PLB chega a 551. Isto motivou o desenvolvimento de um algoritmo de *branch-and-bound* para resolver diretamente o modelo de PLNB, como mostra a próxima seção.

## 4.3 Um algoritmo *branch-and-bound* para resolver os problemas de PNLB

Como visto na seção anterior, a nova formulação de PNLB desenvolvida apresenta bons resultados, mas quando reduzida a um problema de PLB possui um tempo de execução muito elevado. Uma vez que o número de variáveis criadas na linearização do PNLB é grande, buscou-se uma forma de solução direta para o problema não-linear. Como não foram encontrados pacotes livres com rotinas para resolução de problemas de Programação Não-Linear, um algoritmo de *branch-and-bound* foi desenvolvido para o modelo de PNLB.

### 4.3.1 Descrição

O *branch-and-bound* é um método de enumeração implícita utilizado em vários problemas de otimização. O algoritmo foi inicialmente proposto por Land e Doig [LD60] em 1960 para resolver problemas de Programação Inteira e atualmente é aplicado em muitos problemas NP-hard, tais como os problemas de TSP de Programação Não-Linear.

A idéia geral do método consiste em buscar o valor mínimo ou máximo de uma função objetivo  $FO(z)$  sobre um conjunto de valores admissíveis do argumento  $z$ , chamada de região viável. Um algoritmo de *branch-and-bound* requer duas ferramentas. A primeira é uma forma adequada de varrer a região viável a partir de várias sub-regiões viáveis menores (preferencialmente, disjuntas). Isto é chamado ramificação (*branching*) pois o procedimento pode ser repetido recursivamente para cada sub-região e todas as sub-regiões produzidas formam uma estrutura em árvore. Esta estrutura é chamada árvore de busca ou árvore *branch-and-bound*, onde cada nó corresponde a uma sub-região construída. Nesta seção, os termos árvore e nó, quando utilizados, são referência à árvore *branch-and-bound* e seus nós, não devendo ser confundidos com a árvore de um alimentador. A outra ferramenta é o *bounding*, uma forma rápida de se encontrar limites inferiores e superiores para a solução ótima dentro de uma sub-região.

Para uma dada região viável, uma separação eficiente divide o espaço de soluções em um pequeno conjunto de possíveis boas soluções a ser examinado mais profundamente e um conjunto maior de soluções ruins que é descartado. Embora desejável, formas de divisão adequadas são normalmente difíceis de serem encontradas na prática. Por isso a criação de um algoritmo eficiente é altamente dependente da natureza do problema.

O ponto chave desta abordagem é a simples observação que (para um problema de minimização) se o limite inferior de uma sub-região  $A$  é maior que a melhor solução corrente, então

A pode ser seguramente descartada da busca. Este passo é chamado poda (*pruning*). O desempenho do método é diretamente dependente da eficiência dos procedimentos de *branching* e *bounding* empregados: escolhas ruins podem levar a repetidas ramificações, sem nenhuma poda, até que cada sub-região fique muito pequena. Nesta caso, o método se resume a uma enumeração exaustiva do domínio, o que normalmente é impraticável. Mais detalhes sobre a formulação geral do método de *branch-and-bound* e suas aplicações podem ser vistos no capítulo 32 de [Ata98].

O exemplo a seguir busca ilustrar as idéias gerais do método de *branch-and-bound*. Seja  $FO = -5x_1 + 7x_2 - 2x_3$ , onde  $x_i$  são variáveis binárias. Para se minimizar  $FO$ , a melhor solução possível é  $x_1 = 1, x_2 = 0$  e  $x_3 = 1$ , onde  $FO = -7$ . A Figura 4.4 mostra uma árvore de busca que enumera todo o espaço de soluções com a configuração ótima destacada em negrito. A visitação dos nós é feita seguindo a sequência alfabética. Abaixo de cada nó folha encontra-se o valor de  $FO$  obtido quando as variáveis de decisão assumem os valores fixados nas arestas do caminho que vai da folha até a raiz.

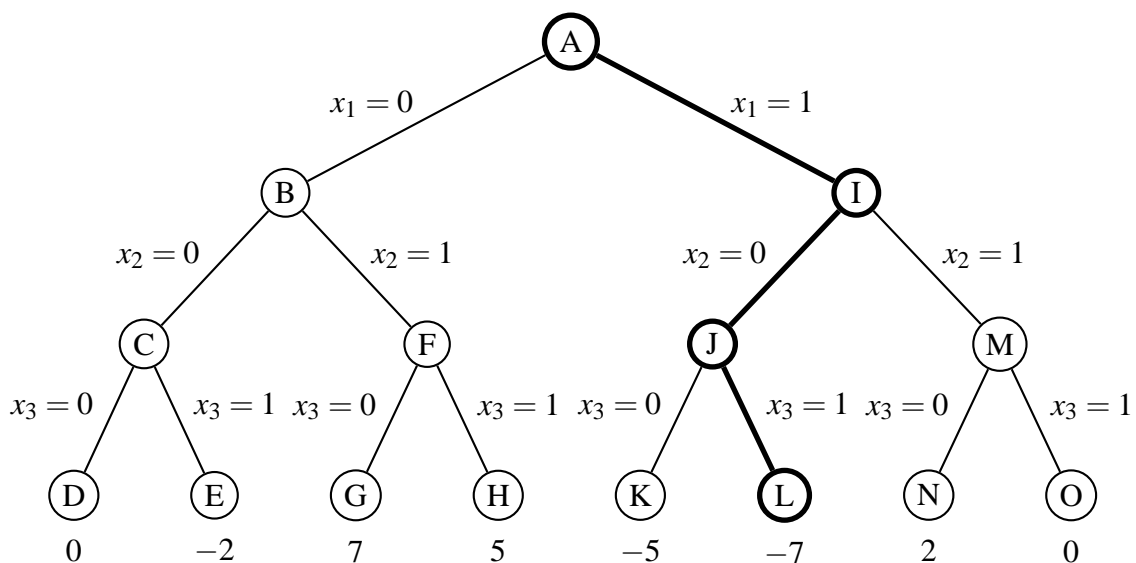


Figura 4.4: Árvore de busca para  $FO$

Uma árvore de busca (binária) que explora todo o espaço de solução possui  $2^{nm+1} - 1$  nós, onde  $nm$  é o número de variáveis de decisão. Nota-se que uma enumeração exaustiva é impraticável quando  $nm$  cresce e portanto é necessário realizar podas para que um algoritmo baseado em uma árvore de busca possa ser utilizado. Estimando-se um limite inferior  $l$  para  $FO$  em cada nó da árvore e tomando  $FO^*$  como o melhor valor de  $FO$  obtido nos nós folhas já visitados é possível fechar alguns nós. Como as variáveis de decisão são binárias, pode-se estimar um limite inferior fazendo todas as variáveis ainda livres com coeficientes negativos igual a 1 e as com coeficientes positivos igual a 0. Um limite superior pode ser calculado com o processo

inverso. A Figura 4.5 mostra uma árvore de busca com os nós F e K podados (sublinhados) e todos limites inferiores calculados.

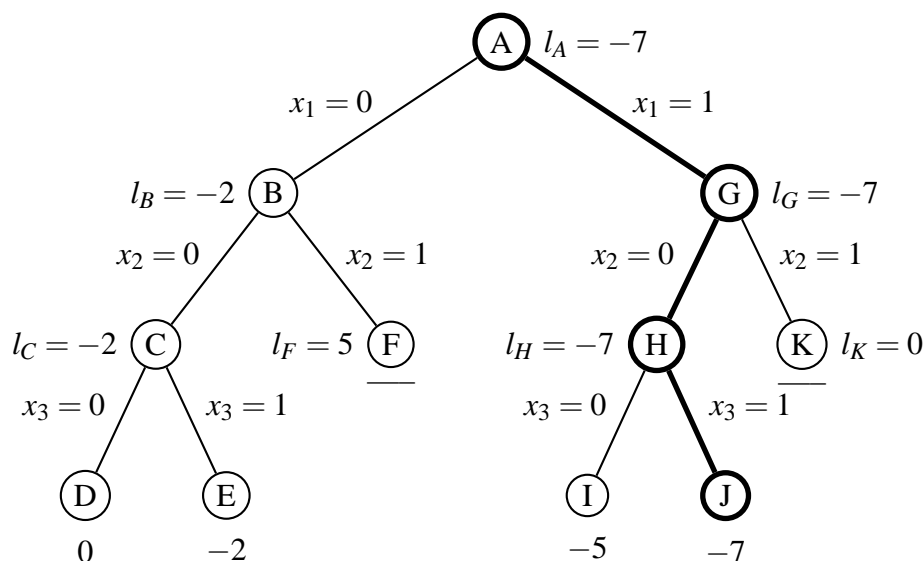


Figura 4.5: Árvore de busca para  $FO$  com poda

Inicialmente não se possui nenhuma solução e  $FO^* \leftarrow \infty$ . Partindo-se do nó A,  $l_A$  é calculado fazendo-se  $x_1$  e  $x_3$  igual a 1 (pois estas variáveis possuem coeficientes negativos) e  $x_2$  igual a 0. Realiza-se um *branching*, criando o nó B, onde  $x_1$  é fixada em 0. Neste caso, as duas variáveis livres  $x_2$  e  $x_3$  recebem 0 e 1 ( $x_1$  está fixo em 0) e  $l_B = -2$ . Este processo é repetido em todos os outros nós. A primeira folha visitada é o nó D. O valor de  $FO$  em D é 0 e como  $FO^* = \infty$ ,  $FO^* \leftarrow 0$ . Da mesma forma, em E,  $FO^*$  passa a valer  $-2$ . Quando a busca chega em F,  $l_F > FO^*$  e o nó pode ser fechado. O mesmo ocorre em K pois o nó J já foi anteriormente visitado e portanto  $FO^* = -7$ .

Até o momento, não se considerou a existência de nenhuma restrição sobre as variáveis de decisão. Supondo que se deseja minimizar  $FO$  obedecendo à restrição que  $x_1 + x_3 \leq 1$ , o valor de  $FO^*$  passa a ser  $-5$  pois a solução do caminho A-G-H-J é inviável (viola a restrição) e a solução ótima fica no ramo A-G-H-I. A escolha da variável a ser fixada em uma ramificação e o valor que ela assumir vai influenciar no número de nós a serem fechados. Olhando para a Figura 4.5 nota-se que se a partir de A,  $x_1$  for fixada primeiro em 1, a solução ótima é encontrada mais rápido, permitindo a poda de mais nós.

O algoritmo de *branch-and-bound* desenvolvido para solução dos problemas de PNLB obtidos a partir do modelo apresentado na seção anterior pode ser visto no Algoritmo 4.1. A

entrada do algoritmo é o problema de PNLB inicial  $P_0$ , que deve possuir o seguinte formato:

$$\min FO = \sum_i c_i \Omega_i$$

sujeito à

$$\sum_{i=1}^m a_i z_i \# b_j, \quad \forall j \in \{1, \dots, m\}, \quad x_i \in \{0, 1\} \quad \text{e} \quad y_i \in \{0, 1\}$$

onde  $c_i$  são os coeficientes de custo,  $a_i$  e  $b_j$  são parâmetros que descrevem as restrições e  $m$  é o número de restrições. O termo  $z_i$  que aparece nas restrições corresponde às variáveis de decisão  $x_i$  e  $y_i$ . No total há  $nn$  variáveis de decisão. O operador  $\#$  pode ser substituído por  $\leq$ ,  $\geq$  ou  $=$ , conforme o caso. Já  $\Omega_i$  corresponde a um termo produto da função objetivo, podendo ser formado por diferentes variáveis  $x$  e  $y$ .

Algoritmo 4.1: *Branch-and-bound* para os problemas de PNLB

1	{Entrada :
2	– $P_0 =$ problema de PNLB inicial
3	– $S_0 =$ uma solução inicial}
4	
5	{Variáveis :
6	– $S^* =$ a melhor solução encontrada até o momento
7	– $FO^* =$ o valor da função objetivo para $S^*$
8	– $l_{bb} =$ o limite inferior do nó $bb$
9	– $P_{bb} =$ problema de PNLB que deve ser resolvido no nó $bb$
10	– $v_{bb} = (z, a)$ , onde $z$ é a variável fixada no nó $bb$ e $a$ é o seu valor
11	– $f_{bb} =$ conjunto dos valores já fixados nos filhos de $bb$
12	– um pilha com os nós atualmente explorados}
13	
14	{Saída: $S^*$ e $FO^*$ indicam a solução ótima}
15	
16	$S^* \leftarrow S_0$
17	$FO^* \leftarrow FO(S_0)$
18	Crie um nó $bb$ com $P_{bb} \leftarrow P_0$
19	Empilhe $bb$
20	<b>Enquanto</b> a pilha não estiver vazia <b>faça</b>
21	$bb$ é o nó que está no topo da pilha
22	<b>Se</b> $ f_{bb}  = 0$ <b>então</b> {Ainda não foi feito branch em $bb$ }
23	<b>Se</b> $P_{bb}$ é inviável para $v_{bb}$ <b>então</b>
24	{Fechamento de nó por violação de restrições}
25	Desempilhe $bb$

```

26   Senão  $\{P_{bb}$  é viável $\}$ 
27       Se todas as variáveis já foram fixadas então  $\{bb$  é uma folha $\}$ 
28           Construa uma solução  $S_{bb}$  partindo do topo da pilha até a base
29            $FO_{bb} \leftarrow FO(S_{bb})$ 
30           Se  $FO_{bb} \leq FO^*$  então  $\{Foi encontrada uma solução melhor\}$ 
31                $FO^* \leftarrow FO_{bb}$ 
32                $S^* \leftarrow S_{bb}$ 
33           Fim se
34           Desempilhe  $bb$ 
35       Senão  $\{bb$  não é folha $\}$ 
36           Atualize  $P_{bb}$ 
37           Atualize  $l_{bb}$ 
38           Se  $l_{bb} > FO^*$  então
39                $\{Fechamento de nó por violação de bounds - pruning\}$ 
40               Desempilhe  $bb$ 
41           Senão  $\{bb$  não pode ser fechado, ramifique para a esquerda $\}$ 
42               Escolha uma variável  $z$  para ser fixada em 0 ou 1 ( $=a$ )
43               Crie um nó  $bb'$ , com  $v_{bb'} \leftarrow (z, a)$  e  $P_{bb'} \leftarrow P_{bb}$ 
44                $f_{bb} \leftarrow \{v_{bb'}\}$ 
45               Empilhe  $bb'$ 
46           Fim se
47       Fim se
48   Fim se
49   Se  $|f_{bb}| = 1$  então  $\{bb$  já foi ramificado para a esquerda, faça a direita $\}$ 
50       Crie um nó  $bb'$ , com  $v_{bb'} \leftarrow \neg u$  e  $P_{bb'} \leftarrow P_{bb}$ , onde  $u \in f_{bb}$ 
51        $f_{bb} \leftarrow f_{bb} + \{\neg u\}$ 
52       Empilhe  $bb'$ 
53   Se  $|f_{bb}| = 2$  então  $\{bb$  possui dois filhos e ambos já foram explorados $\}$ 
54       Desempilhe  $bb$ 
55   Fim se
56 Fim faça

```

A solução inicial  $S_0$  fornecida como entrada é muito importante para o desempenho do algoritmo. Quanto mais próxima  $S_0$  estiver da solução ótima, mais nós da árvore de busca podem ser podados. Nos testes realizados tomou-se como solução inicial o resultado da execução do *Simulated Annealing* sobre o alimentador, com  $r$  religadores. O *Simulated Annealing* foi empregado por fornecer um bom valor para  $S_0$  e possuir um tempo de execução baixo. Caso não

seja possível encontrar uma solução inicial,  $S_0$  corresponde a um conjunto vazio. Neste caso, define-se  $FO(\emptyset) = \infty$  para o cálculo da linha 17 do algoritmo.

O Algoritmo 4.1 possui a mesma estrutura do algoritmo de caminhamento em pré-ordem de árvores binárias apresentado por Knuth [Knu97], exceto pelo fato de nem todos os nós da árvore serem visitados devido às podas. Um caminhamento em pré-ordem primeiro visita a raiz, a seguir a sub-árvore da esquerda e finalmente a sub-árvore da direita. Esta definição é recursiva, mas pode-se evitar um algoritmo recursivo, empregando-se uma pilha de nós. Esta pilha pode ser limitada em  $nn + 1$  nós (o maior caminho possível entre uma folha e a raiz) e, portanto, o algoritmo possui um requisito constante de memória.

O *loop* definido na linha 20 do algoritmo sempre trabalha com o nó  $bb$  que está no topo da pilha. Analizando  $f_{bb}$ , tem-se 3 casos distintos. Quando  $|f_{bb}| = 2$ , tanto a sub-árvore da esquerda de  $bb$  quando da direita já foram exploradas (linha 53) e o nó  $bb$  pode ser descartado (desempilhado). No segundo caso (linha 49),  $|f_{bb}| = 1$  indica que a sub-árvore da esquerda de  $bb$  já foi visitada e deve-se agora visitar a da direita. Aqui  $f_{bb}$  indica qual foi a variável escolhida para ser fixada e qual o valor ela assumiu no ramo esquerdo. O ramo direito que está sendo criado ( $bb'$ ) deve ter a mesma variável  $v_{bb'}$  do ramo esquerdo, com o valor complementar. Isto é, se  $u = (z, a)$ , com  $u \in f_{bb}$ , então  $v_{bb'} \leftarrow \neg u$ , onde  $\neg u$  é definido como  $(z, \neg a)$  (linha 50).

O último caso ( $|f_{bb}| = 0$ , linha 22), ocorre quando o nó  $bb$  foi recém-empilhado, isto é,  $bb$  acabou de ser criado e colocado no topo da pilha. Primeiro, deve-se verificar se  $bb$  é viável (linha 23), isto é, se  $v_{bb}$  não viola alguma restrição do problema  $P_{bb}$ . Por exemplo, se  $v_{bb} = (x_3, 0)$  e  $P_{bb}$  possui uma restrição exigindo que  $x_3 \geq 1$ , então  $bb$  é inviável e deve ser descartado. Se  $bb$  for identificado como um nó folha (linha 27), as variáveis  $v$  dos nós empilhados formam uma solução viável para o problema  $P_0$ . Esta solução é comparada com  $S^*$  e armazenada se for melhor (linhas 28–34). Quando  $bb$  não é uma folha, o problema  $P_{bb}$  deve ser simplificado antes de ser passado para os filhos de  $bb$ . Essa simplificação consiste em substituir  $v_{bb}$  em  $P_{bb}$ , eliminando esta variável do problema. Por exemplo, se  $P_{bb}$  é definido como  $\min FO = 20x_2x_3y_1 - 17y_2y_3 + 12$  sujeito à  $x_2 + x_3 \geq 2$  e  $v_{bb} = (x_3, 1)$ , então  $P_{bb}$  após a atualização passa a ser  $\min FO = 20x_2y_1 - 17y_2y_3 + 12$  sujeito à  $x_2 \geq 1$ . A seguir, o limite inferior do nó  $l_{bb}$  deve ser calculado para verificar se  $bb$  pode ser podado (linhas 37–40). Este cálculo é o mesmo do exemplo apresentado na Figura 4.5, bastando observar que cada termo produto  $\Omega$  da função objetivo pode assumir apenas dois valores (0 ou 1) por ser formado apenas de variáveis binárias. Tomando a  $FO$  do exemplo anterior, tem-se  $l_{bb} = -5$ . Finalmente, quando um nó não puder ser fechado, ele deve ser ramificado à esquerda. Uma variável livre deve ser escolhida e fixada em 0 ou 1 no novo nó  $bb'$  (linha 42). Como esta escolha causa impacto no tempo de execução

do algoritmo, variáveis que permitam podar um maior número de nós devem ser selecionadas primeiro. Para este problema, é conveniente primeiro fixar em 0 todas as variáveis  $x$ , pois a restrição do número de religadores disponíveis pode ser usada para fechar vários nós inviáveis.

### 4.3.2 Resultados

O Algoritmo 4.1 foi implementado no mesmo ambiente de desenvolvimento apresentado no capítulo 3 e os resultados dos testes realizados estão na Tabela 4.5. Como nos testes da Tabela 4.4, todos os alimentadores foram analisados por inteiro ( $d_i = b, \forall i \in B$ ). As colunas NF indicam os índices ótimos obtidos com a nova formulação, os campos ST correspondem aos índices do modelo de Soudi e Tomsovic (seção 4.1) e BE aos da Busca Exaustiva (seção 3.3). O tempo médio em segundos das duas execuções do algoritmo de *branch-and-bound* pode ser visto na coluna  $T_{NF}$ . O número de nós visitados em cada caso também está destacado. Todos os testes foram realizados com  $r = 1$ .

Alimentador	DEC			FEC			Nós visitados	$T_{NF}$
	NF	ST	BE	NF	ST	BE		
A1 $n = 43$	9.27	10.03	11.69	4.15	4.51	5.48	$\approx 2^{25}$ de $2^{87}$	210122.9
A2 $n = 46$	8.84	11.73	8.86	3.37	5.21	3.38	$\approx 2^{21}$ de $2^{93}$	16790.8
B2 $n = 41$	9.62	10.25	10.32	3.12	3.27	3.32	$\approx 2^{25}$ de $2^{83}$	83981.9
B3 $n = 34$	11.78	12.28	12.29	3.81	4.02	4.04	$\approx 2^{10}$ de $2^{69}$	30012.9
C6 $n = 16$	4.23	5.77	5.46	1.36	1.86	1.73	$\approx 2^{10}$ de $2^{33}$	8.9
C7 $n = 8$	1.08	3.53	3.53	0.41	1.21	1.21	$\approx 2^7$ de $2^{17}$	0.1
D4 $n = 38$	8.54	10.77	11.53	2.69	3.46	3.67	$\approx 2^{25}$ de $2^{77}$	243159.9
E1 $n = 37$	6.84	8.05	7.04	2.94	2.98	2.96	$\approx 2^{20}$ de $2^{75}$	3525.1
E6 $n = 48$	6.04	6.60	6.36	2.65	2.94	2.87	$\approx 2^{23}$ de $2^{97}$	255515.3
E9 $n = 23$	5.08	5.19	5.41	1.76	1.86	2.23	$\approx 2^{11}$ de $2^{47}$	18.9
F4 $n = 38$	8.34	8.68	8.78	3.35	3.51	3.59	$\approx 2^{24}$ de $2^{77}$	136939.7
F5 $n = 28$	5.62	5.75	6.02	2.51	2.62	2.81	$\approx 2^{11}$ de $2^{57}$	14.4

Tabela 4.5: Resultados da nova formulação resolvida pelo Algoritmo 4.1

Os índices obtidos com a nova modelagem são sempre melhores que os da formulação de Soudi e Tomsovic e da Busca Exaustiva. Comparando-se o tempo de execução da Tabela 4.4, nota-se que o  $T_{NF}$  com o algoritmo de *branch-and-bound* é consideravelmente menor. Tomando como exemplo o alimentador F5: enquanto a solução do problema linearizado via GLPK leva mais de 28 horas, o algoritmo de *branch-and-bound* apresentado resolve o problema não-linear em 14 segundos. No entanto, a Tabela 4.5 mostra que o esforço computacional do Algoritmo 4.1 para alimentadores com 30 ou mais blocos ainda é alto, apesar do número de nós explorados ser sempre muito menor que o total de nós da árvore de busca. Isto indica que mesmo com o

algoritmo de *branch-and-bound* podando vários ramos da árvore ainda é necessário o uso de heurísticas para simplificar o problema e diminuir o esforço computacional.

## 5 *Conclusão*

Esta dissertação apresenta técnicas de Otimização aplicadas ao problema de minimização de índices de confiabilidade de alimentadores. Dois objetivos principais foram determinados para este trabalho: (1) otimizar a alocação de religadores em um alimentador para melhorar os índices de DEC e FEC e (2) determinar os valores ótimos destes índices quando se realiza a alocação de religadores, fusíveis e chaves de forma conjunta. Os resultados do segundo objetivo servem como um limite inferior e podem ser utilizados como base de comparação com outros métodos de solução.

Para o problema de alocação de religadores em um alimentador partiu-se, inicialmente, para um método de solução força bruta: uma Busca Exaustiva no espaço de soluções. Os resultados obtidos são ótimos mas essa abordagem é ineficiente em termos de tempo de execução. Embora isso impeça o seu emprego em situações que exijam uma resposta em pouco tempo, a Busca Exaustiva ainda é relevante pois permite estabelecer uma base de comparação com os resultados das outras formas de soluções. A seguir, buscou-se adaptar a meta-heurística *Simulated Annealing* para o problema. Nesta etapa, um compromisso entre a qualidade da solução e o tempo empregado em sua busca foi atingido. Os resultados obtidos com esta abordagem destacam-se pois as soluções encontradas pelo *Simulated Annealing* estão sempre muito próximas às da Busca Exaustiva e o tempo de execução da meta-heurística é muito menor. Neste ponto, considerou-se que o objetivo de determinar os melhores pontos para alocação de religadores em um alimentador foi alcançado de forma bastante satisfatória.

Finalmente, partiu-se para a utilização dos modelos de programação matemática. Esta abordagem foi utilizada por permitir a otimização da alocação de religadores, fusíveis e chaves em um alimentador de forma simultânea. Uma vez que o modelo de Programação Linear Binária já existente não apresentou bons resultados para algumas instâncias da massa de teste, uma nova formulação de Programação Não-linear Binária foi desenvolvida. Este modelo é mais abrangente que o original e levou a soluções melhores em todos os testes realizados. No entanto, o tempo de execução necessário para se resolver um problema da nova modelagem era muito alto e por isso criou-se um algoritmo de *branch-and-bound* específico para resolver o novo mo-

delo de PNLB desenvolvido. Este algoritmo diminuiu o tempo necessário para se encontrar uma solução mas ainda exige um esforço computacional considerável para alimentadores com mais de 40 blocos e para a alocação de mais de um religador. Porém, da mesma forma que a Busca Exaustiva, este método de solução serve como base de comparação, permitindo investigar heurísticas que simplifiquem o problema de PNLB. Fica como trabalho futuro verificar como estas heurísticas diminuem o tempo de execução do algoritmo e analisar o seu impacto na qualidade das soluções. Convém destacar que a nova modelagem de PNLB e o algoritmo de *branch-and-bound* são as duas principais contribuições originais deste trabalho.

As formas de solução e os algoritmos apresentados neste trabalho foram implementados e deram origem a uma ferramenta de projeto que pode ser utilizada pelo Engenheiro de Proteção. Esta ferramenta é um programa multi-plataforma (Windows/Linux) que permite ao usuário realizar, de forma interativa, um estudo da rede de proteção de um alimentador. Uma versão inicial deste sistema já foi entregue à concessionária local e está em fase de teste.

Ainda há pontos que podem ser investigados mas, neste momento, os objetivos traçados para este trabalho foram alcançados.

Os resultados obtidos neste trabalho permitem que uma concessionária reestruture e expanda os sistemas de proteção das suas redes de distribuição de uma forma planejada. Desta maneira, os investimentos da empresa passam a ser mais efetivos, levando a um serviço de melhor qualidade sem necessariamente implicar em um aumento de tarifas ao consumidor e de custos para a concessionária. Isto pode levar a benefícios econômicos para ambos.

## 5.1 Trabalhos futuros

O tema abordado nesta dissertação é bastante amplo e oferece várias áreas de pesquisa. Dentre os trabalhos futuros relacionados ao que foi aqui apresentado, destacam-se:

- Estudar outras formas de simulação apresentadas por Brown [Bro02], tal como a Simulação Estocástica. Analisar como esta forma de simulação pode ser utilizada para se estimar os índices de confiabilidade de um alimentador e comparar os resultados com os valores históricos para verificar se este método consegue se aproximar melhor destes índices.
- Investigar como o *Simulated Annealing* pode ser utilizado para otimizar a alocação de todos os dispositivos ao mesmo tempo, como é feito pela programação matemática. Talvez a meta-heurística continue apresentando bons resultados com um tempo de execução menor que o algoritmo de *branch-and-bound*.

- 
- Analisar o impacto das heurísticas adotadas por Soudi e Tomsovic [ST98] na qualidade de solução e tempo de execução da nova modelagem desenvolvida e no algoritmo de *branch-and-bound*. Buscar formas de tentar melhorar o processo de poda do algoritmo para aumentar o seu desempenho.
  - Buscar desenvolver métodos que permitam alocar religadores em uma subestação inteira e não apenas em um alimentador por vez.
  - Utilizar o método de *Goal Programming* empregado por Soudi e Tomsovic em [ST01] para otimizar de forma simultânea os índices de DEC e FEC nos modelos de programação matemática.

## *Referências*

- [ANE00] ANEEL, *Resolução nº 24, de 27 de janeiro de 2000*, 2000. 13, 21
- [Ata98] Mikhail J. Atallah, *Algorithms and theory of computation handbook*, CRC-Press, 1998. 38, 62
- [BJ96] Roy Billinton and Satish Jonnavithula, *Optimal switching device placement in radial distribution systems*, IEEE Transactions on Power Delivery **11** (1996), no. 3, 1646–1651. 15, 38
- [BO98] Richard E. Brown and J. Rafael Ochoa, *Distribution system reliability: Default data and model validation*, IEEE Transactions on Power Delivery **13** (1998), no. 2, 704–709. 23
- [Bro02] Richard E. Brown, *Electric power distribution reliability*, Marcel Dekker, Inc., 2002. 16, 23, 27, 29, 40, 70
- [Dia02] Evaldo Baldin Dias, *Avaliação de indicadores de continuidade e seu impacto no planejamento de sistemas de distribuição*, Master’s thesis, Universidade de São Paulo, 2002. 15
- [dS02] Luis Gustavo Wesz da Silva, *Alocação otimizada de dispositivos de proteção em sistemas de distribuição de energia elétrica*, Master’s thesis, Universidade Estadual Paulista, May 2002. 15, 26, 54
- [dSPM04] Luis G.W. da Silva, Rodrigo A.F. Pereira, and José R.S. Mantovani, *Allocation of protective devices in distribution circuits using nonlinear programming models and genetic algorithms*, Electric Power Systems Research **69** (2004), no. 1, 77–84. 15, 16
- [GK02] Fred W. Glover and Gary A. Kochenberger, *Handbook of metaheuristics (international series in operations research and management science)*, Kluwer Academic Publishers, 2002. 38
- [KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Optimization by simulated annealing*, Science **220** (1983), no. 4598, 671–680. 38
- [Knu97] Donald E. Knuth, *The art of computer programming*, 3rd ed., vol. 1 – Fundamental Algorithms, Addison-Wesley, 1997. 66
- [LD60] A. H. Land and A. G. Doig, *An automatic method for solving discrete programming problems*, Econometrica **28** (1960), no. 3, 497–520. 61
- [ST97] Farajollah Soudi and Kevin Tomsovic, *Towards optimized distribution protection design*, Proceedings of the Third International Conference on Power System Planning and Operations (1997). 15

- 
- [ST98] ———, *Optimized distribution protection using binary programming*, IEEE Transactions on Power Delivery **13** (1998), no. 1, 218–224. , 15, 16, 45, 47, 50, 51, 56, 71
- [ST99] ———, *Optimal distribution protection design: quality of solution and computational analysis*, International Journal on Electric Power and Energy Systems **21** (1999), 327–335. 15
- [ST01] ———, *Optimal trade-offs in distribution protection design*, IEEE Transactions on Power Delivery **16** (2001), no. 2, 292–296. 15, 71

## APÊNDICE A – Testes do Simulated Annealing

Neste apêndice se encontram todos os testes comparativos realizados entre a meta-heurística *Simulated Annealing* (seção 3.4) e a Busca Exaustiva (seção 3.3). Cada tabela corresponde a uma subestação.

As colunas  $DEC_{SA}$  e  $FEC_{SA}$  indicam os melhores valores de DEC e FEC obtidos com o *Simulated Annealing*. As razões  $SA/BE$  comparam estes valores com as soluções ótimas obtidas com a Busca Exaustiva. As colunas  $T_{SA}$  e  $T_{BE}$  correspondem ao tempo de execução em segundos do *Simulated Annealing* e da Busca Exaustiva, respectivamente.

Alimentador		$DEC_{SA}$	$SA/BE$	$FEC_{SA}$	$SA/BE$	$T_{SA}$	$T_{BE}$
A1	$r = 1$	7.95	1.000	5.48	1.000	0.19	0.01
	$r = 2$	6.40	1.000	3.91	1.000	0.39	0.08
	$r = 3$	5.78	1.000	3.27	1.000	0.57	1.07
	$r = 4$	5.34	1.000	2.91	1.000	0.85	10.05
A2	$r = 1$	6.79	1.000	3.37	1.000	0.09	0.01
	$r = 2$	6.15	1.000	2.74	1.000	0.85	0.06
	$r = 3$	5.82	1.000	2.41	1.000	1.19	0.90
	$r = 4$	5.55	1.000	2.14	1.000	1.27	9.39
A3	$r = 1$	38.19	1.000	16.74	1.000	0.82	0.06
	$r = 2$	34.50	1.000	13.05	1.000	1.67	4.57
	$r = 3$	33.55	1.000	12.09	1.000	4.33	178.16
	$r = 4$	32.83	1.000	11.38	1.000	6.72	5826.45
A4	$r = 1$	37.01	1.000	14.66	1.000	1.57	0.16
	$r = 2$	35.08	1.000	12.71	1.000	5.72	9.74
	$r = 3$	33.61	1.000	11.53	1.000	19.09	526.12
	$r = 4$	32.61	1.000	10.22	1.025	36.45	22961.21

Tabela A.1: *Simulated Annealing* × Busca Exaustiva: subestação A

Alimentador		$DEC_{SA}$	$SA/BE$	$FEC_{SA}$	$SA/BE$	$T_{SA}$	$T_{BE}$
B1	$r = 1$	20.30	1.000	11.67	1.000	1.34	0.06
	$r = 2$	15.97	1.000	7.34	1.000	2.53	3.56
	$r = 3$	14.40	1.000	5.77	1.000	4.10	134.29
	$r = 4$	13.39	1.000	5.20	1.000	5.03	4230.43
B2	$r = 1$	9.42	1.000	3.31	1.000	0.07	0.01
	$r = 2$	8.35	1.000	2.97	1.000	0.24	0.05
	$r = 3$	8.07	1.000	2.69	1.000	0.51	0.50
	$r = 4$	7.84	1.000	2.56	1.000	0.56	4.66
B3	$r = 1$	11.19	1.000	4.03	1.000	0.08	0.01
	$r = 2$	10.39	1.000	3.58	1.000	0.26	0.02
	$r = 3$	9.94	1.000	3.23	1.000	0.31	0.25
	$r = 4$	9.66	1.000	3.11	1.000	0.99	1.92
B4	$r = 1$	18.19	1.000	6.20	1.000	0.13	0.04
	$r = 2$	16.68	1.000	5.47	1.000	0.22	0.13
	$r = 3$	15.55	1.000	4.80	1.000	0.56	2.47
	$r = 4$	14.68	1.000	4.39	1.000	0.83	33.85

Tabela A.2: *Simulated Annealing* × Busca Exaustiva: subestação B

Alimentador		$DEC_{SA}$	$SA/BE$	$FEC_{SA}$	$SA/BE$	$T_{SA}$	$T_{BE}$
D1	$r = 1$	18.21	1.000	9.80	1.000	2.45	0.28
	$r = 2$	14.29	1.000	5.87	1.000	7.32	34.04
	$r = 3$	13.84	1.000	5.49	1.000	11.15	2864.03
	$r = 4$	13.45	1.000	5.33	1.000	20.50	168292.20
D2	$r = 1$	28.61	1.000	13.51	1.000	0.96	0.11
	$r = 2$	25.07	1.000	9.97	1.000	1.66	4.71
	$r = 3$	23.24	1.000	8.14	1.000	2.92	203.52
	$r = 4$	22.35	1.000	7.61	1.050	4.79	6514.49
D3	$r = 1$	2.72	1.000	1.75	1.000	0.19	0.01
	$r = 2$	2.50	1.000	1.59	1.000	2.56	0.18
	$r = 3$	2.32	1.000	1.48	1.000	4.37	3.44
	$r = 4$	2.16	1.000	1.40	1.000	4.76	46.71
D4	$r = 1$	11.15	1.000	3.66	1.000	0.02	0.01
	$r = 2$	8.71	1.000	2.76	1.000	0.05	0.02
	$r = 3$	7.63	1.000	2.35	1.000	0.18	0.22
	$r = 4$	6.65	1.000	2.13	1.000	0.29	1.83

Tabela A.3: *Simulated Annealing* × Busca Exaustiva: subestação D

Alimentador		$DEC_{SA}$	$SA/BE$	$FEC_{SA}$	$SA/BE$	$T_{SA}$	$T_{BE}$
C1	$r = 1$	74.41	1.000	19.51	1.000	2.35	0.13
	$r = 2$	72.38	1.000	18.36	1.000	3.59	6.13
	$r = 3$	71.17	1.000	17.48	1.000	6.92	289.99
	$r = 4$	70.10	1.001	16.87	1.000	10.23	10279.55
C2	$r = 1$	68.56	1.000	14.66	1.000	0.17	0.01
	$r = 2$	66.27	1.000	13.27	1.000	0.53	0.22
	$r = 3$	63.98	1.000	12.41	1.000	0.79	3.36
	$r = 4$	61.76	1.000	11.81	1.000	2.52	45.76
C3	$r = 1$	28.21	1.000	8.09	1.000	0.53	0.06
	$r = 2$	26.29	1.000	6.12	1.000	1.02	1.08
	$r = 3$	24.95	1.000	5.74	1.000	1.61	31.62
	$r = 4$	24.58	1.007	5.42	1.000	2.58	704.59
C4	$r = 1$	18.33	1.000	4.77	1.000	0.98	0.09
	$r = 2$	17.04	1.000	4.24	1.000	2.35	2.86
	$r = 3$	16.01	1.000	3.98	1.000	13.47	112.80
	$r = 4$	15.47	1.000	3.76	1.000	18.71	3355.10
C5	$r = 1$	36.62	1.000	13.03	1.000	0.41	0.08
	$r = 2$	34.10	1.000	11.31	1.000	1.85	1.86
	$r = 3$	32.38	1.000	9.64	1.000	2.63	62.37
	$r = 4$	31.95	1.007	9.38	1.000	5.44	1588.22
C6	$r = 1$	3.80	1.000	1.73	1.000	0.01	0.01
	$r = 2$	3.14	1.000	1.06	1.000	0.03	0.02
	$r = 3$	2.76	1.000	0.94	1.000	0.08	0.07
	$r = 4$	2.53	1.000	0.87	1.000	0.20	0.68
C7	$r = 1$	2.15	1.000	1.21	1.000	0.01	0.01
	$r = 2$	1.52	1.000	0.80	1.000	0.01	0.02
	$r = 3$	1.17	1.000	0.45	1.000	0.01	0.03
	$r = 4$	0.99	1.000	0.36	1.000	0.02	0.03
C8	$r = 1$	23.52	1.000	6.09	1.000	0.24	0.01
	$r = 2$	22.31	1.000	5.18	1.000	0.47	0.22
	$r = 3$	21.40	1.000	4.97	1.000	0.79	3.48
	$r = 4$	20.81	1.006	4.80	1.000	1.09	50.08

Tabela A.4: *Simulated Annealing* × Busca Exaustiva: subestação C

Alimentador		$DEC_{SA}$	$SA/BE$	$FEC_{SA}$	$SA/BE$	$T_{SA}$	$T_{BE}$
E1	$r = 1$	6.11	1.000	2.96	1.000	0.27	0.01
	$r = 2$	5.72	1.000	2.53	1.000	0.49	0.06
	$r = 3$	5.38	1.000	2.30	1.000	1.69	0.59
	$r = 4$	5.16	1.000	2.09	1.000	2.27	5.08
E2	$r = 1$	24.64	1.000	13.96	1.000	0.59	0.02
	$r = 2$	23.39	1.000	12.68	1.000	1.98	1.05
	$r = 3$	22.55	1.000	12.16	1.000	3.51	29.02
	$r = 4$	21.99	1.000	11.87	1.011	4.30	592.78
E3	$r = 1$	4.56	1.000	3.69	1.000	0.28	0.01
	$r = 2$	3.76	1.000	2.83	1.000	0.35	0.10
	$r = 3$	3.23	1.000	2.28	1.000	0.54	1.26
	$r = 4$	2.89	1.000	2.07	1.000	2.78	11.24
E4	$r = 1$	6.76	1.000	5.43	1.000	0.71	0.01
	$r = 2$	5.41	1.000	4.00	1.000	1.30	0.57
	$r = 3$	4.16	1.000	2.74	1.000	1.99	12.38
	$r = 4$	3.78	1.000	2.48	1.000	2.24	196.22
E5	$r = 1$	8.85	1.000	6.32	1.000	1.18	0.04
	$r = 2$	7.88	1.000	5.35	1.000	2.28	2.02
	$r = 3$	7.28	1.000	4.72	1.000	3.91	66.13
	$r = 4$	7.03	1.004	4.47	1.000	4.75	1627.94
E6	$r = 1$	5.80	1.000	2.86	1.000	0.17	0.01
	$r = 2$	5.48	1.000	2.70	1.000	0.64	0.13
	$r = 3$	5.35	1.000	2.56	1.000	0.87	1.85
	$r = 4$	5.17	1.008	2.46	1.000	1.05	19.58
E7	$r = 1$	15.02	1.000	9.64	1.000	0.97	0.09
	$r = 2$	12.35	1.000	6.96	1.000	2.75	6.15
	$r = 3$	10.93	1.000	6.20	1.000	8.34	283.06
	$r = 4$	10.16	1.000	5.51	1.000	31.54	10696.31
E8	$r = 1$	10.78	1.000	6.25	1.000	1.89	0.11
	$r = 2$	9.83	1.000	5.26	1.000	3.48	7.50
	$r = 3$	9.20	1.000	4.63	1.000	6.47	341.29
	$r = 4$	8.69	1.000	4.13	1.005	7.38	11760.36
E9	$r = 1$	5.04	1.000	2.23	1.000	0.06	0.01
	$r = 2$	4.82	1.000	1.84	1.000	0.11	0.04
	$r = 3$	4.61	1.000	1.64	1.000	0.19	0.09
	$r = 4$	4.44	1.000	1.55	1.000	0.30	0.28

Tabela A.5: *Simulated Annealing* × Busca Exaustiva: subestação E

Alimentador		$DEC_{SA}$	$SA/BE$	$FEC_{SA}$	$SA/BE$	$T_{SA}$	$T_{BE}$
F1	$r = 1$	20.81	1.000	6.87	1.000	0.42	0.01
	$r = 2$	19.61	1.000	6.25	1.000	1.14	0.27
	$r = 3$	18.84	1.000	5.74	1.000	1.42	5.72
	$r = 4$	18.23	1.000	5.37	1.000	1.94	90.25
F2	$r = 1$	16.00	1.000	7.42	1.000	1.36	0.04
	$r = 2$	15.48	1.000	7.02	1.000	2.58	2.37
	$r = 3$	15.02	1.000	6.70	1.000	3.05	55.58
	$r = 4$	14.62	1.000	6.48	1.000	16.17	1291.05
F3	$r = 1$	16.54	1.000	9.52	1.000	1.10	0.05
	$r = 2$	15.34	1.000	8.27	1.000	11.75	2.08
	$r = 3$	14.24	1.000	7.82	1.000	17.50	71.17
	$r = 4$	13.69	1.000	7.57	1.003	25.22	1463.07
F4	$r = 1$	7.35	1.000	3.58	1.000	0.13	0.01
	$r = 2$	6.70	1.000	2.87	1.000	0.27	0.08
	$r = 3$	6.39	1.000	2.48	1.000	1.63	0.56
	$r = 4$	6.11	1.000	2.35	1.000	2.67	4.32
F5	$r = 1$	5.56	1.000	2.81	1.000	0.09	0.01
	$r = 2$	5.23	1.000	2.53	1.000	0.16	0.05
	$r = 3$	4.91	1.000	2.26	1.000	0.49	0.15
	$r = 4$	4.65	1.000	2.07	1.000	0.85	0.66
F6	$r = 1$	14.13	1.000	6.20	1.000	0.25	0.01
	$r = 2$	13.21	1.000	5.28	1.000	0.71	0.26
	$r = 3$	12.37	1.000	4.63	1.000	0.99	3.79
	$r = 4$	11.71	1.000	4.13	1.043	1.10	46.87
F7	$r = 1$	28.37	1.000	16.03	1.000	0.81	0.10
	$r = 2$	23.44	1.000	11.10	1.000	2.75	2.80
	$r = 3$	22.64	1.000	10.28	1.000	4.78	84.53
	$r = 4$	21.88	1.000	9.53	1.000	6.05	1935.16

Tabela A.6: *Simulated Annealing* × Busca Exaustiva: subestação F