



**Universidade Federal do Ceará**  
**Programa de Pós-Graduação em Engenharia Elétrica**

**José Roberto Bezerra**

**Contribuições para Posicionamento Eficiente de Chaves e Restauração Automática  
de Redes de Distribuição de Energia Elétrica**

**Fortaleza**

**2015**

José Roberto Bezerra

Contribuições para Posicionamento Eficiente de Chaves e Restauração Automática de Redes  
de Distribuição de Energia Elétrica

Tese apresentada ao Programa de Pós- Graduação em Engenharia Elétrica da Universidade Federal do Ceará, como parte dos requisitos para a obtenção do título de Doutor em Engenharia Elétrica.

Prof. Dr. Giovanni Cordeiro Barroso

Profa. PhD. Ruth Pastôra Saraiva Leão

Fortaleza

2015

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca de Pós-Graduação em Engenharia - BPGE

- 
- B469c Bezerra, José Roberto.  
Contribuições para posicionamento eficiente de chaves e restauração automática de redes de distribuição de energia elétrica / José Roberto Bezerra. – 2015.  
115 f. : il. color., enc. ; 30 cm.
- Tese (doutorado) – Universidade Federal do Ceará, Centro de Tecnologia, Departamento de Engenharia Elétrica, Programa de Pós-Graduação em Engenharia Elétrica, Fortaleza, 2015.  
Área de Concentração: Sistemas de Energia Elétrica.  
Orientação: Prof. Dr. Giovanni Cordeiro Barroso.  
Coorientação: Profa. Dra. Ruth Pastôra Saraiva Leão.
1. Engenharia elétrica. 2. Sistemas de energia elétrica. 3. Energia elétrica – Distribuição. 4. Otimização matemática. I. Título.

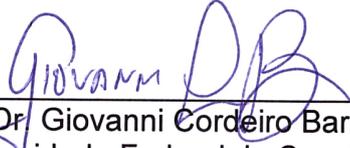
**JOSÉ ROBERTO BEZERRA**

**CONTRIBUIÇÕES PARA O POSICIONAMENTO EFICIENTE DE CHAVES E  
RESTAURAÇÃO AUTOMÁTICA DE REDES DE DISTRIBUIÇÃO  
DE ENERGIA ELÉTRICA**

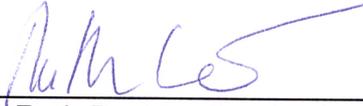
Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Sistemas de Energia Elétrica.

Aprovada em: 16/12/2015

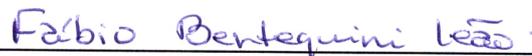
**BANCA EXAMINADORA**



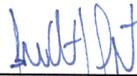
Prof. Dr. Giovanni Cordeiro Barroso (Orientador)  
Universidade Federal do Ceará (PPGEE-UFC)



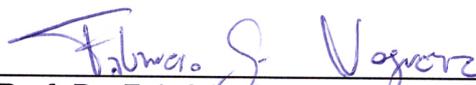
Prof<sup>a</sup>. Dr<sup>a</sup>. Ruth Pastora Saraiva Leão (Coorientadora)  
Universidade Federal do Ceará (PPGEE-UFC)



Prof. Dr. Fábio Bertequini Leão  
Universidade Estadual Paulista (UNESP)



Prof. Dr. Bruno de Athayde Prata  
Universidade Federal do Ceará (DEPRO-UFC)



Prof. Dr. Fabrício Gonzalez Nogueira  
Universidade Federal do Ceará (PPGEE-UFC)

# Agradecimentos

Antes de qualquer agradecimento é preciso dizer que nada do que escrito neste documento vem de mim. Tudo me foi concedido pelo poder superior, divino, supremo, criador ou simplesmente Deus. Servi apenas para receber a inspiração, as ideias e a vontade de fazer esse trabalho.

Agradeço a meus pais, Ivan e Itamê, pela ousadia e insistência em, a partir de quase nada, prover educação para mim e meus irmãos. Em nenhum momento eles tiveram dúvidas em fazer todo o necessário para garantir boa escola para seus filhos. Sinceramente, não sei como eles conseguiram fazer isso diante das circunstâncias difíceis que se apresentaram ao longo da vida e apenas hoje percebo a grandeza disso.

Por fim, agradeço a meus orientadores e parceiros de longa data, Professor Giovanni e Professora Ruth, não apenas pelos conhecimentos, mas também pela confiança e autonomia que me foram dadas.

# Resumo

A demanda por energia elétrica cresce e de igual modo a exigência dos consumidores pela continuidade do seu fornecimento. Diversos índices de confiabilidade são utilizados por agências reguladoras, concessionárias e consumidores para aferir a continuidade dos sistemas elétricos de potência. A melhoria de tais índices vem sendo requisitada por agências reguladoras e perseguida pelas concessionárias de energia. Para tanto, o uso de sistemas de restauração automática vem ganhando destaque como uma ferramenta de apoio a operadores de centros de controle para a restauração rápida e segura de redes elétricas em situações de falta. Este trabalho apresenta duas contribuições para a restauração automática de redes radiais de distribuição de energia. A primeira consiste em um algoritmo para o posicionamento eficiente de chaves telecontroladas em redes de distribuição. Técnicas de otimização multiobjetivo são aplicadas para obter-se como resultado um conjunto de soluções não-dominadas ao invés de uma única solução. Com isso, a tomada de decisão para o posicionamento eficiente de chaves na rede elétrica é facilitado pela flexibilidade das múltiplas soluções oferecidas pelo algoritmo proposto. A segunda contribuição do trabalho consiste em um modelo para restauração automática da rede de distribuição de energia adaptável às mudanças de topologia e que garante a segurança operacional da rede elétrica. O modelo proposto fundamentou-se em Redes de Petri Coloridas como ferramenta de modelagem. Características de reusabilidade e extensibilidade foram buscadas e implementadas no modelo proposto. Como resultado, o tempo necessário para restaurar redes elétricas em condição de falta é reduzido sem comprometer a segurança do sistema elétrico. Estudos de caso são apresentados evidenciando os benefícios para concessionárias de distribuição de energia com a implementação das ferramentas propostas para melhoria da confiabilidade da rede elétrica.

**Palavras-chave:** Sistemas de Energia Elétrica, Energia Elétrica - Distribuição, Otimização Matemática.

# Abstract

The demand for electricity grows in the same way as the consumers requirement for an uninterrupted supply. Many reliability indices are applied by regulatory agencies, utilities and consumers for measuring the power systems reliability. The improvement of such indices has been required by regulatory agencies and pursued by energy distributors. Therefore, the application of automatic restoration systems has gained relevance as a support tool for control center operators for a quick and secure restauration of faulty electric networks. This work proposes two contributions to the restoration process of radial power system networks. The former is a multiobjective algorithm for effective switch placement in distribution networks. Multiobjective optimization techniques are applied to achieve a set of non-dominated solutions rather than an unique one. Therefore, the decision making for switch placement is eased by the flexibility given from the multiple solutions. The latter contribution is a model for automatic restoration of power distribution networks adaptable to the network topology changes and able to ensure the operational security of the power system network. The proposed model is based on coloured Petri nets as the modelling tool. Reusability and extensibility features have been pursued and implemented on the proposed model. As a result, the needed time to restore faulty power system networks is reduced without compromising its operational security. Case studies are presented demonstrating the benefits to the power system distribution utilities obtained with the deployment of the proposed tools to improve the reliability of the power system networks.

**Keywords:** Power Systems, Power System Delivery, Automatic Restoration, Coloured Petri Nets, Particle Swarm Optimization.

# Lista de ilustrações

Figura 2.1 – Fronteira Pareto-ótima. . . . .	25
Figura 2.2 – Representação de uma Rede de Petri Colorida. . . . .	29
Figura 3.1 – Fluxograma do algoritmo MOSP. . . . .	34
Figura 3.2 – RBTS Bus 4. . . . .	38
Figura 3.3 – Resultados da simulação para o RBTS <i>Bus</i> 4. (a). CNS. (b). Número de chaves. . . . .	39
Figura 3.4 – Fronteira de Pareto para o RBTS4. . . . .	40
Figura 3.5 – Rede elétrica simplificada. . . . .	42
Figura 3.6 – Fluxograma do algoritmo MOSP. . . . .	44
Figura 3.7 – Nova fronteira de Pareto para o RBTS4. . . . .	45
Figura 3.8 – Fronteiras de Pareto obtidas com diferentes versões do MOSP. . . . .	46
Figura 4.1 – Rede elétrica com dois alimentadores. . . . .	50
Figura 4.2 – Alimentadores de subestações diferentes. . . . .	50
Figura 4.3 – Hierarquia do modelo PSR. . . . .	54
Figura 4.4 – Interação entre o modelo PSR, o SCADA e o SEP. . . . .	55
Figura 4.5 – Página principal do modelo PSR. . . . .	59
Figura 4.6 – Subpágina LockDetect do modelo PSR. . . . .	59
Figura 4.7 – Subpágina Locate do modelo PSR. . . . .	61
Figura 4.8 – Sistema elétrico de potência. . . . .	62
Figura 4.9 – Subgrafo representando a partida dos relés. . . . .	62
Figura 4.10–Subgrafo para o cenário pós-falta. . . . .	62
Figura 4.11–Subpágina Find Faulty. . . . .	63
Figura 4.12–Subpágina Find Affected. . . . .	64
Figura 4.13–Subpágina Isolate. . . . .	65
Figura 4.14–Subpágina FindAllPaths. . . . .	66
Figura 4.15–Subpágina Reconfigure. . . . .	67
Figura 4.16–Subpágina NetAnalyser. . . . .	69
Figura 4.17–Subpágina Proceed. . . . .	71
Figura 4.18–Subpágina Restore. . . . .	72
Figura 4.19–Sistema de teste TwoBusTS. . . . .	73
Figura 4.20–Estado pós-falta do sistema teste TwoBusTS - Cenário I. . . . .	75
Figura 4.21–Estado pós-reconfiguração do sistema teste TwoBusTS - Cenário I. . . . .	82
Figura 4.22–Estado pós-falta do sistema teste TwoBusTS - Cenário II. . . . .	83
Figura 4.23–Estado pós-falta do sistema teste TwoBusTS - Cenário III. . . . .	85
Figura 4.24–Estado pós-reconfiguração do sistema teste TwoBusTS - Cenário III. . . . .	86

# Lista de tabelas

Tabela 3.1 – Parâmetros do PSO e pesos para os experimentos do RBTS4. . . . .	37
Tabela 3.2 – Conjunto de soluções não-dominadas para o RBTS4. . . . .	40
Tabela 3.3 – Comparativo entre a localização das chaves. . . . .	41
Tabela 4.1 – Dados dos pontos de carga e dos barramentos. . . . .	73
Tabela 4.2 – Dados complementares dos pontos de carga e dos barramentos. . . . .	73
Tabela 5.1 – Artigo publicado em revista. . . . .	94
Tabela 5.2 – Artigo submetido para publicação. . . . .	94
Tabela 5.3 – Artigos publicados e apresentados em congressos. . . . .	94

# Lista de abreviaturas e siglas

AE	Algoritmos Evolucionários
CNS	Clientes Não Supridos
DG	<i>Distributed Generation</i>
ECOST	Custo esperado de uma falta no sistema elétrico
IED	<i>Intelligent Electronic Device</i>
IEEE	<i>Institute of Electric and Electronic Engineers</i>
MA	Multi-Agentes
MILP	<i>Mixed Integer Linear Programming</i>
ML	<i>Modelling Language</i>
MOO	<i>Multi-Objective Optimization</i>
MOSP	<i>Multi-Objective Switch Placement</i>
NA	Normalmente Aberta
NF	Normalmente Fechada
PMU	<i>Phasor Measurement Unit</i>
PSO	<i>Particle Swarm Optimization</i>
PSR	<i>Power System Restoration</i>
RBTS	<i>Roy Billinton Test System</i>
REI	Redes Elétricas Inteligentes
RNA	Redes Neurais Artificiais
RPC	Rede de Petri Colorida
SAIDI	<i>System Average Interruption Duration Index</i>
SAIFI	<i>System Average Interruption Frequency Index</i>
SCADA	<i>Supervisory Control and Data Acquisition System</i>

SED	Sistemas a Eventos Discretos
SEP	Sistema Eléctrico de Potência
SG	<i>Smart Grids</i>
SIN	Sistema Interligado Nacional
SOO	Single Objective Optimization
SRA	Sistema de Restauração Automática
WSM	<i>Weighted Sum Method</i>

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
1.1	Justificativa	15
1.2	Objetivos	15
1.3	Posicionamento de chaves	15
1.4	Restauração automática	17
1.5	Organização do documento	19
<b>2</b>	<b>FERRAMENTAS DE OTIMIZAÇÃO, MODELAGEM E SIMULAÇÃO</b>	<b>21</b>
2.1	Otimização por Enxame de Partículas	21
2.2	Otimização multiobjetivo	23
2.2.1	Fronteira de Pareto	24
2.3	Teoria dos grafos	26
2.3.1	Grafos	26
2.3.2	Grafo simples	26
2.3.3	Percursos, caminhos e ciclos	26
2.3.4	Matriz de adjacência	27
2.3.5	Busca em profundidade	27
2.3.6	Caminhos elementares	28
2.4	Redes de Petri Coloridas	28
2.5	Conclusão	30
<b>3</b>	<b>POSICIONAMENTO DE CHAVES EM REDES DE DISTRIBUIÇÃO</b>	<b>31</b>
3.1	Trabalhos relacionados	31
3.2	Algoritmo MOSP	32
3.3	Descrição do algoritmo	33
3.3.1	Inicialização	33
3.3.2	Busca	35
3.3.3	Classificação	35
3.3.4	Funções objetivo	35
3.4	Estudo de caso	37
3.5	Melhorias de desempenho do algoritmo MOSP	41
3.5.1	Inicialização induzida	41
3.5.2	Múltiplas instâncias em paralelo	43
3.5.3	Nova função objetivo	44
3.5.4	Resultados	45
3.6	Conclusão	46

<b>4</b>	<b>MODELO DE RESTAURAÇÃO DO SISTEMA ELÉTRICO . . . . .</b>	<b>48</b>
<b>4.1</b>	<b>Procedimentos atuais . . . . .</b>	<b>48</b>
<b>4.2</b>	<b>Restrições operacionais . . . . .</b>	<b>49</b>
4.2.1	Ampacidade de cabos . . . . .	49
4.2.2	Capacidade de suprimento de transformadores . . . . .	50
4.2.3	Restrições adicionais . . . . .	50
<b>4.3</b>	<b>Trabalhos relacionados . . . . .</b>	<b>51</b>
4.3.1	Abordagens que utilizam Multi-Agentes . . . . .	51
4.3.2	Método de seccionalização . . . . .	52
4.3.3	Redes neurais artificiais . . . . .	52
4.3.4	Comparativo . . . . .	52
4.3.5	Abordagem adotada . . . . .	53
<b>4.4</b>	<b>Características do modelo PSR . . . . .</b>	<b>53</b>
<b>4.5</b>	<b>Dados de entrada . . . . .</b>	<b>55</b>
<b>4.6</b>	<b>Modelo PSR . . . . .</b>	<b>58</b>
4.6.1	Localização de falta . . . . .	60
4.6.2	Isolamento de falta . . . . .	63
4.6.3	Reconfiguração . . . . .	66
4.6.4	Efetivação . . . . .	70
<b>4.7</b>	<b>Restauração . . . . .</b>	<b>71</b>
<b>4.8</b>	<b>Estudo de Caso . . . . .</b>	<b>73</b>
4.8.1	Cenário I . . . . .	74
4.8.2	Cenário II . . . . .	81
4.8.3	Cenário III . . . . .	85
4.8.4	Cenário IV . . . . .	87
<b>4.9</b>	<b>Conclusão . . . . .</b>	<b>88</b>
<b>5</b>	<b>CONCLUSÕES . . . . .</b>	<b>89</b>
<b>5.1</b>	<b>Algoritmo MOSP . . . . .</b>	<b>89</b>
5.1.1	Destaques . . . . .	89
5.1.2	Limitações . . . . .	90
<b>5.2</b>	<b>Modelo PSR . . . . .</b>	<b>90</b>
5.2.1	Destaques . . . . .	90
5.2.2	Limitações . . . . .	91
<b>5.3</b>	<b>Trabalhos futuros . . . . .</b>	<b>91</b>
5.3.1	Algoritmo MOSP . . . . .	92
5.3.2	Modelo PSR . . . . .	93
<b>5.4</b>	<b>Publicações . . . . .</b>	<b>94</b>

Referências . . . . .	95
<b>APÊNDICES</b>	<b>99</b>
APÊNDICE A – FUNÇÕES DO MODELO PSR . . . . .	100
APÊNDICE B – <i>SCRIPTS</i> .M . . . . .	109

# 1 Introdução

O uso da eletricidade na sociedade moderna cresce a cada dia da mesma forma que a exigência dos consumidores pela continuidade no fornecimento. Imaginar o futuro sem energia elétrica é um exercício difícil de ser feito tamanha é a dependência da eletricidade em todas as atividades cotidianas.

Diversos índices de confiabilidade do sistema elétrico são utilizados para medir a continuidade do fornecimento. Agências reguladoras, consumidores e concessionárias utilizam-se de tais índices com objetivos distintos. As agências reguladoras os utilizam para fins de fiscalização. Clientes podem utilizá-los como parâmetro de escolha entre concessionárias disponíveis. Já as distribuidoras de energia podem utilizar os índices como indicadores para acompanhamento e busca de metas de qualidade.

Entretanto, mesmo os índices de confiabilidade sendo de vital importância em todos os segmentos do sistema elétrico, questões relacionadas ao procedimento operacional das concessionárias em geral contribuem fortemente para que baixos índices de confiabilidade sejam uma realidade.

Atualmente, os procedimentos de restauração do sistema elétrico são realizados de forma manual pelos operadores de centros de controle das concessionárias. Dada a complexidade dos sistemas elétricos de potência e o curto intervalo de tempo disponível para a realização da restauração, é comum que aconteçam erros que ocasionam faltas por longos períodos de tempo e grande quantidade de clientes afetados.

Em face dos recentes esforços de modernização dos sistemas elétricos em todo o mundo e da introdução do conceito de Redes Elétricas Inteligentes (REI), também conhecido como *Smart Grids*, faz-se necessário a adoção de procedimentos automáticos para restauração do sistema elétrico qualquer que seja a extensão da falta.

Diante deste contexto, este trabalho apresenta uma proposta para restauração automática de redes de distribuição de energia elétrica radiais com recurso e também um algoritmo para posicionamento de chaves na rede de distribuição. A proposta tem como objetivo a minimização dos impactos das faltas para os consumidores e como consequência a melhoria dos índices de confiabilidade das redes de distribuição. A proposição de um algoritmo para posicionamento de chaves em conjunto com o sistema de restauração automática incrementa significativamente a qualidade do procedimento de restauração devido à instalação estratégica das chaves, permitindo que o isolamento de trechos defeituosos tenha menor impacto em trechos livres de defeitos. Por esta razão, os dois temas são abordados de forma conjunta neste trabalho.

## 1.1 Justificativa

O presente trabalho justifica-se pela crescente necessidade dos consumidores por fornecimento ininterrupto de eletricidade, seja motivado por necessidades profissionais, conforto, segurança ou ainda pela manutenção da vida. Em face do desafio de fornecimento ininterrupto, pode-se resumir a justificativa para esta proposta nos seguintes pontos:

- Crescente exigência dos consumidores por melhoria na confiabilidade do sistema elétrico;
- Cobrança por parte das agências reguladoras de melhoria nos índices de confiabilidade do sistema elétrico;
- Interesse das concessionárias de agregar melhorias efetivas para o sistema elétrico com o objetivo de garantir a continuidade de fornecimento ou a redução do tempo de restauração e da quantidade de clientes afetados por faltas.

## 1.2 Objetivos

Em face do desafio do fornecimento ininterrupto de energia, este trabalho tem como objetivo principal apresentar contribuições para a melhoria efetiva da confiabilidade das redes elétricas de distribuição. Destacam-se como objetivos específicos desta proposta os seguintes:

- Propor um algoritmo para posicionamento eficiente de chaves em redes elétricas de distribuição que tenha como resultado um conjunto de soluções;
- Validar o algoritmo proposto em um sistema elétrico de teste;
- Propor um modelo em Rede de Petri Colorida para automatizar os procedimentos de restauração automática;
- Validar a modelagem proposta em um sistema elétrico de teste.

## 1.3 Posicionamento de chaves

O uso de Sistemas de Restauração Automática (SRA) pressupõe a existência de chaves para realizar o isolamento dos trechos defeituosos da rede elétrica e ainda modificar a topologia da rede para recomposição das cargas desenergizadas. O desempenho de um SRA é diretamente afetado pela localização destas chaves no Sistema Elétrico de Potência (SEP). Chaves posicionadas em pontos inadequados reduzem a capacidade de manobra do

SRA limitando sua abrangência. Portanto, o posicionamento eficiente de chaves na rede elétrica é fundamental para a melhoria da eficiência do processo de restauração.

Diversos trabalhos são encontrados na literatura sobre o posicionamento de chaves em redes de distribuição (REN et al., 2012; YE; ZHANG; SUTANTO, 2011; SARMADI et al., 2011; BRETAS; PHADKE, 2003; WANG; VITTAL; SUN, 2011), os quais têm como principal objetivo a melhoria da confiabilidade das redes de distribuição apontando uma configuração de chaves única e otimizada. Porém, não foram encontrados na literatura trabalhos que tivessem como foco a melhoria da eficiência de um sistema de restauração automática propriamente dito ou ainda que oferecessem configurações alternativas para a restauração do sistema elétrico. Portanto, o foco principal deste trabalho consiste em oferecer um conjunto de soluções para o problema de posicionamento de chaves ao invés de uma única solução.

Similarmente, as propostas para restauração automática consultadas não prevêm a utilização de nenhuma abordagem específica para o posicionamento de chaves. O uso de um SRA pressupõe uma rede elétrica com capacidade mínima de manobra para realizar isolamentos de ramais defeituosos e transferências de carga. A capacidade máxima de manobra pode ser obtida instalando-se uma chave em cada ramo da rede elétrica considerada. Entretanto, a complexidade das redes de distribuição torna essa prática economicamente inviável devido ao grande número de ramais existentes. Portanto, a localização estratégica de chaves em posições determinantes para o isolamento de trechos defeituosos e a reconfiguração do sistema elétrico a partir de novas fontes é fundamental para a eficiência de um SRA.

Desta forma, o posicionamento eficiente de chaves no SEP é tratado neste trabalho como um problema de otimização com múltiplos objetivos.

O primeiro objetivo é a melhoria da confiabilidade do sistema elétrico a partir da instalação de um determinado número de chaves. Considerando-se que a quantidade máxima de chaves é igual ao número de ramos do sistema elétrico. Pressupõe-se que quanto maior o número de chaves instaladas, maior será a capacidade da rede de isolar menores áreas defeituosas e então restaurar um maior número de clientes a partir de manobras na rede elétrica.

O segundo objetivo diz respeito ao custo, ou seja, a quantidade de chaves necessárias para se obter um determinado nível de confiabilidade da rede elétrica. Pressupõe-se que menores quantidades de chaves, ou a ausência delas, afetam a confiabilidade do sistema, pois reduz-se a capacidade de isolar menores trechos e, por conseguinte, restaurar maior número de trechos.

Portanto, o posicionamento eficiente de chaves tem como objetivo determinar um ponto de equilíbrio entre a confiabilidade e o custo. Sendo a confiabilidade medida através

da quantidade de clientes afetada quando ocorrem faltas na rede elétrica e o custo medido por meio da quantidade de chaves da configuração.

Diante deste contexto, neste trabalho é proposto um novo algoritmo multiobjetivo para o posicionamento de chaves em redes elétricas de distribuição, MOSP (do inglês *Multiobjective Switch Placement*). O algoritmo apresenta como saída um conjunto de soluções não-dominadas ao invés de uma única solução. Esta abordagem busca oferecer flexibilidade para as concessionárias que tenham como meta a melhoria da confiabilidade de suas redes por meio da instalação estratégica de chaves telecontroladas. Estudos de caso comprovando a eficiência do algoritmo MOSP são apresentados.

## 1.4 Restauração automática

A restauração segura de um SEP é uma tarefa que demanda diversos procedimentos operacionais. Normalmente, tais procedimentos são realizados por operadores submetidos a condições de trabalho estressantes, e inerentes à própria função. Desta forma, a restauração de SEP tem grandes chances de ser realizada em um intervalo de tempo inadequado (no caso brasileiro, maior que três minutos) e de que erros possam ser cometidos durante sua execução.

Diversos trabalhos encontrados na literatura, tais como (WANG et al., 2010), (ZIDAN; EL-SAADANY, 2012), (HAJIAN-HOSEINABADI, 2011), (FOUNTAS; HATZIARGYRIOU; VALAVANIS, 1997), (JUNG; KIM; KO, 1993), (KETABI; RANJBAR; FEUILLET, 2002), dentre outros, evidenciam a importância da restauração automática para os SEP e a dificuldade de se apresentar uma solução definitiva para o tema.

Portanto, o uso de SRA por parte das concessionárias de energia vem sendo aceito e aperfeiçoado com intuito de agregar as vantagens de tais sistemas para as concessionárias. Diversos benefícios podem ser atribuídos ao uso de um SRA. Dentre elas, destacam-se:

1. Restauração rápida do sistema elétrico após a ocorrência de faltas;
2. Redução do número de clientes afetados pela falta;
3. Transferência segura de carga entre subestações diferentes;
4. Automatização da priorização de cargas;
5. Monitoramento das restrições operacionais do sistema.

Os SEP atuais, especialmente o brasileiro, têm grandes dimensões e uma alta complexidade. No caso do sistema elétrico brasileiro, tais características são mais marcantes por tratar-se de um sistema interligado e de dimensões continentais (Sistema Interligado Nacional, SIN)

(ONS, 2014). Faltas em uma região do país podem ter consequências em outras regiões. Por outro lado, produção de energia insuficiente em uma região pode ser compensada por outra região trazendo flexibilidade para o equilíbrio do balanço energético. Do ponto de vista da distribuição, o sistema elétrico é ainda mais complexo, visto que fatores externos ao próprio sistema elétrico passam a influenciá-lo mais fortemente como, por exemplo, acidentes urbanos que podem danificar a rede de distribuição predominantemente aérea.

Diante deste nível de complexidade, torna-se cada vez mais difícil o diagnóstico e a restauração segura dos SEP em situações de falta dentro de um intervalo de tempo adequado. Diversas restrições operacionais devem ser observadas pelos operadores de centros de controle para realização de manobras e redistribuição de cargas. A capacidade de condução das linhas de distribuição e dos transformadores, o carregamento dos transformadores no momento da falta, e os consumidores prioritários afetados (hospitais, centrais telefônicas, eletrodependentes, etc.) devem ser observados antes da realização de qualquer ação corretiva. A rápida restauração do sistema elétrico deve ser antecedida da análise destas variáveis e do isolamento do ponto defeituoso. Todas estas ações devem ser planejadas e realizadas pelo operador em um tempo inferior a três minutos (no caso brasileiro) a fim de evitar penalidades para a concessionária. Na maioria das vezes, este tempo é insuficiente para proceder uma restauração segura e adequada, impondo um alto nível de *stress* e pressão aos operadores.

Devido a estas circunstâncias, é comum que a restauração ou a nova configuração do sistema elétrico selecionada pelo operador não seja a melhor configuração, mas sim a configuração possível dentro das restrições de tempo existentes. Melhores resultados podem ser alcançados adotando-se estratégias que utilizam-se de métodos computacionais, pois um número maior de clientes afetados pela falta poderia ser atendido através do novo arranjo da rede elétrica que não foi observado pelo operador.

A restauração é considerada ideal ou ótima quando apenas os clientes conectados ao ramo defeituoso são afetados pela falta. Obviamente, para que um sistema de restauração seja viável, a rede elétrica deve possuir capacidade de realizar manobras que uma rede de distribuição radial convencional não realiza. Para tanto, é necessário instalar chaves ao longo da rede de distribuição. A quantidade e a localização destas chaves deve ser otimizada tendo como objetivo a minimização da quantidade de clientes afetados utilizando-se da menor quantidade de chaves possível.

O atendimento aos clientes afetados de uma determinada subestação, por exemplo, pode demandar energia de uma subestação adjacente. A capacidade de linhas e/ou transformadores deve ser avaliada com cuidado e supervisionada enquanto esta configuração estiver sendo utilizada, pois durante sua utilização, as cargas podem sofrer variações ocasionando sobrecarga nas linhas e/ou transformadores.

Outra questão importante para as distribuidoras é o atendimento a consumidores

prioritários e eletrodependentes. Pessoas que possuem equipamentos de suporte à vida instalados em suas casas e que dependem diretamente deles para sua sobrevivência são prioritários.

Diante deste contexto, neste trabalho é proposto um modelo para restauração automática de redes de distribuição que minimiza a quantidade de clientes afetados. O modelo considera ainda as restrições operacionais do sistema elétrico que devem ser observadas para que a restauração aconteça com segurança. As características de extensibilidade e reusabilidade foram buscadas e implementadas no modelo. Estudos de caso são apresentados para demonstrar a eficiência da modelagem em um sistema elétrico de teste.

## 1.5 Organização do documento

No presente capítulo é apresentada uma breve contextualização do processo de restauração automática de redes de distribuição bem como as vantagens de sua utilização. Similarmente, o posicionamento eficiente de chaves é discutido e evidenciado como sendo de grande valor para a melhoria do desempenho de um SRA.

No Capítulo 2 é feita uma breve introdução às ferramentas matemáticas e de modelagem aplicadas ao algoritmo de posicionamento de chaves e o modelo de restauração automática propostos, respectivamente. Já as técnicas de otimização multiobjetivo em conjunto com a otimização por enxame de partículas fundamentam a abordagem utilizada para o posicionamento eficiente das chaves. As Redes de Petri Coloridas são utilizadas como ferramenta para modelagem do processo de restauração e também são introduzidas neste capítulo.

No Capítulo 3 é apresentado o algoritmo *Multi Objective Switch Placement* (MOSP), proposto para o posicionamento eficiente de chaves. A técnica de Otimização por Enxame de Partículas é aplicada em conjunto com uma abordagem multiobjetivo para determinar o posicionamento e a quantidade de chaves em uma rede de distribuição radial. O resultado é apresentado através de uma curva de Pareto de forma que um conjunto de soluções não-dominadas é apresentado como resultado ao invés de uma única solução ótima sob um determinado ponto de específico. Em estudo de caso é apresentado para demonstrar a eficiência do algoritmo.

No Capítulo 4 é apresentada a proposta de modelo para restauração automática adotado, *Power System Restoration* (PSR). O modelo PSR fundamenta-se em Redes de Petri Coloridas que são utilizadas para modelar toda a dinâmica do processo de restauração. O modelo prevê a análise das restrições operacionais, a priorização de cargas sensíveis e a minimização da quantidade de consumidores afetados. O modelo é plenamente adaptável a diferentes sistemas elétricos, ou seja, modificações na topologia da rede elétrica não invalidam a modelagem adotada demandando apenas modificações na marcação inicial

da rede. Assim, nenhuma alteração na estrutura do modelo é necessária para o correto funcionamento de uma implementação do modelo PSR em uma rede de distribuição. Estudo de caso é apresentado para demonstrar a eficiência do modelo proposto.

No Capítulo 5 são apresentadas as conclusões do trabalho, bem como as principais limitações e restrições que impactaram no seu resultado final. Acrescentou-se ainda um comentário sobre a aplicabilidade do algoritmo MOSP e do modelo PSR em concessionárias de energia na forma de produtos. É apresentada também uma visão de trabalhos futuros a serem desenvolvidos para melhoria dos resultados alcançados ou desenvolvimento de novas propostas relacionadas ao tema. Por fim, são apresentadas as publicações obtidas ao longo do desenvolvimento deste trabalho.

## 2 Ferramentas de Otimização, Modelagem e Simulação

No presente capítulo são abordados de forma simplificada, conceitos matemáticos, técnicas de otimização, simulação e modelagem aplicados neste trabalho. Os temas são abordados apenas para que o leitor tenha familiaridade com a notação e a terminologia empregadas. Um estudo mais detalhado pode ser feito consultando as referências indicadas no documento.

### 2.1 Otimização por Enxame de Partículas

Técnicas de otimização inspiradas no comportamento coletivo de algumas espécies são bastante populares no meio científico devido às suas características de flexibilidade e robustez. Alocação de recursos, projeto de redes, biologia computacional são exemplos de áreas do conhecimento em que estas técnicas podem ser aplicadas (BLUM, 2008). Mais particularmente em SEP, tais técnicas são aplicadas com sucesso em problemas de otimização de reativos (MORI; KOMATSU, 2006), controle dinâmico (VU et al., 2010), posicionamento ótimo de chaves em redes de distribuição (ABIRI-JAHROMI et al., 2012) (MORADI; FOTUHI-FIRUZABAD, 2008), otimização de redes de distribuição de energia (AHMADI; MARTI, 2015), dentre outros.

Otimização por enxame de partículas (PSO, do inglês, *Particle Swarm Optimization*) é uma das técnicas de inteligência de enxame de destaque na literatura. A técnica foi originalmente proposta por Kennedy e Eberhart (1995) como um novo método para otimização de funções contínuas e não lineares. Nesta metodologia, as partículas correspondem aos indivíduos de uma população que tem seu tamanho representado por  $M$  e cada partícula possui dimensão  $N$ . Assim, uma matriz  $M \times N$  é utilizada para representar a população. Cada indivíduo da população representa um agente de busca de soluções para o problema de dimensão  $N$  em questão. Maiores valores de  $M$  indicam mais buscas dentro do espaço de soluções.

Uma versão modificada do PSO original aplicada a valores discretos é proposta por Kennedy e Eberhart (1997), em que os conceitos de posição e velocidade têm uma nova interpretação. A posição de cada partícula assume um valor binário tendo significados diversos dentro do problema de otimização estudado. A quantidade de *bits* da posição é modificada a cada iteração e representa a velocidade da partícula (KENNEDY;

EBERHART, 1997) sendo expressa pela Equação (2.1).

$$V_{i+1} = V_i + c_1 \varphi_{1,i} \Delta V_{1,i} + c_2 \varphi_{2,i} \Delta V_{2,i} \quad (2.1)$$

em que:í

$$\begin{aligned} \Delta V_{1,i} &= pbest_i - X_i \\ \Delta V_{2,i} &= gbest - X_i \end{aligned} \quad (2.2)$$

Os parâmetros  $c_1$  e  $c_2$  são chamados de fatores de aprendizagem ou coeficientes de aceleração e podem assumir valores reais, tipicamente entre 1 e 10. Já os parâmetros  $\varphi_{1,i}$  e  $\varphi_{2,i}$  são valores aleatórios atualizados a cada iteração  $i$ . Também assumem valores reais  $pbest_i$  (*partial best*),  $gbest$  (*global best*) e  $X_i$ . O valor de  $V_i$  corresponde a uma probabilidade ( $\in [0, 1]$ ) obtido através da transformação lógica  $S(V_i)$  como definido pela regra (KENNEDY; EBERHART, 1997):

$$X_i = \begin{cases} 1, & rand(0, 1) < S(V_i) \\ 0, & \text{caso contrário.} \end{cases} \quad (2.3)$$

$$(2.4)$$

$$S(V_i) = \frac{1}{1 + \exp(-V_i)} \quad (2.5)$$

em que:

- $S(V_i)$  transformação sigmóide;
- $rand(0, 1)$  número aleatório distribuído no intervalo  $[0, 1]$ ;
- $V_i$  probabilidade que  $X_i$  seja 1 ou 0.

Para a primeira iteração ( $i = 1$ ), a posição inicial das partículas de cada agente são aleatoriamente é inicializada com valor 1 ou 0 para compor a solução inicial. A adequação de cada solução é calculada por meio da função objetivo, apresentada na Seção 2.2, assim como o valor de  $pbest_i$ . A partir do conjunto de soluções  $pbest_i$  de todos os agentes, é selecionada a melhor delas e atribuída a  $gbest$  (*global best*). Uma nova solução é proposta e atualizada por (2.1) iniciando uma nova iteração. A adequação (*fitness*) da nova solução é calculada através da função objetivo. Caso uma solução mais adequada seja encontrada, o valor de  $gbest$  é substituído. O procedimento é interrompido caso o número de iterações seja alcançado ou um determinado valor da função objetivo seja encontrado.

Os resultados obtidos são expressos apenas por números binários. No caso do problema de posicionamento de chaves, valor 1 em uma determinada posição do vetor  $gbest$  indica a presença de uma chave na referida posição. Valor nulo indica ausência de chave. O melhor resultado de cada iteração ( $i$ ) é mantido juntamente com um registro de suas posições anteriores em uma matriz de soluções de elite,  $P_{id}$ . As soluções contidas nesta matriz são posteriormente classificadas para formar a fronteira de Pareto.

## 2.2 Otimização multiobjetivo

Procedimentos de otimização dizem respeito à determinação de valores máximos ou mínimos de uma ou mais grandezas com o intuito de se obter melhor desempenho na solução de um problema. Estes valores extremos são chamados de soluções ótimas (DEB, 2009). Quando apenas uma grandeza é otimizada, o procedimento é chamado de *Single Objective Optimization* (SOO). Os métodos para solução de problemas de SOO são bem estabelecidos e aplicados a uma vasta gama de problemas. Entretanto, a maioria dos problemas científicos e de engenharia possuem mais de um objetivo (ou grandeza) a ser otimizado, o que ocasionou o desenvolvimento da área de estudo denominada de *Multi Objective Optimization* (MOO).

O que torna problemas de otimização multiobjetivo desafiante é que dois ou mais objetivos conflitantes devem ser manipulados em conjunto para alcançar uma solução ótima (COELLO; LAMONT; VELDHUIZEN, 2007). Custo e conforto são exemplos de objetivos conflitantes que compradores de um veículo se deparam. À medida que itens de conforto são acrescentados em um veículo, mais caro ele se torna. Lucro e custo de produção, custo e confiabilidade são exemplos de objetivos conflitantes encontrados em problemas de engenharia.

Uma das abordagens mais comuns para solucionar problemas de MOO é transformá-los em um problema de SOO. O *Weighted Sum Method* (WSM) é uma forma simples e eficiente de realizar esta tarefa. Com o WSM, cada objetivo é multiplicado por um peso compondo uma soma ponderada de objetivos, conforme a Equação (2.6) (DEB, 2003).

$$F(x) = \sum_{m=1}^M w_m f_m(x) \quad (2.6)$$

em que:

$$w_m \in [0, 1], m \in [1, 2, \dots, M] \text{ e} \quad (2.7)$$

$$\sum_{m=1}^M w_m = 1. \quad (2.8)$$

O valor de cada um dos pesos corresponde à importância ou preferência que o tomador de decisão atribui a cada objetivo. Desta forma, a questão resume-se a otimizar cada objetivo individualmente para obter uma solução. Porém, na maioria dos casos, a escolha dos pesos pode ser subjetiva, baseada em experiência e/ou não trivial.

Além disso, não é possível afirmar que uma determinada solução é ótima para um problema de MOO já que os objetivos, em geral, são conflitantes. Comparando-se duas soluções distintas, uma solução é melhor apenas para um dos objetivos dados ao mesmo tempo que a outra solução é melhor somente para o outro objetivo (DEB, 2009). Para esses casos, não é possível afirmar que uma solução é ótima. O procedimento adequado

consiste em encontrar um conjunto de soluções conflitantes e selecionar dentre elas uma solução que atenda às expectativas de desempenho requeridas.

Algoritmos Evolucionários (AE) são comumente aplicados para solucionar problemas de MOO. AE fundamentam-se no comportamento conjunto de pássaros, insetos ou peixes, convertendo cada indivíduo da população em agentes de busca de soluções para o problema. Portanto, cada elemento da população encontra uma solução individual para o problema. Assim, a cada ciclo do experimento computacional, diversas soluções são encontradas (COELLO; LAMONT; VELDHUIZEN, 2007). Esta característica deixa claro que os AE são adequados para a aplicação em problemas de MOO e para encontrar um conjunto de soluções.

### 2.2.1 Fronteira de Pareto

Apesar de os AE retornarem um conjunto de soluções como resposta, a maioria das soluções não é útil para o problema em questão. Apenas o sub-conjunto das soluções não-dominadas é efetivamente utilizado. Para abordar o conceito de dominância, o problema de posicionamento de chaves em redes de distribuição é adotado como exemplo.

O problema de posicionamento de chaves em redes de distribuição apresenta uma relação intrínseca entre custo e confiabilidade, de forma que melhorias na confiabilidade são alcançadas instalando-se chaves em posições estratégicas da rede de distribuição. À medida que mais chaves são adicionadas aumentam-se os custos envolvidos.

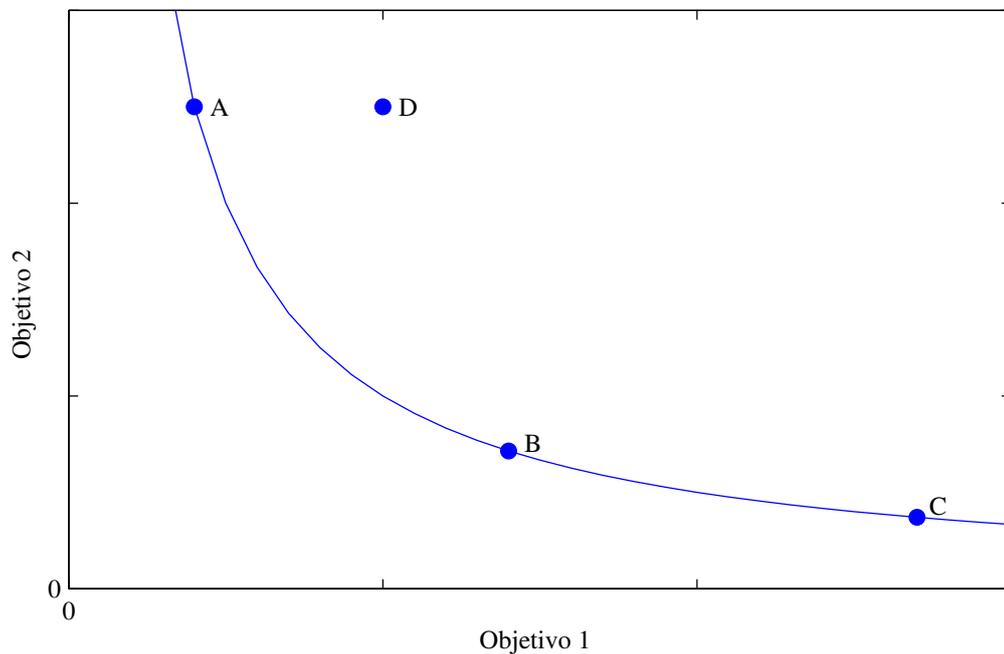
À medida que instalam-se mais chaves, a confiabilidade da rede elétrica também é incrementada, pois a quantidade de manobras disponíveis para isolamento de falhas e/ou transferências de carga é ampliada. Entretanto, uma maior quantidade de chaves implica em maiores custos. Portanto, a questão consiste em encontrar um ponto de equilíbrio entre a máxima confiabilidade para um dado custo como resultado do número de chaves.

A confiabilidade de um sistema pode ser medida de várias formas. Para este trabalho, propõe-se medi-la por meio da quantidade de Clientes Não Supridos (CNS), ou seja, aqueles consumidores que ficam desenergizados quando ocorrem faltas na rede de distribuição, logo, minimizando-se a quantidade de CNS tem-se a máxima confiabilidade.

A Figura 2.1 ilustra um conjunto de soluções para um problema de posicionamento de chaves em que ambos os objetivos devem ser minimizados. Cada letra representa uma solução específica. Comparando-se as soluções  $B$  e  $C$  nota-se que  $B$  é melhor que  $C$  para o objetivo 1 (número de chaves), porém a solução  $C$  é melhor que  $B$  para o objetivo 2 (CNS). Portanto, nenhuma das soluções ( $B$  ou  $C$ ) pode ser declarada a mais adequada ou ótima para ambos objetivos. Cada solução é melhor que a outra apenas em um dos objetivos.

Já o caso das soluções  $A$  e  $D$  é distinta. Ambas têm o mesmo valor para o objetivo

Figura 2.1 – Fronteira Pareto-ótima.



2, porém para o objetivo 1,  $A$  é melhor. Portanto, não há vantagem em utilizar  $D$  como solução em comparação com  $A$ . A solução  $D$  é considerada inferior, pois tem o mesmo valor para o objetivo 2 e é inferior para o objetivo 1. Diz-se que a solução  $D$  é dominada pela solução  $A$  que, por sua vez, não é dominada por nenhuma outra solução. Assim,  $A$  é uma solução não-dominada.

Portanto, o conjunto de soluções não-dominadas para um problema específico compõem a fronteira de Pareto conforme as definições seguintes (DEB, 2009).

**Definição 1** Para um conjunto de soluções  $P$ , o sub-conjunto de soluções não-dominadas  $P'$  são aquelas que não são dominadas por nenhuma outra solução do conjunto  $P$ .

**Definição 2** O conjunto de soluções não-dominadas do espaço de busca  $S$  é chamado de fronteira de Pareto global.

A partir das definições anteriores, nota-se que nenhuma das soluções  $A$ ,  $B$ , e  $C$ , da Figura 2.1, domina nenhuma outra solução e também não é dominada por nenhuma outra. Desta forma, as soluções compõem o conjunto de soluções Pareto-ótimas e a linha formada pelo conjunto de todas as soluções é chamada de fronteira de Pareto global ou simplesmente fronteira de Pareto.

## 2.3 Teoria dos grafos

Os grafos são amplamente utilizados neste trabalho para representar a topologia das redes de distribuição. Portanto, na presente Seção são apresentadas brevemente diversas definições formais sobre teoria dos grafos utilizados na fundamentação deste trabalho. Além disso, são adicionados rápidos comentários para contextualizar o conceito matemático apresentado e sua aplicação neste trabalho.

### 2.3.1 Grafos

Um grafo  $G = (V, E)$  é uma estrutura matemática formada por dois conjuntos finitos  $V$  e  $E$ . Os elementos de  $V$  são chamados de vértices ou nós; e os elementos de  $E$  são chamados de arestas ou arcos. Cada aresta liga dois nós do grafo, chamados pontos finais (GROSS; YELLEN, 2005).

Desta forma, pontos de carga da rede de distribuição, barramentos e transformadores são representados como sendo nós de um grafo. Já os elementos de disjunção, tais como disjuntores, religadores, chaves de qualquer tipo são representados como arestas de um grafo.

### 2.3.2 Grafo simples

Um *auto-loop* é uma aresta que liga um determinado nó a ele mesmo. Já uma multi-aresta é um conjunto de duas ou mais arestas que possui os mesmos pontos finais. Desta forma, é possível definir um grafo simples como sendo um grafo que não possui *auto-loops* nem multi-arestas.

As redes de distribuição radiais são representadas basicamente como grafos simples, uma vez que não há ciclos fechados.

### 2.3.3 Percursos, caminhos e ciclos

Em um grafo  $G$ , um percurso a partir de um nó de origem,  $v_0$ , até um nó de destino,  $v_n$ , é definido como sendo uma sequência alternada de nós e arestas, tal que  $endpoints(e_i) = v_{i-1}, v_i$ , para  $i = 1, \dots, n$  (GROSS; YELLEN, 2005), conforme mostrado na Equação 2.9.

$$P = \langle v_0, e_1, v_1, e_2, \dots, v_{n-1}, e_n, v_n \rangle \quad (2.9)$$

Para o caso de sistemas de distribuição representados por meio de grafos simples, existe apenas uma aresta entre dois nós consecutivos, assim a representação pode ser

abreviada como sendo uma sequência somente de nós, conforme representado pela Equação (2.10).

$$P = \langle v_0, v_1, \dots, v_n \rangle \quad (2.10)$$

Considerando ainda que numa rede de distribuição não haverá nós repetidos, um caminho é definido como sendo um percurso sem repetição de nós (GROSS; YELLEN, 2005).

Por fim, um ciclo é definido como sendo um caminho fechado (GROSS; YELLEN, 2005). Para o caso dos sistemas elétricos radiais, tais caminhos são evitados, uma vez que a característica radial da rede elétrica ficaria comprometida.

### 2.3.4 Matriz de adjacência

Há diversas representações matriciais de estruturas de grafos e sua utilização está associada à necessidade de realizar cálculos e operações diversas (NETO, 2011). A matriz de adjacência  $\mathbf{A}(G)$ , de um grafo  $G(V, E)$ , é uma matriz que possui a mesma ordem de  $\mathbf{G}$ , na qual cada coluna e cada linha está associada a um vértice. Elementos de  $\mathbf{A}$ ,  $a_{ij}$  com valor 1 indicam que existe aresta ligando os vértices  $i$  e  $j$ , já os valores 0 indicam que não há ligação, conforme (2.11).

$$\begin{aligned} \mathbf{A}(G) &= [a_{ij}] \\ a_{ij} &= 1 \Leftrightarrow \exists (i, j) \in E \\ a_{ij} &= 0 \Leftrightarrow \neg \exists (i, j) \in E \end{aligned} \quad (2.11)$$

### 2.3.5 Busca em profundidade

Diversos algoritmos de busca utilizam-se integralmente dos clássicos algoritmos de busca em largura (*breadth-first search*) e busca em profundidade (*depth-first search*) como parte de suas implementações (GROSS; YELLEN, 2005). Basicamente, estes algoritmos diferem em termos da regra para determinar o próxima aresta (*nextEdge*).

A busca em profundidade inicia-se selecionando uma aresta adjacente ao nó inicial, também chamado de raiz, adicionada mais recentemente à árvore junto com o nó de destino desta mesma aresta. O procedimento se repete para o nó de destino da aresta. Cada aresta é também marcada ao ser verificada. Quando não há mais arestas a serem verificadas o algoritmo retrocede (*backtracking*), ou seja, retorna para o nó descoberto mais recentemente e repete o procedimento. O Algoritmo 1 descreve a busca em profundidade através de pseudo-código.

**Algoritmo 1** Busca em profundidade.*Input:* grafo  $G$ , nó inicial  $v$ ,  $S$  conjunto de arestas adjacentes a  $v$ *Output:* árvore de espalhamento  $T$  de  $G$  com raiz  $v$ 


---

```

 $T \leftarrow v$ 
 $S \leftarrow \text{atualizaFronteira}(G, S)$ 
while  $S \neq \emptyset$  do
  fazer  $e = \text{proximaAresta}(G, S)$ 
  fazer  $w$  nó adjacente a  $e$ 
  adicionar  $e$  e  $w$  a  $T$ 
   $\text{atualizaFronteira}(G, S)$ 
end while
return  $T$ 

```

---

### 2.3.6 Caminhos elementares

Dado um grafo  $G(V, E)$  onde o conjunto de vértices é  $V = v_1, v_2, \dots, v_m$  e o conjunto de arestas é  $E = e_1, e_2, \dots, e_n$ , define-se um caminho elementar como sendo uma sequência de vértices e arestas do formato  $e_1v_1e_2v_2e_3$  em que os elementos da sequência são distintos (CHEN, 1997).

Considerando-se um SEP representado por um grafo, um caminho elementar representa uma linha ou parte dela que liga um determinado ponto de carga (representado por um vértice) a uma fonte (outro vértice), por exemplo.

O conceito de caminhos elementares é aplicado no modelo de restauração apresentado no Capítulo 4, mais especificamente aplicado às funções `findPathsFromNodeToActive` (ver A.13) e `findCurrPathsFromNodeToActive` (ver A.14).

## 2.4 Redes de Petri Coloridas

Um Sistema a Eventos Discretos (SED) é um sistema dinâmico em que a evolução no tempo é determinada pela ocorrência de eventos físicos, possivelmente em intervalos regulares. É caracterizado pela mudança de estados, valores discretos, à medida que os eventos acontecem (CASSANDRAS; LAFORTUNE, 2008).

Redes de Petri Coloridas (RPC) consiste em uma linguagem gráfica para a construção de modelos de SED e análise de suas propriedades. RPC é ainda uma linguagem de modelagem que combina as funções das redes de Petri com linguagens de programação de alto nível (JENSEN; KRISTENSEN, 2009). A RPC adota a linguagem de programação CPN ML (do inglês, *Coloured Petri Net Modelling Language*) como a linguagem padrão que, por sua vez, fundamenta-se na linguagem de programação funcional Standard ML (ULLMAN, 1998).

RPC é voltada para o uso prático, pois permite a construção de modelos compactos

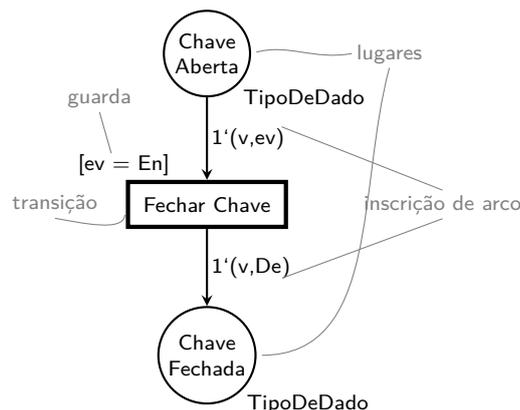
e parametrizados. A vantagem da RPC sobre as redes de Petri é a capacidade de modelar sistemas complexos gerando modelos com alto nível de abstração e melhor representação gráfica (JENSEN; KRISTENSEN, 2009).

Modelos RPC podem conter vários níveis de abstração. Podem ser construídos em um conjunto de módulos ou páginas que interagem entre si através de um conjunto de *interfaces* bem definidas, do mesmo modo que as linguagens de programação. O conceito de páginas em RPC fundamenta-se num mecanismo de estrutura hierárquica, que permite uma página possuir sub-páginas reutilizáveis em diversas partes do modelo.

RPC é uma ferramenta adequada para modelagem de Sistemas Elétricos de Potência (SEP), pois possui características determinantes, tais como modularidade, manutenibilidade e expansibilidade. Estas características permitem a adição de novos processos e novas funções a um modelo pré-existente mantendo a dinâmica do sistema real atualizada no modelo.

Na Figura 2.2 é apresentada a estrutura básica de uma RPC que consiste de lugares, transições e arcos. Os lugares representam os estados do sistema modelado. Os nomes dos lugares são escritos dentro de elipses ou círculos e não possuem significado formal, mas possuem um significado prático na legibilidade do modelo. Cada lugar pode ser marcado com uma ou mais fichas que contém um determinado valor associado. Este valor é chamado de cor. O estado do sistema, ou seja, a marcação do modelo, é representado pela quantidade de fichas, suas cores e em que lugares estas fichas estão contidas.

Figura 2.2 – Representação de uma Rede de Petri Colorida.



As transições representam os eventos que podem ocorrer no sistema. Cada transição é representada por um retângulo com o nome no seu interior. Quando o evento representado pela transição ocorre, diz-se que a transição ocorre ou é disparada. Quando uma transição é disparada, fichas são removidas dos lugares de entrada (lugares de origem de um arco) da transição e fichas são adicionadas aos lugares de saída (lugares de destino de um arco) da transição. A remoção e adição de fichas é determinada pela expressão de arco (inscrições próximas a cada arco). As expressões de arco são escritas utilizando a linguagem CPN ML

(ULLMAN, 1998) e utilizam-se de variáveis, constantes, operadores e funções. Quando todas as variáveis em uma expressão são ligadas ao tipo de dado correto, a expressão pode ser calculada.

Transições também podem ter guardas, ou seja, expressões Booleanas, que inibem o disparo da mesma. Quando uma guarda está presente, ela deve ter valor verdadeiro para que a transição possa estar habilitada, caso contrário, a transição é desabilitada e não pode ocorrer. Portanto, a guarda adiciona uma restrição para o disparo da transição.

## 2.5 Conclusão

O presente capítulo apresentou os conceitos aplicados para o desenvolvimento deste trabalho. No próximo capítulo é apresentada a contribuição para o posicionamento de chaves em redes de distribuição.

## 3 Posicionamento de Chaves em Redes de Distribuição

Neste capítulo é apresentada a contribuição para o posicionamento eficiente de chaves em redes de distribuição radiais visando a minimização do impacto das faltas à menor quantidade de clientes. Uma abordagem multiobjetivo fundamentada em Otimização por Enxame de Partículas é adotada para o desenvolvimento deste trabalho. Um estudo de caso aplicado ao sistema de testes *Roy Billinton Test System* é apresentado.

### 3.1 Trabalhos relacionados

Diversos trabalhos sobre posicionamento de chaves em SEP são encontrados na literatura. A maioria deles utiliza-se de técnicas computacionais para obter a melhor configuração de chaves sob critérios variados. Destacam-se os trabalhos propostos por (BILLINTON; JONNAVITHULA, 1996; CHEN et al., 2006; KONDO et al., 2013; FALAGHI; HAGHIFAM; SINGH, 2009; MORADI; FOTUHI-FIRUZABAD, 2008; ABIRI-JAHROMI et al., 2012).

Dentre as propostas para solução do problema de posicionamento de chaves em SEP, o trabalho proposto por Billinton e Jonnavithula (1996) destaca-se devido ao pioneirismo na utilização de técnicas heurísticas. Os autores propõem o uso de *Simulated Annealing* a uma função objetivo não-linear otimizada que considera os custos de investimento, manutenção e de faltas no SEP. A proposta apresenta como resultado a quantidade e o posicionamento ótimo de chaves no SEP. Similarmente, Chen et al. (2006) aplicam um *Immune Algorithm* para determinar a configuração ótima de chaves que minimiza o custo esperado por falta (ECOST) e o investimento necessário. O resultado é aplicado parcialmente ao SEP da concessionária *Taipower*, de *Taiwan*.

Moradi e Fotuhi-Firuzabad (2008) propõem a aplicação de *Particle Swarm Optimization* (PSO) para determinação do posicionamento e da quantidade ótima de chaves em SEP radiais. A função objetivo utilizada considera o ECOST (custo esperado de uma falta) (IEEE, 2012) para cada configuração de chaves proposta. Estudos de caso aplicando a metodologia proposta nos sistemas de teste IEEE 123 nodes (KERSTING, 2001) e *Roy Billinton Test System* (RBTS) (ALLAN et al., 1991) são apresentados.

A predominância de técnicas heurísticas para solução do problema de posicionamento de chaves em SEP é justificada pela dificuldade em tratar o problema de forma analítica e da característica não-linear das funções que determinam a localização das chaves

no SEP. Assim, os autores de (ABIRI-JAHROMI et al., 2012) propõem o uso de *Mixed Integer Linear Programming* (MILP) para determinar a solução ótima para o problema de posicionamento de chaves. O problema é equacionado para minimização do ECOST em um alimentador típico utilizando-se métodos analíticos.

Apesar das propostas anteriores apresentarem soluções ótimas, ainda que sobre uma perspectiva bem específica do problema de posicionamento de chaves em SEP, tais soluções podem ser inadequadas para aplicação prática nas concessionárias de energia devido ao alto custo (grande número de chaves demandado pelas soluções) ou pela inflexibilidade para sua aplicação. Como resultado, a aplicação destas soluções nas distribuidoras torna-se restrita. Desta forma, em (KONDO et al., 2013) é proposta uma abordagem prática que resulta em um número reduzido de chaves com pequenas perdas na confiabilidade do SEP. Porém, a solução apresentada apenas pode ser apontada como ótima em uma circunstância bem específica para a rede elétrica analisada, ou seja, com as chaves posicionadas exatamente nos pontos especificados na solução.

Os principais trabalhos disponíveis na literatura sobre posicionamento de chaves em SEP aplicam técnicas computacionais heurísticas para sua solução. Porém, as soluções apresentadas são de alto custo, pois demandam uma grande quantidade de chaves. Como consequência, as chances de que tais soluções tenham aplicação prática restrita nas concessionárias são consideráveis, conforme mostrado em (KONDO et al., 2013). Observa-se ainda que o problema de posicionamento de chaves em SEP é um problema de objetivos conflitantes (custo e confiabilidade), logo, não é possível apontar uma única solução ótima e definitiva. Sempre uma solução será superior a outra em um dos objetivos perseguidos. Daí vem a necessidade de aplicar ao problema uma abordagem multiobjetivo em que não apenas uma solução única será apontada como ótima, mas sim um conjunto de soluções. Como resultado, a tomada de decisão é facilitada em relação ao posicionamento de chaves em uma rede elétrica.

## 3.2 Algoritmo MOSP

Um algoritmo multiobjetivo para posicionamento de chaves em redes de distribuição (MOSP, do inglês *Multi Objective Switch Placement*) é proposto neste trabalho. O algoritmo fundamenta-se em PSO (KENNEDY; EBERHART, 1995) e utiliza uma abordagem multiobjetivo para tratar o conflito entre custo (quantidade de chaves) e confiabilidade (quantidade de consumidores afetados) inerente ao problema de posicionamento de chaves.

O algoritmo MOSP exhibe como resultado a configuração de chaves que minimiza o número de Clientes Não Supridos (CNS). Os dados de entrada básicos para o algoritmo são a topologia da rede elétrica e a quantidade de consumidores por ponto de carga.

A topologia da rede elétrica é representada através de Grafos (DIESTEL, 2005).

Uma matriz de adjacência ( $A$ ), conforme definido na Seção 2.3.4, contendo os nós, arestas, pesos e a relação entre eles é definida para a rede elétrica analisada. A partir da matriz  $A$ , é determinada a dimensão ( $D$ ) do problema que corresponde ao número máximo de chaves na rede de distribuição, dada por (3.1),

$$D = NC + NO - CB \quad (3.1)$$

em que  $NC$  é o número de ramos da rede de distribuição,  $NO$  corresponde ao número de chaves normalmente abertas e  $CB$  é o número de disjuntores.

Portanto, cada solução ( $x$ ) é representada por um vetor linha de dimensão  $D$ , em que cada elemento do vetor representa um ramo da rede elétrica. Os elementos de  $x$  podem assumir valores binários (1 ou 0) de forma que 1 indica que a referida posição contém uma chave telecomandada e 0 que não há chave na referida posição.

Uma população com  $N$  indivíduos ou agentes determina um conjunto de soluções que ao final das iterações, é classificada como dominada ou não-dominada. Assim, é feita a composição da fronteira de Pareto indicando as melhores soluções para o problema, ou seja, as posições onde devem ser instaladas chaves telecomandadas.

### 3.3 Descrição do algoritmo

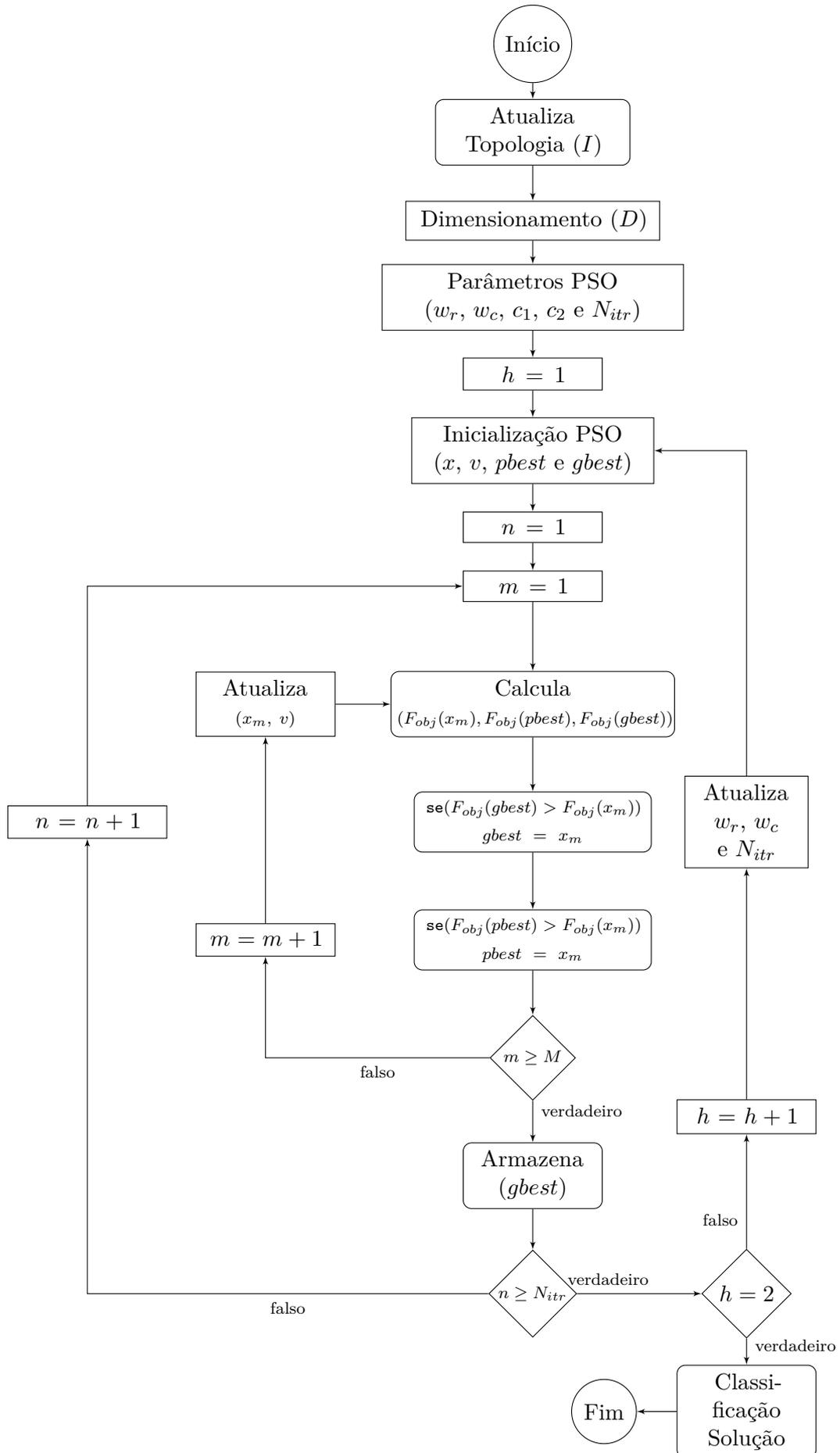
O algoritmo MOSP, cujo fluxograma é apresentado na Figura 3.1, é dividido em três partes: inicialização, busca e classificação. Na primeira etapa é inserida a topologia da rede elétrica de onde são determinados os valores para o restante da execução do algoritmo, como os fatores de aprendizagem ( $c_1$  e  $c_2$ ) e os pesos ( $w_r$  e  $w_c$ ). Na segunda parte, a busca é efetivamente realizada ao longo de um número determinado de iterações definido com base na experiência e testes realizados previamente. Na última parte, a fronteira de Pareto é determinada.

#### 3.3.1 Inicialização

Na primeira parte do algoritmo MOSP é feita, basicamente, a atribuição de valores e ajustes para o início das iterações. Antes do início do experimento computacional os fatores de aprendizagem ( $c_1$  e  $c_2$ ) e os pesos ( $w_r$  e  $w_c$ ) são ajustados. Os valores atribuídos aos pesos  $w_r$  e  $w_c$  são discutidos na subseção 3.3.4, já os valores atribuídos aos fatores de aprendizagem são discutidos a seguir.

Os fatores de aprendizagem, também conhecidos como coeficientes de aceleração, controlam o algoritmo de busca para incremento da diversidade das soluções ou para rápida convergência (GUO; CHEN; FENG, 2006). Adotando valores de  $c_1$  e  $c_2$ , tais que  $c_2 > c_1$ , a convergência é alcançada mais rapidamente. Fazendo o inverso,  $c_1 > c_2$ , as soluções obtidas são mais diversas reduzindo a probabilidade de mínimos locais. Para

Figura 3.1 – Fluxograma do algoritmo MOSP.



compor a fronteira de Pareto, a diversidade é prioritária em relação à rápida convergência, logo, a segunda estratégia de atribuição é adotada ( $c_1 > c_2$ ).

A maioria dos trabalhos ajusta os fatores de aprendizagem para  $c_1 = c_2 = 2$ , Kennedy e Eberhart (1995) Shi e Eberhart (1999). Assim, para este trabalho,  $c_1 = 2$  e  $c_2 = 2$  foram adotados para os testes iniciais. Após diversas simulações para refinamento dos valores, os fatores de aprendizagem foram ajustados em  $c_1 = 4$  e  $c_2 = 2$  para as simulações realizadas. Para  $c_1 > 4$  não ficou evidente uma melhora na diversidade das soluções.

### 3.3.2 Busca

A busca é dividida em dois estágios ( $h$ ) para permitir o ajuste dos pesos ( $w_r$  e  $w_c$ ) e o número de iterações ao longo da simulação. No primeiro estágio é encontrado um valor de referência para a confiabilidade, independentemente do custo para sua obtenção, ou seja, uma aproximação da configuração ótima (que resulta em maior confiabilidade) de chaves independente da quantidade de chaves necessária. A partir desta referência, o segundo estágio do algoritmo conduz à redução do número de chaves.

A solução  $x_{pbest}$  (*partial best*) corresponde à melhor solução encontrada por um indivíduo da população ao final de cada iteração. A solução  $x_{gbest}$  (*global best*) corresponde à melhor solução dentre todas as soluções *pbest*. As soluções *pbest* e *gbest* são aleatoriamente inicializadas, como apresentado na Figura 3.1, e comparadas entre si ao final de cada passo.

### 3.3.3 Classificação

Cada solução *gbest* é armazenada ao final de cada iteração como apresentado na Figura 3.1. Assim, o conjunto de soluções armazenadas é classificado entre soluções dominadas e soluções não-dominadas, formando a base para a composição da fronteira de Pareto.

### 3.3.4 Funções objetivo

A função objetivo é a parte central do algoritmo, pois determina a adequação de cada solução sugerida pelos indivíduos da população a cada iteração. Cada solução ( $x$ ) apresenta um conjunto de chaves que minimiza o número de consumidores afetados por faltas. Duas funções objetivo são aplicadas para determinar a melhor solução a partir da topologia da rede e do número de consumidores em cada ponto de carga. Não são necessários dados de confiabilidade da rede, ou seja, taxas de falha, o que reduz a complexidade das funções. Os trabalhos encontrados na literatura (ABIRI-JAHROMI et al., 2012; KONDO et al., 2013; CHEN et al., 2006; BILLINTON; JONNAVITHULA, 1996;

MORADI; FOTUHI-FIRUZABAD, 2008) utilizam-se frequentemente de taxas de falhas para composição da função objetivo. Porém, em aplicações práticas, observou-se que a disponibilização destes dados pelas concessionárias é rara devido à ausência da informação ou ao uso restrito. Por isso, buscou-se evitar o uso deste dado e observar a qualidade das soluções.

As duas funções objetivo são combinadas para compor a função  $F_{obj}$ , dada em (3.2). A função  $f_{cns}$  calcula o número de CNS (Clientes Não Supridos) pelas faltas. A função  $f_{qsw}$  calcula o número de chaves de uma dada solução.

$$F_{obj}(x) = w_r f_{cns}(x) + w_c f_{qsw}(x) \quad (3.2)$$

em que  $w_r$  e  $w_c$  são definidos como os pesos de confiabilidade e de custo, respectivamente.

Os pesos representam a importância ou prioridade que cada objetivo tem para a simulação. É importante notar que os valores dos pesos são atribuídos segundo as restrições (3.3) e (3.4).

$$w_r + w_c = 1 \quad (3.3)$$

$$w_r \text{ e } w_c \geq 0 \quad (3.4)$$

No primeiro estágio ( $h = 1$ ),  $w_c = 0$  e  $w_r = 1$ , ou seja, a confiabilidade tem prioridade sobre o custo. O objetivo, neste estágio, é obter um valor de referência para o número de chaves. No segundo estágio, é atribuído a  $w_c$  um valor diferente de zero. O valor típico para os pesos ( $w_c = w_r = 0,5$ ) foi adotado em diversas simulações para indicar que custo e confiabilidade são igualmente importantes. Quando são adotados valores, de forma que  $w_r > w_c$ , a fronteira de Pareto tende a apresentar soluções mais confiáveis para o sistema elétrico (menor CNS) e de maior custo (maior número de chaves). Por outro lado, quando  $w_c > w_r$  a fronteira de Pareto tende a apresentar soluções de menor custo e menor confiabilidade para o sistema elétrico.

A função objetivo  $f_{cns}$ , que calcula o número de consumidores não supridos<sup>1</sup>, é dada pela Equação (3.5).

$$f_{cns} = \frac{\sum_{j=1}^b q_j}{CS} \quad (3.5)$$

em que  $b$  corresponde ao número de ramos da rede de distribuição,  $q_j$  ao número de consumidores afetados pela falta no ramo  $j$  e  $CS$  o número total de consumidores atendidos pela rede.

Assim,  $f_{cns}$  calcula o número de CNS quando um determinado conjunto de chaves está instalado na rede de distribuição.

<sup>1</sup> A função  $f_{cns}$  é inspirada no indicador de confiabilidade SAIFI (*System Average Interruption Frequency Index*) (IEEE, 2012)

A função objetivo  $f_{qsw}$ , que avalia o custo da solução  $x$  e está diretamente relacionado com a quantidade de chaves, é dada por (3.6).

$$f_{qsw} = \frac{qsw_m}{NS} \quad (3.6)$$

em que  $qsw_m$  é o número de chaves da solução  $x_m$  e  $NS$  é a quantidade máxima de chaves para qualquer configuração possível.

### 3.4 Estudo de caso

O sistema de testes *Roy Billinton Test System* (RBTS) é comumente adotado na literatura para estudos de confiabilidade de SEP e testes de algoritmos. Parâmetros como descrição de cargas, comprimento de alimentadores, número de consumidores, taxas de falha de equipamentos e a topologia da rede elétrica são disponibilizados em (ALLAN et al., 1991). Também são apresentadas duas redes de distribuição: RBTS *Bus 2* e RBTS *Bus 4* (RBTS4). O algoritmo proposto é aplicado ao RBTS4, apresentado na Figura 3.2. Este sistema de teste possui 7 alimentadores e 4.779 consumidores.

Os resultados do experimento computacional, que é dividido em dois estágios, são apresentados na Figura 3.3. O primeiro estágio busca a configuração de chaves mais confiável sem considerar a quantidade de chaves necessária para isto, que é obtido fazendo-se  $w_c = 0$ . Os demais valores atribuídos para os parâmetros do PSO e os pesos da função objetivo são apresentados na Tabela 3.1.

Os fatores de aprendizagem são mantidos entre os dois estágios,  $c_1 = 4$  e  $c_2 = 2$ , para incremento da diversidade em função da convergência prematura. Estes valores foram determinados empiricamente após diversos experimentos computacionais. Observou-se que valores maiores de  $c_1$  não influenciaram significativamente os resultados, assim como  $c_2$ .

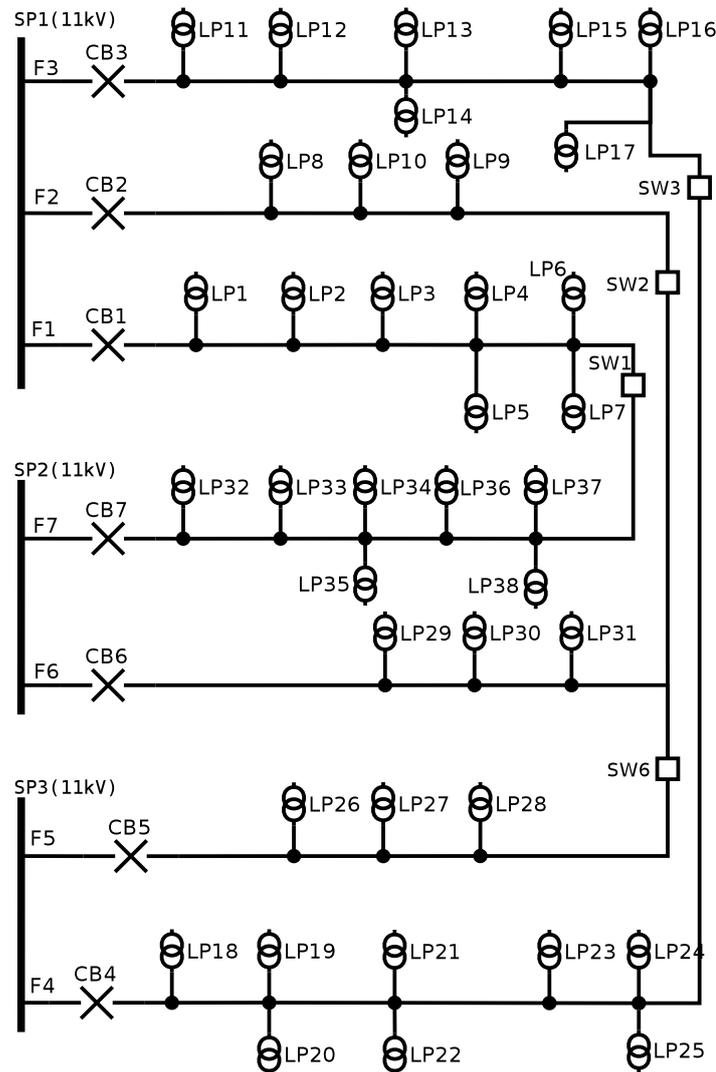
Tabela 3.1 – Parâmetros do PSO e pesos para os experimentos do RBTS4.

Parâmetro	$c_1$	$c_2$	$w_r$	$w_c$	$N_{itr}$
Estágio 1	4	2	1.0	0.0	20
Estágio 2	4	2	0.5	0.5	60

A partir da Figura 3.3 pode-se observar uma tendência de decréscimo no CNS no primeiro estágio (iteração 1 a 20) e incremento no segundo estágio (iteração 21 a 80). O número de chaves cresce no primeiro estágio e é fortemente reduzido no segundo estágio. Este comportamento é explicado pelos valores de  $w_c$  atribuídos, ou seja,  $w_c = 0$  no primeiro estágio e  $w_c = 0.5$  no segundo.

Apesar de o algoritmo MOSP demonstrar ser capaz de reduzir a quantidade de chaves para valores práticos, não é possível afirmar que a quantidade final de chaves

Figura 3.2 – RBTS Bus 4.



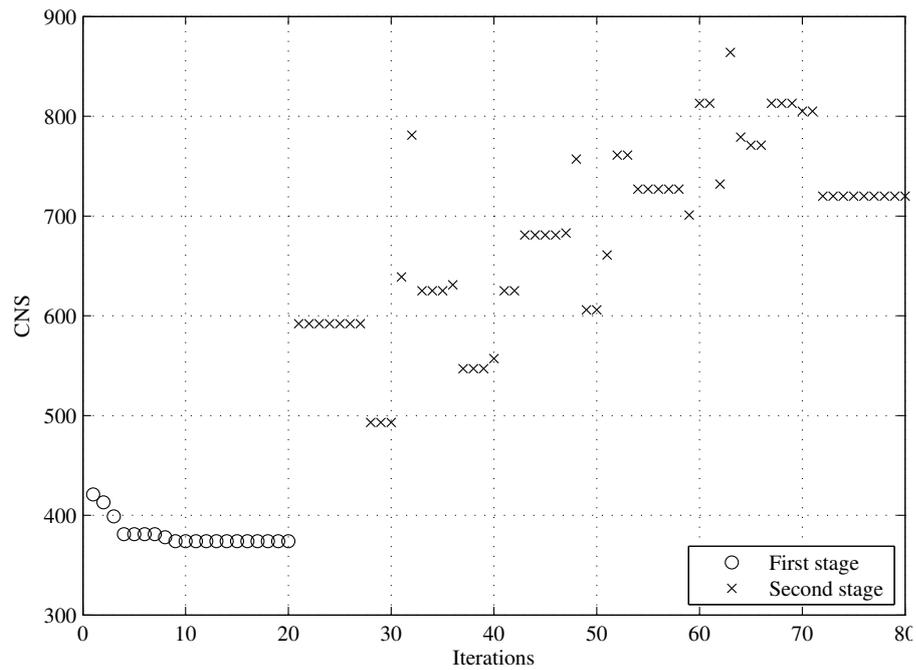
Fonte: Adaptado de (ALLAN et al., 1991).

é ótima. Informações subjetivas são necessárias para estabelecer a melhor solução, tais como o nível de confiabilidade esperado pela concessionária e o custo admissível para a quantidade de chaves necessária.

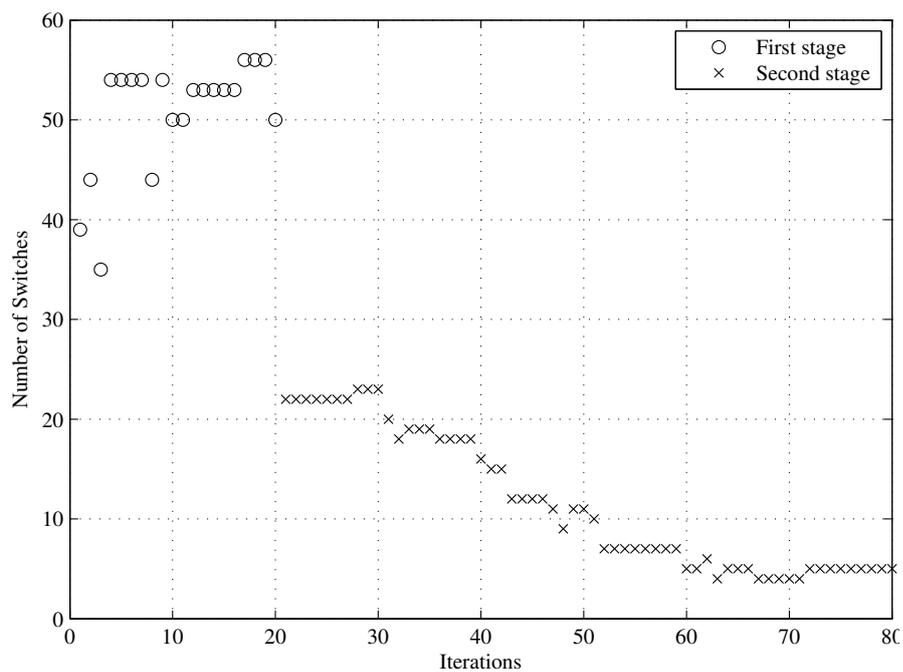
Portanto, a fronteira de Pareto é útil para a escolha de uma solução que atenda às expectativas da concessionária em termos de confiabilidade e custo em um conjunto de soluções não-dominadas. A fronteira de Pareto obtida para o RBTS4 é apresentada na Figura 3.4. Resultados numéricos são apresentados na Tabela 3.2.

Conforme apresentado na Tabela 3.2, a solução #1 é a mais barata e a de menor confiabilidade. Por outro lado, a solução #11 é a de maior custo e a mais confiável. As outras soluções na fronteira de Pareto possuem vantagem em relação a um objetivo e desvantagem em relação a outro objetivo.

Uma comparação é feita em (BEZERRA et al., 2015) entre os resultados obtidos



(a)



(b)

Figura 3.3 – Resultados da simulação para o RBTS *Bus 4*. (a). CNS. (b). Número de chaves.

da aplicação do algoritmo MOSP e os resultados apresentados em (ABIRI-JAHROMI et al., 2012) em que técnicas de programação inteira linear mista (*Mixed-Integer Linear Programming*, MILP) são aplicadas. A técnica de MILP resulta em um conjunto de 22 chaves como quantidade ótima de chaves para o caso em que a configuração radial com recurso é utilizada. Para uma comparação mais fidedigna, o CNS do conjunto de chaves

Figura 3.4 – Fronteira de Pareto para o RBTS4.

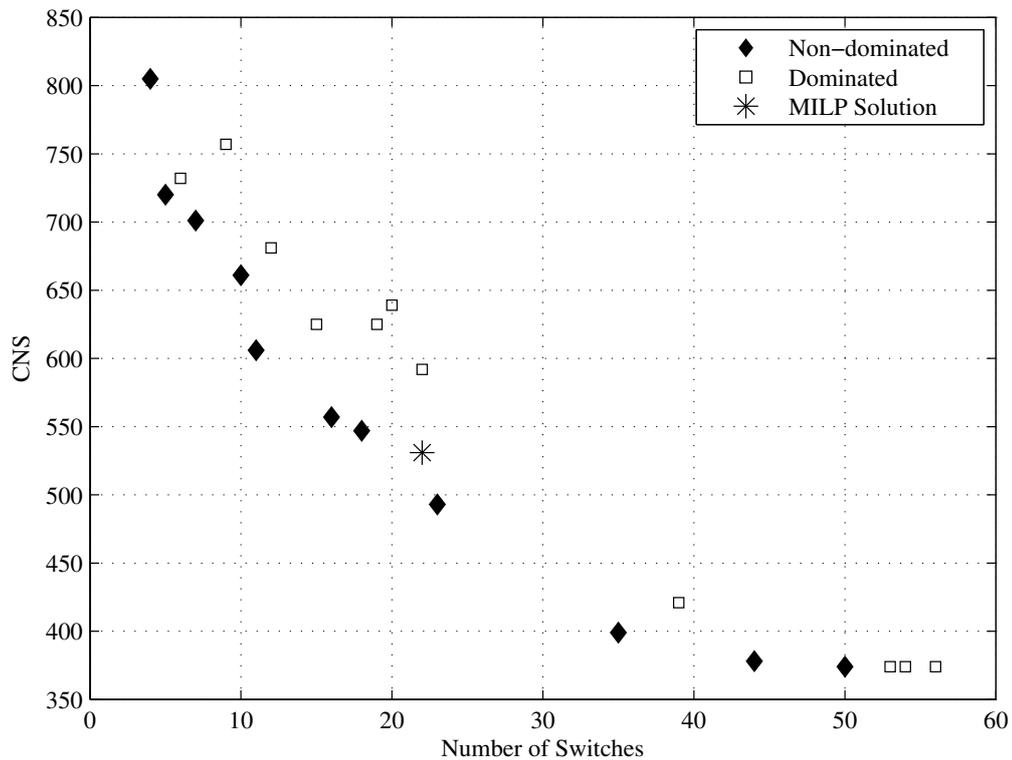


Tabela 3.2 – Conjunto de soluções não-dominadas para o RBTS4.

Solução (#)	Número de chaves	CNS
1	4	805
2	5	720
3	7	701
4	10	661
5	11	606
6	16	557
7	18	547
8	23	493
9	35	399
10	44	378
11	50	374

resultante de (ABIRI-JAHROMI et al., 2012) foi calculado utilizando a mesma função objetivo ( $f_{cns}$ ), dada em (3.5), do algoritmo MOSP, resultando em 531 consumidores. A solução mais próxima, dentre as 11 soluções não-dominadas apresentadas na Tabela 3.2 é a solução #7, conforme apresentado na Figura 3.4. O algoritmo MOSP apresenta como resultado uma quantidade de 18 chaves, inferior a formulação MILP proposta por (ABIRI-JAHROMI et al., 2012). A localização das chaves para as soluções dadas por

ambas as abordagens são apresentadas na Tabela 3.3.

Tabela 3.3 – Comparativo entre a localização das chaves.

Solução	Localização
MOSP #7	17 18 23 10 11 6 7 8 9 48 49 37 26 31 32 SW1 SW6 SW3
MILP (ABIRI-JAHROMI et al., 2012)	19 23 10 11 12 13 5 9 47 51 38 39 40 41 42 33 34 35 36 37 26 28 32

A formulação MILP proposta por (ABIRI-JAHROMI et al., 2012) apresentou resultado ótimo global em um número finito de passos ao invés de ótimo local (HORVATH et al., 2009). Apesar de o algoritmo MOSP não ser capaz de provar a obtenção da solução ótima global, a proximidade das soluções apresentadas demonstra a eficiência do algoritmo em encontrar soluções adequadas para o problema de posicionamento de chaves ao invés de uma solução única (BEZERRA et al., 2015). Solução esta, que, muitas vezes, não atende às necessidades de custo e/ou confiabilidade da concessionária de energia. Assim, o conjunto de soluções não-dominadas traz flexibilidade para a tomada de decisão em relação ao posicionamento de chaves.

## 3.5 Melhorias de desempenho do algoritmo MOSP

Os resultados para o posicionamento eficiente de chaves apresentados até a Seção anterior foram obtidos até a etapa de qualificação. Após esta etapa buscou-se um refinamento do desempenho do algoritmo MOSP. Foram aplicadas basicamente três modificações. A primeira mudança refere-se a um novo procedimento de inicialização do algoritmo, a segunda diz respeito à forma de composição da fronteira de Pareto e a última uma modificação na função objetivo.

### 3.5.1 Inicialização induzida

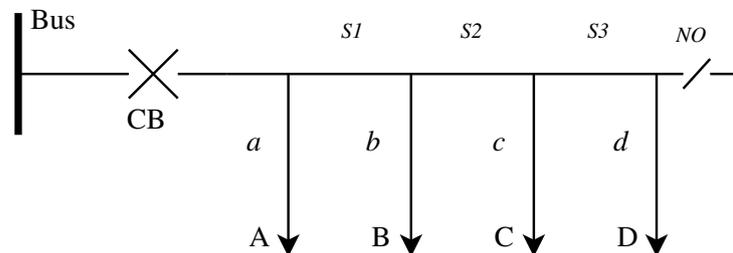
Ao invés de inicializar as chaves em uma posição totalmente aleatória, passou-se a utilizar uma inicialização pseudo-aleatória ou induzida. Sabe-se que as chaves que mais contribuem para o isolamento dos defeitos são aquelas localizadas no ramo principal do alimentador, pois estas possuem maior capacidade de manobra em comparação com as chaves posicionadas nos ramos secundários, que limitam-se a isolar defeitos apenas destes ramos. Assim, adotou-se como melhoria para o procedimento de inicialização privilegiar a escolha das chaves posicionadas no ramal principal (tronco) dos alimentadores, tornando-as preferenciais, em detrimento das chaves posicionadas nos ramos laterais. O que não significa, porém, que ao longo da execução do MOSP chaves não possam ser posicionadas em ramos laterais, pois a modificação trata apenas da posição inicial das chaves na rede elétrica.

Diante disso, buscou-se automatizar o processo de identificação do ramo principal de um alimentador de uma dada rede de distribuição. Porém, os estudos não foram conclusivos e esta identificação foi realizada manualmente e não de forma automatizada. Por meio da inspeção do diagrama unifilar da rede elétrica seleciona-se aquelas posições preferenciais para a instalação das chaves. Portanto, as posições selecionadas são marcadas e representadas utilizando-se de um vetor.

A inicialização é realizada buscando tornar aleatória não apenas a posição das chaves, mas também a quantidade de chaves que compõe a solução. Portanto, uma solução pode conter desde uma única chave até a quantidade máxima de chaves possível. Desta forma, uma distribuição de *Poisson* (WALPOLE et al., 2007) é utilizada para determinar a quantidade inicial de chaves da solução de cada indivíduo. Em seguida, uma distribuição binomial é aplicada para determinar o posicionamento das chaves em posições aleatórias.

Para exemplificar o processo de inicialização considera-se a rede elétrica ilustrada na Figura 3.5, que possui um ramo principal com 3 segmentos ( $S1$ ,  $S2$  e  $S3$ ), 4 ramos laterais ( $a$ ,  $b$ ,  $c$  e  $d$ ) e ainda o segmento que contém a chave  $NO$ , normalmente aberta. Assim, as chaves preferenciais seriam aquelas posicionadas para combinações de  $S1$ ,  $S2$ ,  $S3$  ou  $NO$ .

Figura 3.5 – Rede elétrica simplificada.



Para aplicação do procedimento de inicialização descrito para cada indivíduo da população do enxame, utilizou-se uma estratégia que aplica duas funções de densidade de probabilidade. Primeiramente a distribuição de *Poisson* (WALPOLE et al., 2007), para sugerir a quantidade de chaves ( $q$ ) em uma solução inicial ( $x_0$ ), e em seguida a distribuição de *Bernoulli* (WALPOLE et al., 2007) para posicionar  $q$  chaves em  $x_0$ .

O parâmetro  $\lambda$ , conforme definido na Equação (3.7), corresponde à média e à variância da distribuição (WALPOLE et al., 2007). Como regra prática, adotou-se um valor de  $\lambda$  equivalente a um terço da quantidade de chaves preferenciais<sup>2</sup>. Assim, para a rede elétrica representada na Figura 3.5, que possui 4 chaves preferenciais ( $Q_{PREF}$ ),  $S1$ ,  $S2$ ,  $S3$  e  $NO$ ,  $\lambda = Q_{PREF}/3 = 4/3$ .

$$f(q | \lambda) = \frac{\lambda^q}{q!} e^{-\lambda}; q = 0, 1, 2, \dots, Q_{PREF} \quad (3.7)$$

<sup>2</sup> Tomou-se o cuidado de evitar o valor 0 na inicialização o que representaria a ausência de chaves.

Os segmentos da rede elétrica são representados através de um vetor, como mostrado em (3.8), que representa a rede elétrica da Figura 3.5. O vetor  $x_0$ , em (3.8), indica que os ramos  $S1$  e  $S2$  devem ter uma chave telecontrolada.

$$x_0 = \begin{bmatrix} a & S1 & b & S2 & c & NO & S3 & d \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.8)$$

A variável aleatória  $s$ , definida por meio da função densidade de probabilidade de *Bernoulli* (3.9), atribui valor para cada elemento de  $x_0$ . Valor 1 indica que o segmento associado deve conter uma chave telecontrolada (sucesso) e 0 caso contrário.

$$f(s | p) = p^s(1 - p)^{1-s} \quad (3.9)$$

O valor da probabilidade de sucessos é calculada por meio da Equação (3.10) dividindo-se a quantidade de chaves determinada pela distribuição de *Poisson* pela quantidade de segmentos da rede elétrica ( $Q_{NFNA}$ ), ou seja, a máxima quantidade de chaves possível.

$$p = \frac{f(q | \lambda)}{Q_{NFNA}} \quad (3.10)$$

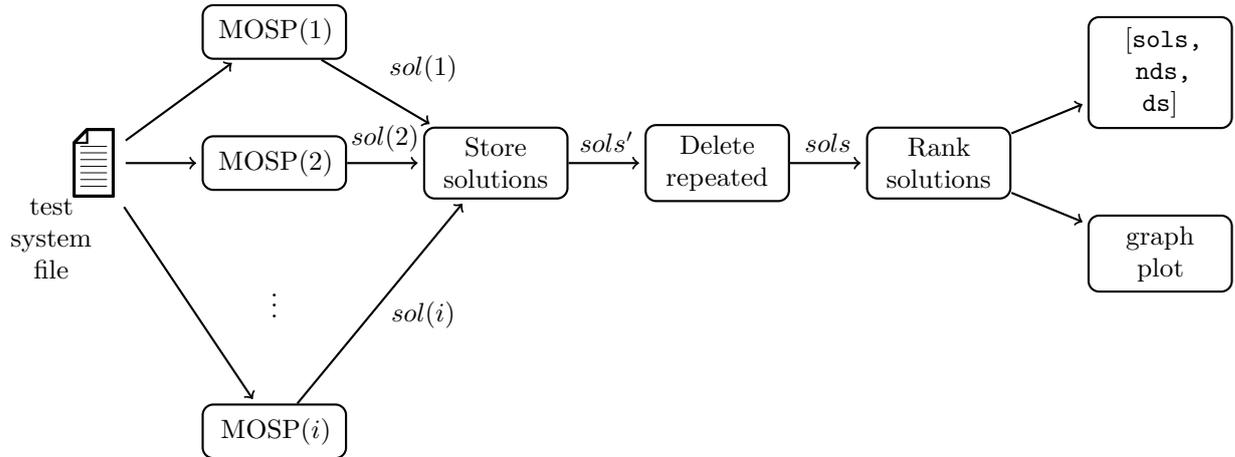
O código fonte do *script* em MATLAB<sup>®</sup> que realiza as operações descritas nesta subseção pode ser conferido no Apêndice B, Código Fonte B.1, `initSols.m`.

### 3.5.2 Múltiplas instâncias em paralelo

A segunda melhoria aplicada diz respeito à forma de composição da fronteira de Pareto. Ao invés de considerar as soluções dominadas e não-dominadas de apenas um ciclo de execução do algoritmo, passa-se a executar diversas instâncias do MOSP em paralelo e armazenar as soluções de cada uma das instâncias. Desta forma, uma quantidade a diversidade das soluções é consideravelmente aumentada. As soluções geradas são comparadas entre si, eliminadas possíveis duplicidades e então agrupadas para compor a fronteira de Pareto. Como resultado, a quantidade de soluções que compõe a fronteira de Pareto não apenas é incrementada numericamente como também qualitativamente, uma vez que as buscas são realizadas em paralelo e em seguida comparadas. Em síntese, com a estratégia de utilizar-se de múltiplas instâncias, a busca é intensificada resultando num maior número de soluções.

O algoritmo MOSP, descrito na Seção 3.3, passa a ser o núcleo de execução de um novo fluxograma, conforme apresentado na Figura 3.6, porém com o novo procedimento de inicialização da população descrito.

Figura 3.6 – Fluxograma do algoritmo MOSP.



O *script* em MATLAB<sup>®</sup> que realiza as operações descritas nesta subseção pode ser conferido no Apêndice B, Código Fonte B.2, *mosp.m*.

### 3.5.3 Nova função objetivo

A função objetivo definida em (3.2) para o algoritmo MOSP sofreu modificações para contemplar a presença de chaves fusíveis nos ramos laterais e de chaves manobradas manualmente por equipes de campo. Estes equipamentos não foram considerados para o cálculo do CNS, mas apenas as chaves telecontroladas e os disjuntores de saída de alimentador. Porém, passando a considerar as chaves fusíveis e as chaves manuais, a representação da rede de distribuição torna-se mais relevante e mais próxima da realidade.

Com isso, o cálculo do CNS é sensivelmente alterado, uma vez que curto-circuitos nos ramos laterais são eliminados pelas chaves fusíveis isolando o restante da rede do defeito, reduzindo a probabilidade de posicionamento de uma chave telecontrolada neste ramo. Desta forma, apenas os consumidores do ramo lateral são contabilizados para o cálculo do CNS reduzindo seu valor numérico. Como consequência, uma chave tem chances reduzidas de ser posicionada nos ramos laterais.

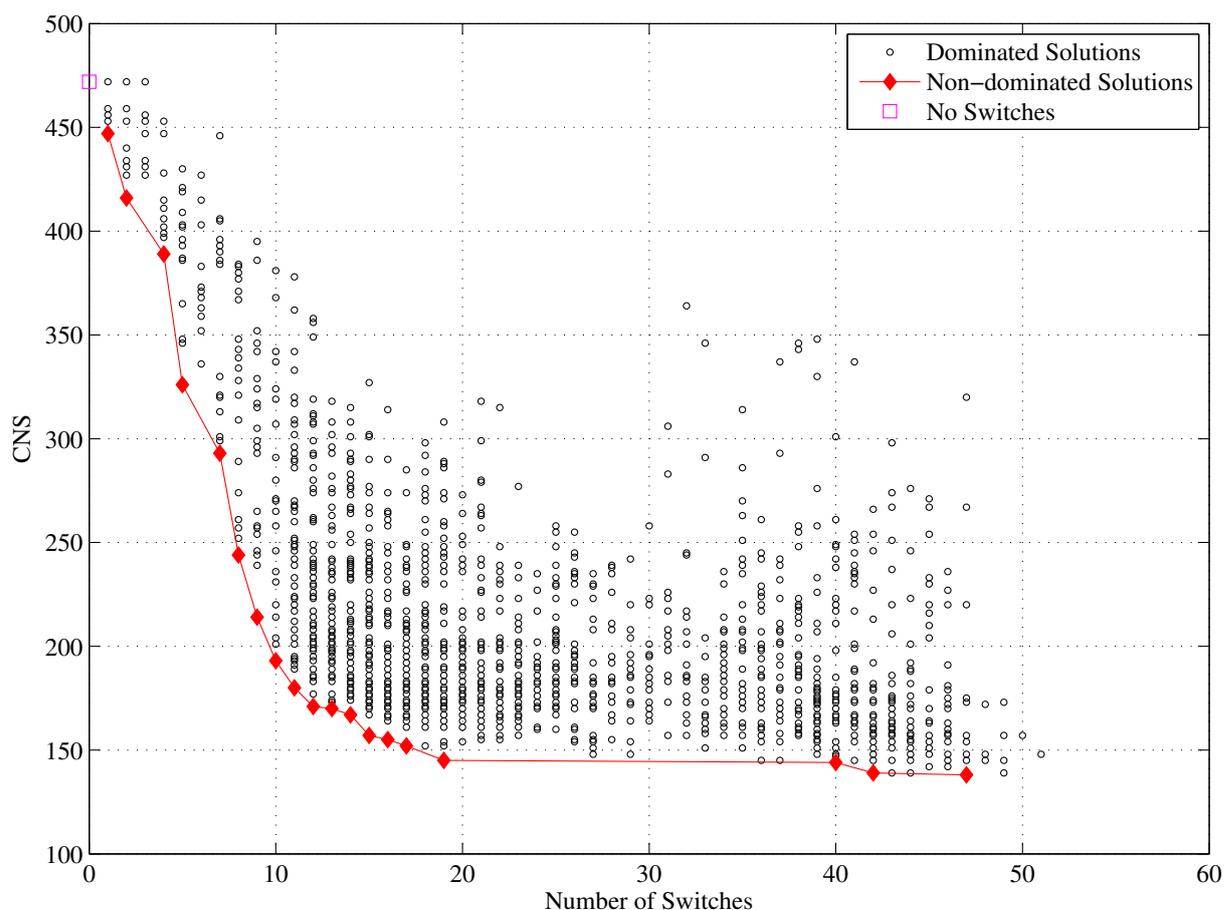
É importante notar que a formulação matemática da função objetivo  $f_{CNS}$ , dada por 3.5, permanece inalterada. Apenas ao considerar chaves fusível e chaves manuais o comportamento da proteção modificado

A versão final da função que realiza o cálculo do CNS pode ser conferida no Apêndice B, Código Fonte B.3, *cns.m*.

### 3.5.4 Resultados

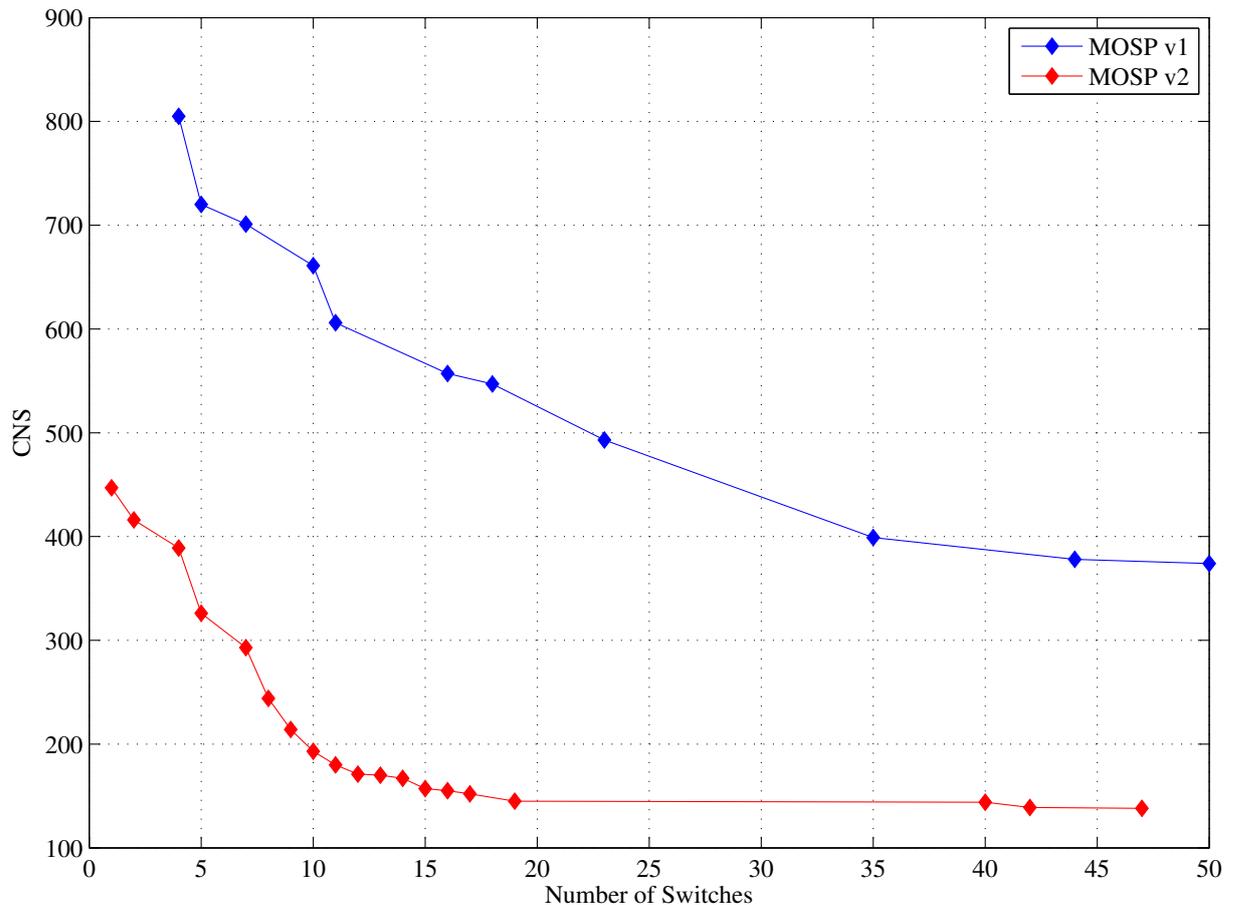
Após aplicadas as melhorias apresentadas anteriormente, o estudo de caso utilizando o sistema de teste RBTS4 foi refeito e novos resultados foram obtidos. Desta forma, 1.688 soluções foram obtidas, das quais 1.667 são dominadas e 19 não-dominadas. A Figura 3.7 apresenta a nova fronteira de Pareto, as soluções dominadas e o valor de CNS quando as chaves não são instaladas na rede elétrica.

Figura 3.7 – Nova fronteira de Pareto para o RBTS4.



Para comparar os resultados obtidos entre a primeira versão (MOSP v1) e a versão melhorada (MOSP v2) do algoritmo MOSP deve ser observado que o procedimento de cálculo do CNS foi modificado para contemplar a operação de chaves fusível e chaves operadas por equipes de campo como detalhado na subseção 3.5.3. Esta modificação torna a comparação entre os resultados injusta, uma vez que funções objetivo distintas estão sendo utilizadas. Assim, com intuito de apresentar uma comparação que ofereça uma percepção qualitativa das melhorias, as condições para executar o algoritmo MOSP antes das melhorias e com a nova função objetivo foram refeitas. A curva de Pareto obtida foi então sobreposta com a curva de Pareto obtida com o algoritmo MOSP melhorado. O resultado é apresentado na Figura 3.8.

Figura 3.8 – Fronteiras de Pareto obtidas com diferentes versões do MOSP.



Inicialmente, pode-se notar que a quantidade de soluções não-dominadas foi bastante ampliada na versão melhorada (MOSP v2) passando de 11 para 19 soluções, conforme mostrado na Figura 3.8. Observa-se ainda uma quantidade maior de soluções entre 1 e 4 chaves, consideradas soluções de menor custo.

Outra observação diz respeito à redução do valor geral do CNS. Enquanto que no MOSP v1 a solução com 5 chaves tem um CNS em torno de 700 consumidores, a respectiva solução obtida com MOSP v2 está em torno de 300 consumidores. Este comportamento é esperado, uma vez que a função objetivo foi modificada entre o desenvolvimento das duas versões do algoritmo, conforme detalhado anteriormente.

## 3.6 Conclusão

Neste capítulo foi apresentado o algoritmo MOSP (*Multi Objective Switch Placement*) para posicionamento de chaves em redes de distribuição de topologia radial. Demonstrou-se a capacidade do algoritmo de fornecer um conjunto de soluções (posicionamento de um conjunto de chaves) que minimiza a quantidade de Clientes Não Supridos (CNS) após a ocorrência de falta.

A aplicação do algoritmo apresentado para melhoria da confiabilidade da rede elétrica é de grande valor para concessionárias e consumidores. A redução do número de CNS tem como efeito a melhoria de diversos índices de confiabilidade adotados por agentes reguladores do sistema elétrico nacional (por exemplo, DEC e FEC ([PRODIST, 2014](#))) e de outros países (por exemplo, SAIDI e SAIFI ([IEEE, 2012](#))). Apesar de não obrigatória, a implementação prévia do algoritmo fornece subsídios que reforçam o desempenho de sistemas de reposição automática, uma vez que as chaves telecontroladas aumentam a eficiência das manobras pós-falta.

No próximo capítulo é apresentado o modelo para restauração automática proposto neste trabalho.

# 4 Modelo de Restauração do Sistema Elétrico

Neste capítulo são apresentados os procedimentos de restauração tradicionalmente adotados pelas distribuidoras de energia elétrica e as restrições operacionais envolvidas. Tais procedimentos e suas restrições fundamentam o modelo proposto neste trabalho para restauração automática de redes de distribuição radiais. Alguns dos principais trabalhos publicados na literatura são apresentados e discutidos. O modelo desenvolvido fundamenta-se em Redes de Petri Coloridas (RPC), uma vez que esta é uma ferramenta adequada para modelagem de sistemas a eventos discretos que permite a simulação, validação e posterior replicação do modelo em um sistema computacional.

## 4.1 Procedimentos atuais

A proteção do Sistema Elétrico de Potência (SEP) tem como função primária a eliminação de anomalias a fim de garantir a segurança operacional da rede elétrica. Este papel é desempenhado por relés de proteção dotados de rápidos microprocessadores com capacidade de comunicação e, em geral, integrados a sistemas de supervisão e controle (SCADA) ((FIRM), 2002). As principais funções destes dispositivos podem ser resumidas em Sallam e Malik (2011):

- Limitar o *stress* térmico, dielétrico e mecânico no SEP;
- Proteger vidas de eventuais faltas no SEP;
- Manter a estabilidade e continuidade de serviço.

Faltas em redes de distribuição são mais comuns do que em linhas de transmissão ou na geração de energia elétrica. A proximidade das linhas de distribuição com o tráfego urbano, a maior densidade de consumidores, a maior capilaridade da rede e desastres naturais ou urbanos são alguns dos fatores que contribuem para que as redes de distribuição sejam mais suscetíveis a faltas (SINGH, 2013; ZAVODA, 2010; BROWN et al., 2001). Desta forma, a restauração automática é um tema de maior relevância quando aplicado às redes de distribuição.

A restauração ou reconfiguração de redes de distribuição pós-falta é realizada basicamente por operadores de centros de controle. Resumidamente, a atividade dos operadores consiste em monitorar o estado dos alimentadores das subestações, avaliar as

circunstâncias em que as faltas acontecem e buscar alternativas para restauração da rede elétrica minimizando os prejuízos aos consumidores e às concessionárias.

O trabalho dos operadores é dificultado, principalmente, pelo curto intervalo de tempo disponível para restauração sem que haja penalidades para as concessionárias. No caso brasileiro, as concessionárias de distribuição de energia estão sujeitas a penalidades sempre que o tempo de desenergização for superior a 3 minutos (PRODIST, 2014). Esta restrição aumenta a pressão psicológica a que os operadores estão submetidos, aumentando também a probabilidade de erros humanos que possam comprometer a segurança do sistema elétrico, dos usuários e profissionais envolvidos na operação e manutenção de redes.

Além do curto período de tempo disponível, outros fatores dificultam a tarefa do operador. A complexidade das redes elétricas atuais, possíveis falhas na comunicação dos dispositivos telecontrolados, diagnóstico de defeitos equivocados, a avalanche de informações apresentadas e possíveis falhas durante a operação de equipamentos são alguns dos fatores que tornam o trabalho do operador bastante complexo.

Desta forma, restrições operacionais envolvidas na restauração dos SEP podem passar despercebidas pelo operador ao realizar manobras na rede elétrica. Ainda é possível que políticas de operação conservadoras adotadas pela concessionária ou pelo próprio operador impeça-o de realizar determinadas manobras que poderiam restaurar a rede elétrica de forma mais eficiente e célere.

## 4.2 Restrições operacionais

Algumas restrições inerentes à operação de um SEP devem ser observadas durante o processo de restauração. Dentre elas, as mais evidentes dizem respeito à capacidade de condução (ampacidade) dos alimentadores envolvidos no processo de restauração e à capacidade de fornecimento dos transformadores para alimentar novas cargas, assumidas em decorrência da recomposição da rede.

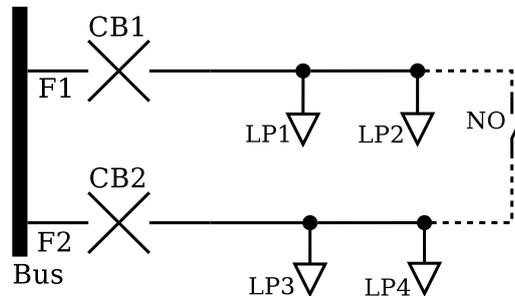
### 4.2.1 Ampacidade de cabos

A máxima corrente que pode ser percorrida por um condutor é referida como a ampacidade do mesmo. Durante o processo de restauração, a rede elétrica sofre modificações devido a manobras e, como consequência, um alimentador pode passar a suprir cargas não previstas originalmente.

A rede elétrica representada na Figura 4.1 auxilia a exemplificar este conceito. Em condições normais de fornecimento, o alimentador F1 atende às cargas LP1 e LP2 e o alimentador F2 atende às cargas LP3 e LP4. Porém, com o uso da chave NO e dos disjuntores CB1 e CB2, é possível realizar manobras fazendo com que as cargas de F1

(LP1 e LP2) sejam supridas por F2 e vice-versa. Assim, é necessário que a ampacidade dos condutores de ambos os alimentadores seja adequada para atendimento das cargas com segurança em qualquer condição de operação.

Figura 4.1 – Rede elétrica com dois alimentadores.

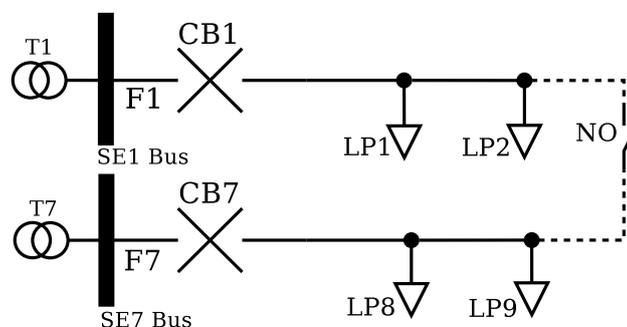


#### 4.2.2 Capacidade de suprimento de transformadores

Além da verificação da ampacidade dos condutores, a capacidade de suprimento dos transformadores constitui uma importante restrição operacional a ser considerada para o processo de restauração. A análise é feita a partir da corrente disponível (corrente de folga) dos transformadores que alimentam o barramento no qual os alimentadores estão conectados, conforme ilustrado na Figura 4.2, em que os transformadores T1 e T7 são considerados como fonte dos barramentos das subestações e conseqüentemente para os alimentadores.

Supondo que haja um defeito no disjuntor CB1 impedindo que a alimentação das cargas de F1 seja feita da forma convencional, através de T1, manobrando-se a chave NO para a posição fechada, LP1 e LP2 podem ser restaurados através de F7. Desta forma, T7 deve ter corrente de folga suficiente para alimentar as cargas transferidas, caso contrário a restauração não pode ser realizada sem comprometer a segurança do sistema elétrico.

Figura 4.2 – Alimentadores de subestações diferentes.



#### 4.2.3 Restrições adicionais

Além das restrições citadas anteriormente, outras restrições devem ser consideradas para garantir a segurança operacional das redes de distribuição. Tais restrições estão

diretamente relacionadas com as características fundamentais de uma rede de distribuição radial, ou seja, sua radialidade. São elas:

1. Cada ponto de carga é alimentado por uma fonte única;
2. Não devem existir caminhos fechados (*loops*) na rede.

Neste trabalho, tais restrições são chamadas de Restrições de Rede e também são consideradas para garantir a restauração segura do sistema elétrico.

### 4.3 Trabalhos relacionados

O problema de restauração automática de redes elétricas, também conhecido como *self-healing*, é tratado em diversos trabalhos na literatura. Dentre as principais propostas para solução deste problema destacam-se aquelas que tratam o tema com uma abordagem descentralizada, em geral, utilizando-se de Multi-Agentes (MA) (REN et al., 2012; YE; ZHANG; SUTANTO, 2011). Há ainda aquelas que utilizam o método de seccionalização que também é amplamente abordado para a restauração de sistemas elétricos, onde, destacam-se, os trabalhos propostos por (SARMADI et al., 2011; WANG; VITTAL; SUN, 2011). O uso de Redes Neurais Artificiais (RNA) também é proposto em (BRETAS; PHADKE, 2003) como solução para o problema de restauração automática em SEP.

A seguir, são discutidos alguns dos trabalhos mais relevantes sobre o tema.

#### 4.3.1 Abordagens que utilizam Multi-Agentes

A aplicação de multi-agentes para solucionar o problema de restauração automática em redes elétricas é adequada quando necessita-se tratar o problema de forma descentralizada. Outra peculiaridade é que a tomada de decisão, em geral, acontece nos próprios dispositivos de campo, ou IEDs (*Intelligent Electronic Device*), sem possibilidade de interferência de operadores humanos durante o procedimento de restauração.

Uma abordagem com MA é proposta em (REN et al., 2012) com dois tipos de agentes: *Bus Agents* e *Coordination Agents*. A interação entre os agentes acontece através das camadas *Social Layer*, *Reactive Layer* e *Proactive Layer*. *Bus Agents* atuam na camada *Proactive* fazendo o monitoramento do sistema. Quando faltas são detectadas, os próprios agentes iniciam o procedimento para determinar as ações de restauração do sistema elétrico. *Coordination Agents* atuam na camada *Social* realizando a troca de informação entre os agentes e, principalmente, definindo dinamicamente o conjunto de agentes responsável pela tomada de decisão para restauração de uma determinada falta ocorrida no SEP. Este mecanismo é chamado de *Dynamic Team Forming Mechanism* e faz parte da proposta de (REN et al., 2012).

Os autores de (YE; ZHANG; SUTANTO, 2011) propõem o uso de um sistema MA híbrido para SEP com diversos geradores. A proposta combina os benefícios da arquitetura centralizada e descentralizada simultaneamente. A aprendizagem por reforço é implementada através de um algoritmo *Q-learning* em um dos agentes.

### 4.3.2 Método de seccionalização

O método de seccionalização consiste em dividir o SEP a ser restaurado em vários sistemas menores e isolados criando diversas ilhas. Cada ilha é reenergizada em separado e, posteriormente, todo o conjunto é novamente sincronizado (SARMADI et al., 2011; WANG; VITTAL; SUN, 2011). Desta forma, diversos pequenos SEP são restaurados com menor esforço quando comparado ao SEP completo.

O método proposto pelos autores (SARMADI et al., 2011) utiliza-se de diversos *Phasor Measurement Units* (PMU) instalados em posições estratégicas para garantir a observabilidade de todo o SEP. Técnicas de programação inteira são empregadas para a otimização da função objetivo proposta.

Já os autores de (WANG; VITTAL; SUN, 2011), definem restrições de geradores (*black-start constraint*) e restrições de balanço energético (*power balance constraint*) para solucionar o problema da restauração. A estratégia de OBDD (*Ordered Binary Decision Diagram*) é proposta pelos autores como procedimento de busca para determinar a melhor combinação de seccionalização para a restauração.

### 4.3.3 Redes neurais artificiais

As Redes Neurais Artificiais (RNA) são propostas por (BRETAS; PHADKE, 2003) como método alternativo aos métodos tradicionais de restauração citados anteriormente. O uso das RNA apresenta como vantagem o pequeno tempo de processamento necessário para que o procedimento de restauração seja inicializado e a capacidade de considerar um grande número de condições de falta com praticamente nenhuma influência no tempo de processamento final. A proposta é aplicada a um SEP de testes com 162 nós.

### 4.3.4 Comparativo

A presente proposta para restauração automática de SEP fundamenta-se, principalmente, na construção de um modelo que seja adaptável a mudanças na topologia do SEP, que tenha tempo de resposta compatível às exigências regulatórias e que tenha dados de entrada oriundos do centro de controle.

A pretensão ao construir o modelo PSR é que este seja a base para uma ferramenta de apoio à operação de SEP capaz de realizar a restauração de redes elétricas de distribuição radiais minimizando a quantidade de clientes afetados pelas faltas. Buscou-se ainda uma

abordagem que demandasse uma quantidade mínima de informações do sistema elétrico, basicamente necessita-se dos estados dos equipamentos, medidas analógicas e a topologia da rede elétrica.

Desta forma, a modelagem utiliza-se de RPC (Rede de Petri Colorida) para a construção do modelo devido ao adequado tratamento que esta ferramenta matemática oferece a sistemas a eventos discretos. O uso de RPC permite ainda a construção de um modelo hierárquico e adaptável às constantes mudanças na topologia do SEP.

### 4.3.5 Abordagem adotada

Apesar de diversas abordagens para restauração disponíveis, nenhuma delas considera integralmente as restrições elétricas envolvidas no processo de restauração conforme abordado no início deste capítulo. Também não está evidente na literatura relacionada a existência de trabalhos que tenham como principal objetivo a minimização da quantidade de clientes afetados pelas faltas ao mesmo tempo que garante a segurança operacional do SEP, contemplando as restrições operacionais citadas. Em geral, os trabalhos buscam otimizar a energia suprida, o custo de faltas, a infra-estrutura existente ou que apresentem o resultado em um tempo adequado.

Devido a caracterização dos SEP como um sistema a eventos discretos, o uso de RPC é adotado como ferramenta para modelagem do processo de restauração. O modelo foi desenvolvido tendo a extensibilidade e a modularidade, dentre outras, como características fundamentais. Detalhes sobre as características do modelo RPC são apresentadas na Seção 4.4.

## 4.4 Características do modelo PSR

Redes de Petri Coloridas (RPC) são utilizadas para criar o modelo *Power System Restoration* (PSR). O principal objetivo deste modelo é realizar a restauração rápida do SEP mediante faltas permanentes, minimizando a quantidade de clientes afetados pela falta e observando as restrições operacionais envolvidas.

Algumas características fundamentais foram definidas como metas a serem alcançadas para o modelo PSR, tais como:

- Flexibilidade;
- Reusabilidade;
- Extensibilidade;
- Modularidade.

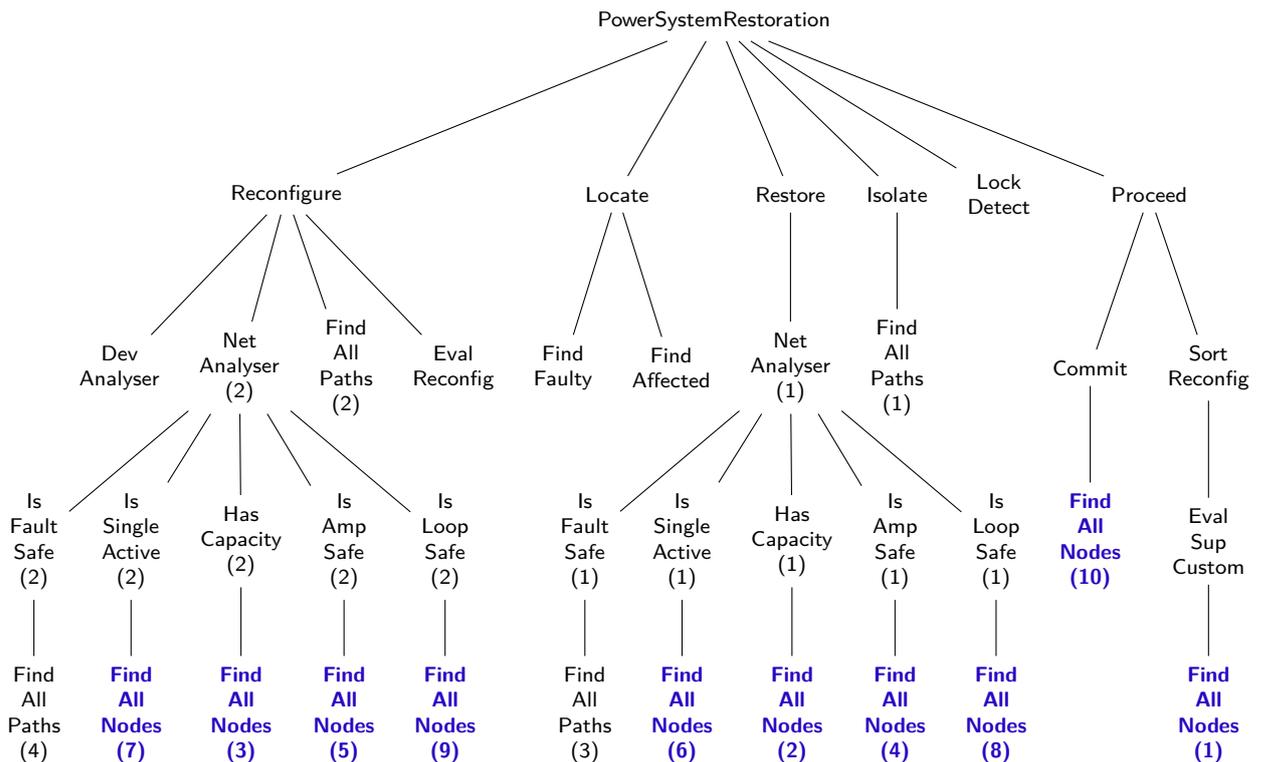
Buscou-se que o modelo fosse flexível, de maneira que novas funcionalidades pudessem ser acrescentadas sem grandes dificuldades.

A reusabilidade foi conferida ao modelo para facilitar a implementação do mesmo em um sistema computacional a ser integrado ao centro de controle de uma concessionária. Tal característica também é resultado da abordagem modular.

O modelo desenvolvido é extensível em termos de topologia de rede, ou seja, mudanças na topologia do SEP não implicam em mudanças estruturais no modelo, mas apenas nos dados de entrada apresentados.

Buscou-se ainda uma modelagem modular organizada hierarquicamente de forma a facilitar a reutilização dos módulos individuais em diferentes partes do modelo. Na Figura 4.3 é ilustrada a estrutura hierárquica do modelo PSR onde é possível notar que alguns módulos são instanciados diversas vezes. Por exemplo, o módulo FindAllNodes (em destaque) é instanciado dez vezes.

Figura 4.3 – Hierarquia do modelo PSR.



O uso do modelo PSR tem como grande vantagem permitir que todo o processo de validação do procedimento de restauração propriamente dito aconteça independente de qualquer implementação de *software* ou sistema computacional. Além disso, a linguagem de modelagem intrínseca às RPC, CPN ML, descreve o procedimento de restauração através de uma linguagem funcional garantindo que a implementação efetiva do modelo PSR possa ser realizada independente de qualquer linguagem de programação. Desta forma,

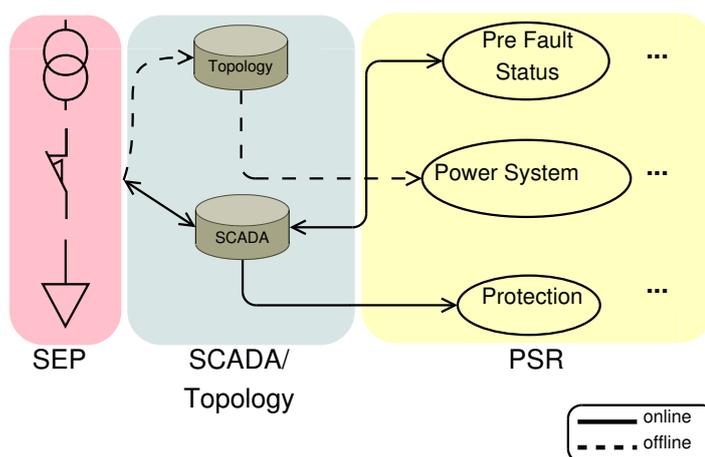
a implementação pode ser feita de forma mais direta, uma vez que possíveis erros de modelagem e concepção podem ser completamente depurados utilizando as ferramentas de modelagem, análise e simulação do *CPN Tools*.

## 4.5 Dados de entrada

O processo de restauração é iniciado quando algum relé de proteção do SEP monitorado passa ao estado bloqueado (*lockout*), indicando que uma falta permanente foi detectada e que eventuais ciclos de religamento não conseguiram eliminar o defeito. Este evento aciona a função de localização e isolamento de falta do modelo. Também é função do modelo PSR determinar os trechos livres de defeitos, mas que estão desenergizados devido a uma falta em outro trecho. Assim, faz-se necessário a reenergização dos trechos sem falta a partir de critérios pré-estabelecidos, tais como prioridade de cargas e uso de fontes preferenciais, observadas as restrições descritas anteriormente na Seção 4.2.

Os dados de entrada do modelo PSR são recebidos a partir do sistema supervisório e de aquisição de dados (SCADA, do inglês *Supervisory Control and Data Acquisition System*) e de uma base de dados que contenha a topologia do sistema elétrico. Na Figura 4.4 é ilustrada a interação entre o modelo PSR, os sistemas técnicos de uma distribuidora de energia e o SEP desta distribuidora. No módulo PSR estão os lugares denominados Pre Fault Status, Power System e Protection no modelo RPC.

Figura 4.4 – Interação entre o modelo PSR, o SCADA e o SEP.



O lugar **Pre Fault Status** representa o ponto de entrada das mensagens recebidas do SCADA contendo os eventos discretos e as medidas do SEP, tais como estado de disjuntores, medidas de corrente e similares. Os eventos discretos também chamados digitais devem ser informados tão logo estejam disponíveis, já as medidas ou valores analógicos apenas precisam ser informados em intervalos mais longos, da ordem de minutos.

O lugar **Protection** recebe especificamente as mensagens do sistema de proteção, ou seja, atuação de funções de proteção diversas, bloqueio de dispositivos e similares. Estas

mensagens também são respassadas ao modelo PSR tão logo são disponibilizadas pelo sistema SCADA.

Já o lugar **Power System** contém uma única ficha que representa toda a topologia e parâmetros físicos da rede elétrica modelada. Esta ficha não é modificada ao longo de uma simulação do modelo, pois considera-se que o sistema não sofre alterações físicas durante uma simulação. Modificações estruturais na rede elétrica ocorrem em intervalos de tempo relativamente longos (comparados ao tempo de processamento do aplicativo) ou mediante alguma forma de anúncio ou intervalo pré-estabelecido. Para chamar atenção para este fato, as ligações entre o *SEP/Topology* e *Topology/Power System* são apresentadas em pontilhado na Figura 4.4.

Detalhes sobre a dinâmica do modelo PSR são apresentadas nas seções posteriores. Ainda com relação aos dados de entrada, pode-se resumí-los como mostrado a seguir.

- Estados de equipamentos;
- Medidas de corrente pré-falta;
- Atuações de dispositivos de proteção;
- Partida (*start*) de dispositivos de proteção;
- Bloqueio de dispositivos (*lockout*).

Estes dados são representados no modelo PSR por meio de dois tipos de fichas **PwrSysSTATUS** e **ProtMSGs**.

**PwrSysSTATUS** é definido como uma lista de estados de todos os nós e ramos da rede elétrica representada. O estado de um ramo, **EDGE\_STATUS**, por sua vez, é definido como um produto de quatro campos:

- nome (**NAME**);
- posição atual (**POSITION**);
- corrente de carga medida (**LOAD**);
- estado (**STATUS**).

Já o estado de um nó (**NODE\_STATUS**), é o produto de dois campos: nome (**NAME**) e estado (**STATUS**). O estado de ramos e nós é unificado para formar o tipo **EDGE\_NODE\_STATUS**.

```
colset NAME = string with "0".."Z"and 0..6;
```

```
colset POSITION = with Open | Open_ | Closed | Closed_ ;
```

```

colset CURRENT = real;

colset LOAD = CURRENT;

colset STATUS = with Active | Inactive | Supplied
| Unsupplied | Unsupplied_ | InService | InService_ | OutOfServ | OutOfServ_ |
Enabled | Disabled | Faulty | Faulty_ | Healthy;

colset EDGE_STATUS = product NAME * POSITION * LOAD * STATUS;

colset NODE_STATUS = product NAME * STATUS;

colset EDGE_NODE_STATUS = union ES:EDGE_STATUS + NS:NODE_STATUS;

colset PwrSysSTATUS = list EDGE_NODE_STATUS;

```

Fichas ProtMSGs são definidas como sendo uma lista contendo eventos associados aos dispositivos de proteção de uma determinada ocorrência no SEP. Cada evento da proteção (ProtMsg), por sua vez, é definido como sendo um produto de três campos: SS (identifica a subestação a qual o dispositivo pertence), IED (identifica o dispositivo) e FUN (indica a função de proteção).

```

colset SS = NAME;

colset IED = NAME;

colset FUN = NAME;

colset ProtMsg = product SS * IED * FUN;

colset ProtMSGs = list ProtMsg;

```

As informações listadas anteriormente correspondem aos dados de entrada em tempo real (*online*) do modelo PSR, ou seja, as informações que devem ser repassadas tão logo ocorrem faltas e/ou modificações no SEP. Porém, para realizar simulações, o modelo PSR deve ser configurado previamente com os parâmetros físicos ou estáticos (*offline*) da rede elétrica a ser restaurada, conforme listado a seguir:

- Topologia da rede elétrica;
- Dispositivos de proteção primária;
- Quantidade de consumidores em cada ponto de carga;
- Seção dos condutores de cada trecho dos alimentadores e demais dados necessários para determinar a Ampacidade dos cabos;
- Capacidade dos transformadores de distribuição (Corrente de folga).

Estes dados são representados no modelo PSR por meio de fichas do tipo PwrSYS que contém basicamente uma lista de segmentos e uma lista de equipamentos da rede

elétrica representada. Cada aresta (**EDGE**) é representada como um produto de quatro campos: nome (**NAME**), terminal 1 (**T1**), terminal 2 (**T2**) e posição padrão (**POSITION**). Os nós (**NODES**) também são representados por um produto de quatro campos: nome (**NAME**), corrente que o nó pode suprir (**SUPPLY**), ampacidade (corrente máxima que o trecho pode conduzir) e número de consumidores conectados ao nó (**CUSTOMER**). Um tipo composto pela união entre ramos e nós (**EDGE\_NODE**) também é definido para formar a lista **PwrSYS**.

```
colset T1 = NAME;
colset T2 = NAME;
colset SUPPLY = CURRENT;
colset CUSTOMER = int;
colset NODE = product NAME * SUPPLY * AMP * CUSTOMER;
colset EDGE = product NAME * T1 * T2 * POSITION;
colset EDGE_NODE = union E:EDGE + N:NODE;
colset PwrSYS = list EDGE_NODE;
```

## 4.6 Modelo PSR

O modelo *Power System Restoration* (PSR) divide o processo de restauração em quatro sub-procedimentos. Cada procedimento é modelado por um módulo específico do modelo PSR. Os procedimentos e os módulos são listados a seguir.

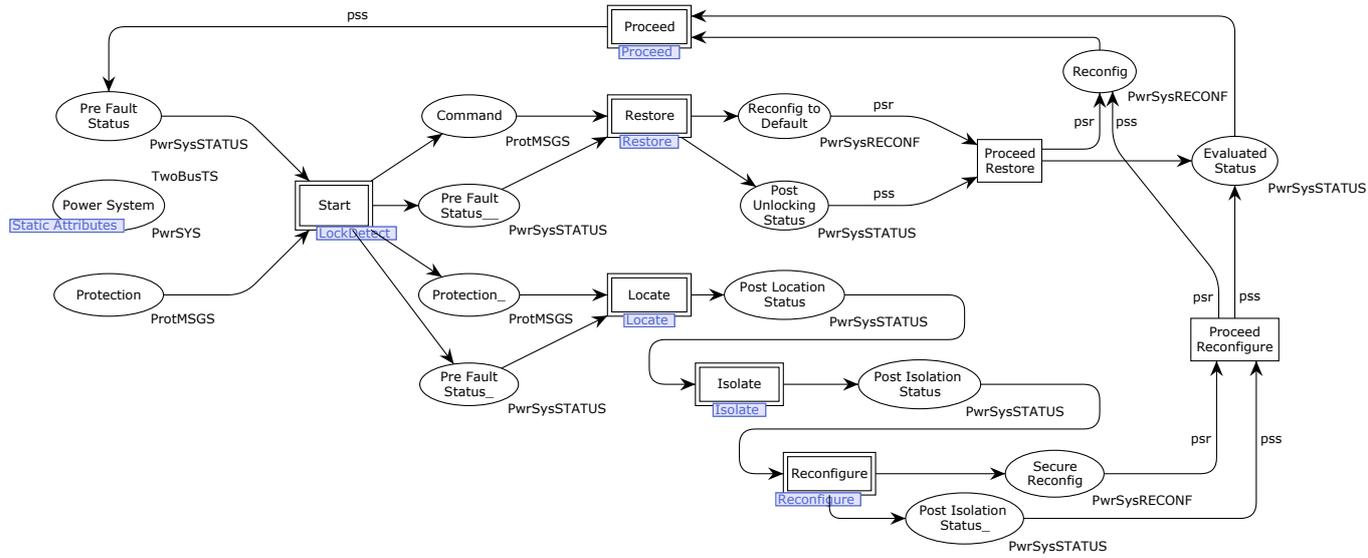
1. Localização de defeito (**Locate**);
2. Isolação de defeito (**Isolate**);
3. Reconfiguração de rede (**Reconfigure**);
4. Restauração de rede (**Restore**).

Além destes procedimentos, outros dois sub-procedimentos auxiliares são definidos: Detecção de bloqueio (**LockDetect**) e Efetivação (**Proceed**). Cada um destes sub-procedimentos é modelado em módulos específicos.

Na Figura 4.5 é apresentada a página principal do modelo PSR onde é mostrado cada um dos módulos e o relacionamento entre eles. Cada módulo é representado por uma transição de substituição ([JENSEN; KRISTENSEN, 2009](#)).

No contexto do modelo de restauração proposto, distingue-se entre Reconfiguração e Restauração. Entende-se por Reconfiguração o procedimento para conduzir o SEP para um estado intermediário em que a falta está isolada e que a quantidade de consumidores

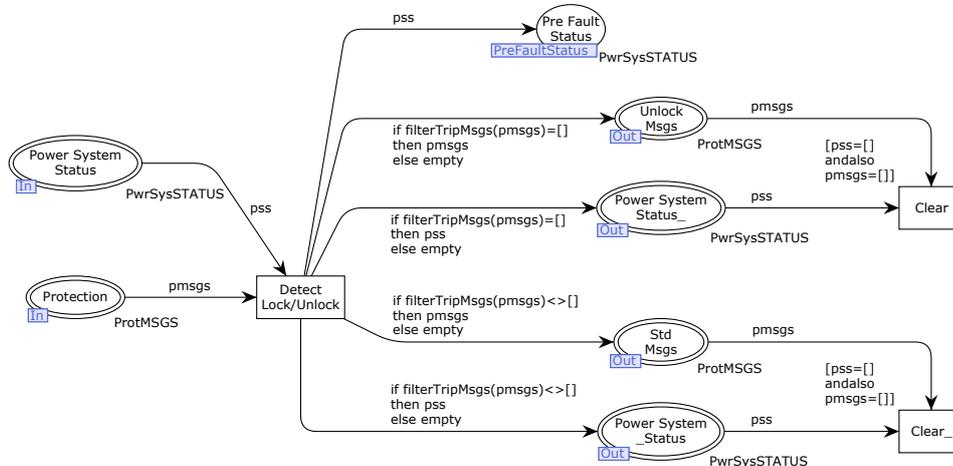
Figura 4.5 – Página principal do modelo PSR.



afetada é minimizada. Já a Restauração é o processo de conduzir o SEP ao seu estado padrão ou considerado normal, em termos topológicos e operacionais do sistema elétrico.

A transição **Start**, mostrada na Figura 4.5, tem como objetivo determinar se a rede elétrica representada na simulação deve ser restaurada ou reconfigurada. A transição **Start** está associada à transição de substituição **LockDetect**. Na Figura 4.6 é apresentada a subpágina com os detalhes da referida transição de substituição que tem como lugares de entrada (*Input Port*) **Pre Fault Status** e **Protection**. Os lugares **Power System Status\_**, **Power System Status** e o lugar de fusão (**JENSEN; KRISTENSEN, 2009**) **Pre Fault Status** são cópias do lugar de entrada **Power System Status**. Fichas são colocadas nestes lugares ao disparo da transição **Detect Lock/Unlock**.

Figura 4.6 – Subpágina **LockDetect** do modelo PSR.



Se existir ao menos uma mensagem de *trip*, representada por uma ficha do tipo **ProtMSGs** no lugar **Protection**, então fichas representando o estado do SEP (**PwrSysSTATUS**),

mensagens de proteção (`ProtMSGs`), uma ficha vazia (`[]`) e outra ficha vazia são colocadas nos lugares `Power System _Status`, `Std Msgs`, `Unlock Msgs` e `Power System Status_`, respectivamente. Caso existam apenas fichas representando o desbloqueio (*unlock*) de dispositivos, então fichas não vazias são colocadas em `Unlock Msgs` e `Power System Status_`; e fichas vazias em `Std Msgs` e `Power System _Status`. Toda esta lógica é realizada basicamente por meio das funções de arco, conforme apresentado na Figura 4.6.

O código fonte de cada uma das funções escritas para o desenvolvimento do modelo PSR é apresentado no Apêndice A. Assim, as funções utilizadas na subpágina, `filterTripMsgs` (ver A.2) e `isFuncTrip` (ver A.1), podem ser conferidas no referido apêndice.

O lugar de fusão `Pre Fault Status` é utilizado posteriormente e sua função será detalhada em momento mais oportuno ao longo do texto. Por ora, pode-se dizer que o mesmo mantém uma cópia do estado pré-falta do SEP durante a simulação e que é possível que não seja necessário utilizá-lo.

As variáveis `pss` e `pmsgs` são definidas como segue.

```
var pss : PwrSysSTATUS;  
var pmsgs : ProtMSGs;.
```

Os demais módulos do modelo PSR são apresentados nas subseções seguintes.

### 4.6.1 Localização de falta

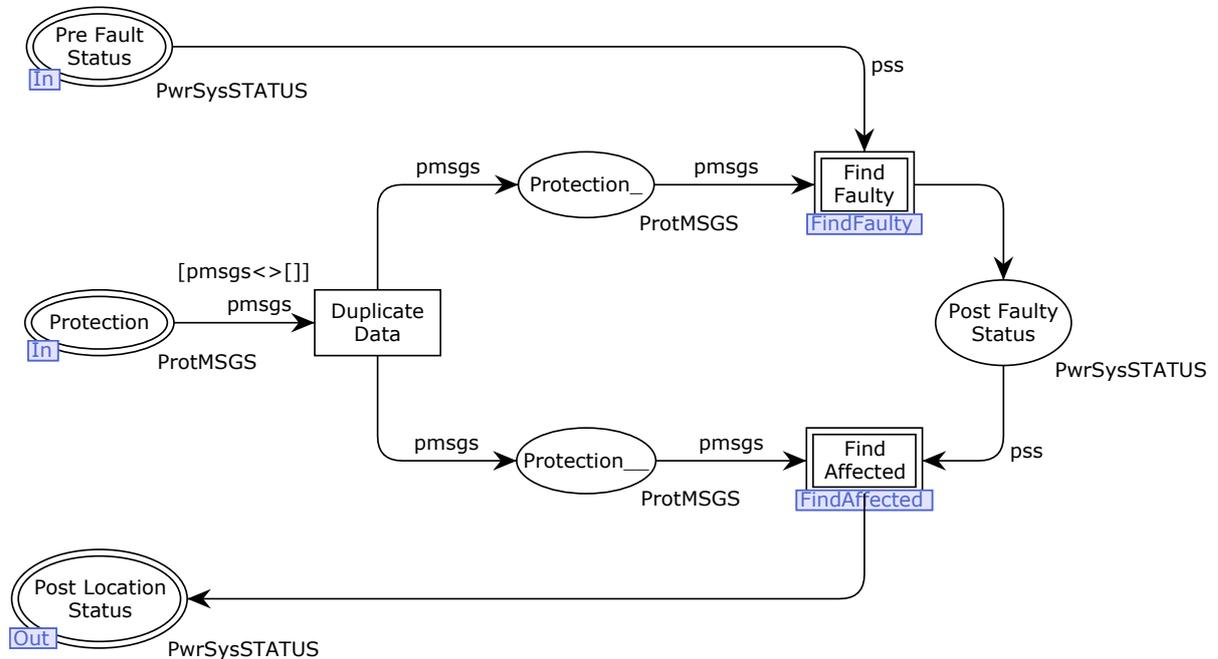
Uma vez determinado que o SEP necessita ser reconfigurado, o passo seguinte corresponde à localização da falta. Esta tarefa é realizada pela transição de substituição `Locate` como apresentado na Figura 4.5. Como resultado, o estado atualizado do SEP é mostrado no lugar `Post Isolation Status`, que contém o estado inicial do sistema elétrico que é representado pela marcação do lugar de entrada `Pre Fault Status_`, porém com o campo `status` igual a `Faulty_1` para os pontos de carga defeituosos conforme as operações realizadas pela transição `Locate`.

Na Figura 4.7 é apresentada a subpágina com os detalhes da transição de substituição `Locate` tendo como Portas de Entrada (*Input Ports*) os lugares `Pre Fault Status` e `Protection`. Ao disparo da transição `Duplicate Data`, fichas do tipo `ProtMSGs`, que representam os eventos da proteção do sistema elétrico, são colocadas nos lugares de saída da transição. Tais lugares correspondem aos lugares de entrada de outras duas transições de substituição: `Find Faulty` e `Find Affected`. A primeira tem a função de determinar efetivamente o ponto de carga faltoso a partir dos eventos de proteção. A segunda transição de substituição, `FindAffected`, tem como objetivo determinar os pontos de carga saudáveis (li-

<sup>1</sup> Estados com sufixo "`_`" são usados para indicar um estado provisório, que foi determinado, mas que ainda não foi efetivado.

vres de defeito), mas que foram afetados por eventuais mudanças (manobras) na topologia do SEP como consequência das operações da proteção do sistema elétrico.

Figura 4.7 – Subpágina **Locate** do modelo PSR.



O princípio de funcionamento da transição **Find Faulty** consiste em encontrar o nó mais profundo do subgrafo que tem como nós aqueles ligados pelas arestas que representam os relés de proteção que partiram e ainda o nó que representa a fonte. Para exemplificar este princípio, considerando-se o SEP mostrado na Figura 4.8a e seu respectivo grafo, apresentado na Figura 4.8b, num cenário de ocorrência do sistema de proteção em que uma falta se apresenta no ponto de carga LP3. Os relés CB, R1 e R3 detectam a falta e partem (*start*). Em uma atuação perfeita do sistema de proteção R3 abriria para eliminar a falta, porém deseja-se exemplificar o caso em que, por descoordenação entre os relés de proteção, R1 abre para eliminar a falta. Para este cenário, os seguintes eventos de proteção são esperados:

- Partida (*start*) dos relés CB, R1 e R3;
- Atuação da função de proteção de sobrecorrente em R1;
- Abertura (*trip*) do religador R1.

Diante do cenário exposto, na Figura 4.9 é apresentado o subgrafo para o SEP que contém as arestas que representam os relés que partiram (CB, R1 e R3) e os nós ligados a estas arestas (T, Bus, LP1 e LP3). O nó mais profundo deste subgrafo é determinado como sendo o ponto de carga faltoso para uma dada ocorrência. Portanto, para esta ocorrência,

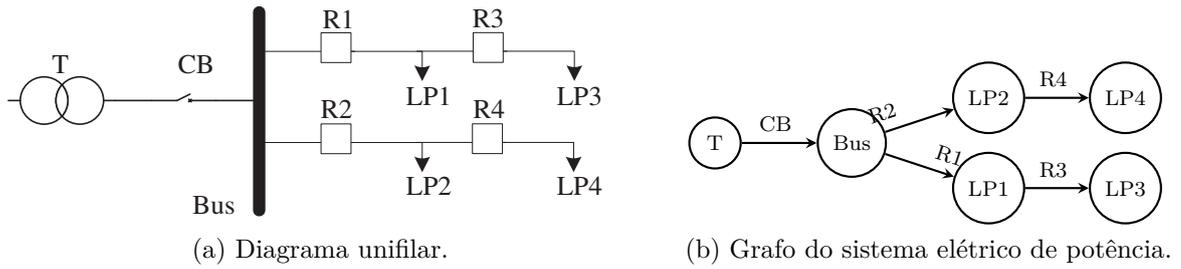
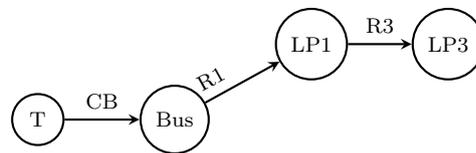


Figura 4.8 – Sistema elétrico de potência.

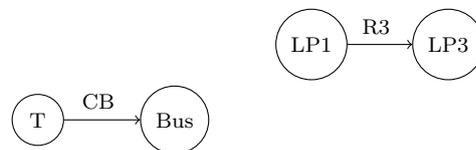
LP3 é determinado como sendo o ponto de carga faltoso, uma vez que este é o nó mais profundo do subgrafo apresentado na Figura 4.9.

Figura 4.9 – Subgrafo representando a partida dos relés.



Além da determinação do ponto de carga faltoso também é necessário determinar os pontos de carga afetados pelas ações do sistema de proteção. Para tanto, basta atualizar o grafo eliminando as arestas correspondentes às chaves que foram abertas após a atuação da proteção. Assim, os nós isolados da fonte representam aqueles que foram afetados pela falta. Para o caso em questão, como mostrado na Figura 4.10, R1 é removido do grafo. Logo, LP1 e LP3 correspondem aos pontos de carga afetados. Como a falta está localizada em LP3, apenas LP1 é classificado como afetado. Esta tarefa no modelo é realizada pela transição de substituição Find Affected atualizando o estado dos pontos de carga afetados para "não suprido"(STATUS=Unsupplied\_).

Figura 4.10 – Subgrafo para o cenário pós-falta.

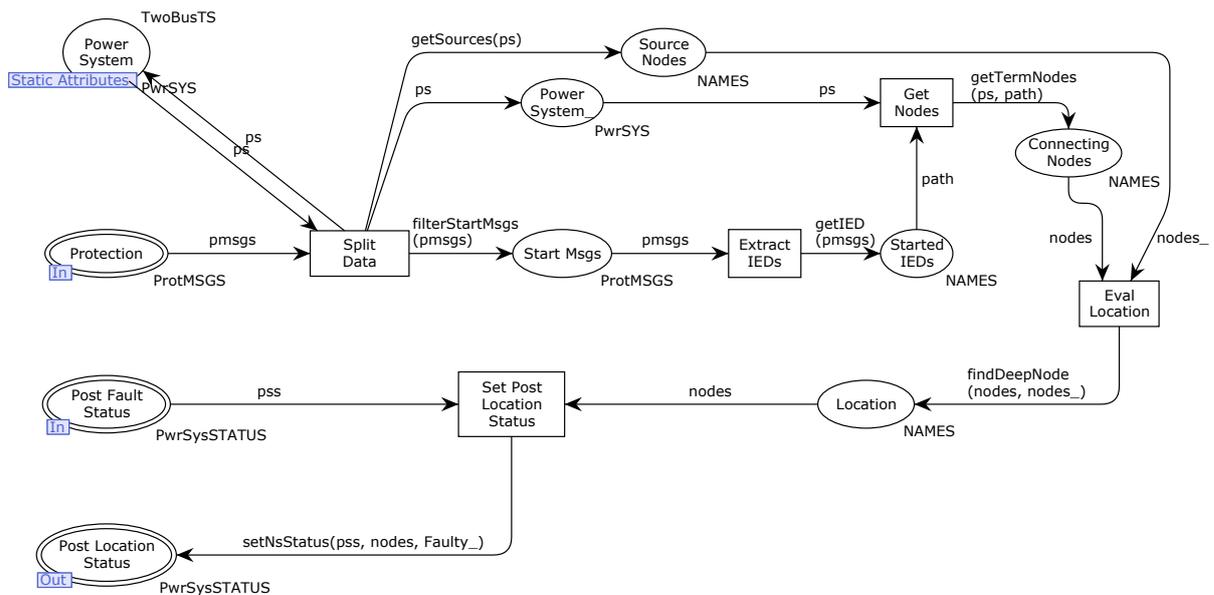


Nas Figuras 4.11 e 4.12 são apresentadas as subpáginas Find Faulty e Find Affected, respectivamente. O código-fonte das funções, em linguagem ML, utilizadas nas referidas subpáginas são listadas a seguir:

- `isSource` (ver A.3);
- `getSources` (ver A.4);
- `filterStartMsgs` (ver A.5);

- `getTermNodes` (ver A.6);
- `getIED` (ver A.7);
- `findDeepNode` (ver A.8);
- `setNsStatus` (ver A.9).

Figura 4.11 – Subpágina Find Faulty.

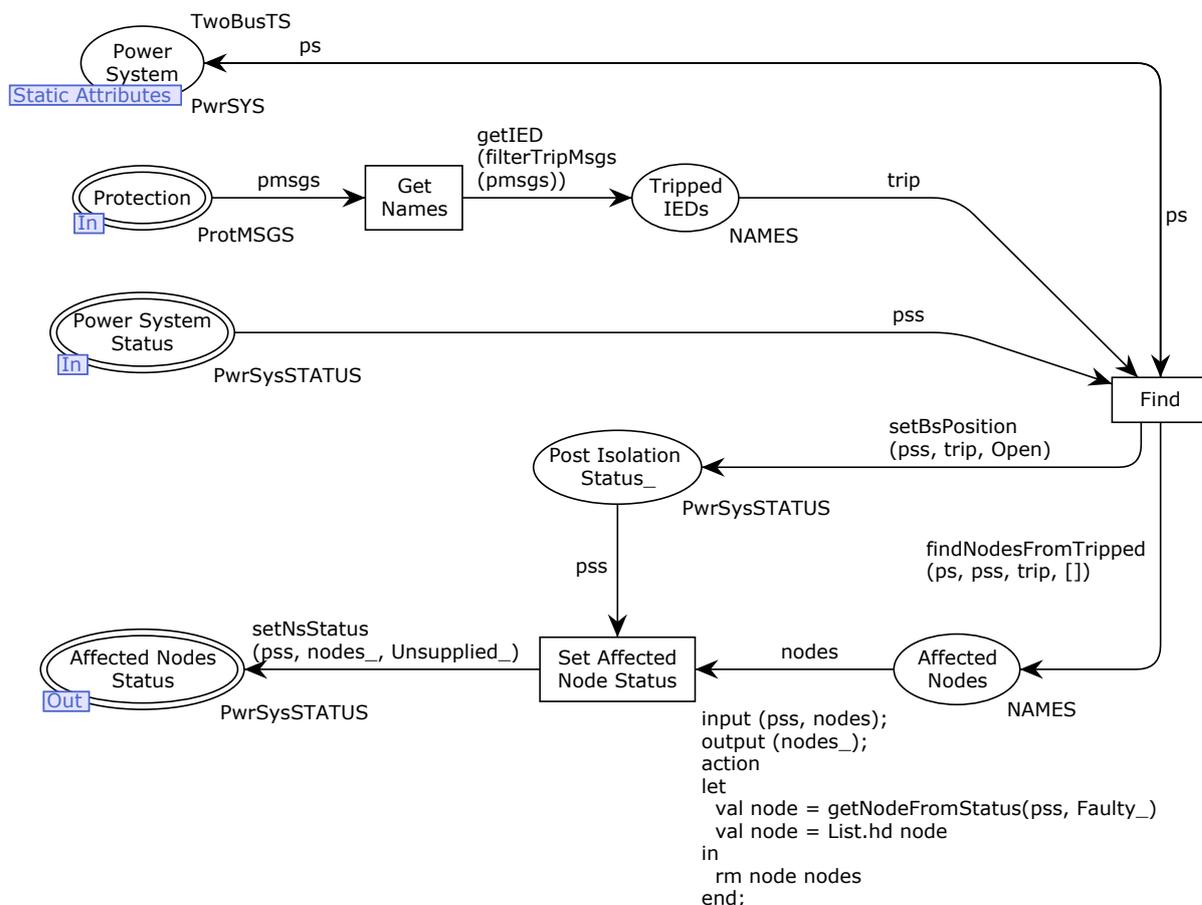


#### 4.6.2 Isolamento de falta

O próximo passo para o processo de restauração consiste em isolar o defeito abrindo as chaves (disjuntores, religadores e similares) anterior e posterior ao ponto de carga faltoso identificado na etapa anterior. O isolamento ideal consiste em abrir as duas chaves imediatamente vizinhas ao ponto de carga faltoso, porém é possível que tais chaves não estejam disponíveis para manobras devido a defeitos na própria chave, ou que estejam desabilitadas para manutenção ou ainda desabilitadas devido a uma falta ocorrida anteriormente e ainda a ser corrigida na rede. Portanto, faz-se necessário buscar chaves alternativas em caso de indisponibilidade das chaves consideradas ideais.

A função de isolamento é realizada no modelo PSR pela transição de substituição **Isolate**, mostrada na Figura 4.13. Os lugares **Prior Isolation Status** e **Post Isolation Status** correspondem às portas de entrada e saída da subpágina, respectivamente. As fichas nestes lugares são do tipo **PwrSysSTATUS**. Basicamente, o disparo de **Split Data** extrai da ficha no lugar de entrada as seguintes informações: o ponto de carga defeituoso (**Recent Faulty Node**), a lista de dispositivos que estão operando normalmente (**STATUS=InService**) e a lista de dispositivos que estão abertos (**POSITION=Open**).

Figura 4.12 – Subpágina Find Affected.

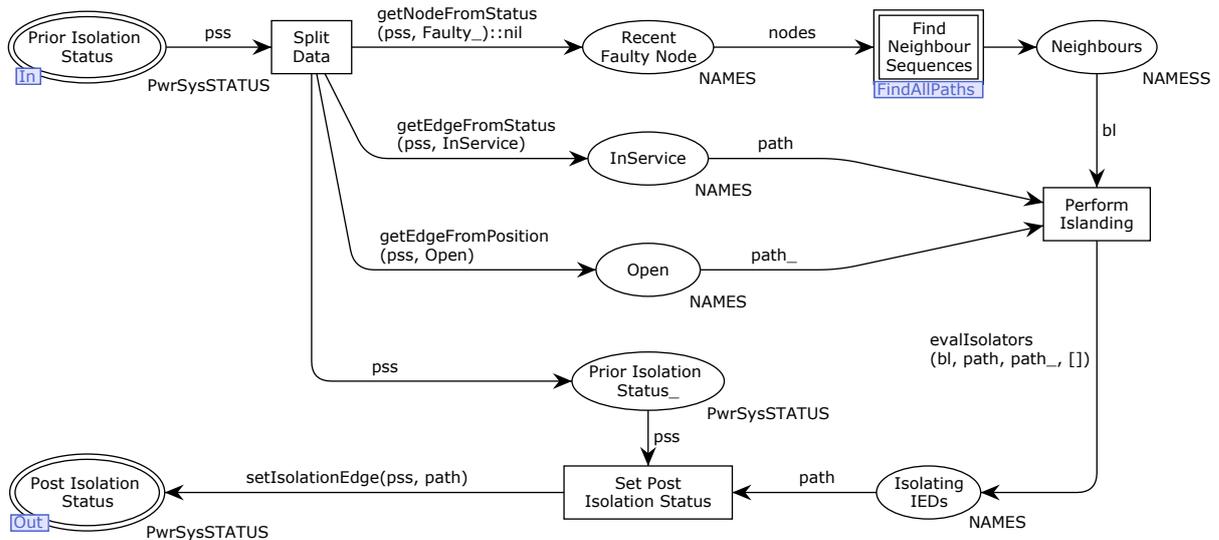


O lugar Recent Faulty Node corresponde à porta de entrada da transição de substituição Find Neighbour Sequences que tem a função de encontrar os caminhos entre o ponto de carga faltoso e as fontes, independentemente de seu estado e da posição atual dos dispositivos. Como resultado, uma ficha contendo a lista de caminhos entre o ponto de carga faltoso e as fontes é colocada no lugar Neighbours. Cada elemento desta lista é uma lista ordenada que contém as chaves entre o ponto de carga faltoso e uma fonte.

Na sequência, o disparo da transição Perform Islanding determina a lista de isoladores considerando agora aqueles que efetivamente podem operar (indicado pela ficha no lugar InService) e aqueles que já se encontram abertos (indicado pela ficha no lugar Open). O resultado é colocado no lugar de saída da transição, Isolating IEDs. Por fim, a transição Set Post Isolation Status atualiza o estado do sistema elétrico com as chaves selecionadas para isolar o defeito, que passam a ter seu estado individual indicando que os mesmos estão fora de operação (STATUS=OutOfServ\_).

Sobre a transição de substituição Find Neighbour Sequences, a mesma corresponde a uma instância da subpágina FindAllPaths que é reutilizada em diversas partes do modelo PSR. Basicamente, a transição FindAllPaths realiza a operação de travessia (do

Figura 4.13 – Subpágina Isolate.



inglês, *traverse*) em um grafo. Tal grafo corresponde à rede de distribuição considerada (representado por uma ficha *PwrSYS*) e o nó de partida corresponde ao ponto de carga faltoso.

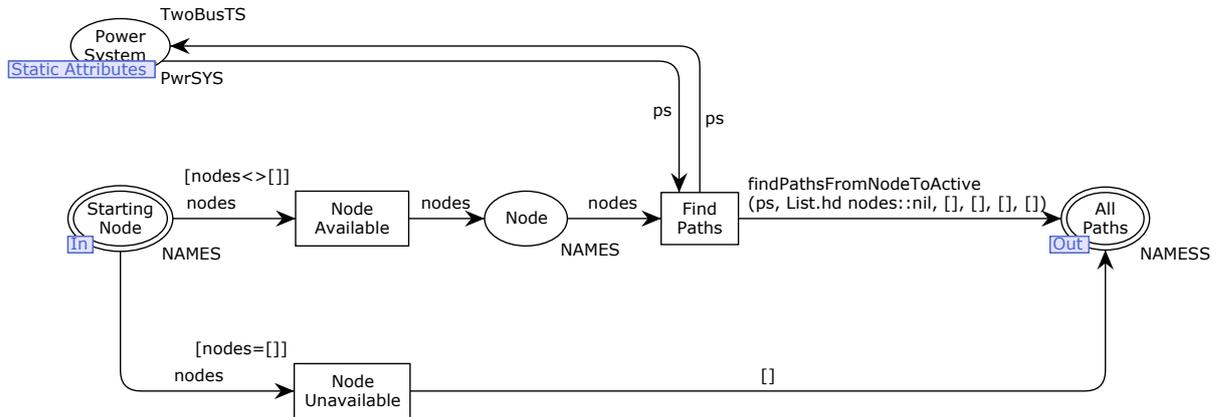
*FindAllPaths* implementa algoritmos de busca tradicionais, similares a busca em largura (*breadth-first search*) e a busca em profundidade (*depth-first search*). Na Figura 4.14 é apresentada a subpágina *FindAllPaths* e o código-fonte das funções utilizadas nesta subpágina e em *Isolate* são listadas a seguir:

- `getNodeFromStatus`, (ver A.15);
- `getEdgeFromStatus`, (ver A.17);
- `getEdgeFromPosition`, (ver A.16);
- `evalIsolators`, (ver A.18);
- `setIsolationEdge`, (ver A.19);
- `getEdgesFromNode`, (ver A.20);
- `createRepList`, (ver A.21);
- `findPathsFromNodeToActive`, (ver A.13).

Os tipos *NAMES*, *NAMESS* e as variáveis *path*, *path\_* e *nodes* são declarados como mostrado a seguir:

```
colset NAMES = list NAME;
```

Figura 4.14 – Subpágina FindAllPaths.



```
colset NAMESS = list NAMES;
var path, path_, nodes : NAMES;
```

### 4.6.3 Reconfiguração

A reconfiguração do sistema elétrico é a parte mais delicada do processo de restauração automática, uma vez que, nesta etapa são definidas as opções disponíveis para restauração do suprimento após uma falta. Em síntese, a tarefa consiste em encontrar os caminhos possíveis entre os pontos de carga saudáveis (*healthy*) e as fontes. Tais caminhos são referidos como o conjunto de caminhos simples ou elementares entre dois nós de um grafo, conforme já detalhado na Seção 2.3.6.

Depois de determinados o conjunto de caminhos simples, cada um destes é avaliado quanto a dois tipos de restrições:

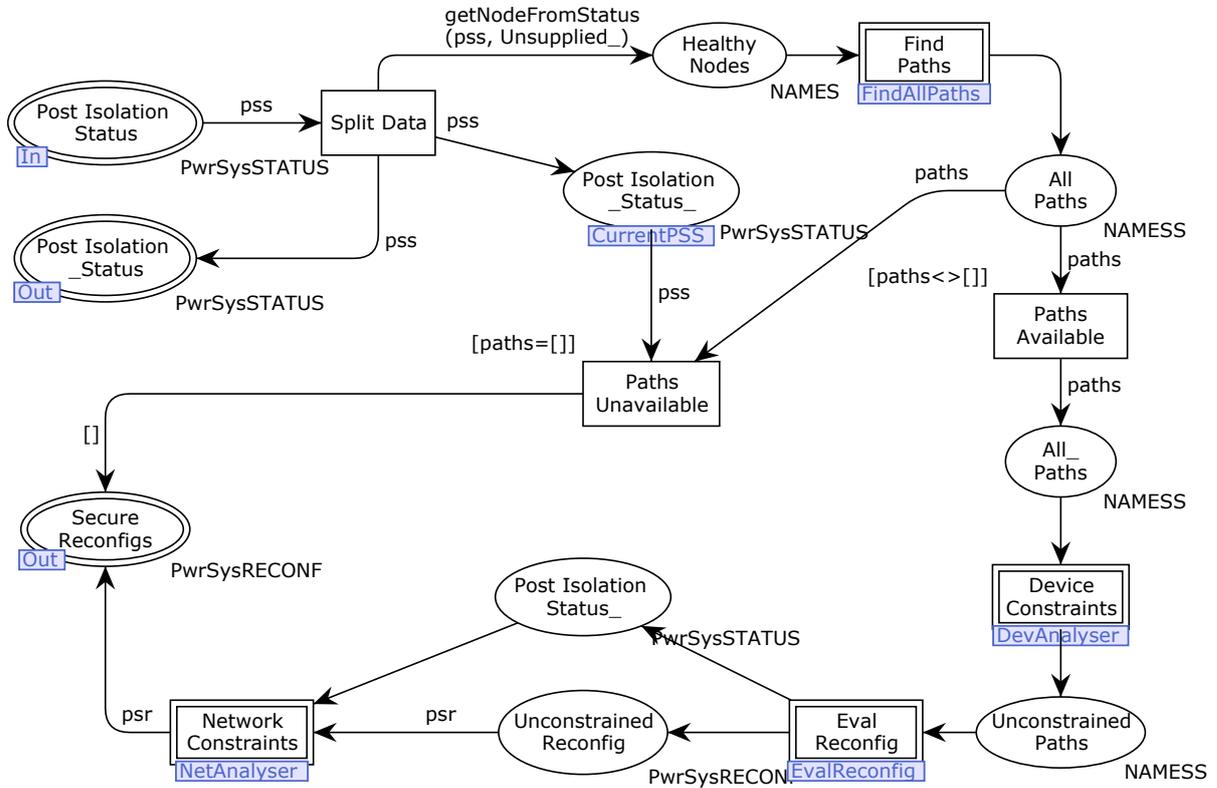
- Restrições de dispositivo (*Device constraints*);
- Restrições de rede (*Network constraints*).

Restrições de dispositivo são observadas para cada chave integrante de um caminho do ponto de vista de sua disponibilidade para ser manobrada. Para que um caminho seja válido, em termos topológicos, todas as chaves deste caminho devem estar fechadas. Se uma das chaves do caminho estiver inoperante ou selecionada anteriormente para isolar um defeito, o caminho é irrealizável e passa a ser desconsiderado no processo de restauração. Já as restrições de rede são aquelas relacionadas às restrições do próprio sistema elétrico, tais como ampacidade, capacidade de suprimento das fontes e similares, conforme descrito na Seção 4.2.

A função de reconfiguração é realizada pela transição de substituição **Reconfigure**, mostrada na Figura 4.5. Os detalhes da subpágina **Reconfigure** são apresentados na Figura

4.15 em que podem ser observadas quatro outras transições de substituição: Find Paths, Device Constraints, EvalReconfig e Network Constraints. Cada uma destas transições realiza uma subfunção da etapa de reconfiguração como mostrado a seguir.

Figura 4.15 – Subpágina Reconfigure.



A transição Find Paths, mostrada na Figura 4.15, é uma instância da subpágina FindAllPaths, apresentada anteriormente. Sua função na subpágina Reconfigure é encontrar os caminhos simples entre os pontos de carga saudáveis indicados pelas fichas no lugar Healthy Nodes e as fontes. Como resultado, uma ficha do tipo NAMESS (que representa uma lista de caminhos) é colocada no lugar All Paths. É possível que tais caminhos não existam, resultando assim em uma ficha vazia ([]) em All Paths. Neste caso, a transição Paths Unavailable é habilitada e ao ser disparada, uma ficha vazia, é colocada no lugar Secure Reconfigs indicando que não existe uma reconfiguração segura para o cenário em questão. Caso contrário, a ficha em All Paths é diferente de vazio, a transição Paths Available é disparada e a ficha com os caminhos é colocada em All\_Paths.

No modelo PSR, cada caminho (*path*) é representado por uma lista, como por exemplo:

$$p_1 = [ CB1, SW7, SW2, R3 ];$$

$$p_2 = [ CB1, SW7, SW3, R4 ].$$

O conjunto de caminhos é representado como uma lista de listas. No caso, fichas

do tipo **NAMESS** são usadas para este propósito. A declaração de **NAMESS** é mostrada na seção anterior.

Depois de determinados o conjunto de caminhos, cada chave de cada um dos caminhos é avaliada pela transição de substituição **Device Constraints** quanto às restrições de dispositivos. Aqueles caminhos que possuem uma ou mais chaves com restrições são descartados. Os caminhos restantes são representados novamente em uma ficha **NAMESS** e colocados no lugar **Unconstrained Paths**. Desta forma, cada caminho,  $p_i$ , é avaliado pela transição **Eval Reconfig** para determinar sua respectiva reconfiguração,  $r_i$ . Por exemplo:

$$p_1 \Rightarrow r_1 = [(CB1, Close), (SW7, Close), (SW2, Close), (R3, Close)]$$

$$p_2 \Rightarrow r_2 = [(CB1, Close), (SW7, Close), (SW3, Close), (R4, Close)]$$

O tipo **PwrSysRECONF** é definido para representar o conjunto de manobras (*manoeuvres*) que deve ser realizado no SEP.

```
colset MANOEUV = product NAME * POSITION;
```

```
colset RECONF = list MANOEUV;
```

```
colset PwrSysRECONF = list RECONF;
```

Às reconfigurações são adicionados os dispositivos definidos previamente na sub-página **Isolate** como sendo os isoladores do ponto de carga faltoso na posição aberta. Considerando como exemplo de isoladores as chaves SWF e SWB, as reconfigurações  $r_1$  e  $r_2$  passam a ser:

$$r_1 = [(SWF, Open), (SWB, Open), (CB1, Close), (SW7, Close), (SW2, Close), (R3, Close)]$$

$$r_2 = [(SWF, Open), (SWB, Open), (CB1, Close), (SW7, Close), (SW3, Close), (R4, Close)]$$

Em conjunto com o estado atual do sistema elétrico (representado pelo lugar de fusão **CurrentPSS**), a posição atual de cada chave é verificada. Aqueles chaves que já se encontram na posição determinada, são removidas da reconfiguração. Se, por exemplo, CB1 encontra-se na posição fechada, temos:

$$r_1 = [(SWF, Open), (SWB, Open), (SW7, Close), (SW2, Close), (R3, Close)] \text{ e}$$

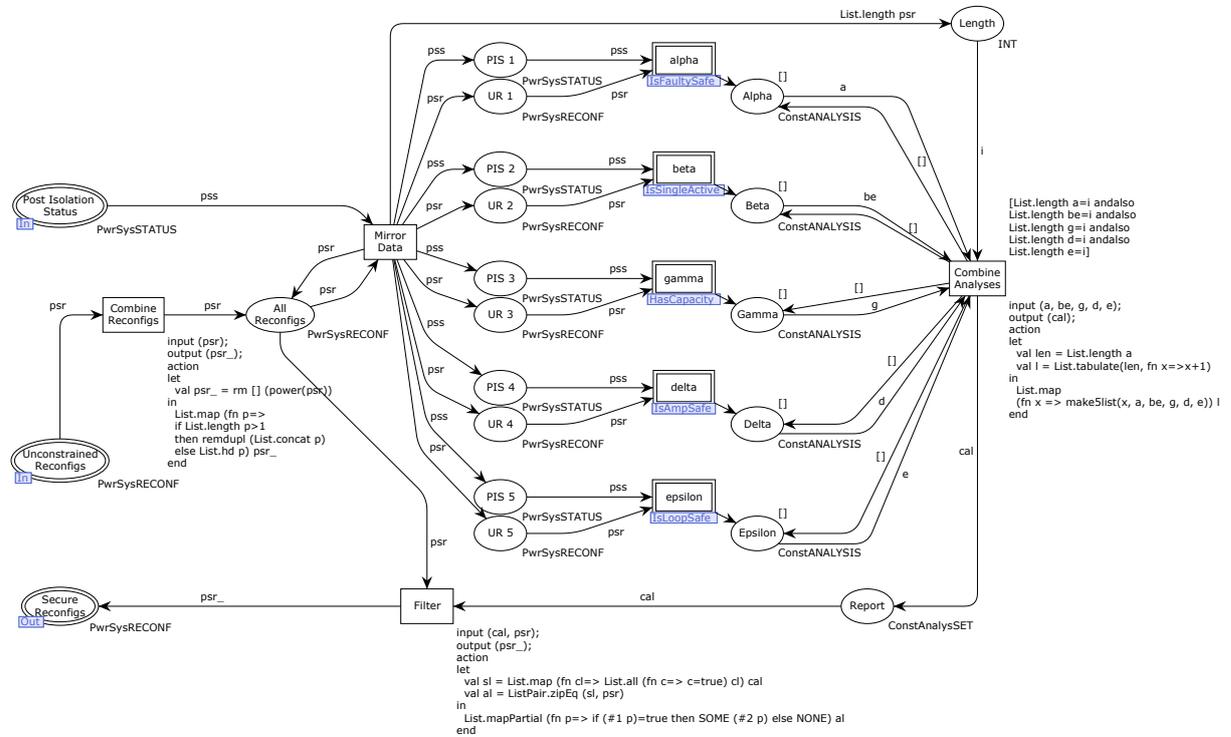
$$r_2 = [(SWF, Open), (SWB, Open), (SW7, Close), (SW3, Close), (R4, Close)].$$

Desta forma, a ficha no lugar **Unconstrained Reconfig** contém as reconfigurações irrestritas que serão avaliadas pela transição **Network Constraints**.

As restrições de rede são consideradas pela transição de substituição **Network Constraints** que, por sua vez, é uma instância da transição de substituição **NetAnalyser**. Esta transição é reutilizada na etapa de restauração, apresentada na Seção 4.7. As portas

de entrada de NetAnalyser, Post Isolation Status e Unconstrained Reconfigs, podem ser vistas na Figura 4.16, assim como a porta de saída, Secure Reconfigs.

Figura 4.16 – Subpágina NetAnalyser.



São verificadas por NetAnalyser cinco tipos de restrições:

- Isolação de defeito ( $\alpha$ );
- Fonte única ( $\beta$ );
- Capacidade de suprimento ( $\gamma$ );
- Ampacidade ( $\delta$ );
- Ciclo fechado ( $\epsilon$ ).

A verificação é feita pelas subpáginas indicadas entre parênteses que correspondem a instâncias das subpáginas IsFaultSafe, IsSingleActive, HasCapacity, IsAmpSafe e IsLoopSafe, respectivamente. Todas as subpáginas apresentam uma saída do mesmo tipo, ConstANALYSIS.

```
colset ConstANALYSIS = list BOOL;
colset ConstAnalysisSET = list ConstANALYSIS;
```

Cada elemento de uma ficha do tipo ConstANALYSIS tem valor verdadeiro (true) quando a reconfiguração tem uma condição válida para a restrição analisada e falso

(**false**) quando a reconfiguração viola a restrição analisada. Assim, cada ficha nos lugares  $\alpha \dots \epsilon$ , apresentados na Figura 4.16, tem comprimento igual ao número de reconfigurações representadas pela ficha no lugar **Unconstrained Reconfigs**. Então, as reconfigurações são replicadas nos lugares de entrada URx de cada uma das transições  $\alpha \dots \epsilon$  pelo disparo da transição **Mirror Data**, como mostrado na Figura 4.16.

Quando as fichas nos lugares  $\alpha \dots \epsilon$  têm o comprimento indicado no lugar **Length**, a transição **Combine Analyses** é habilitada e, ao seu disparo, uma ficha é colocada no lugar **Report** para representar o resultado final, em uma ficha do tipo **ConstAnalysisSET**. Uma ficha desse tipo contém uma lista do tipo **ConstANALYSIS** para cada reconfiguração indicando sua condição em relação às restrições analisadas em sequência. Aquelas reconfigurações que não possuem nenhuma restrição, todos os valores verdadeiros, são representadas na porta de saída **Secure Reconfigs**.

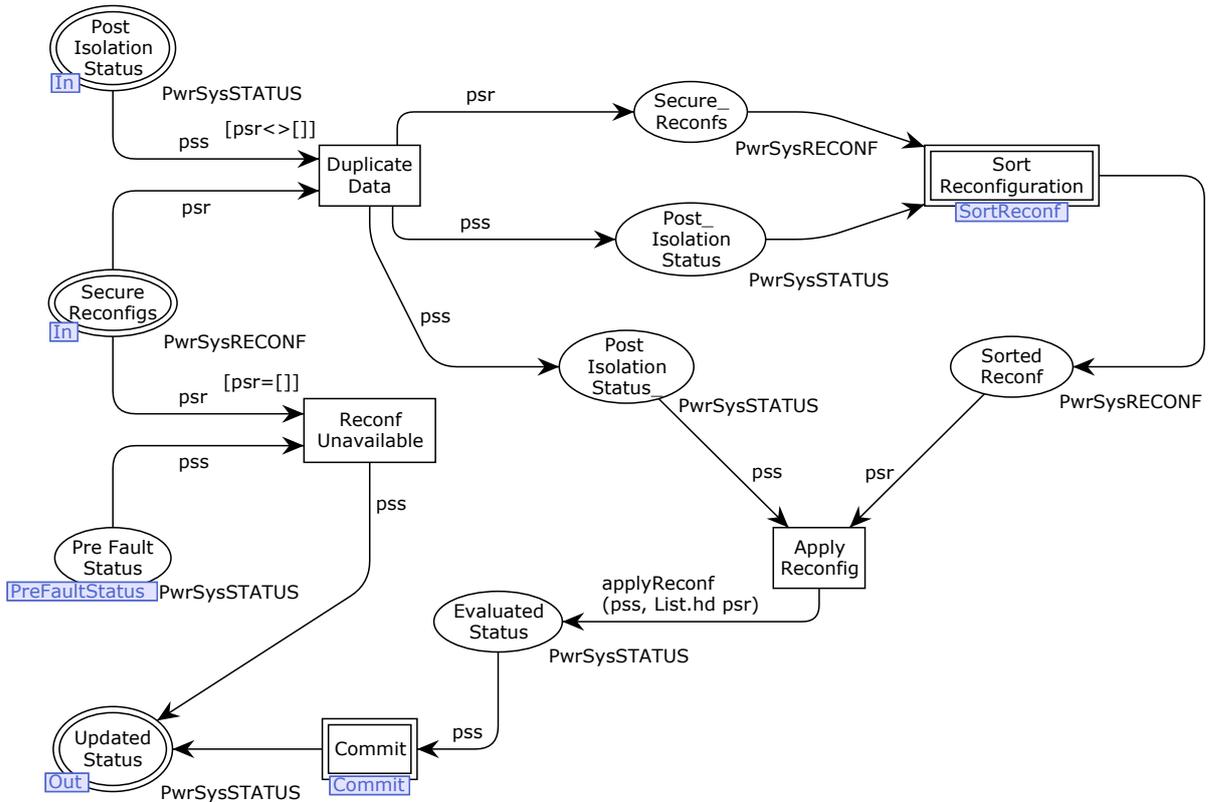
#### 4.6.4 Efetivação

Depois de determinado o conjunto de reconfigurações seguras realizadas na etapa anterior, a escolha da reconfiguração e sua realização ocorre na etapa de efetivação. Estas tarefas são realizadas pela transição de substituição **Proceed**, apresentada na Figura 4.5. A subpágina de mesmo nome é apresentada na Figura 4.17, onde são mostradas duas transições de substituição: **SortReconf** e **Commit**. A primeira é responsável por ordenar o conjunto de reconfigurações seguras representadas pela ficha do tipo **PwrSysRECONF** no lugar **Secure Reconfigs\_**. A segunda aplica efetivamente a reconfiguração no SEP e atualiza o estado de cada um dos elementos.

A escolha da reconfiguração otimizada, realizada na subpágina **Sort Reconf**, é feita segundo o critério de mínimo CNS (Clientes Não Supridos). A reconfiguração que apresentar o menor CNS é selecionada. O cálculo do CNS para cada reconfiguração é feito seguindo os passos mostrados a seguir:

1. Aplica-se a referida reconfiguração para obtenção de um novo estado do SEP (**PwrSysSTATUS**);
2. Determina-se a lista de pontos de carga supridos para o novo estado ( $L_S = [lp_1, \dots, lp_n]$ );
3. Consulta-se a quantidade de consumidores de cada ponto de carga ( $q_i$ ) em  $L_S$  para obter a lista  $Q = [q_1, \dots, q_n]$ ;
4. Calcula-se a quantidade de consumidores atendidos pela rede, ou seja, todos os consumidores conectados ao SEP como sendo  $CS$ ;
5. O CNS para a reconfiguração é obtido fazendo  $CNS = CS - \sum_{i=1}^n q_i$ .

Figura 4.17 – Subpágina Proceed.



Como resultado, a ficha contendo a reconfiguração selecionada é colocada no lugar Sorted Reconf. A partir desta reconfiguração, um novo estado do SEP é calculado, sendo representado pela ficha no lugar Evaluated Status que, por sua vez, é submetida à transição de substituição Commit. Basicamente, esta transição verifica e atualiza o estado dos pontos de carga após a reconfiguração. O resultado é o estado atualizado do SEP representado numa ficha do tipo PwrSysSTATUS no lugar Updated Status, mostrado na Figura 4.17.

## 4.7 Restauração

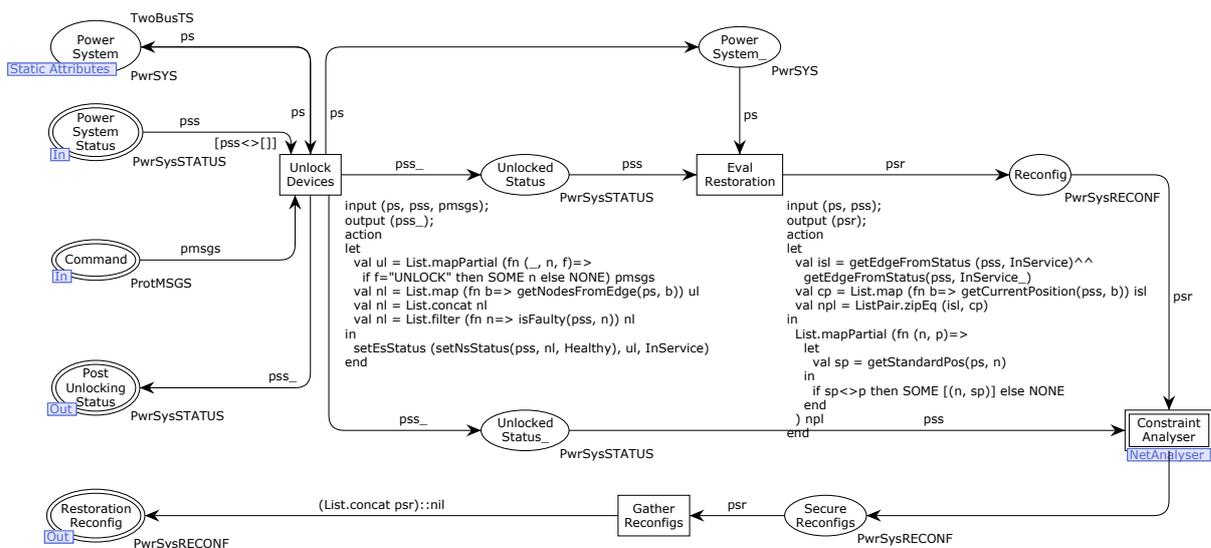
A etapa de reconfiguração e restauração aparentam realizar as mesmas ações, porém é feita uma diferenciação no modelo PSR. A primeira conduz o sistema a um estado de mínimas perdas após a ocorrência de uma falta permanente. Já a restauração ocorre depois de serem feitas todas as correções físicas na rede elétrica que ocasionaram a falta, assim a rede pode operar novamente em sua condição padrão ou normal. Tais correções são representadas pelo desbloqueio (*unlock*) dos dispositivos anteriormente bloqueados durante o processo de reconfiguração.

A função de restauração no modelo PSR é realizada pela transição de substituição Restore mostrada na página principal do modelo (Figura 4.5). A decisão entre reconfigurar ou restaurar é feita pela transição Start (também mostrada na Figura 4.5), como expli-

cado anteriormente na Seção 4.6. Se apenas comandos de desbloqueio são apresentados pelo sistema de proteção, a restauração é selecionada, caso contrário, se mensagens da proteção são apresentadas (por exemplo, *trip*, atuação de funções de proteção e similares) a reconfiguração é selecionada.

Portanto, a etapa de restauração tem como entrada o estado atual do SEP e o desbloqueio dos equipamentos; e como saída o conjunto de manobras que devem ser realizadas no SEP para conduzi-lo a seu estado padrão. A transição de substituição Restore, mostrada na Figura 4.18, representa esta etapa no modelo PSR. O disparo da transição Unlock Devices, dentre outras ações, coloca uma ficha no lugar Unlocked Status representando o estado atualizado do SEP. Ao disparo da transição Eval Restoration uma ficha do tipo PwrSysRECONF é colocada no lugar Reconfig representando as manobras que devem ser realizadas no SEP para reconduzi-lo ao estado normal. Esta reconfiguração (conjunto de manobras) é avaliada pela transição Constraint Analyser, que é uma instância de NetAnalyser, explicada anteriormente na Seção 4.6.3.

Figura 4.18 – Subpágina Restore.



Como resultado, as portas de saída Post Unlocking Status e Restoration Reconfig contêm fichas representando o estado atualizado do SEP e o conjunto de manobras seguras a ser aplicada.

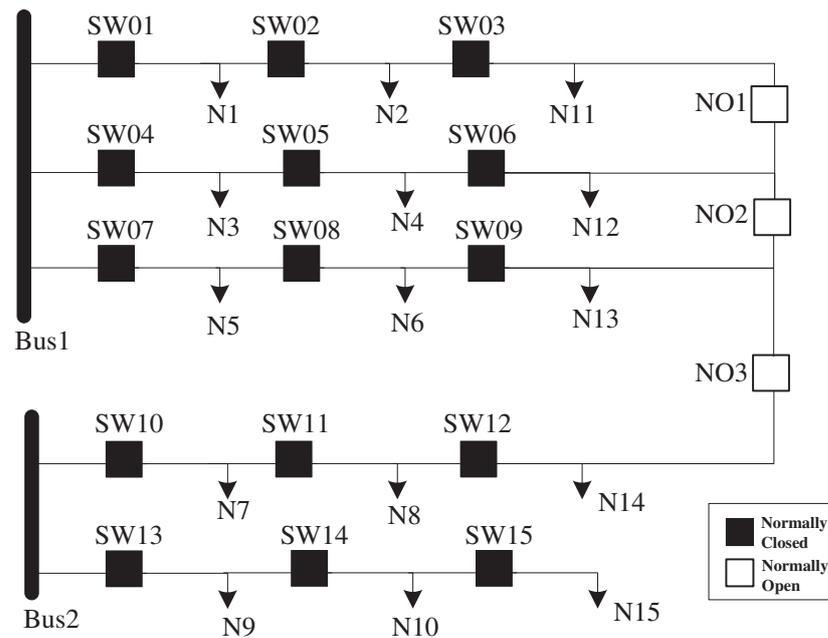
As funções listadas a seguir foram utilizadas na subpágina Restore:

- `getCurrentPosition`, (ver A.10);
- `getStandardPosition`, (ver A.11);
- `setBsStatus`, (ver A.12).

## 4.8 Estudo de Caso

Com intuito de demonstrar a eficiência do modelo PSR, nesta seção são apresentados estudos de caso aplicados a um SEP com as características típicas de uma rede de distribuição de energia elétrica radial. A rede é denominada **TwoBusTS** e é apresentada na Figura 4.19, onde são mostrados dois barramentos (Bus1 e Bus2) e 15 pontos de carga (N1 ... N15).

Figura 4.19 – Sistema de teste TwoBusTS.



A distribuição dos consumidores nos pontos de carga e outros parâmetros da rede são mostrados nas Tabelas 4.1 e 4.2.

Tabela 4.1 – Dados dos pontos de carga e dos barramentos.

Ponto de carga	BUS1	BUS2	N1	N2	N3	N4	N5	N6	N7
Suprimento (A)	90	90	0	0	0	0	0	0	0
Ampacidade (A)	90	90	30	20	30	20	30	20	30
Consumidores	0	0	750	300	790	350	600	280	330

Tabela 4.2 – Dados complementares dos pontos de carga e dos barramentos.

Ponto de carga	N8	N9	N10	N11	N12	N13	N14	N15
Suprimento (A)	0	0	0	0	0	0	0	0
Ampacidade (A)	20	20	11	10	10	10	10	5
Consumidores	170	210	150	250	200	220	130	110

Uma ficha do tipo PwrSYS, conforme apresentado na Seção 4.5, representa a topologia da rede, a capacidade de suprimento de cada ponto de carga, a ampacidade de

cada trecho e a quantidade de consumidores de cada trecho. Desta forma, uma variável com o mesmo nome do sistema de teste, `TwoBusTS`, e contendo os dados apresentados nas Tabelas 4.1 e 4.2 é mostrada a seguir:

```

val TwoBusTS : PwrSYS =
[
E ("SW01", "BUS1", "N1", Closed),
E ("SW02", "N1", "N2", Closed),
E ("SW03", "N2", "N11", Closed),
E ("SW04", "BUS1", "N3", Closed),
E ("SW05", "N3", "N4", Closed),
E ("SW06", "N4", "N12", Closed),
E ("SW07", "BUS1", "N5", Closed),
E ("SW08", "N5", "N6", Closed),
E ("SW09", "N6", "N13", Closed),
E ("SW10", "BUS2", "N7", Closed),
E ("SW11", "N7", "N8", Closed),
E ("SW12", "N8", "N14", Closed),
E ("SW13", "BUS2", "N9", Closed),
E ("SW14", "N9", "N10", Closed),
E ("SW15", "N10", "N15", Closed),
E ("N01", "N11", "N12", Open),
E ("N02", "N12", "N13", Open),
E ("N03", "N13", "N14", Open),
N ("BUS1", 90, 90, 0),
N ("BUS2", 90, 90, 0),
N ("N1", 0, 30, 750),
N ("N2", 0, 20, 300),
N ("N3", 0, 30, 790),
N ("N4", 0, 20, 350),
N ("N5", 0, 30, 600),
N ("N6", 0, 20, 280),
N ("N7", 0, 30, 330),
N ("N8", 0, 20, 170),
N ("N9", 0, 20, 210),
N ("N10", 0, 11, 150),
N ("N11", 0, 10, 250),
N ("N12", 0, 10, 200),
N ("N13", 0, 10, 220),
N ("N14", 0, 10, 130),
N ("N15", 0, 5, 110)
];

```

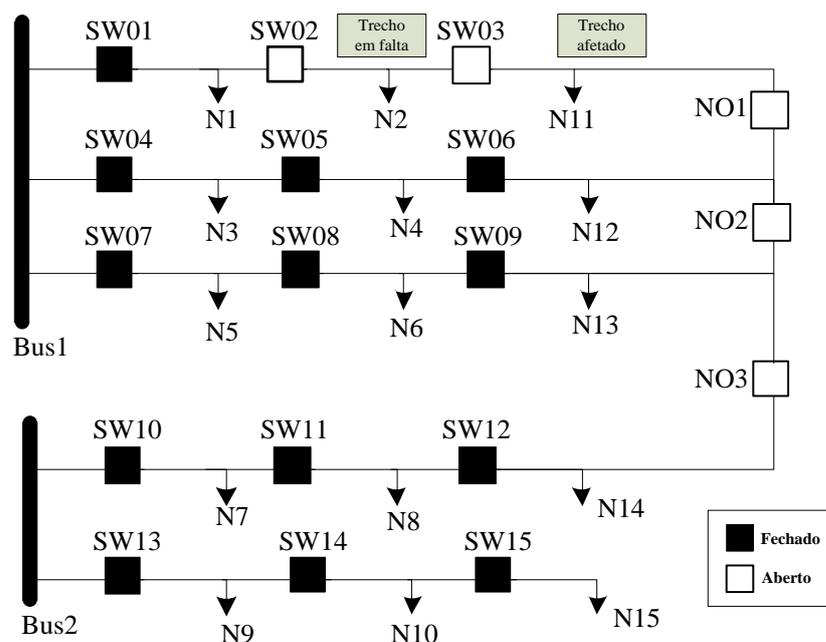
Nas subseções seguintes são apresentados cenários simulando faltas em pontos específicos do sistema de teste proposto para observação do desempenho do modelo PSR. Os cenários seguintes representam uma sequência de faltas (cenários I a III) e, ao final das mesmas, o sistema é restaurado a seu estado padrão (cenário IV).

#### 4.8.1 Cenário I

O primeiro cenário representa a rede de distribuição `TwoBusTS`, inicialmente em seu estado normal, conforme apresentado na Figura 4.19, e após a ocorrência de uma falta na linha entre as chaves SW02 e SW03, afetando diretamente os pontos de carga N2

e N11, conforme apresentado na Figura 4.20. Ao longo do texto esta falta será referida apenas como "falta em N2", apenas para simplificação, apesar de não necessariamente a falta localizar-se exatamente no ponto de carga, mas sim em qualquer localização entre as chaves especificadas. O cenário pós-falta da ocorrência descrita é ilustrado por meio do diagrama unifilar apresentado na Figura 4.20.

Figura 4.20 – Estado pós-falta do sistema teste TwoBusTS - Cenário I.



A atuação dos dispositivos de proteção para a referida ocorrência pode ser resumida pelos eventos mostrados cronologicamente como segue:

- Partida (*start*) dos relés a montante da falta SW02 e SW01;
- Atuação da função de proteção do relé SW02;
- Abertura (*trip*) da chave SW02.

Inicialmente, cada relé que detecta corrente de falta reporta a anomalia na rede e inicia (*start*) a contagem de tempo para possível atuação de alguma das funções de proteção, tais como sobrecorrente, subtensão e similares. Considerando a atuação correta da proteção, a função de proteção do relé SW02 atua e a chave SW02 abre (*trip*) para eliminar a falta<sup>2</sup>.

Os eventos da proteção são representados pela ficha TwoBusTS\_PROT1 de tipo ProtMSGs, como mostrado a seguir:

<sup>2</sup> Ao longo do texto, é utilizada a mesma nomenclatura para o relé e a chave (disjuntor, religador e similares) que é comandada por ele.

```

val TwoBusTS_PROT1 : ProtMSGs =
[
("BUS1", "SW01", "ST"),
("BUS1", "SW02", "ST"),
("BUS1", "SW02", "51"),
("BUS1", "SW02", "52")
];

```

Os dois primeiros elementos da lista representam a partida de SW01 e SW02 (ST), seguidos da atuação da função de sobrecorrente (51). O último elemento representa a abertura de SW02.

Os dados pré-falta da ocorrência são representados através de uma ficha do tipo PwrSysSTATUS, conforme explicado anteriormente na Seção 4.5. Assim, a variável TwoBusTS\_STATUS1, mostrada a seguir, contém o estado inicial (pré-falta) da rede para a ocorrência. É importante observar que todos os relés estão operacionais (STATUS=InService), as chaves estão na posição padrão, as fontes estão operando normalmente (STATUS=Active) e todos os pontos de carga sendo supridos normalmente (STATUS=Supplied).

```

val TwoBusTS_STATUS1 : PwrSysSTATUS =
[
ES ("SW01", Closed, 10, InService),
ES ("SW02", Closed, 2, InService),
ES ("SW03", Closed, 3, InService),
ES ("SW04", Closed, 14, InService),
ES ("SW05", Closed, 5, InService),
ES ("SW06", Closed, 2, InService),
ES ("SW07", Closed, 17, InService),
ES ("SW08", Closed, 8, InService),
ES ("SW09", Closed, 1, InService),
ES ("SW10", Closed, 10, InService),
ES ("SW11", Closed, 2, InService),
ES ("SW12", Closed, 1, InService),
ES ("SW13", Closed, 13, InService),
ES ("SW14", Closed, 5, InService),
ES ("SW15", Closed, 1, InService),
ES ("N01", Open, 0, InService),
ES ("N02", Open, 0, InService),
ES ("N03", Open, 0, InService),
NS ("BUS1", Active),
NS ("BUS2", Active),
NS ("N1", Supplied),
NS ("N2", Supplied),
NS ("N3", Supplied),
NS ("N4", Supplied),
NS ("N5", Supplied),
NS ("N6", Supplied),
NS ("N7", Supplied),
NS ("N8", Supplied),
NS ("N9", Supplied),
NS ("N10", Supplied),
NS ("N11", Supplied),
NS ("N12", Supplied),
NS ("N13", Supplied),

```

```
NS ("N14", Supplied),
NS ("N15", Supplied)
];
```

As variáveis citadas anteriormente `TwoBusTS_PROT1` e `TwoBusTS_STATUS1`, além de `TwoBusTS`, são utilizadas na subpágina *Power System Reconfiguration* (mostrada na Figura 4.5) como marcação inicial dos lugares *Protection*, *Pre Fault Status* e *Power System*, respectivamente, iniciando assim a simulação.

A transição *Start*, mostrada na Figura 4.5, coloca fichas nos lugares `Protection_` e `Pre Fault Status_` indicando que o SEP deve ser reconfigurado e não restaurado, conforme já explicado na Seção 4.5. Desta forma, a transição *Locate* é habilitada para realizar a localização da falta. Como resultado, uma ficha é colocada no lugar *Post Location Status*, mostrada a seguir, representando o novo estado do SEP em que o ponto de carga N2 é classificado como `Faulty_`<sup>3</sup> e o ponto de carga N11 como `Unsupplied_`.

```
1 '[
    ES(("SW01", Closed, 10, InService)),
    ES(("SW02", Open, 2, InService)),
    ES(("SW03", Closed, 3, InService)),
    ES(("SW04", Closed, 14, InService)),
    ES(("SW05", Closed, 5, InService)),
    ES(("SW06", Closed, 2, InService)),
    ES(("SW07", Closed, 17, InService)),
    ES(("SW08", Closed, 8, InService)),
    ES(("SW09", Closed, 1, InService)),
    ES(("SW10", Closed, 10, InService)),
    ES(("SW11", Closed, 2, InService)),
    ES(("SW12", Closed, 1, InService)),
    ES(("SW13", Closed, 13, InService)),
    ES(("SW14", Closed, 5, InService)),
    ES(("SW15", Closed, 1, InService)),
    ES(("N01", Open, 0, InService)),
    ES(("N02", Open, 0, InService)),
    ES(("N03", Open, 0, InService)),
    NS(("BUS1", Active)),
    NS(("BUS2", Active)),
    NS(("N1", Supplied)),
    NS(("N2", Faulty_)),
    NS(("N3", Supplied)),
    NS(("N4", Supplied)),
    NS(("N5", Supplied)),
    NS(("N6", Supplied)),
    NS(("N7", Supplied)),
    NS(("N8", Supplied)),
    NS(("N9", Supplied)),
    NS(("N10", Supplied)),
    NS(("N11", Unsupplied_)),
    NS(("N12", Supplied)),
    NS(("N13", Supplied)),
```

<sup>3</sup> O sufixo `_` é utilizado para indicar um estado calculado pelo modelo PSR, mas não necessariamente efetivado fisicamente na rede elétrica.

```

    NS(("N14", Supplied)),
    NS(("N15", Supplied))
]

```

A ficha de saída no lugar **Post Location Status** habilita a transição **Isolate** para que a mesma realize o isolamento da falta. Logo, os caminhos entre o nó faltoso (N2) e as fontes (BUS1 e BUS2) são determinados pela transição **Find Neighbour Sequences** e o resultado é mostrado a seguir:

```

1 '[
["SW03", "N01", "N02", "N03", "SW12", "SW11", "SW10"],
["SW03", "N01", "N02", "SW09", "SW08", "SW07"],
["SW03", "N01", "SW06", "SW05", "SW04"],
["SW02", "SW01"]
]

```

As três primeiras listas indicam o caminho entre o ponto de carga N2 e a fonte BUS1. A primeira chave isoladora deve ser a chave mais próxima e disponível para operar. Como SW03 está disponível para operação (**STATUS=InService**), segundo a ficha no lugar **Post Location Status** mostrada anteriormente, e ainda é a chave mais próxima da falta, ela é selecionada. O último caminho, mostrado na ficha anterior, indica SW02 como sendo a chave mais próxima disponível que é selecionada como a segunda isoladora. Caso SW02 estivesse fora de serviço (**STATUS=OutOfService**), SW01 seria selecionada. Portanto, SW03 e SW02 são os isoladores de N2 e passam a ter **STATUS=OutOfServ\_** e **POSITION=Open\_**. Como resultado, uma nova ficha com a atualização do estado da rede elétrica é colocada no lugar **Post Isolation Status** como mostrado a seguir:

```

1 '[
ES(("SW01", Closed, 10, InService)),
ES(("SW02", Open_, 2, OutOfServ_)),
ES(("SW03", Open_, 3, OutOfServ_)),
ES(("SW04", Closed, 14, InService)),
ES(("SW05", Closed, 5, InService)),
ES(("SW06", Closed, 2, InService)),
ES(("SW07", Closed, 17, InService)),
ES(("SW08", Closed, 8, InService)),
ES(("SW09", Closed, 1, InService)),
ES(("SW10", Closed, 10, InService)),
ES(("SW11", Closed, 2, InService)),
ES(("SW12", Closed, 1, InService)),
ES(("SW13", Closed, 13, InService)),
ES(("SW14", Closed, 5, InService)),
ES(("SW15", Closed, 1, InService)),
ES(("N01", Open, 0, InService)),
ES(("N02", Open, 0, InService)),
ES(("N03", Open, 0, InService)),
NS(("BUS1", Active)),
NS(("BUS2", Active)),
NS(("N1", Supplied)),
NS(("N2", Faulty_)),
NS(("N3", Supplied)),

```

```

NS(("N4", Supplied)),
NS(("N5", Supplied)),
NS(("N6", Supplied)),
NS(("N7", Supplied)),
NS(("N8", Supplied)),
NS(("N9", Supplied)),
NS(("N10", Supplied)),
NS(("N11", Unsupplied_)),
NS(("N12", Supplied)),
NS(("N13", Supplied)),
NS(("N14", Supplied)),
NS(("N15", Supplied))
]

```

A marcação anterior habilita o disparo da transição **Reconfigure**, mostrada na Figura 4.5, para determinar a reconfiguração a ser aplicada no SEP. Na Figura 4.15 são apresentados os detalhes da subpágina **Reconfigure** em que os caminhos entre os pontos de carga saudáveis, porém desenergizados (N11), e as fontes são determinados pela transição **Find Paths**. Para o caso em análise, quatro caminhos,  $p_1 \dots p_4$ , são determinados:

- $p_1 = ["NO1", "NO2", "NO3", "SW12", "SW11", "SW10"];$
- $p_2 = ["NO1", "NO2", "SW09", "SW08", "SW07"];$
- $p_3 = ["NO1", "SW06", "SW05", "SW04"];$
- $p_4 = ["SW03", "SW02", "SW01"].$

Cada caminho tem cada uma de suas chaves avaliada quanto às restrições de dispositivo, ou seja, se a chave está disponível para ser manobrada ou não. Esta tarefa é realizada pela transição **Device Constraints** e o resultado é representado pela ficha no lugar **Unconstrained Paths**. Os caminhos que contém qualquer das chaves selecionadas para isolar N2, SW02 e SW03, são descartados do restante da análise. Assim,  $p_4$  é descartado.

Os caminhos restantes são passados para a transição **Eval Reconfig** para que as respectivas reconfigurações,  $r_1 \dots r_3$ , sejam determinadas. A tarefa desta transição é remover as chaves que já se encontram na posição fechada, portanto não precisam ser manobradas, bem como acrescentar as chaves isoladoras na posição aberta. Logo:

- $r_1 = [("SW02", Open_), ("SW03", Open_), ("NO1", Closed_), ("NO2", Closed_), ("NO3", Closed_)];$
- $r_2 = [("SW02", Open_), ("SW03", Open_), ("NO1", Closed_), ("NO2", Closed_)];$
- $r_3 = [("SW02", Open_), ("SW03", Open_), ("NO1", Closed_)].$

As reconfigurações acima são avaliadas pela transição **Network Constraints**, mostrada na Figura 4.16, quanto às restrições de rede (apresentadas na Seção 4.2): Isolação de defeito

( $\alpha$ ), Fonte única ( $\beta$ ), Capacidade de suprimento ( $\gamma$ ), Ampacidade ( $\delta$ ) e Ciclo fechado ( $\epsilon$ ). A checagem das restrições é feita pelas transições de substituição **alpha**, **beta**, **gamma**, **delta** e **epsilon**, apresentadas anteriormente na Figura 4.16. O resultado para cada uma das restrições é mostrado a seguir:

- $\alpha = [true, true, true]$ ;
- $\beta = [false, true, true]$ ;
- $\gamma = [true, true, true]$ ;
- $\delta = [true, true, true]$ ;
- $\epsilon = [false, false, true]$ .

O resultado para restrição  $\alpha$  (**IsFaultSafe**) é verdadeiro para as três reconfigurações analisadas, ou seja,  $r_1$ ,  $r_2$  e  $r_3$  mantêm o ponto de carga faltoso (N2) isolado.

A reconfiguração  $r_1$  é falsa para as restrições  $\beta$  e  $\epsilon$ , pois ao fechar as chaves NO1, NO2 e NO3 vários pontos de carga passam a ser supridos simultaneamente por BUS1 e BUS2 (violando a restrição  $\beta$ ) e ainda vários caminhos fechados (*loops*) passam a existir para BUS1 (violando a restrição  $\epsilon$ ).

Nenhuma das reconfigurações sobrecarrega qualquer das fontes, BUS1 ou BUS2, logo  $\gamma$  é verdadeiro para todas as reconfigurações. Da mesma forma, todos os trechos têm sua ampacidade respeitada,  $\delta$  é verdadeiro para todas as reconfigurações.

Apenas  $r_3$  é livre de *loops*, como mostrado por  $\epsilon$ . Desta forma,  $r_3$  é selecionado como sendo a única reconfiguração segura que pode ser aplicada. Então o lugar **Secure Reconfigs** da subpágina **NetAnalyser** (Figura 4.16) passa a ter a seguinte marcação:

```
1' [[("SW02",Open_), ("SW03",Open_), ("N01",Closed_)]]
```

A marcação anterior em conjunto com a marcação do lugar **Post Isolation Status\_**, idêntica a do lugar **Post Isolation Status** e já mostrada na presente subseção, compõe a saída da transição **Reconfigure** (Figura 4.5). Este resultado é então efetivado pelas transições **Proceed Reconfigure** e **Proceed** mostradas na Figura 4.5. O estado do SEP é então atualizado em definitivo pela transição **Proceed** que passa a ser representado pela marcação do lugar **Pre Fault Status** como apresentado a seguir:

```
1' [
ES(("SW01",Closed,10,InService)),
ES(("SW02",Open,2,OutOfServ)),
ES(("SW03",Open,3,OutOfServ)),
ES("SW04",Closed,14,InService)),
ES(("SW05",Closed,5,InService)),
ES(("SW06",Closed,2,InService)),
```

```

ES("SW07",Closed,17,InService)),
ES(("SW08",Closed,8,InService)),
ES(("SW09",Closed,1,InService)),
ES("SW10",Closed,10,InService)),
ES(("SW11",Closed,2,InService)),
ES(("SW12",Closed,1,InService)),
ES("SW13",Closed,13,InService)),
ES(("SW14",Closed,5,InService)),
ES(("SW15",Closed,1,InService)),
ES(("N01",Closed,0,InService)),
ES(("N02",Open,0,InService)),
ES(("N03",Open,0,InService)),
NS(("BUS1",Active)),
NS(("BUS2",Active)),
NS(("N1",Supplied)),
NS(("N2",Faulty)),
NS(("N3",Supplied)),
NS(("N4",Supplied)),
NS(("N5",Supplied)),
NS(("N6",Supplied)),
NS(("N7",Supplied)),
NS(("N8",Supplied)),
NS(("N9",Supplied)),
NS(("N10",Supplied)),
NS(("N11",Supplied)),
NS(("N12",Supplied)),
NS(("N13",Supplied)),
NS(("N14",Supplied)),
NS(("N15",Supplied))
]

```

Estão destacadas todas as mudanças entre o estado pré-falta, no início da simulação, e a saída apresentada, o estado pós-reconfiguração. Como resultado, o ponto de carga N11, que estava afetado pela falta em N2, é reenergizado reduzindo o número de CNS de 550 para 300, conforme o número de consumidores indicado nas Tabelas 4.1 e 4.2. O estado pós-reconfiguração do sistema teste `TwoBusTS` é mostrado na Figura 4.21.

#### 4.8.2 Cenário II

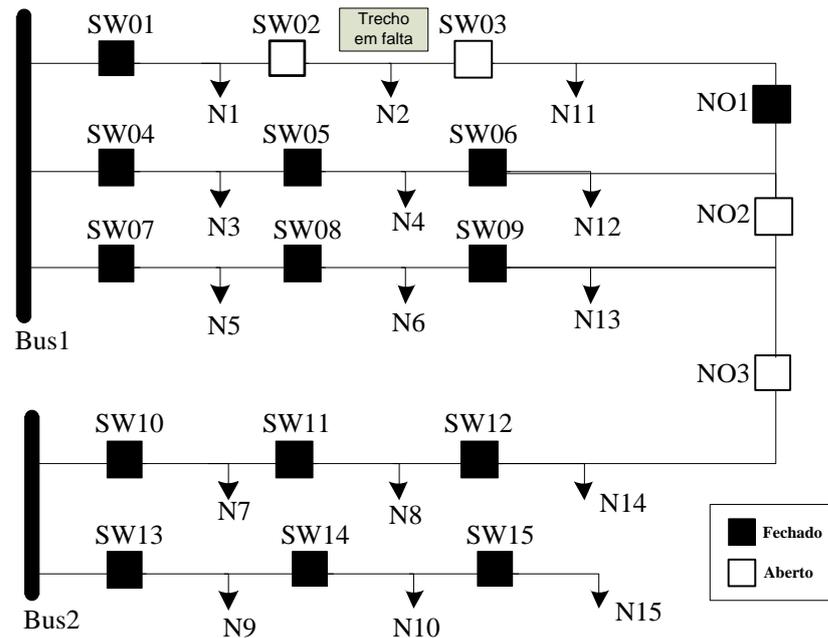
Ainda utilizando-se do sistema teste denominado `TwoBusTS`, o cenário seguinte inicia-se a partir do estado final do cenário anterior, mostrado na Figura 4.21. Neste novo cenário uma falta é simulada no ponto de carga N11 sendo a atuação do sistema de proteção representada pela ficha no lugar `Protection`, conforme mostrado a seguir:

```

val TwoBusTS_PROT2 =
[
("BUS1", "SW04", "ST"),
("BUS1", "SW05", "ST"),
("BUS1", "SW06", "ST"),

```

Figura 4.21 – Estado pós-reconfiguração do sistema teste TwoBusTS - Cenário I.



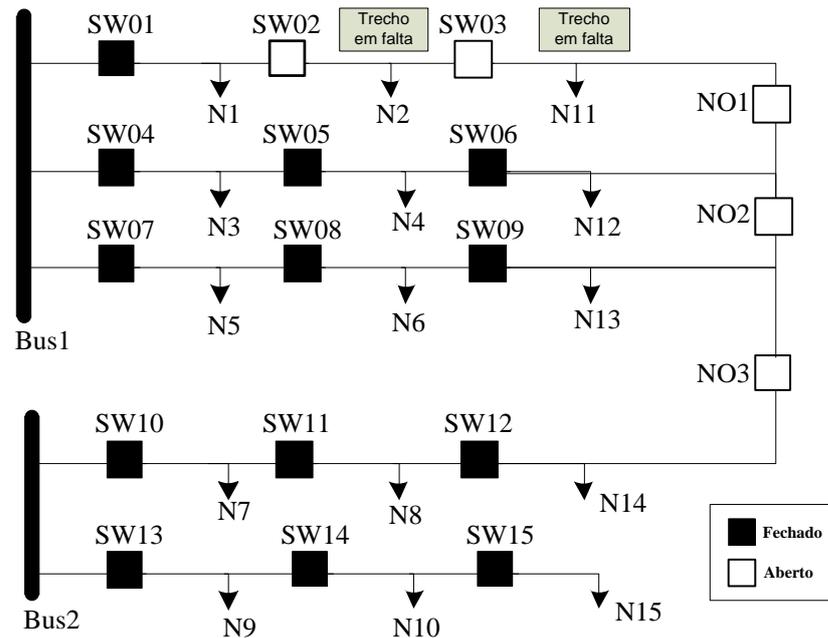
```
( "BUS1", "NO1", "ST" ),
( "BUS1", "NO1", "51" ),
( "BUS1", "NO1", "52" )
];
```

A ficha mostrada representa a partida dos relés SW04, SW05, SW06 e NO1; a atuação da função de proteção de sobrecorrente (51); e a abertura do religador NO1, ou simplesmente da chave NO1. O estado pós-falta do SEP neste novo cenário é apresentado na Figura 4.22.

A simulação é iniciada com a transição **Start** (apresentada na Figura 4.5) determinando que o SEP deve ser reconfigurado, logo, a transição **Locate** é habilitada e o estado posterior à localização da falta é calculado como mostrado a seguir:

```
1 '[
ES("SW01",Closed,9,InService)),
ES("SW02",Open,0,OutOfServ)),
ES("SW03",Open,0,OutOfServ)),
ES("SW04",Closed,18,InService)),
ES("SW05",Closed,9,InService)),
ES("SW06",Closed,6,InService)),
ES("SW07",Closed,17,InService)),
ES("SW08",Closed,8,InService)),
ES("SW09",Closed,1,InService)),
ES("SW10",Closed,10,InService)),
ES("SW11",Closed,2,InService)),
ES("SW12",Closed,1,InService)),
ES("SW13",Closed,13,InService)),
ES("SW14",Closed,5,InService)),
ES("SW15",Closed,1,InService)),
```

Figura 4.22 – Estado pós-falta do sistema teste TwoBusTS - Cenário II.



```

ES(("N01",Open,4,InService)),
ES(("N02",Open,0,InService)),
ES(("N03",Open,0,InService)),
NS("BUS1",Active)),
NS("BUS2",Active)),
NS("N1",Supplied)),
NS("N2",Faulty)),
NS("N3",Supplied)),
NS("N4",Supplied)),
NS("N5",Supplied)),
NS("N6",Supplied)),
NS("N7",Supplied)),
NS("N8",Supplied)),
NS("N9",Supplied)),
NS("N10",Supplied)),
NS("N11",Faulty_)),
NS("N12",Supplied)),
NS("N13",Supplied)),
NS("N14",Supplied)),
NS("N15",Supplied))
]

```

Observa-se que nenhum dos pontos de carga, N1 ... N15, possui STATUS=Unsupplied\_ o que indicaria que o ponto de carga foi afetado pela falta no ponto de carga N11, que, por sua vez, é o ponto de carga faltoso (STATUS=Faulty\_).

A transição *Isolate* dá prosseguimento à simulação determinando quais as chaves para isolamento de N11. As chaves ideais para o isolamento seriam SW03 e NO1. Como a primeira já encontra-se aberta e ainda fora de operação (STATUS=OutOfServ), apenas NO1 é selecionada como mostra a marcação do lugar de saída de *Isolate* (ver Figura 4.5), mostrada a seguir:

```

1 '[
ES(("SW01",Closed,9,InService)),
ES(("SW02",Open,0,OutOfServ)),
ES(("SW03",Open,0,OutOfServ)),
ES(("SW04",Closed,18,InService)),
ES(("SW05",Closed,9,InService)),
ES(("SW06",Closed,6,InService)),
ES(("SW07",Closed,17,InService)),
ES(("SW08",Closed,8,InService)),
ES(("SW09",Closed,1,InService)),
ES(("SW10",Closed,10,InService)),
ES(("SW11",Closed,2,InService)),
ES(("SW12",Closed,1,InService)),
ES(("SW13",Closed,13,InService)),
ES(("SW14",Closed,5,InService)),
ES(("SW15",Closed,1,InService)),
ES(("N01",Open_,4,OutOfServ_)),
ES(("N02",Open,0,InService)),
ES(("N03",Open,0,InService)),
NS(("BUS1",Active)),
NS(("BUS2",Active)),
NS(("N1",Supplied)),
NS(("N2",Faulty)),
NS(("N3",Supplied)),
NS(("N4",Supplied)),
NS(("N5",Supplied)),
NS(("N6",Supplied)),
NS(("N7",Supplied)),
NS(("N8",Supplied)),
NS(("N9",Supplied)),
NS(("N10",Supplied)),
NS(("N11",Faulty_)),
NS(("N12",Supplied)),
NS(("N13",Supplied)),
NS(("N14",Supplied)),
NS(("N15",Supplied))
]

```

Como não existem pontos de carga não supridos e livres de defeito, não existem reconfigurações possíveis a serem realizadas. Como resultado, a transição **Reconfigure** coloca uma ficha vazia no lugar **Secure Reconfig**. A simulação prossegue para recolocar o novo estado do sistema elétrico, praticamente inalterado, no lugar **Pre Fault Status** mantendo o modelo PSR apto para a simulação de uma nova ocorrência no SEP. O cenário pós-falta deste cenário mantêm-se como apresentado na Figura 4.22.

O presente cenário evidencia a eficiência do modelo em preservar a segurança operacional do sistema elétrico evitando possíveis erros por parte de operadores como, por exemplo, a tentativa de reenergizar N11 por meio do fechamento de SW02 e SW03, que é um caminho que apresenta uma falta anterior e ainda não corrigida na rede elétrica.





Similarmente, em caso de indisponibilidade de Bus2 e dependendo do carregamento,  $r_2$  pode ser considerada inválida quanto à capacidade das fontes, restrição  $\gamma$ .

#### 4.8.4 Cenário IV

Considerando a correção de todos os defeitos que ocasionaram as faltas no sistema TwoBusTS e estudadas nos casos anteriores, faz-se necessário que o operador informe manualmente quais as chaves que deixam a condição de bloqueio, voltando ao estado operacional. Esta ação também é representada por meio da ficha no lugar **Protection**. A ficha mostrada a seguir representa o desbloqueio dos equipamentos bloqueados pelas ações do modelo PSR nos cenários apresentados anteriormente.

```
val TwoBusTS_PROT4 =
[
("BUS1", "SW02", "UNLOCK"),
("BUS1", "SW03", "UNLOCK"),
("BUS1", "SW10", "UNLOCK"),
("BUS1", "SW11", "UNLOCK"),
("BUS1", "NO1", "UNLOCK")
];
```

Com isso, a transição **Start** determina que o sistema elétrico deve ser restaurado, levando-o a seu estado normal ou mais aproximado do normal. Para isso, uma reconfiguração (conjunto de manobras) deve ser determinada e aplicada ao SEP. A transição **Restore**, apresentada na Figura 4.18, determina a reconfiguração que precisa ser realizada como sendo:

```
[("SW02",Closed),("SW03",Closed),("SW10",Closed),
("SW11",Closed),("NO3",Open)]
```

Basicamente, a reconfiguração realizada altera a posição das chaves que estavam fora de operação para sua posição padrão, como indicado no campo **POSITION** da ficha de tipo **PwrSYS**. A transição **Restore** ainda utiliza-se de uma instância de **Net Analyser** para verificar se a reconfiguração não possui alguma restrição. Por fim, a reconfiguração é aplicada no sistema TwoBusTS trazendo-o à sua topologia padrão, como representado através da ficha **PwrSysSTATUS**, mostrada a seguir:

```
[
ES(("SW01",Closed,10,InService)),
ES(("SW02",Closed,0,InService)),
ES(("SW03",Closed,0,InService)),
ES(("SW04",Closed,13,InService)),
```

```

ES(("SW05",Closed,5,InService)),
ES(("SW06",Closed,2,InService)),
ES(("SW07",Closed,16,InService)),
ES(("SW08",Closed,10,InService)),
ES(("SW09",Closed,6,InService)),
ES(("SW10",Closed,0,InService)),
ES(("SW11",Closed,0,InService)),
ES(("SW12",Closed,2,InService)),
ES(("SW13",Closed,8,InService)),
ES(("SW14",Closed,5,InService)),
ES(("SW15",Closed,1,InService)),
ES(("N01",Open,0,InService)),
ES(("N02",Open,0,InService)),
ES(("N03",Open,3,InService)),
NS(("BUS1",Active)),
NS(("BUS2",Active)),
NS(("N1",Supplied)),
NS(("N2",Supplied)),
NS(("N3",Supplied)),
NS(("N4",Supplied)),
NS(("N5",Supplied)),
NS(("N6",Supplied)),
NS(("N7",Supplied)),
NS(("N8",Supplied)),
NS(("N9",Supplied)),
NS(("N10",Supplied)),
NS(("N11",Supplied)),
NS(("N12",Supplied)),
NS(("N13",Supplied)),
NS(("N14",Supplied)),
NS(("N15",Supplied))
]

```

A marcação anterior mostra que todos os pontos de carga estão supridos e todas as chaves estão na posição padrão, conforme apresentado na Figura 4.19.

## 4.9 Conclusão

Neste capítulo foi apresentado o modelo *Power System Restoration*, fundamentado em redes de Petri coloridas, bem como detalhes e estudo de caso que demonstra as principais características do modelo: reusabilidade e extensibilidade. Os estudos realizados e apresentados no capítulo sugerem que a implantação do modelo no centro de controle é de grande valor para as distribuidoras de energia elétrica, ao passo que garante a segurança do sistema elétrico em circunstâncias críticas para os operadores do sistema. Como resultado, uma rede mais confiável, melhores índices de confiabilidade, melhor percepção de qualidade por parte dos consumidores e a não ocorrência de penalidades dos órgãos reguladores é factível.

## 5 Conclusões

O presente capítulo apresenta as conclusões finais do trabalho bem como os pontos de destaque da pesquisa e suas limitações. As observações são apresentadas para ambas as contribuições: o algoritmo MOSP (*Multi-Objective Switch Placement*) para posicionamento de chaves e para o modelo de restauração automática PSR (*Power System Restoration*). É apresentada uma visão de trabalhos futuros e melhorias que podem ser realizadas em cada uma das contribuições. Por fim, são apresentadas as publicações obtidas ao longo do desenvolvimento deste trabalho.

### 5.1 Algoritmo MOSP

#### 5.1.1 Destaques

O algoritmo MOSP desenvolvido neste trabalho destaca-se como sendo uma ferramenta valiosa para concessionárias de distribuição de energia que buscam melhoria da confiabilidade pelo uso da automação da distribuição. O MOSP auxilia no delicado processo de posicionamento de chaves na rede elétrica resultando num planejamento e tomada de decisão mais seguros para a concessionária e profissionais envolvidos. O uso do algoritmo proposto confere flexibilidade ao processo de posicionamento de chaves, uma vez que oferece como resultado final um conjunto de soluções com diferentes quantidades de chaves. Desta forma, a tomada de decisão resume-se a um processo de análise financeira em relação à confiabilidade esperada pela companhia. Ou seja, a viabilidade financeira para a instalação das chaves permite a instalação de quantas chaves? Para esta quantidade de chaves a melhoria na confiabilidade atende às expectativas da companhia?

Além de facilitar o processo de tomada de decisão para o posicionamento de chaves, o algoritmo MOSP requer dados de entrada bastante simplificados. O processo de coleta de dados no ambiente de concessionárias de energia elétrica não é um processo simples. Não é raro o desconhecimento de determinados dados pela própria concessionária. Pode-se citar, como o exemplo, o parâmetro de taxa de falhas dos segmentos da rede elétrica. Este parâmetro é comumente utilizado na literatura para o posicionamento de chaves. Porém, durante a participação em projeto de Pesquisa e Desenvolvimento em parceria com uma concessionária brasileira observou-se o desconhecimento da companhia sobre tal dado o que inviabilizaria a aplicação de muitas das propostas existentes na literatura. Assim, buscou-se uma estratégia de desenvolvimento do algoritmo que demandasse dados de entrada simplificados, tornando realizável o desenvolvimento de uma proposta inovadora para o problema. Como resultado, o algoritmo MOSP desenvolvido demanda apenas a

topologia da rede elétrica e a quantidade de consumidores nos pontos de carga desta rede.

### 5.1.2 Limitações

O algoritmo MOSP possui algumas limitações em termos do tempo de resposta. Para os experimentos computacionais foi utilizado um microcomputador equipado com processador Intel® Core™i5 @ 2.66GHZ. Observou-se que o tempo para realizar um experimento computacional é proporcional à quantidade de ramos da rede elétrica em estudo, do tamanho da população e da quantidade de instâncias do algoritmo rodando em paralelo. O tempo para realizar o experimento computacional para o sistema teste RBTS4, por exemplo, foi de 70 a 90 minutos para uma população de tamanho igual à quantidade de ramos desse sistema (71 ramos) com 4 instâncias em paralelo. Porém, considerando que o algoritmo se destina a uma etapa de planejamento do sistema elétrico e não a uma operação em tempo real, o tempo de resposta apresentado é aceitável.

O algoritmo MOSP também limita-se a redes elétricas de topologia radial. Outras topologias, tais como malhada (*mesh*), interconectada, dentre outras, não são contempladas pelo algoritmo. Porém, a imensa maioria das redes de distribuição de energia elétrica são radiais ou operam radialmente. Por isso, escolheu-se contemplar inicialmente este tipo de rede.

## 5.2 Modelo PSR

### 5.2.1 Destaques

O modelo *Power System Restoration* tem como principal destaque o fato de que seu funcionamento é independente da topologia da rede elétrica. Pode-se afirmar que o modelo PSR está apto a realizar a restauração de qualquer rede de distribuição elétrica radial. Esta propriedade do modelo é conseguida devido a todos os parâmetros físicos da rede, tais como quantidade de consumidores, localização de consumidores, potência e quantidade de transformadores, dentre outros, estarem contidos nos dados de entrada (fichas de entrada do modelo RPC) e desacoplados da estrutura do modelo. Como resultado, quaisquer duas redes de distribuição distintas podem ser simuladas utilizando exatamente o mesmo modelo PSR tendo como única diferença as fichas de entrada. Esta característica torna o modelo PSR bastante extensível a qualquer rede elétrica radial de qualquer concessionária de energia.

Outro ponto de destaque é a estratégia de desenvolvimento modularizada e hierárquica adotada. A estrutura hierárquica e modular facilita a implementação do modelo PSR em um sistema computacional. Desta forma, o aproveitamento e reuso de módulos do sistema é facilmente percebida na estrutura hierarquizada do modelo. Com isso, grandes

benefícios para a implementação efetiva do modelo PSR em um *software* podem ser alcançados. Toda a depuração é facilitada, uma vez que durante a etapa de modelagem propriamente dita a maioria das falhas de funcionamento já foram corrigidas facilitando a etapa de implementação.

O modelo PSR também se destaca por estar imune a possíveis falhas ocasionadas pela descoordenação do sistema de proteção. Como resultado, tais falhas da proteção não comprometem o processo de restauração apresentado pelo modelo PSR.

### 5.2.2 Limitações

Assim, como o algoritmo MOSP, o modelo PSR aplica-se apenas a redes de distribuição de topologia radial. Não estão previstas outras topologias de rede, uma vez que os algoritmos de busca implementados não são capazes de realizar esta busca em outras topologias de rede.

O modelo também não contempla a ocorrência de múltiplas faltas no sistema elétrico, ou seja, a ocorrência de uma nova falta antes da finalização do procedimento de restauração já iniciado não está prevista no modelo. Pressupõe-se, para o correto funcionamento do modelo PSR, que as faltas acontecem em intervalos de tempo suficiente para que as manobras sejam realizadas na sua totalidade.

Também não está previsto no modelo PSR a adaptação do sistema de proteção à nova topologia pós-reconfiguração do sistema elétrico. Ao efetuar mudanças na topologia da rede elétrica é necessário adequar também o sistema de proteção de cada alimentador envolvido, pois a rede vista pelos religadores pode ter sido acrescida de um novo trecho ou ter tido algum trecho removido.

Com relação à confiabilidade dos dispositivos, sistema de proteção e do sistema de comunicação considera-se que estes sejam confiáveis, ou seja, que todos os dispositivos sempre operam corretamente e que não há falhas no sistema de comunicação. Assim, o modelo PSR não prevê anomalias no funcionamento dos equipamentos de maneira geral considerando que suas atuações sempre estão corretas. Entretanto, o caso de descoordenação da proteção está plenamente contemplado pelo modelo PSR não comprometendo o resultado final.

## 5.3 Trabalhos futuros

A seção atual apresenta uma visão de melhorias que podem ser aplicadas às contribuições resultantes deste trabalho. São apresentadas melhorias para o algoritmo MOSP e para o modelo de restauração automática PSR.

### 5.3.1 Algoritmo MOSP

Tornar o algoritmo MOSP compatível com outras topologias de rede elétrica, tais como malhada. Desta forma, o algoritmo se tornaria ainda mais flexível para adequar-se a redes de distribuição heterogêneas, compostas por topologias diversas. Para alcançar este objetivo, um estudo propondo a adequação do algoritmo MOSP deve ser realizado para adaptar ou substituir os algoritmos de busca para sistemas elétricos em diferentes topologias. A dificuldade desta tarefa consiste em fazer com que o modelo se adeque a topologias que podem ter características conflitantes. Como alternativa pode-se desenvolver versões do algoritmo para diferentes topologias, abrindo-se mão, em parte, da característica de generalidade do algoritmo.

Apesar de o algoritmo MOSP se destinar a uma ação considerada de planejamento, seria desejável menores tempos de resposta, o que facilitaria a análise de um maior número de configurações da rede elétrica em um menor intervalo de tempo. O uso de técnicas de computação paralela e/ou distribuída mais sofisticadas, o uso de equipamentos de maior capacidade computacional e a otimização do código fonte são ações que contribuiriam significativamente para a redução do tempo de resposta. Uma estratégia ainda mais efetiva para redução do tempo de resposta que pode ser adotada é a implementação do algoritmo utilizando uma linguagem de programação (C, C#, Java e similares) ao invés de *softwares* como o MATLAB<sup>®</sup>, empregado para o desenvolvimento desta primeira versão do algoritmo MOSP.

Do ponto de vista da qualidade das soluções apresentadas pelo algoritmo MOSP, é desejável que haja pelo menos uma solução com uma única chave, duas chaves, etc, até a quantidade máxima de chaves admissível para a rede elétrica. A necessidade desta característica é reforçada pelo fato de o algoritmo MOSP propor um conjunto de soluções como resultado. Para isso, faz-se necessário ampliar a diversidade das soluções obtidas pelo algoritmo, mas de forma direcionada, em regiões específicas do espaço de busca. Este refinamento do algoritmo pode ser alcançado por meio de estudos mais detalhados sobre a técnica de *Particle Swarm Optimization* ou ainda a proposição de uma técnica específica que tenha esta característica.

Ainda tratando da questão da qualidade das soluções do algoritmo, refinamentos relacionados à otimização multiobjetivo podem ser adotados. Dentre eles, destaca-se a adoção de técnicas e algoritmos mais eficientes em substituição ao método atualmente empregado, WSM (do inglês, *Weighted Sum Method*) que apresenta-se como o mais tradicional e simplificado. Métodos clássicos tais como, *Weighted Metric Method*, o método de *Benson* e o *Goal Programming Method* merecem ser investigados com mais profundidade para comparação de desempenho.

### 5.3.2 Modelo PSR

Quando alterações na topologia da rede elétrica ocasionadas por manobras executadas pelo modelo PSR são realizadas, a nova topologia continua a ser protegida por relés ajustados para a topologia padrão da rede elétrica e conseqüentemente desatualizada. Na maioria dos casos a nova topologia demanda também novos ajustes no sistema de proteção seguindo as alterações da topologia. Portanto, faz-se necessário que o sistema de proteção adapte-se ao dinamismo conferido pelo modelo PSR ao sistema elétrico garantindo seu funcionamento adequado. Para isso, o modelo PSR necessitaria de uma funcionalidade de Mudança de Grupos de Ajuste dos relés de proteção. Esta nova funcionalidade tornaria o modelo PSR mais completo, melhor adaptado e conferiria maior segurança aos SEP.

Outra questão relevante a ser tratada pelo modelo PSR é o uso de geração distribuída e/ou microrredes. Considerando que o uso destas fontes passe a ser mais comum no futuro, a adaptação do modelo PSR de forma a considerar estes elementos na rede elétrica deve ser perseguida. Uma vez que uma fonte alternativa de energia está disponível para atender um determinado cliente, a análise do modelo PSR é afetada, já que a presença de outras fontes no sistema altera inteiramente o comportamento dos algoritmos de busca utilizados no modelo PSR. Assim, um estudo aprofundado visando contemplar o uso de múltiplas fontes no sistema elétrico agregaria maior capacidade ao modelo PSR.

Outra questão relevante, mas que não é tratada pelo modelo PSR é a ocorrência de faltas múltiplas, ou seja, antes que uma falta tenha seu tratamento concluído pelo modelo PSR, uma nova falta se apresenta no SEP. No estágio atual de desenvolvimento do modelo não está prevista a ocorrência de múltiplas faltas e, caso isso aconteça, resultados inconsistentes podem ser apresentados. Apesar de pouco provável, a ocorrência de faltas múltiplas no sistema elétrico pode ocorrer e comprometeria o resultado apresentado.

Para o desenvolvimento do modelo PSR foram consideradas apenas as chaves telecontroladas. Chaves manuais, ou seja, manobradas por equipes de campo, não são consideradas no modelo. Porém, estas chaves possuem um papel fundamental nas redes elétricas. Assim mesmo, não há indícios de que o estado de tais chaves venha a ser monitorado através do centro de controle. Portanto, considerar este tipo de chave tornaria o modelo PSR mais representativo, aproximando-se ainda mais de um SEP real.

Por fim, a aplicação do modelo PSR ao sistema elétrico de uma concessionária de energia elétrica ou mesmo a uma fração de sua rede com a devida integração aos sistemas de informação validaria de forma irrefutável a modelagem proposta e traria benefícios imensuráveis para a validação do modelo PSR, seu refinamento e aprimoramento. Além disso, os benefícios de melhoria da confiabilidade apresentados neste trabalho poderiam ser contabilizados por meio de cálculos reais dos índices de confiabilidade.

## 5.4 Publicações

Os resultados obtidos ao longo do desenvolvimento deste trabalho foi publicado em revista científica especializada, conforme apresentado na Tabela 5.1.

Tabela 5.1 – Artigo publicado em revista.

<b>Título</b>	<b>Publicação</b>	<b>DOI</b>
<i>Multiobjective Optimization Algorithm for Switch Placement in Radial Power Distribution Networks</i>	<i>IEEE Transactions on Power Delivery</i> (ISSN 0885-8977)	10.1109/TPWRD.2014.2317173

Outros resultados referentes a esta proposta foram recentemente submetidos para a revista *Control Engineering Practice*, conforme apresentado na Tabela 5.2.

Tabela 5.2 – Artigo submetido para publicação.

<b>Título</b>	<b>Publicação</b>	<b>Status</b>
<i>Hierarchical CPN Model Applied for Automatic and Secure Restoration of Power Distribution Networks</i>	<i>Control Engineering Practice</i> (ISSN 0967-0661)	<i>Under review</i>

Por fim, os resultados obtidos também foram apresentados e publicados em anais de congressos internacionais conforme apresentado na Tabela 5.3.

Tabela 5.3 – Artigos publicados e apresentados em congressos.

<b>Título</b>	<b>Evento</b>	<b>DOI</b>
<i>Switch Placement Algorithm for Reducing Customers Outage Impacts on Radial Distribution Networks</i>	<i>2012 IEEE Region 10 Conference</i> (Cebu, Philippines, 2012)	10.1109/TEN-CON.2012.6412213
<i>Power Distribution Network Planning Application Based on Multi-Objective Binary Particle Swarm Optimization Algorithm</i>	<i>Fourth International Conference on Swarm Intelligence</i> (Harbin, China 2013)	10.1007/978-3-642-38703-6_31

## Referências

- ABIRI-JAHROMI, A. et al. Optimized sectionalizing switch placement strategy in distribution systems. *Power Delivery, IEEE Transactions on*, v. 27, n. 1, p. 362–370, Jan. 2012. ISSN 0885-8977. Citado 8 vezes nas páginas 21, 31, 32, 35, 36, 39, 40 e 41.
- AHMADI, H.; MARTI, J. Distribution system optimization based on a linear power-flow formulation. *Power Delivery, IEEE Transactions on*, v. 30, n. 1, p. 25–33, Feb 2015. ISSN 0885-8977. Citado na página 21.
- ALLAN, R. et al. A reliability test system for educational purposes-basic distribution system data and results. *Power Systems, IEEE Transactions on*, v. 6, n. 2, p. 813–820, May 1991. ISSN 0885-8950. Citado 3 vezes nas páginas 31, 37 e 38.
- BEZERRA, J. et al. Multiobjective optimization algorithm for switch placement in radial power distribution networks. *Power Delivery, IEEE Transactions on*, v. 30, n. 2, p. 545–552, April 2015. ISSN 0885-8977. Citado 2 vezes nas páginas 38 e 41.
- BILLINTON, R.; JONNAVITHULA, S. Optimal switching device placement in radial distribution systems. *Power Delivery, IEEE Transactions on*, v. 11, n. 3, p. 1646–1651, Jul 1996. ISSN 0885-8977. Citado 3 vezes nas páginas 31, 35 e 36.
- BLUM, X. L. C. *Swarm intelligence introduction and applications*. Berlin London: Springer, 2008. ISBN 9783540740896. Citado na página 21.
- BRETAS, A.; PHADKE, A. Artificial neural networks in power system restoration. *Power Delivery, IEEE Transactions on*, v. 18, n. 4, p. 1181–1186, Oct 2003. ISSN 0885-8977. Citado 3 vezes nas páginas 16, 51 e 52.
- BROWN, R. et al. Assessing the reliability of distribution systems. *Computer Applications in Power, IEEE*, v. 14, n. 1, p. 44–49, Jan 2001. ISSN 0895-0156. Citado na página 48.
- CASSANDRAS, C. G.; LAFORTUNE, S. *Introduction to Discrete Event Systems*. Boston, MA: Springer Science+Business Media, LLC, 2008. ISBN 9780387333328 0387333320 9780387686127 0387686126. Disponível em: <<http://dx.doi.org/10.1007/978-0-387-68612-7>>. Citado na página 28.
- CHEN, C.-S. et al. Optimal placement of line switches for distribution automation systems using immune algorithm. *Power Systems, IEEE Transactions on*, v. 21, n. 3, p. 1209–1217, 2006. ISSN 0885-8950. Citado 3 vezes nas páginas 31, 35 e 36.
- CHEN, W.-K. *Graph Theory and Its Engineering Applications*. World Scientific Publishing, 1997. (Advanced Series in Electrical and Computer Engineering, v. 5). ISBN 9810218591,9789810218591. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=7E6BA660DB2C49587C9FC573EE3C0E15>>. Citado na página 28.
- COELLO, C. C.; LAMONT, G. B.; VELDHUIZEN, D. A. van. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. [S.l.]: Springer, 2007. ISBN 0387332545. Citado 2 vezes nas páginas 23 e 24.

DEB, K. Multi-objective evolutionary algorithms: Introducing bias among pareto-optimal solutions. In: GHOSH, A.; TSUTSUI, S. (Ed.). *Advances in Evolutionary Computing*. [S.l.]: Springer Berlin Heidelberg, 2003, (Natural Computing Series). p. 263–292. ISBN 978-3-540-43330-9. Citado na página 23.

DEB, K. *Multi-Objective Optimization Using Evolutionary Algorithms (Wiley Paperback)*. [S.l.]: Wiley, 2009. Citado 2 vezes nas páginas 23 e 25.

DIESTEL, R. *Graph Theory (Graduate Texts in Mathematics)*. [S.l.]: Springer, 2005. Hardcover. ISBN 3540261826. Citado na página 32.

FALAGHI, H.; HAGHIFAM, M.-R.; SINGH, C. Ant colony optimization-based method for placement of sectionalizing switches in distribution networks using a fuzzy multiobjective approach. *Power Delivery, IEEE Transactions on*, v. 24, n. 1, p. 268–276, Jan. 2009. ISSN 0885-8977. Citado na página 31.

(FIRM), A. *Network Protection & Automation Guide*. ALSTOM, 2002. ISBN 9782951858909. Disponível em: <<http://books.google.com.au/books?id=1WMbAAAACAAJ>>. Citado na página 48.

FOUNTAS, N.; HATZIARGYRIOU, N.; VALAVANIS, K. Hierarchical time-extended petri nets as a generic tool for power system restoration. *Power Systems, IEEE Transactions on*, v. 12, n. 2, p. 837–843, May 1997. ISSN 0885-8950. Citado na página 17.

GROSS, J. L.; YELLEN, J. *Graph Theory and Its Applications, Second Edition (Discrete Mathematics and Its Applications)*. [S.l.]: Chapman & Hall/CRC, 2005. ISBN 158488505X. Citado 2 vezes nas páginas 26 e 27.

GUO, W.; CHEN, G.; FENG, X. A new strategy of acceleration coefficients for particle swarm optimization. In: *Computer Supported Cooperative Work in Design, 2006. CSCWD '06. 10th International Conference on*. [S.l.: s.n.], 2006. p. 1–5. Citado na página 33.

HAJIAN-HOSEINABADI, H. Impacts of automated control systems on substation reliability. *Power Delivery, IEEE Transactions on*, v. 26, n. 3, p. 1681–1691, July 2011. ISSN 0885-8977. Citado na página 17.

HORVATH, D. et al. Local vs. global search strategies in evolutionary grid-based conformational sampling and docking. In: *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*. [S.l.: s.n.], 2009. p. 247–254. Citado na página 41.

IEEE. IEEE guide for electric power distribution reliability indices. *IEEE Std 1366-2012 (Revision of IEEE Std 1366-2003)*, p. 1–43, May 2012. Citado 3 vezes nas páginas 31, 36 e 47.

JENSEN, K.; KRISTENSEN, L. M. *Coloured Petri Nets - Modelling and Validation of Concurrent Systems*. [S.l.]: Springer, 2009. I-XI, 1-384 p. ISBN 978-3-642-00283-0. Citado 4 vezes nas páginas 28, 29, 58 e 59.

JUNG, K.-H.; KIM, H.; KO, Y. Network reconfiguration algorithm for automated distribution systems based on artificial intelligence approach. *Power Delivery, IEEE Transactions on*, v. 8, n. 4, p. 1933–1941, Oct 1993. ISSN 0885-8977. Citado na página 17.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: *Neural Networks, 1995. Proceedings., IEEE International Conference on*. [S.l.: s.n.], 1995. v. 4, p. 1942–1948 vol.4. Citado 3 vezes nas páginas 21, 32 e 35.

KENNEDY, J.; EBERHART, R. A discrete binary version of the particle swarm algorithm. In: *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*. [S.l.: s.n.], 1997. v. 5, p. 4104–4108 vol.5. ISSN 1062-922X. Citado 2 vezes nas páginas 21 e 22.

KERSTING, W. Radial distribution test feeders. In: *Power Engineering Society Winter Meeting, 2001. IEEE*. [S.l.: s.n.], 2001. v. 2, p. 908–912 vol.2. Citado na página 31.

KETABI, A.; RANJBAR, A.; FEUILLET, R. Analysis and control of temporary overvoltages for automated restoration planning. *Power Delivery, IEEE Transactions on*, v. 17, n. 4, p. 1121–1127, Oct 2002. ISSN 0885-8977. Citado na página 17.

KONDO, D. et al. A methodology for reclosers allocation in distribution networks. In: *Innovative Smart Grid Technologies Latin America (ISGT LA), 2013 IEEE PES Conference On*. [S.l.: s.n.], 2013. p. 1–8. Citado 4 vezes nas páginas 31, 32, 35 e 36.

MORADI, A.; FOTUHI-FIRUZABAD, M. Optimal switch placement in distribution systems using trinary particle swarm optimization algorithm. *Power Delivery, IEEE Transactions on*, v. 23, n. 1, p. 271–279, Jan. 2008. ISSN 0885-8977. Citado 4 vezes nas páginas 21, 31, 35 e 36.

MORI, H.; KOMATSU, Y. Power network decomposition with new ant colony optimization. In: *Probabilistic Methods Applied to Power Systems, 2006. PMAPS 2006. International Conference on*. [S.l.: s.n.], 2006. p. 1–6. Citado na página 21.

NETO, P. O. B. *Grafos: teoria, modelos, algoritmos*. 5a. ed. São Paulo SP: E. Blucher, 2011. Citado na página 27.

ONS. *ONS - Conheça o sistema*. 2014. <[http://www.ons.org.br/conheca\\_sistema/o\\_que\\_e\\_sin.aspx](http://www.ons.org.br/conheca_sistema/o_que_e_sin.aspx)>. Accessed Fev 4, 2014. Citado na página 18.

PRODIST Módulo 8, PRODIST. [S.l.]: ANEEL, Brasília, Brasil, 2014. Citado 2 vezes nas páginas 47 e 49.

REN, F. et al. Conceptual design of a multi-agent system for interconnected power systems restoration. *Power Systems, IEEE Transactions on*, v. 27, n. 2, p. 732–740, May 2012. ISSN 0885-8950. Citado 2 vezes nas páginas 16 e 51.

SALLAM, A. A.; MALIK, O. P. *Electric Distribution Systems (IEEE Press Series on Power Engineering)*. Wiley-IEEE Press, 2011. ISBN 9780470943854. Disponível em: <<http://www.amazon.com/Electric-Distribution-Systems-Series-Engineering-ebook/dp/B0050PIAZ2%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3DB0050PIAZ2>>. Citado na página 48.

SARMADI, S. N. et al. A sectionalizing method in power system restoration based on wams. *Smart Grid, IEEE Transactions on*, v. 2, n. 1, p. 190–197, March 2011. ISSN 1949-3053. Citado 3 vezes nas páginas 16, 51 e 52.

- SHI, Y.; EBERHART, R. Empirical study of particle swarm optimization. In: *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*. [S.l.: s.n.], 1999. v. 3, p. –1950 Vol. 3. Citado na página 35.
- SINGH, K. *Electricity Distribution Network Expansion Planning*. [S.l.]: Department of Engineering Science, University of Auckland, 2013. <<http://www.epoc.org.nz/papers/KaviORSNZPaper.pdf>>. Citado na página 48.
- ULLMAN, J. D. *Elements of ML programming - ML 97 edition*. [S.l.]: Prentice Hall, 1998. I-XIII, 1-383 p. ISBN 978-0-13-790387-0. Citado 2 vezes nas páginas 28 e 30.
- VU, P. et al. A novel weight-improved particle swarm optimization algorithm for optimal power flow and economic load dispatch problems. In: *Transmission and Distribution Conference and Exposition, 2010 IEEE PES*. [S.l.: s.n.], 2010. p. 1–7. Citado na página 21.
- WALPOLE, R. E. et al. *Probability & statistics for engineers and scientists*. 8th. ed. Upper Saddle River: Pearson Education, 2007. Citado na página 42.
- WANG, C. et al. Ptdf-based automatic restoration path selection. *Power Systems, IEEE Transactions on*, v. 25, n. 3, p. 1686–1695, Aug 2010. ISSN 0885-8950. Citado na página 17.
- WANG, C.; VITTAL, V.; SUN, K. Obdd-based sectionalizing strategies for parallel power system restoration. *Power Systems, IEEE Transactions on*, v. 26, n. 3, p. 1426–1433, Aug 2011. ISSN 0885-8950. Citado 3 vezes nas páginas 16, 51 e 52.
- YE, D.; ZHANG, M.; SUTANTO, D. A hybrid multiagent framework with q-learning for power grid systems restoration. *Power Systems, IEEE Transactions on*, v. 26, n. 4, p. 2434–2441, Nov 2011. ISSN 0885-8950. Citado 3 vezes nas páginas 16, 51 e 52.
- ZAVODA, F. Advanced distribution automation (ada) applications and power quality in smart grids. In: *Electricity Distribution (CICED), 2010 China International Conference on*. [S.l.: s.n.], 2010. p. 1–7. Citado na página 48.
- ZIDAN, A.; EL-SAADANY, E. A cooperative multiagent framework for self-healing mechanisms in distribution systems. *Smart Grid, IEEE Transactions on*, v. 3, n. 3, p. 1525–1539, Sept 2012. ISSN 1949-3053. Citado na página 17.

# Apêndices

## APÊNDICE A – Funções do modelo PSR

Este Apêndice contém o código fonte de todas as funções escritas em ML para utilização no modelo PSR. Comentários são inseridos no próprio código para facilitar o entendimento.

A nomenclatura atribuída ao nome de cada função segue as seguintes regras:

- Inicia com um verbo em letra minúscula, que sugere a ação que a função realiza (*get*, *set*, etc);
- As palavras que vêm a seguir possuem a primeira letra maiúscula;
- Não são usados espaços em branco.

### Código-fonte A.1 – Função `isFuncTrip`.

```
(*
Returns true if the given ProtMsg
is a trip (it is not a Start, it is not
a position change neither Unlocking
*)
fun isFuncTrip (pm : ProtMsg) =
if #3 pm <> "ST" andalso
#3 pm <> "52" andalso
#3 pm <> "UNLOCK"
then true
else false;
```

### Código-fonte A.2 – Função `filterTripMsgs`.

```
(*
Filter the trip messages from a given
ProtMSGs list
*)
fun filterTripMsgs (pml: ProtMSGs) =
List.filter (fn x => isFuncTrip(x)) pml;
```

### Código-fonte A.3 – Função `isSource`.

```
(*
Evaluates true if the given node has non-zero supply current (i)
and node status is Active in the given power system (ps).
```

*REMARK: Non-zero supply current means the node is a power source. The current function does not check if the source is available.*

```
*)
fun isSource (ps : PwrSYS, node : NAME) =
let
  val nl = List.filter (fn x => BRANCH_NODE.of_N(x)) ps
in
  List.exists (fn N (n, i, _, _) => n=node andalso i<>0) nl
end;
```

#### Código-fonte A.4 – Função getSources.

```
(*
Returns the name of the power sources in the
given PwrSysSTATUS in a list
*)
fun getSources (ps : PwrSYS) =
let
  val nl = List.filter (fn x => BRANCH_NODE.of_N(x)) ps
in
  List.mapPartial (fn N (n, _, _, _)=>
    if isSource(ps, n) then SOME n else NONE) nl
end;
```

#### Código-fonte A.5 – Função filterStartMsgs.

```
(*
Extracts the start messages from the
given ProtMSGs list (pml) returning a list
that is a subset of the original list
*)
fun filterStartMsgs (pml: ProtMSGs) =
List.filter (fn x => isFunc("ST", x)) pml;
```

#### Código-fonte A.6 – Função getTermNodes.

```
(*
Returns the node names connected at terminals T1 and T2
from the branches in the given list (l) matching the given
power system (ps).
*)
fun getTermNodes (ps : PwrSYS, l : NAMES) =
let
```

```

val b1 = List.filter (fn x => BRANCH_NODE.of_B(x)) ps
val b1 = List.filter (fn B (b, _, _, _) =>
    List.exists (fn x => x=b) l) b1
val l1 = List.map (fn B (_, n1, _, _) => n1) b1
val l2 = List.map (fn B (_, _, n2, _) => n2) b1
in
  l1^^l2
end;

```

## Código-fonte A.7 – Função getIED.

```

(*
Get the IED names in the given
ProtMSGs list
*)
fun getIED (pml:ProtMSGs) =
List.map (fn (_, ied, _) => ied) pml;

```

## Código-fonte A.8 – Função findDeepNode.

```

(*
Finds the deepest node from a list.
REMARK: The deepest node is considered the one which
has no spring nodes or the node which is the leaf of
the graph.
*)
fun findDeepNode (n : NAMES, s : NAMES) =
let
  val n_ = List.filter (fn x => not(
    let
      val t = rm x n
    in
      List.exists (fn y => x=y) t
    end)) n
in
  List.filter (fn x => not(List.exists (fn y => y=x) s)) n_
end;

```

## Código-fonte A.9 – Função setNsStatus.

```

(*
Set the status field of the NODE_STATUS to a new given value (st_)
in the given power system status (pss) returning a modified pss
*)

```

```

fun setNsStatus (pss : PwrSysSTATUS, nsl : NAMES, st_ : STATUS) =
let
  val bl = List.filter (fn x => BRANCH_NODE_STATUS.of_BS(x)) pss
  val nl = List.filter (fn x => BRANCH_NODE_STATUS.of_NS(x)) pss
  val nl = List.map (fn NS (n, st) =>
    if List.exists (fn x => x=n) nsl
    then NS (n, st_)
    else NS (n, st)
  ) nl
in
  bl^^nl
end;

```

## Código-fonte A.10 – Função getCurrentPosition.

```

(*
Returns the current position of the given branch (b)
in a given power system status (pss).
*)
fun getCurrentPosition (pss : PwrSysSTATUS, b : NAME) =
let
  val bl = List.filter (fn x => BRANCH_NODE_STATUS.of_BS(x)) pss
in
  List.hd (List.mapPartial (fn BS (n, p, _, _)=>
    if n=b then SOME p else NONE) bl)
end

```

## Código-fonte A.11 – Função getStandardPos.

```

(*
Returns the standard position from the given branch (b)
in a given power system (ps).
*)
fun getStandardPos (ps : PwrSYS, b : NAME) =
let
  val bl = List.filter (fn x => BRANCH_NODE.of_B(x)) ps
in
  List.hd (List.mapPartial (fn B (n, _, _, p)=>
    if n=b then SOME p else NONE) bl)
end

```

## Código-fonte A.12 – Função setBsStatus.

```

(*

```

*Set the BRANCH\_STATUS status to a new given value (s\_) in a given power system status (pss) returning a modified pss.*

```

*)
fun setBsStatus (pss : PwrSysSTATUS, bs1 : NAMES, s_ : STATUS) =
let
  val n1 = List.filter (fn x => BRANCH_NODE_STATUS.of_NS(x)) pss
  val b1 = List.filter (fn x => BRANCH_NODE_STATUS.of_BS(x)) pss
  val b1 = List.map (fn BS (n, p, i, s) =>
    if List.exists (fn x => x=n) bs1
    then BS (n, p, i, s_)
    else BS (n, p, i, s)
  ) b1
in
  b1^^n1
end;

```

#### Código-fonte A.13 – Função findPathsFromNodeToActive.

```

(*)
Finds all the elementary paths or simple paths from the given load point (node) to the sources (active nodes) in the given power system (ps). p represents the current path under analysis, v the visited nodes, s is the stack and all contains the results.
*)
fun findPathsFromNodeToActive
(ps : PwrSYS, node : NAMES, v : NAMES, p : NAMES, s : NAMES, all : NAMESS
if node=[]
then node2edge(ps, all, [])
else
  if isSource(ps, List.hd node)
  then findPathsFromNodeToActive (ps, (List.tl node), v, s, s,
    (rev ((List.hd node)::p))::all)
  else
let
  val v = (List.hd node)::v
  val p = (List.hd node)::p
  val n = List.concat (List.map (fn b=>getNodesFromEdge(ps, b))
    (getEdgesFromNode (ps, (List.hd node))))
  val n = rmall (List.hd node) n
  val un = List.filter (fn n_=> not(mem v n_)) n
  val node = un^(List.tl node)
  val s = if List.length un > 1 then p else s

```

```

in
  findPathsFromNodeToActive (ps, node, v, p, s, all)
end

```

## Código-fonte A.14 – Função findCurrPathsFromNodeToActive.

```

(*)
Finds the available paths (which all switches are closed) in the given
power system (ps) under the current power system status (pss)
from the given load point to the sources (active nodes).
REMARK: The current function is based on
the findPathsFromNodeToActive.
*)
fun findCurrPathsFromNodeToActive
(ps : PwrSYS, pss : PwrSysSTATUS, node : NAMES, curr : NAMESS) =
let
  val curr = findPathsFromNodeToActive (ps, node, [], [], [], [])
in
  List.filter (fn p=> not(isPathOpen(pss, p))) curr
end;

```

## Código-fonte A.15 – Função getNodeFromStatus.

```

(*)
Returns a list of node names that match a given status (st_)
in a given power system status (pss).
*)
fun getNodeFromStatus (pss : PwrSysSTATUS, st_ : STATUS) =
let
  val nl = List.filter (fn x => EDGE_NODE_STATUS.of_NS(x)) pss
in
  List.mapPartial (fn NS (n, st) =>
    if st=st_
    then SOME n else NONE) nl
end;

```

## Código-fonte A.16 – Função getEdgeFromPosition.

```

(*)
Returns the list of edges which have a given position (p_) in
the given power system status (pss).
*)
fun getEdgeFromPosition (pss : PwrSysSTATUS, p_ : POSITION) =
let

```

```

val pss = List.filter (fn x => EDGE_NODE_STATUS.of_ES(x)) pss
val pss = List.filter (fn ES (_, p, _, _) => p=p_) pss
in
List.map (fn ES (n, _, _, _) => n) pss
end;

```

## Código-fonte A.17 – Função getEdgeFromStatus.

```

(*
Returns the list of EDGE_STATUS names that match a given
status (s_) in the given power system status (pss).
*)
fun getEdgeFromStatus (pss : PwrSysSTATUS, s_ : STATUS) =
let
val pss = List.filter (fn x => EDGE_NODE_STATUS.of_ES(x)) pss
in
List.mapPartial (fn ES (n, _, _, s) =>
if s=s_
then SOME n else NONE) pss
end;

```

## Código-fonte A.18 – Função evalIsolators.

```

(*
Chooses the devices capable (which are InService or Closed) to
isolate a node from any active node.
*)
fun evalIsolators(bl : NAMESS, isl : NAMES, ol : NAMES, l : NAMES) =
if bl = []
then remdupl l
else
let
val hbl = List.hd bl
val bl = rm hbl bl
val hhbl = List.hd hbl
val hbl = rm hhbl hbl
val isOpen = List.exists (fn x => x=hhbl) ol
val isInServ = List.exists (fn x => x=hhbl) isl
in
if isOpen=true
then if isInServ=true
then evalIsolators (bl, isl, ol, hhbl::l)
else evalIsolators (bl, isl, ol, l)

```

```

    else if isInServ=true
        then evalIsolators (bl, isl, ol, hhbl::l)
        else evalIsolators (hbl::bl, isl, ol, l)
end;

```

## Código-fonte A.19 – Função setIsolationEdge.

```

(*
Sets the position (to Open_) and status (to OutOfServ_) for
the isolating edges which names match the given list (ibl).
*)
fun setIsolationEdge (pss : PwrSysSTATUS, ibl : NAMES) =
let
    val bl = List.filter (fn x => EDGE_NODE_STATUS.of_ES(x)) pss
    val nl = List.filter (fn x => EDGE_NODE_STATUS.of_NS(x)) pss
    val bl = List.map (fn ES (n, p, i, s) =>
        if List.exists (fn y => y=n) ibl
        then ES (n, Open_, i, OutOfServ_)
        else ES (n, p, i, s)
    ) bl
in
    bl^^nl
end;

```

## Código-fonte A.20 – Função getEdgesFromNode.

```

(*
Returns the list of edges connected to a given node (node)
in a given power system (ps).
*)
fun getEdgesFromNode (ps : PwrSYS, node : NAME) =
let
    val bl = List.filter (fn x => EDGE_NODE.of_E(x)) ps
in
    if isSource(ps, node)
    then []
    else
        List.mapPartial (fn E (n, n1, n2, _) =>
            if n1=node orelse n2=node then SOME n else NONE) bl
end;

```

## Código-fonte A.21 – Função createRepList.

```

(*

```

```
Creates a list (l) which contains n repeated lists (l_).
*)
fun createRepList (n : INT, l_ : NAMES, l : NAMES) =
if n=0
then l
else
let
  val n = n - 1
in
  createRepList(n, l_, l_^^1)
end;
```

## APÊNDICE B – *Scripts .m*

Este Apêndice contém o código fonte dos *scripts* que implementam o algoritmo MOSP em sua versão final. Comentários inseridos no próprio código auxiliam na compreensão dos mesmos.

### Código-fonte B.1 – *initSols.m*

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %% Initial Solutions
3  %
4  % Induces a set of initial solutions for M agents
5  % to the given test system (ts) regarding the
6  % preferred switches (prefsw)
7  %%
8  function sol = initSols(ts,M)
9  %% error handling
10 if nargin ~= 2
11     error('Myapp:argChk','Wrong number of input parameters.')
12 end
13 %% loads the file containing test system definitions (.mat)
14 testSystem = load(ts);
15 QNA = testSystem.QNA;
16 QNF = testSystem.QNF;
17 prefsw = testSystem.prefsw;
18 %%
19 % sum of NC and NO switches
20 QNFNA = QNA+QNF;
21 % solutions set size
22 sol = zeros(M,QNA+QNF);
23 % number of preferred switches
24 QPREF = sum(prefsw);
25 % number of switches for a single solution
26 q = random('Poisson',QPREF/3,1,M);
27 % change zeros by ones to avoid solutions without switches
28 q(q==0) = 1;
29 % creates M solutions
30 for i=1:M
31     % random solution guess
32     rsol = random('Binomial',1,q(i)/QNFNA,1,QNFNA);
33     % selects the preferred switches

```

```

34     sol(i,:) = and(rsol,prefsw);
35     % avoid solutions without switches
36     while ~any(sol(i,:),2)
37         sol(i,:) = random('Binomial',1,q(i)/QPREF,1,QNFNA);
38     end
39 end
40 %
41 end

```

## Código-fonte B.2 – mosp.m

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %% MULTI-OBJECTIVE SWITCH PLACEMENT algorithm
3  %
4  % ts => applied Test System
5  % DG => DG scheme
6  %
7  function placement = mosp(ts)
8  %% error handling
9  if nargin == 0 || nargin < 1
10     error('Myapp:argChk','Invalid number of input arguments.')
11 end
12 %% loads test system file (.mat)
13 testSystem = load(ts);
14 Q = testSystem.Q;
15 E = testSystem.E;
16 QNA = testSystem.QNA;
17 QNF = testSystem.QNF;
18 %% initial values
19 QNFNA = QNF + QNA;
20 % swarm size
21 M = E;
22 % number of supplied customers
23 CS = sum(Q);
24 % fator de aprendizagem
25 % c(2,1) -> c1 do segundo estagio (social)
26 c = [3 2; 2 2];
27 % number of iterations
28 Nitr = [E 3*E];
29 % weights
30 w = [1 0 ; .5 .5];
31 % switch placement
32 placement = zeros(sum(Nitr),QNFNA+2);

```

```
33 % CNS when no automatic swith is available
34 x = zeros(1,QNFNA);
35 CNS_0 = cns(ts,x);
36 %% BINARY DISCRETE PARTICLE SWARM OPTIMIZATION
37 %
38 for h=1:2 % estagios
39     %% initial solution, avoiding trivial solutions
40     % induced starting solution
41     x = initSols(ts,M);
42     % random starting solution
43     %x = round(rand(M,QNFNA));
44     % starting velocity
45     v = rand(M,QNFNA);
46     % starting pbest
47     pbest = initSols(ts,M);
48     % data to the first stage running
49     if h == 1
50         % starting gbest
51         gbest = x(1,:);
52         cns_gbest = cns(ts,gbest);
53     end
54     %
55     for n=1:Nitr(h)
56         % rand1 and rand2
57         r1 = rand();
58         r2 = rand();
59         %
60         for m=1:M % number of agents
61             if w(h,2) == 0
62                 qswx = 0;
63                 qsw_pbest = 0;
64                 qsw_gbest = 0;
65             else
66                 % number of switches on the current solution (x)
67                 qswx = w(h,2) * nnz(x(m,:)) / (QNFNA);
68                 % number of switches on the pbest solution
69                 qsw_pbest = w(h,2) * nnz(pbest(m,:)) / (QNFNA);
70                 % number of switches on the gbest solution
71                 qsw_gbest = w(h,2) * nnz(gbest) / (QNFNA);
72             end
73             % current solution fitness
74             fitness_x = w(h,1) * cns(ts,x(m,:)) / CNS_0 + qswx;
```

```

75         % pbest fitness
76         fitness_pbest = w(h,1) * cns(ts,pbest(m,:)) / CNS_0 + qsw_pbest;
77         % compare with pbest
78         if (fitness_pbest > fitness_x)
79             pbest(m,:) = x(m,:);
80         end
81         % compare with gbest
82         fitness_gbest = w(h,1) * cns_gbest / CNS_0 + qsw_gbest;
83         if (fitness_gbest > fitness_x)
84             gbest = x(m,:);
85             cns_gbest = cns(ts,gbest);
86         end
87         %% updates x and V
88         % update x and avoids trivial solutions
89         x(m,:) = lt(rand(1,QNFNA),sigmoid(v(m,:)));
90         while ~any(x(m,:),2)
91             x(m,:) = lt(rand(1,QNFNA),sigmoid(v(m,:)));
92         end
93         % evaluates deltaV
94         dV(1,:) = pbest(m,:) - x(m,:);
95         dV(2,:) = gbest - x(m,:);
96         % update V
97         v(m,:) = v(m,:) + c(h,1)*r1*dV(1,:) + c(h,2)*r2*dV(2,:);
98     end
99     % store gbest
100    if h == 1
101        placement(n,:) = [gbest round(cns_gbest*CS) nnz(gbest)];
102    else
103        placement(n+Nitr(1),:) = [gbest round(cns_gbest*CS) nnz(gbest)];
104    end
105    end
106 end

```

## Código-fonte B.3 – cns.m

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %
3  %% Customers NOT Supplied function
4  %
5  % Evaluates the number of CNS affected by faults
6  % using the Test System 'ts' the set of switches 'x'
7  %
8  % ts => Test System file applied

```

```

9  % x => Set of switches installed
10 function cns_ = cns(ts,x)
11 %% error handling
12 if nargin < 2
13     error('Myapp:argChk','Wrong number of input parameters.')
14 end
15 %% loads the file containing test system definitions (.mat)
16 testSystem = load(ts);
17 I = testSystem.I; % I = sparse(S,D,1,V,V);
18 Q = testSystem.Q;
19 T = testSystem.T;
20 SW = testSystem.SW;
21 %LP = testSystem.LP;
22 SOURCES = testSystem.SOURCES;
23 state0 = testSystem.state0;
24 %% valores iniciais
25 %x = [1,1,0,1,1,1,1,1];
26 cns_ = 0;
27 % total de clientes atendidos
28 CS = sum(Q);
29 %% updates the SW and T matrices with the automatic switches in x
30 [SW T] = updateswt(SW,T,x);
31 clear T;
32 %% set of graph edges
33 b = find(state0);
34 [i j] = ind2sub(size(state0),b);
35 % distancia dos vertices a fonte
36 DIST = graphshortestpath(I,SOURCES); % graphshortestpath is applicable to s
37 shortcircuit = zeros(1,2); %#ok<NASGU>
38 %% evaluates the CNS for each contingency
39 for k=1:max(size(b))
40     % nodes on each the short-circuit is simulated
41     shortcircuit = [i(k) j(k)];
42     % source side node connected to the short circuit
43     [dsc_near,vsc_near] = min(DIST(shortcircuit)); %#ok<NASGU>
44     % far source node connected to the short circuit
45     [dsc_far,vsc_far] = max(DIST(shortcircuit));
46     % path from source to source side node
47     [dssc,path1] = graphshortestpath(I,SOURCES,shortcircuit(vsc_far));
48     p = length(path1);
49     % finds the closest switch to the short circuit(isolation I)
50     fuse = 0;

```

```
51     turn = 0;
52     for m=p:-1:2
53         s_ = path1(m-1);
54         d_ = path1(m);
55         type = SW(s_,d_);
56         if type ~= 1
57             %
58             s1 = s_;
59             d1 = d_;
60             I(s1,d1) = 0;
61             if type == 3
62                 fuse = 1;
63             end
64             if type == -1
65                 turn = 1;
66             end
67             break;
68         else
69             continue;
70         end
71     end
72     %
73     Iu = direc2undirec(I);
74     % search paths to reach the source(restoration)
75     if ~fuse
76         sw_no = find(SW==-1);
77         if ~turn
78             %
79             I(sw_no) = 1; %#ok<FNDSB> % more than one switch may be closed
80             Iu = direc2undirec(I);
81         end
82         [dssc,path2] = graphshortestpath(Iu,SOURCES,shortcircuit(vsc_far))
83         p = length(path2);
84         for m=p:-1:2
85             s_ = path2(m);
86             d_ = path2(m-1);
87             type = SW(s_,d_);
88             if type ~= 1
89                 %
90                 s2 = s_;
91                 d2 = d_;
92                 Iu(s2,d2) = 0;
```

```
93         Iu(d2,s2) = 0;
94         break;
95     else
96         continue;
97     end
98 end
99 end
100 % supplied vertices
101 sv = graphtraverse(Iu,SOURCES);
102 cns_inc = (CS - sum(Q(sv)))/CS;
103 cns_ = cns_ + cns_inc;
104 %cns_ = cns_inc*CS;
105 %% restart topology
106 I = state0;
107 end
108 %%
109 cns_ = cns_ * inv(size(b,1));
110 %
```