Filipe de Oliveira Saraiva

Aplicação de Sistemas Multiagentes para Gerenciamento de Sistemas de Distribuição tipo Smart Grids

São Carlos – São Paulo 23 de março de 2012

Filipe de Oliveira Saraiva

Aplicação de Sistemas Multiagentes para Gerenciamento de Sistemas de Distribuição tipo Smart Grids

Dissertação apresentada à Escola de Engenharia de São Carlos - Universidade de São Paulo como parte dos requisitos exigidos para a obtenção do título de Mestre em Ciências, Programa de Engenharia Elétrica.

Área de Concentração: Sistemas Elétricos de Potência.

Orientador:

Prof. Dr. Eduardo Nobuhiro Asada

MESTRADO EM ENGENHARIA ELÉTRICA DEPARTAMENTO DE ENGENHARIA ELÉTRICA ESCOLA DE ENGENHARIA DE SÃO CARLOS UNIVERSIDADE DE SÃO PAULO

São Carlos - São Paulo

23 de março de 2012

Trata-se da versão corrigida da dissertação. A versão original se encontra disponível na EESC/USP que aloja o Programa de Pós-Graduação de Engenharia Elétrica

Ficha catalográfica preparada pela Seção de Tratamento da Informação do Serviço de Biblioteca - EESC/USP

Saraiva, Filipe de Oliveira

S243a

Aplicação de sistemas multiagentes para gerenciamento de sistemas de distribuição tipo *smart grids*. / Filipe de Oliveira Saraiva ; orientador Eduardo Nobuhiro Asada. São Carlos, 2012.

Dissertação (Mestrado - Programa de Pós-Graduação em Engenharia Elétrica e Área de Concentração em Sistemas Elétricos de Potência)-- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2012.

 Gerenciamento de sistemas elétricos de distribuição.
 Smart grids .
 Sistemas multiagentes.
 Sistemas distribuídos.
 Título.

FOLHA DE JULGAMENTO

Candidato: Bacharel FILIPE DE OLIVEIRA SARAIVA

Título da dissertação: "Aplicação de sistemas multiagentes para gerenciamento de sistemas de distribuição tipo *Smart Grids*".

Data da defesa: 23/03/2012

Comissão Julgadora:

Resultado:

Prof. Dr. Eduardo Nobuhiro Asada (Orientador)

APROVADO

(Escola de Engenharia de São Carlos/EESC)

Prof. Dr. Antonio Padilha Feltrin

APROVADO

(Universidade Estadual Paulista Júlio de Mesquita Filho/UNESP-campus de Ilha Solteira)

Prof. Dr. **João Bosco Augusto London Júnior** (Escola de Engenharia de São Carlos/EESC)

Aprovado

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica: Prof. Titular **Denis Vinicius Coury**

A minha mãe e meu pai, dona Dêca e seu Saraiva, a meu irmão e minha irmã, Diego e Livia; por todo carinho, dedicação, compreensão, amor. Que o esforço investido neste trabalho se traduza em orgulho e alegria para vocês.

Agradecimentos

Este trabalho está carregado de sentimentos por pessoas queridas. Enquanto escrevia esta dissertação, ou ainda em meus momentos de reflexão para prepará-la, era-me impossível não recordar delas. Enumero-as então, dedicando esta obra:

- Ao meu orientador, Prof. Dr. Eduardo Asada, por todos os conselhos dados e pelas provocações que partilhamos na busca de criarmos algo original, instigante e desafiador.
- A Aracele Torres, por todo o carinho, força e aprendizados que trocamos em nossa cumplicidade.
- Aos amigos piauienses em São Carlos, pelo qual o simples fato de partilharmos o mesmo sotaque já traz alento diante da grande saudade por estar longe da terra. Muito obrigado a Aldir Silva, Daniel Mário, Hermes Renato e Valdemiro Júnior.
- A meus e minhas camaradas de laboratórios, pelos momentos de aprendizado partilhados.
 Em especial, Felipe Mattos, Eduardo Wesley, Alex Vilarindo, Wellington.
- Meus sinceros agradecimentos a todos os membros das diversas comunidades de software livre que produziram as tecnologias livres que alicerça este trabalho. São muitos os envolvidos, mas cito as comunidades: KDE, OpenJDK, Jade, Python, GNU, Linux, Git, Mageia, Apache, Eclipse, Mozilla, Google, Scilab, Libre Office, Gimp, Inkscape, LaTeX, Beamer e ABNTex.
- À CAPES, financiadora deste projeto, que me dá suporte através da bolsa de estudos do programa PROEX.
- E finalmente, meus agradecimentos aos brasileiros, financiadores desta grande instituição que é o ensino superior do país. A Universidade tem o dever de assumir seu papel de agente transformador desta sociedade, através de ações de pesquisa e desenvolvimento de caráter socialmente referenciável.

"Uma maneira de definirmos nossa identidade, talvez a principal, seja em contraposição ao
"outro", em primeiro lugar aos vizinhos. Essa forma de definição é válida tanto para
disciplinas quanto para nações. Também elas têm seus "campos", suas culturas, suas tribos e
territórios. Apesar disso, a inovação intelectual é muitas vezes o resultado da burla da polícia
de fronteira e da invasão do território dos outros, ou, para variar a metáfora, de tomar
empréstimos dos vizinhos em vez de mantê-los a distância"
- Peter Burke
"Dê adeus aos futuros usados. Aqui está um novo mundo inteiro,
intenso como um choque elétrico."

- Bruce Sterling

Resumo

SARAIVA, F. de O. (2012). Aplicação de Sistemas Multiagentes para Gerenciamento de Sistemas de Distribuição tipo *Smart Grids*. São Carlos, 2012. 115 p. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo.

Os smart grids são tidos como a nova geração dos sistemas elétricos de potência, combinando avanços em computação, sistemas distribuídos e inteligência artificial para prover maiores funcionalidades sobre acompanhamento em tempo real da demanda e do consumo de energia elétrica, gerenciamento em larga escala de geradores distribuídos, entre outras, a partir de um sistema de controle distribuído sobre a rede elétrica. Esta abordagem alteraria fundamentalmente a maneira como se dá o planejamento e a operação de sistemas de distribuição, e há grandes possibilidades de pesquisa e desenvolvimento possibilitada pela busca de implementação destas funcionalidades. Com esse cenário em vista, o presente trabalho utiliza uma abordagem a partir do uso de sistemas multiagentes para estudar o gerenciamento de sistemas de distribuição, do ponto de vista da reconfiguração da topologia da rede, simulando as características de um smart grid. Nesta dissertação, foi desenvolvido um sistema multiagente para simulação computacional de um sistema de distribuição elétrico do tipo smart grid, buscando executar a reconfiguração topológica do sistema a partir de dados de carga capturados de forma distribuída pelos agentes dispersos na rede elétrica. Espera-se que o desenrolar da pesquisa conduza à vários estudos sobre algoritmos e técnicas que melhor implementem tais funcionalidades a serem transpostas para um ambiente de produção.

Palavras-chave: Gerenciamento de Sistemas Elétricos de Distribuição, *Smart Grids*, Sistemas Multiagentes, Sistemas Distribuídos

Abstract

SARAIVA, F. de O. (2012). Application of Multiagent Systems for Management of Distribution Systems like Smart Grids. São Carlos, 2012. 115 p. Dissertação (Master of Science) - Escola de Engenharia de São Carlos, Universidade de São Paulo.

Smart grids are taken as the new generation of electric power systems, combining advances in computing, distributed systems and artificial intelligence to provide more features on real-time monitoring of demand and consumption of electricity, managing large-scale distributed generators, among others, from a distributed control system on the grid. This approach fundamentally alter the way how is the planning and operation of distribution systems, and there are great possibilities for research and development offered in the quest to implement these features. With that environment, this text uses an approach through the use of multi-agent systems to study the management of the distribution system, from the reconfiguration of grid topology, simulating the characteristics of a smart grid. In this text, was developed a multiagent system to computational simulation of a distribution system like smart grid to topological reconfiguration, from datas collected for agents in electrical grid. It is expected that the conduct of research leads to several studies about better algorithms and techniques that would implement such functionality in a production environment.

Keywords: Management of Electrical Distribution Systems, Smart Grids, Multiagent Systems, Distributed Systems

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

1	Intr	odução	p. 23
	1.1	As Redes Inteligentes e a Mudança de Paradigma dos Sistemas Elétricos de Potência	p. 23
	1.2	Objetivos	p. 25
	1.3	Metodologia	p. 26
	1.4	Estrutura do Trabalho	p. 27
2	Sma	rt Grids	p. 28
	2.1	A Nova Geração dos Sistemas Elétricos de Potência	p. 28
	2.2	Países com Projetos para Smart Grids	p. 33
	2.3	A Simulação Computacional de Smart Grids	p. 36
3	Gere Red	enciamento dos Sistemas Elétricos de Distribuição via Reconfiguração da	p. 42
	3.1	Conceitos sobre Reconfiguração de Sistemas Elétricos de Distribuição	p. 42
	3.2	Aplicações de Reconfiguração de Sistemas Elétricos de Distribuição	p. 44

	3.3	Smart Grids e Reconfiguração de Sistemas Elétricos	p. 45
4	Siste	emas Multiagentes	p. 49
	4.1	Inteligência Artificial e Agentes	p. 49
	4.2	Inteligência Artificial Distribuída e Sistemas Multiagentes	p. 51
	4.3	Smart Grids e Sistemas Multiagentes	p. 55
5	Sim	ulação de Sistemas de Distribuição através de Sistemas Multiagentes	p. 58
	5.1	Proposta de Simulação de Sistemas de Distribuição através de Sistemas Multiagentes	p. 58
	5.2	Algoritmo de Dijkstra Distribuído através de Sistemas Multiagentes	p. 61
	5.3	Modelando Controle Distribuído em Sistemas de Distri-buição via Sistemas Multiagentes	p. 68
	5.4	Reconfiguração de Sistemas de Distribuição tipo <i>Smart Grids</i> através de Sistemas Multiagentes	p. 71
		5.4.1 Modelagem do Sistema Multiagente	p. 72
		5.4.2 Implementação do Sistema Multiagente	p. 85
		5.4.3 Simulações Computacionais	p. 86
6	Con	clusões e Trabalhos Futuros	p. 107
	6.1	Conclusões	p. 107
	6.2	Trabalhos Futuros	p. 108
Re	eferên	icias Bibliográficas	p. 110

Lista de Figuras

2.1	Smart Grids em camadas	p. 29
2.2	Sistema de Distribuição com Geração Centralizada	p. 32
2.3	Sistema de Distribuição com Geração Distribuída	p. 32
2.4	Modelos de sistemas distribuídos - arquitetura cliente-servidor	p. 38
2.5	Modelos de sistemas distribuídos - arquitetura P2P (par-a-par)	p. 38
3.1	Configuração topológica de um sistema de distribuição	p. 43
4.1	Componentes de interação entre um agente e o ambiente	p. 50
4.2	Agentes em um sistema multiagente e a representação de interações	p. 53
4.3	Agentes dispersos em um sistema elétrico de distribuição	p. 57
5.1	Iterações do algoritmo de Dijkstra. O vértice marcado com dois círculos será aquele que terá a vizinhança avaliada naquela iteração; os marcados com um retângulo serão aqueles que já tiveram sua vizinhança avaliada	p. 63
5.2	(a) Resultado da aplicação do algoritmo de Dijkstra para $v_i = v1$ e $v_j = v3$. (b) Caminho mínimo entre $v1$ e $v3$	p. 66
5.3	Diagrama de classe dos principais agentes do sistema	p. 72
5.4	Diagrama de classe para o Electrical Agent	p. 73
5.5	À esquerda, exemplo de sistema elétrico completo - grafo completo; a direita, subgrafo que forma a coalizão dos agentes nas cargas 3, 4, 5 e 6	p. 74
5.6	À esquerda, exemplo de sistema elétrico completo - grafo completo; a direita, o subgrafo que forma o caminho elétrico para coalizão da Figura 5.5	p. 75

5.7	Diagrama de classe para o Feeder Agent	p. 75
5.8	Fluxograma de execução dos agentes feeder	p. 77
5.9	Diagrama de classe para o Loader Agent	p. 78
5.10	Fluxograma de execução dos agentes <i>loader</i>	p. 81
5.11	Diagrama de classe para o Switcher Agent	p. 82
5.12	Fluxograma de execução dos agentes switcher	p. 82
5.13	Diagrama de sequência com a ordem de trocas de mensagens entre os agentes	p. 83
5.14	Fluxograma de execução dos agentes	p. 84
5.15	Sistema de testes - 10 barras	p. 87
5.16	Sistema de testes - 11 barras e 2 chaves	p. 88
5.17	Sistema de testes - 11 barras e 3 chaves	p. 91
5.18	Sistema de testes - 16 barras e 6 chaves	p. 93

Lista de Tabelas

2.1	Características próximas entre sistemas distribuídos e smart grids	 	 p. 39
3.1	Progressão da quantidade de estados a partir do número de chaves	 •	 p. 45
5.1	Instâncias de teste do Algoritmo de Dijkstra distribuído	 •	 p. 67
5.2	Protocolos das Mensagens	 	 p. 74
5.3	Sistema de 10 barras	 •	 p. 87
5.4	Resultado final para o sistema de 10 barras	 	 p. 88
5.5	Sistema de 11 barras e 2 chaves	 •	 p. 89
5.6	Resultado final para o sistema de 11 barras e 2 chaves	 	 p. 90
5.7	Sistema de 11 barras e 3 chaves	 •	 p. 91
5.8	Resultado final para o sistema de 11 barras e 3 chaves	 •	 p. 92
5.9	Sistema de 16 barras e 6 chaves	 •	 p. 94
5.10	Limite máximo de corrente nos trechos	 •	 p. 95
5.11	Características dos Momentos	 •	 p. 95
5.12	Estado Inicial	 •	 p. 96
5.13	Agentes 12, 13 e 14 com 20% a mais de energia	 •	 p. 97
5.14	Agentes 12, 13 e 14 com 50% a mais de energia	 •	 p. 97
5.15	Agentes 7, 8, 9, 10, 12, 13 e 14 com 20% a mais de energia	 	 p. 98
5.16	Agentes 7, 8, 9, 10, 12, 13 e 14 com 50% a mais de energia	 	 p. 98
5.17	Agentes 7, 8, 9 e 10 com 20% a mais de energia	 	 p. 99

5.18	Agentes 7, 8, 9 e 10 com 50% a mais de energia p. 99
5.19	Agentes 3, 4 e 5 com 50% a mais de energia p. 100
5.20	Agentes 3, 4, 5, 7, 8, 9 e 10 com 50% a mais de energia p. 100
5.21	Agentes 3, 4, 5, 7, 8, 9, 10, 11, 12, 13 e 14 com 50% mais de energia p. 101
5.22	Agentes 3, 4, 5, 7, 8, 9, 10, 11, 12, 13 e 14 com 20% mais de energia p. 101
5.23	Agentes 3, 4, 5, 11, 12, 13 e 14 com 50% mais de energia; 7, 8, 9 e 10 com 20% mais de energia
5.24	Chaves fechadas por Momentos
5.25	Tempos de execução por momento

Lista de Algoritmos

5.1	Algoritmo de Dijkstra	p. 62
5.2	Ações do Agente Ativo	p. 65
5.3	Algoritmo de Dijkstra no Agente Reativo	p. 66
5.4	Algoritmo do feederAgent	p. 69
5.5	Algoritmo dos loaderAgents	p. 70
5.6	Busca em profundidade - encontra coalizão	p. 78
5.7	Algoritmo de Dijkstra com múltiplos pontos finais	p. 79

1 Introdução

1.1 As Redes Inteligentes e a Mudança de Paradigma dos Sistemas Elétricos de Potência

Os sistemas elétricos de potência, em especial os sistemas de distribuição, estão passando por uma mudança radical em seu conceito e projeto. A tentativa de redução do uso de fontes de energia muito poluentes, a busca por uma maior eficiência energética, a necessidade de utilização de geração distribuída e a desregulação do mercado impulsionam grupos de pesquisa para estudos que visam adicionar uma "camada de informação" à rede de distribuição.

Estes cientistas e pesquisadores almejam que esta camada de informação possa prover à rede de distribuição funcionalidades que auxiliem no gerenciamento do sistema em si, tornando- o mais confiável e transparente para os operadores, além de permitir que o próprio sistema execute tarefas tais como reconfiguração, acompanhamento pormenorizado da demanda e do consumo, redução das perdas, entre outras. Seriam estes os *smart grids* - as "redes inteligentes" dos sistemas elétricos de potência.

Este tema tornou-se rotineiro na agenda de governos e organizações com interesse na questão energética. Por exemplo, o painel sobre mudanças climáticas da ONU realizado em 2010 em Copenhague, contou com um evento satélite direcionado a esta tecnologia (DUAN; DECONINCK, 2010). A Comissão de Energia da União Europeia (*European Commission Energy*), em seu relatório sobre o planejamento energético para 2020, aponta que investirá maciçamente em uma nova iniciativa onde (ENERGY, 2011):

(...)smart grids to link the whole electricity grid system, from the offshore wind farms in the North Sea, solar plants in the South and existing hydro-electric dams, to individual households, while making power networks more intelligent, efficient

and reliable¹

Apesar de haver esse projeto para médio prazo na União Europeia, já existem iniciativas que utilizam algumas das funcionalidades dos *smart grids* hoje. A cidade portuguesa de Évora tornou-se um laboratório para o projeto de redes inteligentes da EDP - Energias de Portugal, empresa do mercado energético daquele país. Os habitantes dessa cidade contam com medidores inteligentes que permitem monitoramento e acompanhamento mais eficiente do consumo (inclusive através da internet), e a empresa consegue detectar e resolver anomalias da rede com maior rapidez. Este acompanhamento permitirá à concessionária personalizar um plano de consumo para aqueles clientes que se encontram em um perfil específico. Além disso, há um plano de incentivo para a microprodução de energia em casas residenciais, através de painéis fotovoltaicos ou pequenas turbinas eólicas ligadas diretamente ao sistema de distribuição. Neste cenário, os outrora consumidores de energia passarão também a serem fornecedores para a concessionária footnoteA EDP mantém um site sobre o projeto de Évora, com várias informações, etapas já realizadas e próximos passos. Disponível em http://www.inovcity.pt/ (acessado dia 07/11/2011).

A adequação destas funcionalidades pressupõe uma radical modificação na estrutura atual dos sistemas de distribuição. Além disso, outro impacto significativo que esta tecnologia deve causar está na forma como engenheiros e cientistas da área trabalham suas pesquisas em tópicos como operação e expansão dos sistemas de distribuição. Dado as profundas mudanças que os *smart grids* trarão tanto para a topologia quanto para a operação das redes, os atuais métodos computacionais deverão ser atualizados para este novo paradigma, outros métodos serão criados e novas tecnologias para simulação desses sistemas serão avaliadas.

Uma das tecnologias candidatas a integrar essa nova fase de pesquisa em sistemas elétricos de potência são os sistemas multiagentes (WEISS, 1999; WOOLDRIDGE, 2009; SHOHAM; LEYTON-BROWN, 2009). Estes sistemas são compostos por peças de software e/ou hardware distribuídos em um dado ambiente, onde os mesmos conseguem comunicar-se entre si através de um protocolo específico para troca de mensagens e, a partir de então, interagir para alcançar um dado comportamento do grupo ou objetivo.

A característica distribuída somada à possibilidade de utilização de técnicas de inteligência artificial (RUSSEL; NORVIG, 2004) nos sistemas multiagentes os tornam uma interessante ferramenta técnica para ser utilizada nos trabalhos de simulação computacional nesse cenário

¹O relatório completo é disponibilizado pela comissão, e pode ser encontrado em http://ec.europa.eu/energy/strategies/2010/2020_en.htm (acessado dia 14/02/2011)

1.2 Objetivos 25

da nova geração dos sistemas elétricos de potência.

1.2 Objetivos

A aplicação de sistemas multiagentes para simulação computacional de *smart grids* é motivada pela busca da validação de uma tecnologia que permitirá a implementação e teste de algoritmos em um ambiente que emula as características dos novos sistemas elétricos de potência.

Com este intuito, o presente trabalho focará no gerenciamento de sistemas de distribuição através da reconfiguração da topologia da rede (BARAN; WU, 1989), para verificar a possibilidade de agentes distribuídos poderem prover informações para execução deste tipo de funcionalidade. A reconfiguração topológica de sistemas elétricos é uma ferramenta indispensável para várias funcionalidades destes sistemas, sendo utilizada tanto para otimização quanto para proteção do mesmo. No cenário dos *smart grids*, ela também terá fundamental importância para muitas das características esperadas por estas redes.

O gerenciamento do sistema elétrico de distribuição visa manter a topologia da rede em uma configuração viável, dado um conjunto de restrições sobre o mesmo. Discretizando uma sequência temporal onde, por exemplo, as cargas do sistema mudam de um momento para o outro, o sistema multiagente aqui proposto e aplicado deve ser versátil o suficiente para alterar a configuração da rede e mantê-la dentro dos limites de restrições impostas.

Será estudado o modelo conceitual de sistemas multiagentes e *smart grids*. Após isso, um sistema multiagente será modelado para aplicação da funcionalidade de gerenciamento do sistema elétrico de distribuição. Os equipamentos dos sistemas elétricos que, imagina-se, serão responsáveis pela reconfiguração topológica do sistema, serão modelados como agentes, bem como as interações por eles realizadas para executarem coletivamente a tarefa proposta.

Os métodos a serem implementados no sistema multiagente combinarão algoritmos de otimização com técnicas de inteligência artificial em ambiente distribuído.

Finalmente, o trabalho procura através destas pesquisas validar a aplicação de sistemas multiagentes para estudos relacionados com esta nova geração dos sistemas elétricos de potência. Acreditamos que as características dos sistemas multiagentes os tornam fortes candidatos para as pesquisas neste campo no futuro próximo. Assim, este trabalho também tem um teor pedagógico na busca pela contribuição ao desenvolvimento das pesquisas e estado da arte da área aqui tratada.

1.3 Metodologia 26

1.3 Metodologia

A metodologia deste trabalho consistiu, em um primeiro momento, no estudo de artigos científicos e leitura de textos sobre *smart grids*, afim de conhecer quais as características destes sistemas e o que os compõem.

Com esses dados, fez-se a escolha da funcionalidade específica do *smart grid* a ser estudada e qual tecnologia seria utilizada para simulação e consequente estudos computacionais sobre o tema. Decidiu-se, afim de se conseguir simular as redes elétricas inteligentes com uma certa fidelidade, os sistemas multiagentes. Cabe comentar que a seleção da tecnologia utilizada não foi anterior à definição do projeto. Uma explicação mais aprofundada sobre esta escolha encontra-se nos Capítulos 2, 4 e 5 desta obra.

Em seguida, foi pesquisado em livros e artigos como o gerenciamento da topologia da rede e o problema de reconfiguração do sistema de distribuição é tratado hoje nas atuais redes elétricas, e como ele poderia ser tratado em um sistema distribuído cuja comunicação entre os equipamentos é possível e a automação sobre o controle da rede está presente.

Também fora estudada a teoria sobre sistemas multiagentes, sua relação com a inteligência artificial e sistemas distribuídos, suas teorias organizacionais e as tecnologias de programação que implementam o paradigma orientado a agentes (CASTRO; ALENCAR; SILVA, 2006). Estes assuntos estavam presentes em artigos, livros e *surveys*. Também foi escolhida e estudada a linguagem de programação, bibliotecas e tecnologias relacionadas a serem utilizadas neste trabalho.

A partir deste levantamento, foram feitos alguns estudos preliminares sobre a distribuição do cálculo do algoritmo de Dijkstra e a modelagem de controle distribuído usando sistemas multiagentes. O primeiro estudo serviu de base para maiores desenvolvimentos sobre a tomada de decisão distribuída e posterior implementação na realidade dos sistemas elétricos de distribuição; o segundo, foi específico para um maior amadurecimento da modelagem computacional da proposta aqui apresentada.

Após estes estudos preliminares, foi feita a modelagem de equipamentos de sistemas elétricos que realizassem o monitoramento de cargas, alimentadores e chaves como agentes, bem como as interações entre os mesmos via trocas de mensagens, afim de que o modelo global atendesse ao objetivo de simulação do gerenciamento da rede via reconfiguração topológica do sistema.

Com a implementação computacional do modelo, o software foi executado em alguns conjuntos de testes que tivessem as características que propiciassem a verificação das funcionalidades esperadas.

1.4 Estrutura do Trabalho

A estruturação deste trabalho divide-se da seguinte forma:

O Capítulo 1 fez uma breve descrição da área de estudos, motivação da pesquisa e metodologia utilizada.

O Capítulo 2 apresentará um pouco mais de informações, possibilidades e alguns projetos em andamento na área de *smart grids* ao redor do mundo, discutindo as mudanças imaginadas para as redes convencionais de energia e como isso afeta as pesquisas em simulações computacionais de sistemas de distribuição.

O Capítulo 3 foca no problema de gerenciamento de sistemas de distribuição. Este foi o problema escolhido para estudar a implementação de algoritmos computacionais em um cenário onde o sistema de distribuição deve buscar informações de seu estado e executar tarefas de reconfiguração de forma autônoma. Este capítulo também apresenta uma revisão bibliográfica sobre reconfiguração de sistemas elétricos e evidencia de que formas as pesquisas nessa área serão modificadas pelas novas possibilidades fornecidas pelas redes inteligentes.

O Capítulo 4 trata da teoria dos sistemas multiagentes através de uma revisão bibliográfica sobre o assunto. Tópicos como definição, características, aspectos organizacionais e aplicações em sistemas elétricos de potência são abordados. Também há uma exposição sobre a proximidade de características entre sistemas multiagentes e *smart grids*.

O Capítulo 5 começa discorrendo sobre a importância de simulações e como um pesquisador deve encará-las. Em seguida, são apresentados dois trabalhos preliminares desenvolvidos para dar subsídios ao trabalho central desta obra, que é detalhadamente apresentado tanto em seu modelo quanto em sua implementação, com resultado dos teses aplicados.

O Capítulo 6 encerra estre trabalho apresentando algumas reflexões sobre o tema e discorre sobre os resultados dos testes realizados pelos autores. Finalmente, são discutidas algumas possibilidades de trabalhos futuros que darão prosseguimento aos estudos aqui desenvolvidos.

2 Smart Grids

2.1 A Nova Geração dos Sistemas Elétricos de Potência

Grupos de pesquisa situados em universidades e empresas ao redor do mundo se debruçam no estudo de uma proposta para a nova geração de sistemas elétricos de potência, que integre avanços tecnológicos em áreas como computação, telecomunicações, sistemas distribuídos e inteligência artificial às etapas de geração, transmissão e distribuição da energia elétrica.

Existem várias denominações para referenciar os projetos com essa finalidade, entretanto a mais comumente utilizada hoje chama-se *smart grids* - as "redes inteligentes". A combinação das tecnologias citadas anteriormente tem potencial para prover uma camada de comunicação às redes elétricas, permitindo a troca de informações sobre o estado da rede entre equipamentos alocados ao sistema.

Estes dados coletados e trocados, além de permitirem um controle mais eficaz sobre o estado do sistema e uma melhor observação das características da rede para os operadores, também alimentarão bancos de dados e/ou bases de conhecimento que serão utilizados por sistemas especialistas para desempenhar alguma função específica do *smart grid*, notadamente auxiliando o processo de tomada de decisão do sistema durante intervenções autônomas da rede.

Esta nova geração de sistemas elétricos de potência transformaria os sistemas elétricos convencionais em um grande "sistema de informação" computacional, onde as características já mencionadas de controle distribuído e conhecimento do estado da rede possibilitariam interessantes funcionalidades. Entre estas, destacam-se (BROWN, 2008; WEI; YU-HUI; JIE-LIN, 2009):

- Auto-recuperação do sistema;
- Alta qualidade da energia entregue;

- Resistência à ataques cibernéticos;
- Possibilidade de utilização em larga escala de geração distribuída;
- Acomodação de equipamentos para estoque de energia;
- Controle e acompanhamento pormenorizado do consumo por parte do usuário;
- Maior acompanhamento da demanda dos usuários pelas empresas de distribuição;
- Reconfiguração do sistema de distribuição a partir de medições observadas "on-line";
- Minimização dos custos de operação e manutenção.

A Figura 2.1, apresenta um *smart grid* dividido em camadas, a saber, a camada do consumidor, camada de comunicação e camada da substação. O fluxo de informação flui nos dois sentidos entre camada do consumidor e a camada da substação via camada de comunicação. Cada camada tem equipamentos e tecnologias próprias. Em especial, naquela a qual faz parte o consumidor, existem diversos equipamentos componentes de um *smart grid*. Percebe-se que todos eles se utilizam da rede para finalidades tais como consumo, geração, fornecimento de energia ao sistema de distribuição e/ou comunicação entre os componentes.

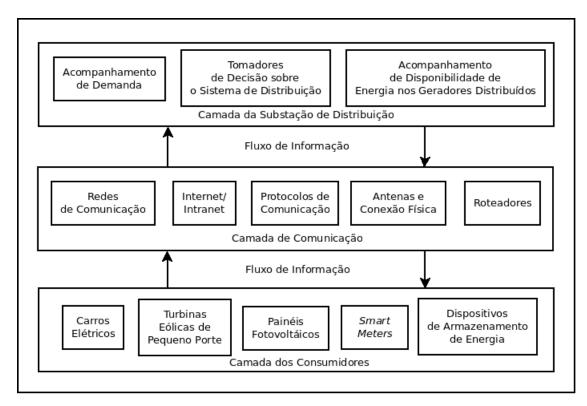


Figura 2.1: Smart Grids em camadas

Espera-se que, com o contínuo desenvolvimento, pesquisa e implementação desta tecnologia, mais funcionalidades deverão ser imaginadas e posteriormente estudadas para implementação no *smart grid*. Esta linha de estudos é bastante recente e repleta de possibilidades para pesquisa e inovação.

O esforço empreendido no desenvolvimento desta tecnologia é motivado por dois movimentos que ganharam grande visibilidade na última década entre os atores do mercado global de energia - a crescente desregulação deste mercado e a busca pela eficiência energética a partir dos preceitos da responsabilidade ambiental (WEI; YU-HUI; JIE-LIN, 2009).

A desregulação do mercado energético retirou, através das privatizações, muito da responsabilidade que o Estado exercia sobre a geração, transmissão e distribuição de energia elétrica¹. Criou-se então um cenário em que existem diversas empresas especializadas em um ou mais ramos do mercado energético, competindo entre si pelos consumidores a partir de regras de mercado próprias da área.

Entretanto, estas diferentes empresas se utilizam de um mesmo sistema elétrico que conectaas tanto a seus consumidores quanto entre si. Logo, operações que envolvem recursos pertencentes a diferentes empresas, em especial aquelas responsáveis por linhas de distribuição, são comuns.

Podemos citar, como exemplo, a permuta de uma chave de distribuição que roteará a energia que passava em um trecho da linha para um outro trecho, dessa vez sob responsabilidade de uma empresa diferente daquela que executou a reconfiguração.

Neste caso, um sistema distribuído que observa o estado e demais medidas da rede seria interessante para este novo cenário fortemente descentralizado. Assim, cada empresa teria a possibilidade de, levando em conta os dados observados, acompanhar o desempenho de sua respectiva tarefa e programar equipamentos para realizarem uma ação a partir de um estado específico do sistema. Estes sistemas teriam autonomia para executar suas funcionalidades, sempre remetendo a seus responsáveis as descrições sobre suas ações e as causas que os levaram a realizá-las.

O segundo movimento que impulsiona a pesquisa e desenvolvimento na área de *smart grids*, a busca pela eficiência energética a partir dos preceitos da responsabilidade ambiental, surge como consequência de vários debates e seminários patrocinados por empresas, governos e instituições sociais nos últimos anos sobre a questão ambiental, aquecimento global e limitações

¹Apesar de, em alguns países, o Estado manter agências reguladoras para fiscalizar os serviços destas empresas

das reservas energéticas dos países.

Esta preocupação é própria de países cuja matriz energética é prioritariamente constituída por fontes não-renováveis ou bastante poluentes, como alguns países europeus, uma realidade muito diferente da brasileira onde a principal fonte de energia é a hidrelétrica ².

Nestes países, a possibilidade de utilização de fontes de energia renováveis e não poluentes implicaria, dependendo das características de cada região, no uso de energia eólica, solar, biomassa, etc.

Porém, a utilização e manuseamento destes tipos de fontes de energia são ainda pouco maduras em seu desenvolvimento (como a energia solar) ou muito dependentes de condições climáticas/ambientais bastante inconstantes (como a força dos ventos). A utilização destes tipos de fonte de energia demandaria um sistema de transmissão e/ou distribuição que conseguisse gerenciar o fluxo de energia dessas fontes a partir da disponibilidade de geração das mesmas em determinado tempo.

Para potencializar a adoção deste tipo de fonte de energia, seria interessante utilizá-las de forma distribuída em um dado espaço geográfico. Por exemplo, pequenas turbinas eólicas e placas fotovoltaicas em coberturas de edifícios de uma metrópole, responsáveis pela geração de energia e posterior injeção da mesma na rede de distribuição. Essa forma de obtenção de energia chama-se geração distribuída (RODRIGUES; BORGES; FALCÃO, 2007).

Abaixo, apresentam-se duas imagens que ilustram as diferenças conceituais entre geração tradicional - centralizada - e a geração distribuída. A Figura 2.2 apresenta um único gerador e as linhas de distribuição em sentido único, alimentando indústria, área residencial e comércio. Já a Figura 2.3 apresenta um grande gerador central e vários pequenos geradores. Percebe-se que os fluxos representados pelos arcos chegam de várias direções a um mesmo lugar (no comércio, por exemplo) ou mesmo chegam a sair de convencionais consumidores de energia (indústria, no exemplo). Isso exemplifica a possibilidade de geração de energia também por parte de alguns consumidores, através do uso de painéis fotovoltáicos ou pequenas turbinas de energia eólica. Estas imagens foram baseadas nos originais de (WEI; YU-HUI; JIE-LIN, 2009).

Entretanto, o modelo convencional de redes de distribuição que existem nas cidades e é trabalhado pela engenharia elétrica não comporta este tipo de gerenciamento de forma simples.

²Apesar de existirem alguns estudos sobre a produção de gases poluentes nos lagos criados por grandes barragens em usinas hidrelétricas, levaremos em conta neste trabalho que o processo da energia hidrelétrica é pouco poluente se comparada a energias geradas em termoelétricas, por exemplo. Maiores detalhes sobre este assunto estão em (KEMENES; FORSBERG; MELACK, 2007).

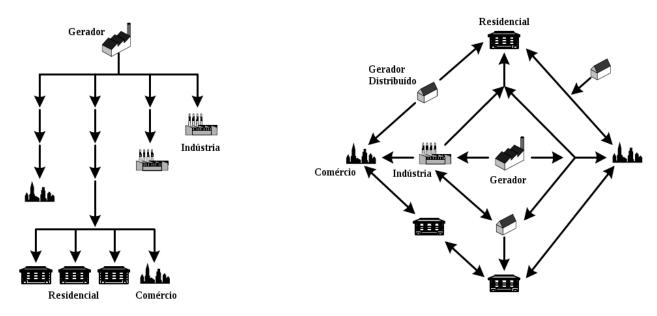


Figura 2.2: Sistema de Distribuição com Geração Centralizada

Figura 2.3: Sistema de Distribuição com Geração Distribuída

Em muitos casos, a geração distribuída pode prejudicar o sistema de distribuição ou transmissão como um todo a partir de uma ocasional diferença de tensão, por exemplo. E como a rede não tem automação para executar ações corretivas em casos como este, torna-se muito complicado, ou mesmo impraticável, o uso deste tipo de geração.

Logo, um sistema de controle e avaliação distribuído ao longo da rede poderia gerenciar estas múltiplas fontes de energia sem desestabilizar o sistema de distribuição ou transmissão, utilizando a geração distribuída a partir do acompanhamento temporal da disponibilidade de geração das fontes, injetando energia no sistema sempre que isso não incorrer em problemas técnicos para a estrutura ou para o provimento de seus respectivos serviços.

O desenvolvimento da tecnologia de *smart grids* e suas funcionalidades impactará no planejamento dos futuros sistemas de distribuição. Se hoje temos sistemas prioritariamente radiais e sem (ou com pouca) automação, no futuro o controle distribuído da rede permitirá que a mesma tenha uma topologia malhada, permitindo uma maior flexibilidade para a reconfiguração do sistema (BROWN, 2008).

Apesar da retórica "futurista" ou, diriam alguns, "sonhadora demais", muitos países criaram projetos de implantação de *smart grids*, e inclusive alguns já começaram as primeiras fases de construção dos mesmos, mostrando que a aposta nesta tecnologia é séria e está mobilizando a indústria energética em geral.

2.2 Países com Projetos para Smart Grids

Foi comentado na introdução deste trabalho sobre o projeto da Comissão de Energia da União Europeia e seus estudos sobre as perspectivas energéticas dos países daquele bloco até 2020. No relatório, a comissão prevê investimentos no desenvolvimento das tecnologias de redes inteligentes como uma forma de atingir a eficiência energética, utilizando prioritariamente fontes alternativas de energia não poluidoras através de uma rede que seja "mais inteligente, eficiente e robusta" (ENERGY, 2011).

Com este objetivo, a União Europeia iniciou ainda em 2005 o seu projeto de estudo e desenvolvimento de redes inteligentes, chamado ETP SmartGrids - *European Technology Platform SmartGrids* ³ (COMMISSION, 2006). As características buscadas pelo bloco de países para suas redes de energia são flexibilidade quanto a reconfiguração, expansão de acessibilidade ao sistema para os usuários, uso prioritário de fontes de energia renováveis e busca pelo gerenciamento eficiente da energia (JIANG et al., 2009).

Novamente na introdução deste trabalho, foi comentado sobre a implantação de *smart grids* na cidade portuguesa de Évora, um projeto da empresa EDP em parceria com outras empresas, centros de pesquisa e universidades. Este projeto já encontra-se em um interessante estado de operação, permitindo aos consumidores o acompanhamento do consumo de energia de sua casa via internet, lâmpadas de iluminação pública inteligentes que adequam sua luminosidade (e, por consequência, seu consumo) a partir de relógios e sensores de presença, possibilidade de criação de um perfil de consumo personalizado para um dado grupo de usuários, e mais.

Apesar desses avanços, o projeto de Évora ainda está em sua fase inicial. Muitas funcionalidades ainda estão previstas para serem implementadas, entre elas, a criação de uma rede de abastecimento de carros elétricos e a possibilidade de gestão da geração distribuída a partir de microgeradores instalados nas residências e escritórios comercias dos próprios consumidores, com disponibilidade inclusive de venda da energia produzida para a concessionária de distribuição.

Além destes projetos desenvolvidos no âmbito dos países europeus, existem muitas outras iniciativas de estudo e implantação de *smart grids* sendo realizadas por governos e laboratórios de pesquisa em outros continentes.

Por exemplo, o Departamento Americano de Energia (U.S. Department of Energy) iniciou

³A União Europeia mantém um site específico sobre seu projeto ETP SmartGrids, com vários documentos acessíveis para leitura. http://www.smartgrids.eu/ (acessado dia 17 de fevereiro de 2011).

o projeto GridWise, que tem como objetivo modernizar a infraestrutura e operação das redes de distribuição, adicionando um fluxo de mão-dupla tanto de eletricidade quanto de informação entre consumidores e subestações (BROWN, 2008).

Este projeto acabou fomentando a criação de outros dois subprojetos complementares: o GridApps e o GridWorks, ambos patrocinados pelo Departamento Americano de Energia.

GridApps é um consórcio de empresas de tecnologia que trabalham para fornecer aplicações que visam modernizar a operação dos sistemas elétricos de transmissão e distribuição. As tecnologias utilizadas neste projeto podem ser divididas em três tipos: monitoramento e gerenciamento de redes, novos dispositivos e engenharia de integração de sistemas para otimização de performance.

Já GridWorks tem como alvo fortalecer a confiabilidade dos sistemas elétricos através da modernização de cabos, condutores, subestações e sistemas de proteção, utilizando principalmente a eletrônica de potência. A implementação deste projeto inclui as etapas de desenvolvimento das tecnologias e instalação imediata das mesmas nos sistemas elétricos convencionais já existentes.

Também nos Estados Unidos, o projeto DV2010 (*Distribution Vision 2010*) tem o objetivo de, através da combinação de redes de comunicação de alta velocidade, dispositivos de chaveamento de redes e controladores inteligentes, criar uma rede de distribuição virtualmente "à prova de falhas" (FANNING, 2005; BROWN, 2008).

Entretanto, pelo menos em um primeiro momento, os resultados obtidos pelo DV2010 não seriam implementados em todo o sistema de distribuição. Seria criada uma área "premium" onde a operação do sistema teria estas características funcionais. Esta área seria destinada para consumidores com necessidades críticas de energia (grandes indústrias, data centers e outras) e que poderiam pagar um valor adicional por um serviço de alta qualidade.

A EPRI - *Electric Power Research Institute*, centro de pesquisa californiano, iniciou em 2001 o projeto IntelliGrid, com o objetivo de criar uma nova infraestrutura de distribuição de energia integrando avanços na comunicação, computação e eletrônica ⁴ (JIANG et al., 2009).

Esta iniciativa almeja o desenvolvimento, integração e aplicação destas tecnologias para tornar a infraestrutura dos sistemas de distribuição mais eficientes, com alta qualidade, segurança e robustez. Este projeto acabou evoluindo e gerando outro com objetivos um pouco mais ousa-

⁴A ERPI mantém um site específico para o projeto IntelliGrid, com muitas informações, artigos, e mais. http://intelligrid.epri.com/ (acessado dia 17 de fevereiro de 2011).

dos. O ADA - Advanced Distribution Automation visa criar os sistemas de distribuição do futuro, onde a infraestrutura será totalmente automatizada através de uma arquitetura de operação flexível, gerenciada por sistemas de controle que se comunicam através de um protocolo aberto (BROWN, 2008).

Também existem várias outras iniciativas de desenvolvimento de *smart grids* em países do continente asiático. Citar-se-á neste texto, como exemplo, as experiências da China e Coréia do Sul.

A China é atualmente um dos países que mais se expandem economicamente no mundo. Com o crescimento da economia, aumenta também a demanda por energia. Estima-se que o país, que atualmente conta com uma capacidade de geração instalada de 792 milhões de kilowatts, deverá necessariamente dobrá-la até 2020 (BI et al., 2010).

O governo chinês, então, aposta na geração distribuída a partir de fontes renováveis de energia como uma forma de complementar aos esforços para suprir sua demanda energética neste espaço de tempo. Logo, o país necessitará de uma rede de distribuição que dê suporte à integração de diferentes tecnologias de geração de energia, garantindo também a eficiência da operação.

Em 2007, a *East China Power Grid Company* iniciou as pesquisas para avaliação da viabilidade do desenvolvimento de *smart grids* para o país. Os resultados obtidos foram a constatação desta viabilidade, e o direcionamento do plano para desenvolvimento da mesma deveria se pautar na criação de equipamentos de alta tecnologia e na habilidade de inovação dos grupos de pesquisa do país (LU; AI, 2009).

Foi então idealizada uma estratégia de 3 passos para a implantação de *smart grids* no leste chinês: primeiramente, a construção de centrais para monitoramento das redes de distribuição a partir de 2010; em seguida, a implementação de uma camada de comunicação na rede de distribuição com capacidade de automação preliminar até 2020; e por último, a capacidade do sistema de auto-recuperar-se já em funcionamento a partir de 2030 (LU; AI, 2009).

Em adição, a *Shangai Municipal Electric Power Company* e a *North China Power Grid Company* também lançaram projetos de redes inteligentes, em 2008 e 2009, respectivamente, focando medição inteligente, automação da distribuição, interação com o usuário e mais outras funcionalidades (BI et al., 2010).

O desenvolvimento de *smart grids* na Coréia do Sul é capitaneado pela empresa KEPCO -

Korea Electric Power Corporation entre outras organizações de pesquisa. Através do projeto Power IT, a empresa utiliza tecnologias de informação e comunicação para prover os já comentados controle e monitoramento distribuído do sistema de distribuição (KWON et al., 2008).

Além dos países citados neste capítulo, vários outros investem na tecnologia de *smart grids* a partir de um plano político próprio para os sistemas elétricos. Diante da quantidade de governos e do montante em investimentos por eles executados, o movimento para a nova geração dos sistemas elétricos de potência, em escala mundial, parece ser irreversível. Os países que não aderirem a ele nesta década, certamente o farão no futuro próximo, visto a quantidade de tecnologia e *expertise* que está sendo desenvolvida agora.

2.3 A Simulação Computacional de Smart Grids

A adição de uma camada de comunicação ao sistema elétrico de distribuição possibilitará que equipamentos e demais artefatos técnicos alocados neste sistema troquem informações relevantes sobre o estado do mesmo, permitindo a execução de ações de controle sobre a rede.

Esta é uma característica marcante que aponta o grande diferencial entre as redes *smart grids* e as redes elétricas convencionais, que não possibilitam a comunicação entre equipamentos e, por consequência, não oferecem subsídios para a tomada de decisão e automação na reconfiguração ou gerenciamento do sistema de distribuição.

Neste cenário de sistemas elétricos tipo *smart grids*, percebe-se que a mudança conceitual introduzida por esta tecnologia é muito forte e impacta sobretudo, entre outras áreas, na maneira como as pesquisas em sistemas elétricos de potência que exigem simulação computacional são desenvolvidas.

Com diversos tipos de equipamentos acoplados ao sistema de distribuição, monitorando características do mesmo, e a possibilidade de troca de informação entre eles, se faz necessário pensar em métodos computacionais de simulação condizentes com esta nova arquitetura para verificar a viabilidade de novas funcionalidades, em especial as de controle e gerenciamento, que as redes elétricas poderão implementar nessa nova geração de sistemas elétricos de potência.

Em termos computacionais, podemos pensar os *smart grids* como um típico sistema distribuído (não confundir com sistemas elétricos de distribuição). Estes sistemas caracterizam-se por serem compostos de hardware ou software, localizados em sistemas computacionais interligados em rede, que se comunicam e coordenam suas ações apenas enviando mensagens entre

si (COULOURIS; DOLLIMORE; KINDBERG, 2005). O conceito de sistemas distribuídos é amplamente trabalhado na ciência computação, nas linhas de pesquisa de redes, processamento distribuído, etc.

A principal motivação do desenvolvimento de sistemas distribuídos vem da necessidade do compartilhamento de recursos e escalonamento de processamento para tarefas que demandam muitas requisições, onde técnicas distribuídas poderão levar o sistema a adquirir uma maior eficiência. Exemplos de sistemas desse tipo podem ser encontrados na estrutura de servidores da *web*, onde vários computadores processam e respondem às requisições de algum serviço específico.

Entre algumas características dos sistemas distribuídos, podemos citar (COULOURIS; DOL-LIMORE; KINDBERG, 2005):

- Concorrência tarefas são processadas concorrentemente no ambiente, ou seja, quando uma tarefa tem seus pré-requisitos completos e recursos disponíveis, ela estará sendo executada em uma máquina. Caso outra tarefa também esteja pronta para o processamento, a mesma poderá ser executada em outra máquina pertencente ao sistema.
- Sem horário global quando programas precisam cooperar para resolver determinada tarefa, eles fazem isso através de trocas de mensagens. Isso é executado dessa forma por conta da dificuldade em sincronizar o tempo nas máquinas interligadas.
- Falhas independentes pode ocorrer de algumas máquinas alocadas no sistema distribuído falharem em suas tarefas. Isso não impedirá que demais máquinas continuem seus respectivos processamentos. Apesar de, em alguns casos, a falha de uma máquina levar à falha total do sistema, é possível criar metodologias de detecção e tratamento destas quando detectadas as chamadas técnicas de tolerância à falhas.
- Heterogeneidade não é necessário que o sistema computacional distribuído seja construído apenas com equipamentos de um mesmo tipo. Pelo contrário, um sistema deste tipo pode ser composto por máquinas para diversas finalidades, que cooperarão entre si para atingir um objetivo dado.
- Protocolos abertos de comunicação é necessário que os diversos componentes do sistema consigam comunicar-se entre si. Para tanto, torna-se imprescindível que protocolos abertos de comunicação sejam utilizados, possibilitando o envio, recebimento e entendimento de mensagens entre os componentes.

 Transparência - para o usuário, sistemas distribuídos não aparentam ser formados por várias máquinas interconectadas. Normalmente, aquele que utiliza o sistema trabalha apenas com uma máquina, e o sistema gerencia a divisão das tarefas autonomamente.

Existem também vários tipos de arquiteturas de sistemas distribuídos. Por exemplo, a Figura 2.4 apresenta a arquitetura cliente-servidor, onde uma máquina "central" é responsável por distribuir as tarefas entre as demais máquinas processadoras. Já a Figura 2.5 apresenta uma rede onde a distribuição do processamento é feita par-a-par, também conhecida como arquitetura P2P (peer-to-peer).



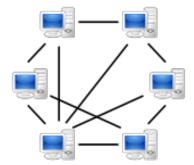


Figura 2.4: Modelos de sistemas distribuídos - arquitetura cliente-servidor

Figura 2.5: Modelos de sistemas distribuídos - arquitetura P2P (par-a-par)

Estas arquiteturas de sistemas distribuídos são planejadas a partir do problema que deseja-se abordar. Cada uma delas tem características próprias que as definem como melhor solução para um dado problema e não para outro. Também é possível arquiteturas híbridas - por exemplo, poderíamos ter uma arquitetura P2P que automatiza uma dada tarefa mas que sempre reporta seus procedimentos para um computador central, como forma de criação de um *log* de informações e decisões tomadas ao longo da vida útil do sistema.

Com a conceituação e as características dos sistemas distribuídos exposta, podemos agora fazer uma comparação entre as características desejadas de um *smart grid* e as presentes em um sistema distribuído computacional. A Tabela 2.1 apresenta alguns desses elementos:

Dada as similaridades de características, pode-se aferir que as técnicas de sistemas distribuídos são fortes candidatas para servirem de ferramentas à simulações computacionais mais fiéis aos sistemas de distribuição *smart grids*. Entretanto, falta-nos avaliar mais uma característica fundamental dos sistemas elétricos de nova geração: a capacidade de resolver problemas de forma autônoma, a partir da observação do estado do mesmo.

Tabela 2.1: Características próximas entre sistemas distribuídos e *smart grids*

Sistemas Distribuídos

- Concorrência processos sendo executados simultaneamente em máquinas diferentes.
- Heterogeneidade diferença entre máquinas no mesmo sistema distribuído.
- Sem horário global sistemas coordenam suas atividades a partir da troca de mensagens.
- Falhas independentes uma falha poderá afetar parte de um sistema distribuído, sem necessariamente afetar o sistema inteiro.
- Protocolos abertos de comunicação
 é necessário que os diversos componentes do sistema consigam comunicar-se entre si através de um protocolo comum.

Smart Grids

- Concorrência processos diferentes executados em máquinas diferentes do sistema de distribuição.
- Heterogeneidade máquinas com funções distintas estarão alocadas na mesma rede de distribuição.
- Sem horário global equipamentos do sistema usam a camada de informação para coordenar tarefas.
- Falhas independentes falha de equipamentos específicos de uma tarefa podem não afetar os relacionados à outra tarefa.
- Protocolos abertos de comunicação os diferentes equipamentos alocados ao sistema devem utilizar um protocolo comum para troca de mensagens.

Um *smart grid* tem diversas funcionalidades a serem executadas pelos sistemas nele instalados. Cada funcionalidade dessas pode ser vista como um objetivo a ser atingido, como por exemplo, redução das perdas, recuperação do sistema, gerenciamento da demanda, entre outras. Os equipamentos instalados observarão o estado da rede, comunicar-se-ão entre si caso necessário e executarão uma ação afim de alcançar seu propósito.

Esta descrição é muito similar ao conceito de inteligência artificial proposto por alguns autores ⁵. Por exemplo, para Russel e Norvig, inteligência artificial é o estudo e implementação de agentes racionais. Agentes racionais são abstrações para um conjunto de software ou hardware que procura atingir seus objetivos através de suas crenças e comportamentos (RUSSEL; NORVIG, 2004).

Existe uma área específica da inteligência artificial que trabalha sistemas inteligentes nãoisolados, que se relacionam entre si. Esta área chama-se Inteligência Artificial Distribuída (do inglês, *Distributed Artificial Intelligence*), e seu foco de pesquisa se dá no estudos de modelos e técnicas onde a característica distribuída dos agentes é inerente à arquitetura do sistema (GARCIA; SICHMAN, 2003).

Em sistemas de inteligência artificial distribuída, existe uma ênfase maior no "comportamento social" dos agentes. A perspectiva é que técnicas dessa área possam dar subsídios para coordenar um conjunto de agentes, alguns inclusive com funcionalidades diferentes entre si, para simular um determinado comportamento de uma sociedade cujo objetivo é alcançado a partir das interações entre os agentes. Fala-se que, a partir do comportamento dessa sociedade, emerge a ação que o sistema deverá executar para atingir seu objetivo.

É neste contexto da inteligência artificial distribuída que surgem os chamados sistemas multiagentes (WOOLDRIDGE, 2009; SHOHAM; LEYTON-BROWN, 2009; WEISS, 1999). Nestes sistemas, cada agente é autônomo e projetado para atingir um dado objetivo. Entretanto, ele também tem a capacidade de se relacionar com outros agentes, seja através do envio de mensagens, seja através da repercussão no ambiente a partir da execução de determinada tarefa. O projeto de sistemas multiagentes deve se preocupar em garantir que os comportamentos dos agentes consigam gerar um encadeamento de ações que resolverão o problema proposto ou atingirão o objetivo imaginado (GARCIA; SICHMAN, 2003). Este tópico será tratado em maiores detalhes no Capítulo 4 desta dissertação.

⁵O conceito de inteligência artificial ainda suscita muita divergência entre os teóricos da área, além de haver aqueles que defendem que o termo é usado de maneira incorreta, preferindo "inteligência computacional" ou até mesmo "sistemas especialistas". Maiores detalhes sobre esta discussão, além de um apanhado sobre como outras áreas tratam o termo, podem ser vistos em (RUSSEL; NORVIG, 2004)

Assim, com os conceitos de sistemas distribuídos, a relação entre as características de *smart grids* e sistemas distribuídos, inteligência artificial distribuída e sistemas multiagentes dissertados, foi apresentado o embasamento sobre como este conjunto de tecnologias e conceitos funcionariam muito bem como uma alternativa de simulação computacional mais próxima às características das redes *smart grids*.

De fato, sistemas multiagentes já são empregados na engenharia elétrica de potência já há algum tempo. Existem muitos trabalhos que utilizam esta tecnologia publicados na área (MCARTHUR et al., 2007a, 2007b; SILVA, 2009).

Entretanto, apenas recentemente esta técnica está sendo pensada a partir do seu potencial de simular as características que os sistemas de distribuição deverão assumir nos próximos anos (RUMLEY et al., 2008). Logo, estudos que utilizem sistemas multiagentes para simular funcionalidades de *smart grids* deverão ganhar espaço nos próximos anos em publicações científicas especializadas da área.

Na busca por validar a hipótese de que sistemas multiagentes são importantes ferramentas para a simulação computacional de *smart grids*, esta tecnologia será utilizada para aplicação ao problema de gerenciamento do sistema de distribuição. O próximo capítulo aprofundará melhor este problema, formalizando-o, discutindo sua importância, comentando as técnicas usuais empregadas para sua resolução e como o ambiente de *smart grids* criará um novo cenário onde a característica distribuída deste, proporcionará desafios para o desenvolvimento de uma nova e interessante maneira de implementá-lo.

3 Gerenciamento dos Sistemas Elétricos de Distribuição via Reconfiguração da Rede

3.1 Conceitos sobre Reconfiguração de Sistemas Elétricos de Distribuição

A partir do momento em que, na etapa de geração, um tipo de energia é transformada em energia elétrica, o fluxo de corrente flui pelos equipamentos e cabos de transmissão ou distribuição, em direção aos centros consumidores ou às cargas do sistema elétrico.

As redes elétricas do sistema de distribuição são, em sua maioria, configuradas para operarem em uma topologia radial, onde cada carga alocada recebe energia de apenas um ramo do sistema. Esse tipo de topologia facilita fatores relacionados à proteção, coordenação e atenuação de correntes de curto-circuito além de diminuir custos de equipamentos (MANTOVANI; CA-SARI; ROMERO, 2000).

Entretanto, apesar das facilidades, a topologia radial também pode ocasionar problemas em situações de contingência ou sobrecarga, nas quais as vezes se faz necessário o corte de cargas, ilhamentos de algumas partes do sistema elétrico, entre outras (GUERRA; UNSIHUAY-VILA; KAGAN, 2011).

Nesse contexto, em uma parte dos sistemas de distribuição, apesar da operação normalmente acontecer em uma topologia radial, a construção dos sistemas de distribuição é feita em forma de malha. A configuração radial é, então, obtida a partir de operações sobre equipamentos conhecidos como "chaves", que conectam trechos dos sistemas elétricos e permitem a passagem de corrente entre estes.

Quando uma chave está fechada, o circuito também fecha, possibilitando a passagem de

corrente entre os trechos interligados. Caso a chave esteja aberta, o circuito também configurase como aberto e não há passagem de corrente pela ligação monitorada pela chave.

A Figura 3.1, baseada em (RUGTHAICHAROENCHEEP; SIRISUMRANNUKUL, 2008), exemplifica estes conceitos. Nela, existem duas subestações de distribuição, *Substação 1* e *Substação 2*. Os símbolos "□" representam as chaves abertas e os símbolos "□" representam as chaves fechadas. Cada chave é referenciada por um "CB" sucedido por um número entre 1 e 9. Já os símbolos "•" nas linhas representam cargas alocadas ao sistema.

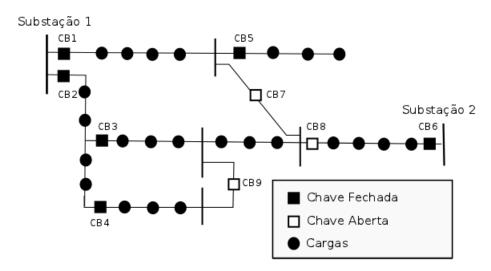


Figura 3.1: Configuração topológica de um sistema de distribuição.

Percebe-se na figura que o sistema elétrico, como um todo, tem uma topologia malhada. Encontrar um ponto de operação cuja configuração seja radial será possível a partir de operações de abertura ou fechamento das chaves CB's.

Na topologia apresentada na Figura 3.1, as chaves CB1, CB2, CB3, CB4, CB5 e CB6 estão fechadas, permitindo que haja passagem do fluxo de carga nas secções das linhas de distribuição unidas por estes equipamentos. As chaves CB7, CB8 e CB9 encontram-se abertas, impedindo a transferência de corrente entre suas secções.

Novas topologias radiais podem ser obtidas a partir de novas combinações de estados aberto ou fechado das chaves. Esta alteração topológica é chamada de reconfiguração de sistemas elétricos.

A busca por novas topologias radiais no sistema de distribuição visam atender algum objetivo específico, normalmente relacionados a aspectos sobre otimização, controle ou gerenciamento do estado das redes elétricas.

3.2 Aplicações de Reconfiguração de Sistemas Elétricos de Distribuição

Como já citado, a construção dos sistemas elétricos de forma malhada, onde a configuração topológica radial é obtida a partir de operações de abertura e/ou fechamento de chaves, permite um interessante grau de flexibilidade para reconfiguração da rede elétrica, afim de atingir algum ou uma combinação de objetivos e comportamentos.

Entre alguns comportamentos do sistema elétrico que operações de reconfiguração possibilitam atingir, temos aqueles relacionados com proteção elétrica, coordenação e atenuação de correntes de curto-circuito (MANTOVANI; CASARI; ROMERO, 2000), cortes de cargas e ilhamentos de partes do sistema (GUERRA; UNSIHUAY-VILA; KAGAN, 2011), encontrar rotas para o fluxo de carga que cumpram requisitos de tensão mínima e outras relacionadas a legislação, possibilitar ao fluxo de carga atingir determinado conjunto de cargas utilizando uma rota diferente, alternar alimentadores que enviam energia para determinadas cargas, balanceamento entre os alimentadores acoplados ao sistema, entre outros.

Outras aplicações recorrentes de reconfiguração de sistemas elétricos dizem respeito a problemas de otimização característicos do sistema. Por exemplo, diferentes topologias radias, compostas por trechos com impedâncias diferentes, resultam em níveis de perdas elétricas diferentes. Logo, a reconfiguração pode ser empregada para minimizar as perdas elétricas no sistema (BARAN; WU, 1989). Além desta, outras aplicações como minimização dos custos de construção da rede durante a etapa de planejamento e balanceamento de carga, são possíveis (DELBEM; CARVALHO; BRETAS, 2005).

Apesar das aplicações de reconfiguração de sistemas serem vastas e importantes para os sistemas elétricos, o custo computacional para resolvê-las na melhor configuração possível é bastante elevado e, em geral para diferentes instâncias de problemas, proibitivo.

A aplicação de reconfiguração da topologia em sistemas de distribuição para qualquer destes fins é um problema computacional de difícil solução, pois não existe um algoritmo de complexidade polinomial capaz de resolvê-lo (BARAN; WU, 1989; CIVANLAR et al., 1988). Podemos inferir que o número de redes possíveis (incluindo aquelas que violam as restrições) cresce exponencialmente de acordo com o número de chaves, a partir da função 2^n , onde n refere-se à quantidade de chaves (SCHMIDT et al., 2005). No exemplo da Figura 3.1, que é um sistema de pequeno porte, existem 512 configurações topológicas diferentes. Por conta do extenso espaço de busca gerado pelo problema e da impossibilidade de utilizar um algoritmo de

complexidade polinomial para resolvê-lo, a configuração topológica do sistema de distribuição é um exemplo de problema computacional do tipo NP-Árduo (CORMEN et al., 2009).

Nestes problemas, o esforço computacional aumenta de forma exponencial a medida que a instância do problema cresce de forma polinomial. A Tabela 3.1 apresenta a progressão do número de chaves e do número de estados possíveis a partir das diferentes combinações de estados entre as mesmas:

Número de Chaves	Número de Estados
10	1.024
11	2.048
12	4.096
15	32.768
20	1.048.576
25	33.554.432
30	1.073.741.824

Tabela 3.1: Progressão da quantidade de estados a partir do número de chaves

Percebe-se que um aumento polinomial no tamanho da instância provoca um aumento exponencial no número de estados, impactando decisivamente no aumento do custo da resolução computacional. Chama a atenção que o aumento da instância de 10 para 30 chaves implicou no aumento em mais de 10⁶ vezes na quantidade de combinações de estados em relação à instância original. Este fenômeno é característico dos problemas NP-Árduo, e é referenciado na literatura como "explosão combinatória" (RAYWARD-SMITH et al., 1996).

Do exposto, é possível perceber que a reconfiguração de sistemas elétricos tem grande importância em uma variada gama de aplicações relacionadas aos sistemas de distribuição. No contexto das redes elétricas inteligentes, esta ferramenta também é fundamental para muitas das promessas que este novo tipo de sistemas elétricos acena - mas é possível que os métodos e abordagens de resolução devam ser retrabalhados para se adaptarem às condições deste novo cenário.

3.3 Smart Grids e Reconfiguração de Sistemas Elétricos

Para os problemas citados neste texto onde a aplicação de reconfiguração de sistemas elétricos é uma ferramenta de importância fundamental, a topologia é bastante dependente do perfil de demanda do conjunto de consumidores em um dado momento definido no tempo. Quando há

variação da demanda, pode ocorrer daquela topologia que está em uso e configurada naquele momento não cumprir mais o objetivo a qual havia sido empregada.

Essa situação é bastante comum, pois o perfil de demanda do conjunto de consumidores não é constante e uniforme: ele varia, tanto em função das características das cargas quanto por influência de fatores outros como dia da semana, temperatura, estação do ano, etc (KAGAN; OLIVEIRA; ROBBA, 2005).

Além disso, o perfil de demanda com o qual a operadora de distribuição trabalha em um dado momento é aferido a partir de uma previsão baseada em observações e medições realizadas no passado, normalmente aplicando métodos inteligentes como redes neurais a um conjunto de dados e estimando-se os possíveis valores que seriam constatados no futuro.

Certamente, o método mais eficaz de acompanhar a demanda do conjunto de consumidores seria através de uma medição *online* ou em uma janela de tempo que mantenha um bom
nível de precisão dos valores reais, além de comunicação entre os equipamentos que fazem as
medições e os computadores que tomariam a decisão de reconfiguração topológica do sistema
de distribuição. Entretanto, esta abordagem ainda é bastante custosa e complexa, visto o alto
custo dos equipamentos e o próprio tratamento ao montante de dados gerados por uma medição
dinâmica, distribuída e com necessidade de comunicação em tempo real.

Por conta destas dificuldades, a maioria dos atuais métodos de criação do perfil de demanda dos consumidores baseiam-se na aplicação de métodos inteligentes e estocásticos. Apesar dessa alternativa não conduzir necessariamente aos resultados reais obtidos posteriormente, esta abordagem quando bem empregada garante uma boa margem de precisão entre os resultados estimados e os valores observados durante a medição. Entretanto, dificilmente estes valores são utilizados para a reconfiguração da topologia do sistema de distribuição durante a operação normal do sistema.

Essa realidade está cada vez mais perto de ser mudada, caso as promessas da nova geração dos sistemas elétricos sejam cumpridas. Com diversos grupos pesquisas trabalhando e o montante de investimento no desenvolvimento do conceito e da tecnologia de *smart grids*, espera-se que a medição da demanda e a reconfiguração do sistema aconteçam corriqueiramente e sem maiores complicações, cumprindo as restrições e garantindo uma boa qualidade do serviço de distribuição de energia.

A possibilidade de medição e comunicação de dados na rede elétrica em "tempo real", poderá indicar um ponto de ruptura na maneira como se dá a resolução de problemas em sistemas elétricos de distribuição convencionais e em sistemas *smart grids*, incluindo o gerenciamento e a reconfiguração do sistema.

Normalmente, a maneira como pesquisadores trabalham os problemas relacionados com os sistemas elétricos se dá em configurações estáticas do perfil da rede - dados de cargas, correntes elétricas e demais componentes são imutáveis, não variam de acordo com o tempo. Logo, por exemplo, aplicações de métodos inteligentes como as metaheurísticas algoritmos genéticos (MICHALEWICZ, 1996), busca tabu (GLOVER, 1990) ou outra, são normalmente direcionadas a um tipo de estado do sistema elétrico. Se esse estado muda - por exemplo, altera-se as tensões em um conjunto de cargas -, os métodos deverão ser aplicados novamente para este novo estado, acarretando em nova execução de todas as etapas que compõe os métodos.

No caso dos *smart grids*, é necessária a adaptação destes métodos comentados anteriormente, ou a utilização de novos métodos, para trabalhar as possibilidades da característica de troca de informações entre os componentes do sistema elétrico. Este tipo de rede requer tratamento rápido dos dados, obtenção de soluções para os problemas propostos e coordenação de ações nos diferentes equipamentos para reconfiguração da rede. Estes requisitos permitem entrever os desafios que o cenário de *smart grids* oferecem para pesquisadores da área de sistemas elétricos de potência.

Certamente, o avanço das "redes inteligentes" poderá propiciar, em um certo nível, a fusão do planejamento da operação baseada no perfil de carga dos consumidores, onde a configuração da rede se dá a partir destas informações, com a própria operação do sistema, onde durante a execução do serviço a rede poderá ter autonomia para reconfigurar a si mesma a partir das informações sobre a variação do consumo obtida e transmitida pelos equipamentos medidores inteligentes dos *smart grids*.

Em artigo publicado no ano de 1988, Civanlar *et al* comentaram sobre a importância da aplicação da reconfiguração do sistema de distribuição para o gerenciamento da rede em tempo real. Segundo os autores (CIVANLAR et al., 1988):

Distribution feeder reconfiguration can be used as a planning tool as well as a real-time control tool. Modifying the radial structure of the distribution feeders from time to time, by changing the open/closed states of switches o transfer loads from one feeder to another, may significantly improve the operating conditions of the overall system.

Na época, os autores mostraram-se entusiasmados com o avanço da miniaturização de computadores e processadores, além do crescente desenvolvimento das tecnologias de telecomunicações. Imaginava-se que o emprego destas tecnologias no sistema de distribuição possibilitariam maiores avanços na automação do mesmo, através de aplicações e processos de controle sobre a rede.

Este artigo ainda pontua algumas experiências pioneiras na utilização destas tecnologias em sistemas de pequena escala, através da aquisição distribuída de dados e automação da reconfiguração. Os projetos citados eram desenvolvidos por *Carolina Power & Light Company*, *Oak Ridge National Laboratory* e *Texas Electric Service Company*.

De certa forma, a ideia presente no conceito de *smart grids* faz um resgate dessa projeção quanto ao uso de tecnologias de processamento computacional e comunicação nos sistemas de distribuição.

No próximo capítulo será apresentado em maiores detalhes a ferramenta computacional utilizada para implementação do método de gerenciamento topológico dos sistemas elétricos de distribuição no contexto dos *smart grids*, os sistemas multiagentes.

4 Sistemas Multiagentes

4.1 Inteligência Artificial e Agentes

O campo de estudos sobre inteligência artificial é bastante recente quando comparado a outras ciências, além de ter um caráter fortemente interdisciplinar. Diversas áreas se dedicam ao estudo de algum aspecto da inteligência artificial: ela é objeto de apreciação de cientistas da computação, matemáticos, economistas, neurocientistas, psicólogos, cientistas sociais e outros (RUSSEL; NORVIG, 2004).

Talvez por conta desta característica interdisciplinar, definir de maneira unificada o termo "inteligência artificial" é uma tarefa bastante complicada, pois suscita diferentes questionamentos e debates. Inclusive há aqueles que defendem outra nomenclatura para a ciência, por exemplo, "inteligência computacional", "soft computing", e outras.

Apesar disso, é possível se trabalhar com a ideia de que os sistemas ditos "com inteligência artificial", ou sistemas inteligentes, são capazes de executar alguma função que, quando realizada pelo homem, exige conhecimento e raciocínio (REZENDE, 2003).

A exigência de conhecimento humano equivaleria, em termos computacionais, a uma base de conhecimento onde o sistema inteligente guardaria informações importantes sobre os processos que o mesmo modela, podendo manipulá-las à medida que o mesmo se defronta com situações que ocorrem em seu ambiente de aplicação. Já a ideia de raciocínio humano teria sua contraparte nos métodos de busca e tomada de decisão dos sistemas inteligentes.

Mesmo com algumas dificuldades em se trabalhar com conceitos unificados na área de inteligência artificial, uma abordagem didática da área, presente no livro de Russel e Norvig, vem ganhando muitos adeptos desde o final da década de 90. Ela unifica as diversas concepções da área a partir do conceito de agentes.

Um agente é uma entidade que percebe o ambiente em que está imerso e age sobre este

ambiente, afim de atingir algum objetivo (RUSSEL; NORVIG, 2004). Podemos visualizar um agente como um tipo específico de software ou como um robô, se o ambiente for o meio físico. A principal característica do agente é que ele age de forma autônoma, sem um controle executado por qualquer ente exterior a ele próprio.

Um agente inteligente possui as características de um sistema inteligente: uma base de conhecimentos que servirá para embasar a resolução do tipo de problema a qual o mesmo fora modelado, e métodos de busca e manipulação dessa base, que servirão para encontrar as informações que proverão a tomada de decisão da entidade.

O agente também é composto por duas partes que possibilitam a interação entre ele e o ambiente em que o mesmo se encontra. Os sensores são as ferramentas que o agente tem para perceber o ambiente; já os atuadores são as partes do agente que, de fato, executam a ação resultante da tomada de decisão feita a partir do estado que o agente modelou e da busca feita na base de conhecimentos.

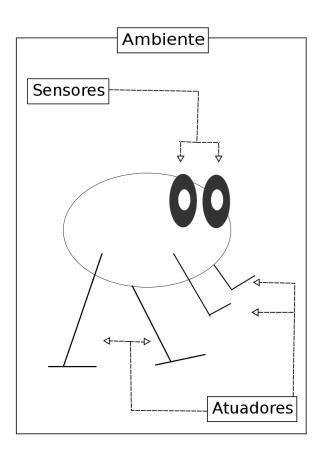


Figura 4.1: Componentes de interação entre um agente e o ambiente

A Figura 4.1 apresenta um rascunho de agente e os componentes que fazem a interface entre a entidade e o ambiente em que o mesmo está imerso. O sensores poderiam representar, dependendo do agente e do tipo de aplicação para que o mesmo foi projetado, visão compu-

tacional, sonar, medidores, etc. Para os atuadores, poderiam ser pernas ou braços mecânicos, funções que disparassem ações de controle sobre o sistema, dentre outras.

Normalmente, quando tratamos de inteligência artificial e sistemas inteligentes, nos referimos a um único sistema capaz de simular o processo de aprendizado e tomada de decisão humana para resolver algum problema. Entretanto, existe outra área de estudos e aplicações onde a dinâmica de aprendizado simulada é mais próxima da interação entre agentes do que propriamente da obtenção de conhecimento realizada por um agente apenas, isolado de outros sistemas inteligentes. Essa área é conhecida como inteligência artificial distribuída, e suas teorias e técnicas formam as principais bases para o desenvolvimento de sistemas multiagentes.

4.2 Inteligência Artificial Distribuída e Sistemas Multiagentes

Alguns problemas em que ferramentas computacionais são aplicadas para obtenção de uma solução são de natureza distribuída, seja na obtenção dos dados, no controle ou no cálculo necessário para o tratamento do mesmo.

Para estes casos, existe um domínio de estudos próprios, com teorias, práticas e técnicas desenvolvidas para lidar com estas características. A essa área dá-se o nome de sistemas distribuídos.

Pode-se caracterizar um sistema como do tipo distribuído se seus componentes estiverem localizados em uma rede de computadores, aos quais seus recursos estão compartilhados, seu processamento é feito de forma concorrente pelos componentes e existe coordenação para execução de uma dada tarefa (COULOURIS; DOLLIMORE; KINDBERG, 2005). O principal exemplo de sistema distribuído atual, e certamente um dos maiores existentes, são as aplicações que executam na internet, como a *web*.

Similarmente, há problemas cuja aplicação de sistemas inteligentes é mais propícia quando trabalhada em um conjunto de agentes, e não em apenas uma única entidade computacional que centraliza o processo de obtenção de dados, tomada de decisão ou ambos. Entretanto, trabalhar sistemas inteligentes em ambientes distribuídos é uma mudança de paradigma que exige ponderações teóricas diferentes e uma gama nova de técnicas para a implementação e obtenção de resultados satisfatórios para os mesmos.

A área de estudos que preocupa-se com a aplicação de sistemas inteligentes em ambientes

cuja característica distribuída, seja física ou funcional, é inerente, chama-se inteligência artificial distribuída (GARCIA; SICHMAN, 2003). Ela se utiliza de estudos nas áreas de inteligência artificial, ciência da computação, ciências sociais, economia, organização e gerenciamento para embasar seu corpo teórico (WEISS, 1999). Nesta especialidade, são modelados conjuntos de sistemas inteligentes que devem interagir entre si de alguma forma, afim de atingir um dado objetivo.

A questão da interação pode ser dividida, a grosso modo, entre cooperação ou competição. Sistemas que cooperam entre si buscam atingir um objetivo global compartilhado pelos seus componentes. Em modelos de interação competitivos, sistemas disputam um mesmo recurso ou cada componente tem um objetivo pelo qual, para ser atingido, deve prejudicar o desempenho do cumprimento de objetivo de outros agentes.

Voltando ao conceito de agente citado anteriormente, os sistemas multiagentes seriam sistemas em que temos um conjunto de agentes em um ambiente computacional distribuído que implementam as técnicas de inteligência artificial distribuída (WOOLDRIDGE, 2009). Logo, em um sistema multiagente, cada agente pode implementar técnicas de inteligência artificial, obtenção de dados, entre outras - a diferença é que, neste tipo de sistema, a interação entre os agentes é necessária para a aplicação ao qual o mesmo foi modelada.

Neste contexto, onde a interação é uma característica essencial do sistema, o aprendizado em sistemas multiagente a partir de uma visão macro do modelo, é deslocado para estas interações. Aqui, não há mais agentes que interagem apenas com o ambiente e o processo de aprendizado acontece a partir das situações que o mesmo encontra.

Para sistemas multiagentes, a troca de informações entre os agentes é também uma forma de aprendizado. Por exemplo, um sistema de controle gerenciado por vários agentes, quando entra em um estado instável, um agente poderia perceber a perturbação e enviar mensagens aos demais, para que ações fossem tomadas afim de estabilizar o controle novamente.

As mensagens entre os agentes são trocadas via redes de telecomunicações, como *ethernet* convencional, redes wi-fi, redes de celulares, etc (SHOHAM; LEYTON-BROWN, 2009). Os protocolos de transporte das mensagens são os comumente encontrados em sistemas *web*, como o HTTP e XMPP (KUROSE et al., 2007), MTP (BELLIFEMINE; CAIRE; GREENWOOD, 2007), entre outros. Quanto a estruturação das mensagens, a comunidade de desenvolvedores e pesquisadores de sistemas multiagentes definiu alguns protocolos padrões para esta finalidade, como KQML e FIPA (WOOLDRIDGE, 2009). Entretanto, outros protocolos comuns da área de desenvolvimento para *web* também podem ser utilizados, como XML (BRAY et al., 2000) e

JSON (CROCKFORD, 2006).

A Figura 4.2 apresenta um conjunto de agentes imersos em um mesmo ambiente. As setas entre os agentes representam as trocas de mensagens entre os mesmos, que é a maneira pela qual estes entes conseguem trocar informações e subsidiar a tomada de decisão distribuída.

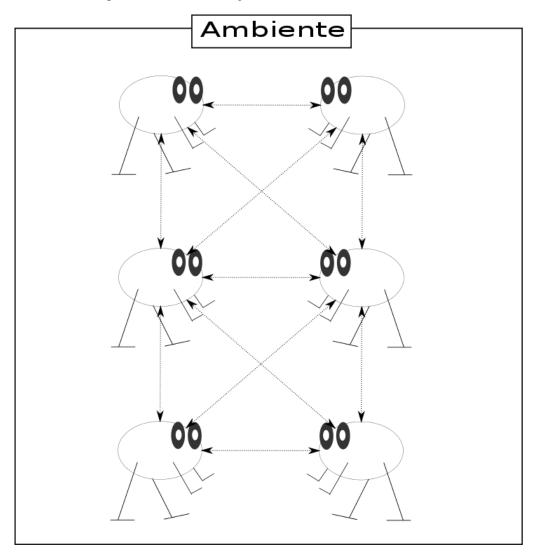


Figura 4.2: Agentes em um sistema multiagente e a representação de interações

Interações entre os agentes permitem a observação de comportamentos que "emergem" da sociedade de agentes. Decisões tomadas coletivamente a partir de análise e obtenção distribuída de dados reforça a característica de "aprendizado social" em sistemas multiagentes, ao contrário do aprendizado isolado sem interação inter-agentes dos sistemas inteligentes convencionais.

Inclusive, para Woodridge, a simulação da inteligência humana sem levar em conta as interações sociais é falha. Como seres humanos que vivem em comunidade, muito de nosso aprendizado e tomada de decisão constituem-se como fenômenos passíveis de influência ou intervenção externa ao homem que as executa. Em termos de agentes, o aprendizado híbrido

da inteligência artificial convencional com a proporcionada pela interação em ambientes multiagentes é mais próxima da observada em seres humanos que a aquisição de conhecimento ou tomada de decisão performada isoladamente. Por conta disso, o citado autor defende que a inteligência artificial seria uma subárea da inteligência artificial distribuída, pois esta última se preocupa com assuntos que englobam a inteligência artificial mais outras linhas de pesquisa (WOOLDRIDGE, 2009).

Dada a importância das maneiras como os agentes se relacionam, existem várias teorias e modelos organizacionais para mediar o processo de interação e tomada de decisão nos sistemas multiagentes. Entre alguns desses modelos, podemos citar (HORLING; LESSER, 2004):

- Hierarquia modelo centralizado onde os agentes colhem estímulos de seus sensores e enviam os dados para um agente central, que tratará de tomar a decisão e reenviar a resposta para os demais agentes do sistema;
- Holoarquia baseado na ideia de hólon (KOESTLER, 1969), cada grupo de agentes é
 responsável por uma secção de um dado nível de organização; o conjunto de secções de
 um mesmo nível forma um hólon que, agrupado a outros hólons, formam o próximo nível
 da pirâmide organizacional;
- Sociedade modelo organizacional em que se pretende observar o comportamento dos agentes diante de situações de dinâmica social, não necessariamente voltada para resolução de problemas. Normalmente implementada em sistemas abertos;
- Federação um dado agente é líder de um subgrupo de agentes. Todas as interações entre agentes de grupos diferentes devem ser realizadas sob mediação dos líderes, que também são responsáveis pela coordenação da execução de tarefas de seus respectivos times;
- Mercado aqui os agentes interagem a partir de regras de mercado, como oferta-eprocura. Cada agente tem uma função que pretende maximizar (lucros) e todas as interações são realizadas em termos de negociação;
- Modelos híbridos formas de organização que implementam características de uma ou mais outras formas organizacionais.

Para auxiliar na tarefa de implementação de sistemas multiagentes, existem várias linguagens de programação, bibliotecas e *frameworks* que encapsulam API's diversas que possibilitam

as interações entre os agentes, como acesso a redes de computadores, simulação de comportamento, programação *multithreading* (TANENBAUM, 2005) e outras características. Um interessante levantamento sobre estas tecnologias pode ser encontrado em (BORDINI et al., 2006).

Na área de sistemas elétricos de potência, a grande área onde este trabalho se insere, o uso de sistemas multiagentes já vem sendo trabalhado há alguns anos, apesar de muitas possibilidades ainda estarem abertas. Interessantes estudos sobre esta tecnologia em sistemas de potência, com detalhes teóricos e alguns exemplos de trabalhos na área podem ser encontrados em (MCARTHUR et al., 2007a, 2007b).

4.3 Smart Grids e Sistemas Multiagentes

Certamente, a quantidade de dados a ser adquirida e trabalhada em um sistema de distribuição é bastante grande. Além desta característica, os próprios dados estão dispersos no sistema, devendo ser adquiridos de forma distribuída e transmitidos entre os equipamentos do mesmo para embasar a tomada de decisão do sistema como um todo.

Manipular uma grande quantidade de dados pode inviabilizar a resolução do problema em tempo hábil. Métodos utilizados para superar esta dificuldade são normalmente técnicas de inteligência computacional para extração de conhecimento. Em um ambiente distribuído, os dados também estão distribuídos, sendo necessário sistemas especializados no tratamento de diferentes tipos de dados trabalhando paralelamente para a obtenção e processamento destes.

Após a fase de obtenção e pré-processamento dos dados, é chegado o momento de usar estas informações para a tomada de decisão. Novamente, técnicas de inteligência computacional deverão ser utilizadas para tratar o volume de dados que chegam de forma temporalmente assíncrona e de regiões geograficamente dispersas do sistema de distribuição. A decisão que o sistema terá que realizar é baseada nestes dados e visa atingir um objetivo específico, ao qual o sistema por completo fora previamente modelado para trabalhar.

Estas fases de obtenção e pré-processamento dos dados, tomada de decisão e execução, descritas anteriormente, são características dos sistemas ditos "inteligentes" (RUSSEL; NORVIG, 2004). Quando estes sistemas são empregados em ambientes distribuídos, temos que usufruir dos avanços da área e das técnicas de inteligência artificial distribuída, que encontram nos sistemas multiagentes seus principais representantes (WEISS, 1999).

A descrição das funcionalidades buscadas pelos smart grids revela um sistema de controle

fortemente distribuído sobre a topologia da rede, onde diversos equipamentos ficariam responsáveis pelo monitoramento (coleta de dados) e tomada de decisão sobre ações no sistema.

Por exemplo, uma das funcionalidades dos *smart grids* seria a auto-recuperação do sistema (BROWN, 2008), ou seja, a possibilidade da rede elétrica perceber a falta de energia em um dado conjunto de linhas de distribuição e, de maneira autônoma, reconfigurar-se para fazer com que o fluxo de carga atinja aquela região antes em falta. Nesta funcionalidade, é imperativo que equipamentos consigam perceber a falta de fluxo de carga em dadas linhas e, ao se depararem com esta situação, enviar para outros equipamentos da rede a informação da falta. Em seguida, os equipamentos responsáveis pela configuração do sistema deverão encontrar uma topologia que envie corrente de volta para a linha em falta (se possível), buscando atingir ainda algum outro objetivo, como redução das perdas.

Abordagem similar pode ser utilizada para todas as demais funcionalidades do *smart grid* que lidem com controle da rede, inclusive para o problema de gerenciamento da rede via reconfiguração da topologia. Seguindo o modelo anterior, equipamentos alocados no sistema poderiam medir a demanda em "tempo-real" e, dentro de uma janela de tempo, ativar o processo de reconfiguração da rede para atingir o objetivo modelado.

Buscando gerar simulações do comportamento e das condições técnicas de operação de um *smart grids*, há a necessidade de um método computacional que englobe sistemas distribuídos e tenha flexibilidade para implementação de técnicas de inteligência computacional, afim de que sejam testados algoritmos para as funcionalidades da rede que neste trabalho, em especial, será o gerenciamento da rede através da reconfiguração da topologia de distribuição.

Com estes requisitos postos, a utilização de sistemas multiagentes mostra-se uma decisão interessante. Os agentes poderiam simular os diversos equipamentos da rede de distribuição e suas relações, como numa sociedade de agentes, com comunicação e tomada de decisão emergindo a partir da iteração entre os mesmos (WOOLDRIDGE, 2009).

A Figura 4.3 apresenta um desenho esquemático de um sistema de distribuição, baseado em (CIVANLAR et al., 1988), com agentes dispersos geograficamente nos componentes deste sistema, tanto nas cargas, alimentadores e chaves, simbolizados respectivamente por "•", "\"
" " "."

Os agentes são flexíveis ao ponto de servirem como "metáfora" do ente que eles simulam. Por exemplo, um agente que simula um equipamento de medição de consumo - o software poderia simular o processo através da leitura de um arquivo texto, ou através da "escuta" de

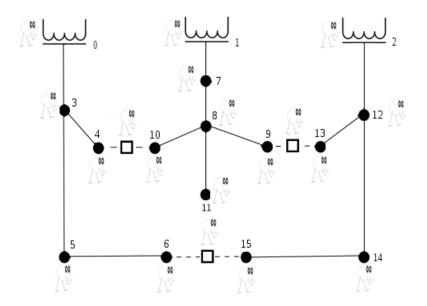


Figura 4.3: Agentes dispersos em um sistema elétrico de distribuição

uma porta de rede de computador com mensagens sendo transmitidas por um servidor. Quando houvesse variação no consumo, uma mensagem seria enviada ao agente que atualizaria o valor que ele controla. Quando a variação atingisse certa taxa, o agente poderia informar os demais agentes que, em conjunto, avaliariam se existiria outra topologia para o sistema de distribuição que garantisse a entrega de energia para aquele momento.

Assim, serão simulados *smart grids* através de sistemas multiagentes objetivando testar algoritmos computacionais aplicados ao problema de gerenciamento do sistema elétrico de distribuição nesse novo cenário, onde a aquisição de dados é distribuída, o transporte de informações e iteração entre os equipamentos é condição necessária para o bom funcionamento do sistema e o uso de ferramentas de inteligência computacional é indispensável.

O próximo capítulo apresenta o método desenvolvido e sua aplicação, bem como estudos preliminares realizados para criação desta abordagem.

5 Simulação de Sistemas de Distribuição através de Sistemas Multiagentes

5.1 Proposta de Simulação de Sistemas de Distribuição através de Sistemas Multiagentes

Dado o alto investimento e esforço em termos de pesquisa e inovação tecnológica em várias universidades e centros de pesquisa de diversos países, a implantação dos sistemas elétricos de nova geração, os ditos *smart grids*, apresenta-se como iminente em alguns anos.

De fato, como relatado neste trabalho, já existem algumas experiências em sistemas de pequeno ou médio porte, como por exemplo em Portugal, na cidade de Évora. A medida que novas técnicas e potenciais aplicações para *smart grids* forem surgindo, certamente as mesmas serão incorporadas a sistemas de distribuição já em produção, para avaliação de suas possibilidades no ambiente real de funcionamento.

Entretanto, antes da instalação de *smart grids* em ambiente de produção, mesmo em sistemas de pequena escala, é necessário que simulações e testes de suas características sejam realizados anteriormente. Isso possibilita um melhor estudo das formas de execução de um dado conjunto de ações, análises comparativas entre métodos diferentes para uma mesma tarefa e mais, tudo isso a um baixo custo, pois o ambiente destas simulações seria computacional.

Algumas linhas de atuação da área de ciência da computação trabalham simulações computacionais de problemas que existem no mundo real. Por exemplo, a otimização de sistemas, com suas aplicações de métodos exatos ou heurísticos, é uma delas. Entretanto, no jargão coloquial, ao referir-se a uma aplicação da área, convencionou-se chamar de "resolução do problema X por meio do método Y".

A rigor, em termos de uma "epistemologia da computação", o que fora realizado foi a simulação computacional do problema X e a resolução desta simulação pelo método Y. Para se criar um ambiente de simulação computacional, o cientista da computação deve abstrair os dados do problema que será tratado e modelá-lo em termos matemáticos, que é o objeto de análise e cálculo do computador.

Portanto, falar em resolução de um problema do mundo real a partir da modelagem e simulação computacional, é na verdade dizer que os métodos implementados resolveram satisfatoriamente a simulação do modelo do problema, com seus dados e características abstraídos para tratamento computacional. Nesta perspectiva, se o problema atacado for um problema do mundo real, a resolução da simulação do modelo permite ao autor averiguar se aquela técnica ou resultado é interessante ou não para ser transplantada do "mundo virtual" para o "mundo real".

Da exposição, torna-se possível afirmar que a simulação computacional de *smart grids* é uma frente de pesquisa que se insere nesta metodologia de simulações já presente e bastante trabalhada na ciência da computação.

Interessante também comentar sobre a decisão quanto a utilização de sistemas multiagentes para este trabalho. No Capítulo 2 dissertou-se sobre as similaridades entre *smart grids* e sistemas distribuídos. Aquelas semelhanças justificam a opção pela tecnologia de sistemas multiagentes, pois a mesma tem potencial para abstrair um modelo de *smart grids* o mais próximo possível do produto final que pretende-se levar ao ambiente de produção, por sua característica distribuída e também pela ênfase na automação e tomada de decisão.

Existem várias ferramentas computacionais que podem ser usadas para simulação de técnicas ou buscas de solução. Há métodos exatos, heurísticos, metaheurísticos, inteligência artificial, sistemas distribuídos, e outros. O que cabe ao cientista da computação é selecionar aquela que melhor se adapta ao modelo abstraído do problema que se está trabalhando, tentando tornar a simulação a mais verossímil possível.

A busca pela simulação mais próxima do problema real é importante porque ela garante que dificuldades e desafios que existem no problema e se fazem presentes na simulação, caso contornados ou tratados nela, também servirão para o ambiente de produção.

Smart grids são redes formadas por equipamentos de monitoramento e controle da rede, que permitem um nível de automação da operação do sistema de distribuição em tópicos como reconfiguração do sistema, minimização das perdas, disponibilidade para gerenciamento em

larga escala de geradores elétricos distribuídos, etc. Estas funcionalidades são possíveis devido a comunicação que ocorre entre estes equipamentos, pois os mesmos estão distribuídos ao longo da topologia do sistema elétrico.

Mapear um *smart grid* através de sistemas multiagentes mostra-se uma boa escolha pois estes últimos são capazes de emular as características das redes elétricas inteligentes enunciadas acima. Pode-se criar um agente para cada funcionalidade específica da rede elétrica - medidor, chaveador, entre outros. A comunicação via rede de computadores e composição de mensagens entre os agentes simula a comunicação entre os equipamentos do *smart grid*. E por fim, a tomada de decisão distribuída no sistema multiagente tem sua contraparte no controle distribuído do sistema elétrico.

Assim, com um ambiente computacional próximo ao ambiente real ideal¹, podem-se fazer testes de técnicas existentes, desenvolver novas, fazer estudos comparativos entre técnicas diferentes entre outros estudos. A base para simulação permite desenvolver a tecnologia da área a partir destes tipos de estudos.

Entre as muitas funcionalidades dos *smart grids*, uma que chama atenção pelo potencial de estudos no campo de sistema multiagentes e tomada de decisão distribuída é a reconfiguração topológica do sistema de distribuição. Como o *smart grid* é formado por vários equipamentos distribuídos ao longo do sistema de distribuição, estudos que viabilizem a tomada de decisão de forma distribuída acerca da reconfiguração do sistema contribuirá, de forma fundamental, para a iniciativa de controle distribuído buscada pelos visionários desta tecnologia.

Logo, acreditamos que a aplicação de sistemas multiagentes para a simulação da reconfiguração topológica de um sistema de distribuição do tipo *smart grids* é uma importante contribuição para o desenvolvimento das tecnologias relacionadas à nova geração dos sistemas elétricos de potência, assim como abre possibilidades de vários estudos e desenvolvimento de algoritmos próprios para o desempenho das funcionalidades que se esperam dos sistemas elétricos do futuro.

¹De fato, mesmo buscando conferir ao ambiente de simulação a maior semelhança possível, é impraticável reproduzi-lo exatamente igual ao ambiente de produção. Ainda assim, quanto mais próximo conseguirmos simulálo, mais condizente com a realidade ele será e os trabalhos feitos sobre este ambiente terão mais chance de serem transpostos e executados com sucesso no ambiente de produção.

5.2 Algoritmo de Dijkstra Distribuído através de Sistemas Multiagentes

Na busca por maneiras de trabalhar a tomada de decisão distribuída, optamos por fazer um estudo preliminar acerca da distribuição do cálculo do algoritmo de Dijkstra (RABUSKE, 1992). Existem alguns problemas específicos, como para algumas aplicações deste algoritmo, cuja a resolução pode se beneficiar dos avanços atuais da computação distribuída, onde tanto o processamento quanto a própria obtenção dos dados para o cálculo se dá em um ambiente descentralizado, disperso geograficamente. Os atuais ambientes de computação modernos são distribuídos e de grande porte. Estes ambientes permitem tanto o balanceamento da carga do processamento quanto o acompanhamento descentralizado do estado do sistema modelado.

Estes ambientes necessitam de algoritmos adaptados a suas características. Somente assim será possível usufruir as vantagens dos ambientes distribuídos, estudar e averiguar suas possibilidades, e contribuir com o amadurecimento em pesquisas desse campo. Com esta justificativa, propõem-se uma adaptação do algoritmo de Dijkstra, distribuindo-o para execução no modelo de grafo, através do uso de sistemas multiagentes. O objetivo é que a proposta consiga solucionar os problemas para topologias de grafo estáticas com a mesma qualidade que o algoritmo de Dijkstra convencional consegue atingir.

O algoritmo de Dijkstra é um clássico algoritmo computacional para o cálculo da distância mínima entre um vértice e todos os demais vértices de um grafo. Foi desenvolvido pelo premiado² matemático holandês Edsger Dijkstra (1930 - 2002) e publicado na edição número 1 do periódico *Numerical Mathematics*, em 1959 (DIJKSTRA, 1959).

Define-se um grafo $G = \{V, E\}$, onde V é o conjunto dos vértices e E o conjunto das arestas; w será a função custo de cada aresta, onde $\forall w(u, v) >= 0$, tal que a aresta $(u, v) \in E$, $\forall u, v \in V$. Tem-se também um vértice v_i que será o vértice-fonte ao qual se quer calcular a distância mínima entre ele e todos os demais vértices do grafo (os chamados vértices-alvo).

Durante a execução do algoritmo de Dijkstra, cada vértice terá uma variável chamada custo relativo (que simbolizaremos como Q_i , onde 1 <= i <= |V|) que será a distância total, naquela iteração, entre este vértice e o vértice-fonte. Estes custos começarão com um valor tão grande quanto possível³, exceto o custo relativo do vértice-fonte que será 0. Também haverá para cada vértice uma variável chamada vértice anterior (que chamaremos de A_i , onde 1 <= i <= |V|), a

²Dijkstra foi laureado em 1972 com o Prêmio Turing, considerado o "prêmio Nobel da computação"

³Utilizaremos neste texto um "valor infinito" para este propósito.

qual guardará a referência para o vértice em que está ligada naquela iteração.

O algoritmo manterá também um conjunto de vértices S que indicará os vértices que já tiveram suas vizinhanças analisadas pelo método. Durante a execução, o algoritmo selecionará o vértice $u \in V - S$ com o menor custo relativo naquela iteração para ter a sua vizinhança analisada, até que o critério S = V seja atingido, indicando que todos os vértices foram avaliados e o valor da distância mínima entre o vértice-fonte e cada vértice-alvo do grafo foi obtido (CORMEN et al., 2009).

Abaixo, no Algoritmo 5.1 temos o algoritmo de Dijkstra representado em forma de pseudocódigo:

```
Algoritmo 5.1: Algoritmo de Dijkstra
```

```
Entrada: G, v_i
 1: Q_i = \text{Inicializar-Custos-Relativos}(G), \forall i
 2: A_i = \text{Inicializar-no-Anterior}(), \forall i
 3: S = \emptyset
 4: R = \emptyset
 5: D = \emptyset
 6: temp = 0
 7: noAvaliado = v_i
 8: enquanto S \neq V faça
       Q_i = \text{Calc-Custo-Relativo}(G, noAvaliado), \forall i
       S = S \bigcup noAvaliado
10:
       temp = noAvaliado
11:
       noAvaliado = Minimo((V - S), Q)
12:
13:
       A_{noAvaliado} = temp
14: fim enquanto
15: para todo v \in V faça
       D = D \bigcup (v, A_v)
17: fim para
18: R = (V, D)
19: retorne R
```

De forma descritiva, podemos visualizar o funcionamento do algoritmo de Dijkstra a partir da seguinte sequencia de operações. Passando como argumento o grafo G e o vértice-fonte v_i pertencente ao grafo, inicializam-se variáveis chamadas custos relativos (Q_i) para cada vértice do grafo com o valor ∞ , excetuando-se o custo relativo do vértice v_i que será igual a 0. No algoritmo, este passo corresponde à chamada de função Inicializar-Custos-Relativos, na linha 1. Inicializam-se também as variáveis A_i durante a chamada da função Inicializar-no-Anterior, linha 2, sendo $A_{v_i} = 0$ e para os demais casos um valor que represente a falta de um antecessor. Na Figura 5.1 (a), o vértice v1 terá sua vizinhança avaliada. Em todas as figuras aqui

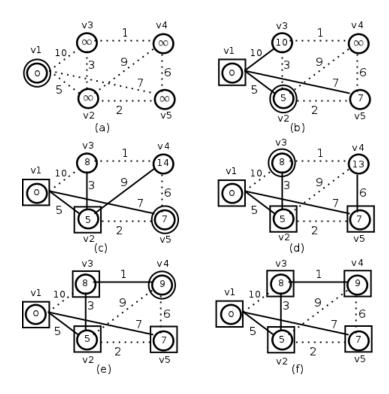


Figura 5.1: Iterações do algoritmo de Dijkstra. O vértice marcado com dois círculos será aquele que terá a vizinhança avaliada naquela iteração; os marcados com um retângulo serão aqueles que já tiveram sua vizinhança avaliada.

apresentadas, os vértices que estiverem circulados representarão aqueles que serão avaliados nesta iteração. As arestas pontilhadas representam as ligações entre os vértices, já as contínuas significam aquelas que formam a solução para aquela iteração.

O algoritmo então fará uma busca na vizinhança de v_i , atualizando o custo relativo de todos os vértices diretamente conectados ao vértice v_i com o respectivo valor da aresta que cria a ligação entre estes dois vértices somado ao valor do custo relativo do vértice v_i . Este passo é simbolizado pela chamada de função Calc-Custo-Relativo. Na Figura 5.1 (b), a verificação da vizinhança de v1 atualizou os custos relativos de v2, v3 e v5.

Agora, marca-se o vértice v_i como um vértice que já teve sua vizinhança avaliada e executase outra iteração de avaliação de vizinhança, desta vez no vértice com menor custo relativo atual. Utilizaremos um retângulo nos vértices que já tiveram suas vizinhanças avaliadas.

Caso durante a iteração algum vértice vizinho ao vértice avaliado tenha seu custo relativo atual maior que a soma da aresta que o conecta mais o custo relativo do vértice avaliado, o custo relativo deste vértice vizinho será então atualizado para o valor dessa soma. Isso acontece, por exemplo, com o nó v3 onde, na Figura 5.1 (b) o valor de seu custo relativo é 10 e na Figura 5.1 (c) é 8. Também ocorre o mesmo com o vértice v4 quando o valor do seu custo relativo muda

de 14 para 13 nas Figuras 5.1 (c) e 5.1 (d) e finalmente assume o valor 9 a partir da Figura 5.1 (e).

O algoritmo de Dijkstra executa estas iterações até que todos os vértices $v \in V$ do grafo tenham sido selecionados para terem suas vizinhanças avaliadas. Neste momento, temos o caminho mínimo entre o vértice-fonte e todos os demais vértices do grafo. No exemplo, a Figura 5.1 (e) é o momento em que o nó v5 é selecionado para avaliação. A Figura seguinte, 5.1 (f), representa a solução para o problema nesta instância já com a vizinhança do vértice v5 avaliada.

A instrução que começa na linha 15 do Algoritmo 5.1 indica que, neste momento, serão colocados no conjunto D as arestas que ligam um vértice v à seu antecessor A_v , $\forall v \in V$. Em seguida, na linha 18 é construído o grafo R que será a resposta para o problema, a partir do conjunto de vértices V e de arestas D.

A implementação distribuída do algoritmo de Dijkstra é realizada a partir do conceito e tecnologia de sistemas multiagentes. A partir da topologia do grafo, aloca-se um agente em cada vértice do mesmo. Como o algoritmo de Dijkstra necessita de um vértice-fonte, utilizou-se um modelo organizacional hierárquico dos agentes (HORLING; LESSER, 2004) onde podemos classificá-los em dois tipos: o agente ativo que demanda processamento para os demais agentes é o que está alocado no vértice-fonte; e os demais agentes são reativos, recebendo demandas de processamento do agente no vértice-fonte e respondendo aquilo que é pedido.

Cada agente tem uma representação completa do grafo em sua memória, que pode ser atualizada pelo envio de mensagens do agente no vértice-fonte caso aconteça a inserção de um novo vértice ou novas arestas no grafo. Esta funcionalidade provê ao sistema a possibilidade de tratar cenários dinâmicos, onde a topologia do grafo pode mudar com o decorrer do tempo.

Assim que se iniciam os agentes, o agente no vértice-fonte envia uma mensagem para todos os demais agentes requerendo que os mesmos calculem o menor caminho entre o vértice-fonte e o vértice onde se encontra o agente que recebeu a mensagem. Este passo corresponde à chamada da função Enviar-Grafo-Para-Agentes, no Algoritmo 5.2, onde M representa o conjunto de todos os agentes alocados nos vértices-alvo do grafo. Neste caso, o agente do vértice-fonte tanto atualiza a estrutura do grafo nos agentes dos demais vértices como faz o requerimento para o cálculo do menor caminho. Em seguida, o agente receptor da mensagem executa o algoritmo de Dijkstra, iniciando a análise de vizinhança a partir do vértice-fonte. Os passos desta implementação estão explicitados no Algoritmo 5.3.

Algoritmo 5.2: Ações do Agente Ativo

Entrada: G, M

1: Enviar-Grafo-Para-Agentes(G, M)

2: $N = \emptyset$

3: $R = \emptyset$

4: enquanto $N \neq M$ faça

5: $R = R \cup \text{Resposta-Mensagem-Grafo}()$

6: $N = N \cup \text{Resposta-Mensagem-Emissor}()$

7: fim enquanto

8: retorne R

O critério de parada do método, neste caso, diferentemente do Algoritmo 5.1, ocorre quando o vértice a ser avaliado é o vértice onde se encontra o agente. Neste momento, pode ocorrer da árvore com o caminho entre os dois vértices conter vértices desnecessários, que serviriam para ligar outros vértices ao vértice-fonte.

Para tratar isto, cada vértice do grafo terá uma variável própria que referenciará o vértice com o qual se está ligada naquela iteração, o A_i . Assim, ao atingir o critério de parada, basta-se percorrer os A_i 's a partir do vértice-alvo. O caminho será então obtido, sem vértices desnecessários. Este passo corresponde à instrução que começa a partir da linha 17 do Algoritmo 5.3. Nela, o conjunto W que representa os vértices do menor caminho entre os nós v_i (vértice-fonte) e v_j (vértice-alvo onde se encontra o agente) começa a adicionar os vértices anteriores A_{temp} a partir do vértice v_j . Também, o conjunto D adiciona as arestas que conectam (v, A_{temp}) .

Ao atingir o critério de parada da linha 17 do Algoritmo 5.3, o grafo R formado por (W, D) na linha 22 terá o caminho mínimo entre o vértice-fonte v_i e o vértice-alvo v_j . A Figura 5.2 (a) apresenta o resultado da aplicação do algoritmo de Dijkstra onde $v_i = v1$ e $v_j = v3$. Na Figura 5.2 (b), temos o caminho mínimo entre v1 e v3 após ser retirado os vértices desnecessários para a resposta, no caso, v4 e v5.

Com a resposta R obtida, o agente no vértice-alvo a envia para o agente no vértice-fonte, que está esperando as respostas chegarem. Este agente, então, vai sobrepondo cada resposta enviada por cada agente, até que a resposta completa para o grafo como um todo seja obtida. Esta operação se dá na linha 5 do Algoritmo 5.2, onde o resultado da função Resposta-Mensagem-Grafo é o caminho mínimo entre v_i e algum v_j . Na linha seguinte, o agente marca os demais agentes que já responderam à requisição.

Após todos enviarem suas contribuições, o agente no vértice-fonte termina a execução do algoritmo e o *R* resultante representa a solução do problema completo.

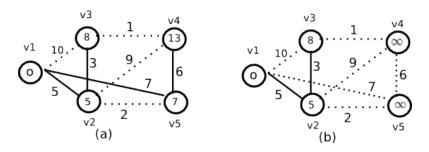


Figura 5.2: (a) Resultado da aplicação do algoritmo de Dijkstra para $v_i = v1$ e $v_j = v3$. (b) Caminho mínimo entre v1 e v3.

Algoritmo 5.3: Algoritmo de Dijkstra no Agente Reativo

```
Entrada: G, v_i, v_j
 1: Q_i = \text{Inicializar-Custos-Relativos}(G), \forall i
 2: A_i = Inicializar-no-Anterior(), \forall i
 3: S = \emptyset
 4: R = \emptyset
 5: D = \emptyset
 6: W = \bigcup v_i, v_j
 7: temp = 0
 8: noAvaliado = v_i
 9: enquanto noAvaliado \neq v_i faça
       Q = \text{Calc-Custo-Relativo}(G, noAvaliado)
10:
       S = S \bigcup noAvaliado
11:
       temp = noAvaliado
12:
       noAvaliado = Minimo((V - S), Q)
13:
14:
       A_{noAvaliado} = temp
15: fim enquanto
16: temp = v_i
17: enquanto temp \neq v_i faça
18:
       W = W \bigcup A_{temp}
       D = D \bigcup (temp, A_{temp})
19:
       temp = A_{temp}
20:
21: fim enquanto
22: R = (W, D)
```

23: retorne R

Grafos de Teste				
Instância	V	E	S. Ótima	S. Encontrada
1	6	10	43	43
2	6	12	25	25
3	8	12	20	20
4	9	15	701	701
5	10	19	253	253
6	12	23	34	34
7	14	22	703	703
8	16	33	23	23
9	16	33	29	29
10	48	77	858	858

Tabela 5.1: Instâncias de teste do Algoritmo de Dijkstra distribuído.

Este método foi testado em um conjunto de grafos que variam de 6 vértices e 10 arestas até 48 vértices e 77 arestas. Estes exemplos foram retirados do *website* do professor Ph.D. Kenji Ikeda da Universidade de Tokushima, Japão⁴. Na Tabela 5.2, a coluna V indica o número de vértices da instância; E o número de arestas, S. Ótima a solução ótima da instância e S. Encontrada a solução encontrada pelo algoritmo distribuído de Dijkstra.

Percebe-se que, para todas as instâncias, o resultado encontrado pelo algoritmo distribuído de Dijkstra atingiu o valor da solução ótima.

Apesar dos resultados positivos, é importante ressaltar que a principal contribuição desta abordagem reside no fato da adição da característica de processamento distribuído ao sistema. Isso permitiria que o sistema modelado pudesse lidar com topologias dinâmicas, distribuísse o processamento e possibilitasse um controle distribuído sobre o mesmo.

Este método e os resultados obtidos foram descritos em forma de artigo e apresentados em 2011 durante o X Simpósio Brasileiro de Automação Inteligente (X SBAI), em São João del-Rei, Minas Gerais (SARAIVA; SOUSA; ASADA, 2011).

⁴Disponível em http://www-b2.is.tokushima-u.ac.jp/~ikeda/suuri/dijkstra/Dijkstra.shtml, acessado dia 4 de março de 2011

5.3 Modelando Controle Distribuído em Sistemas de Distribuíção via Sistemas Multiagentes

A principal característica dos *smart grids* que permitirá a implementação das novas funcionalidades no sistema elétrico é a possibilidade de comunicação entre os equipamentos. Por exemplo, os medidores de consumo instalados nas cargas do sistema devem enviar dados sobre o consumo e outras medidas para um sistema alojado na central de distribuição afim de que estas informações possam subsidiar funcionalidades como a reconfiguração do sistema, autorecuperação, entre outras.

Podemos modelar agentes em um sistema multiagente para simular alguma destas funcionalidades. Existem várias linguagens e *frameworks* para sistemas multiagente (BORDINI et al., 2006) que podem ser utilizados para a implementação, mas como neste trabalho nos interessa tão somente o algoritmo em si, utilizaremos pseudocódigo para exemplificar o modelo.

Todo agente em um sistema multiagente tem um identificador único (BELLIFEMINE; CAIRE; GREENWOOD, 2007), que serve como uma espécie de endereço (WEISS, 1999) para recebimento de mensagens. Isso permite que os agentes troquem as mensagens entre si de forma correta, sempre enviando a mensagem para um agente ou para um conjunto de agentes que se deseja proceder a comunicação. Na modelagem aqui apresentada teremos os agentes que ficam nas cargas de um sistema elétrico de distribuição chamados de loaderAgent'n', onde n = 1, 2, 3, ..., k, sendo k o número total de cargas do sistema.

Assim, serão representados os agentes *loaderAgent1*, *loaderAgent2*, *loaderAgent3*, ..., *loaderAgent'k'*. O agente que representará o equipamento na central do sistema de distribuição será chamado de *feederAgent*.

O feederAgent deverá conhecer de antemão quantos agentes existem nas cargas da rede elétrica e qual é a atual configuração topológica do sistema de distribuição. Isso permitirá que ele identifique facilmente qual agente está em qual carga, uma informação importante para algumas funcionalidades.

O feederAgent receberá mensagens com as medições feitas pelos agentes loaderAgents nas cargas do sistema. Sistemas multiagentes, por herdarem características dos sistemas distribuídos, são sistemas assíncronos (WOOLDRIDGE, 2009; COULOURIS; DOLLIMORE; KINDBERG, 2005) por conta da dificuldade em se definir um horário global comum para os componentes. Logo, o tratamento das ações a serem realizadas pelos agentes deve levar esta

característica em consideração.

Pode-se programar o feederAgent para definir um tempo máximo t em que ele espera os dados enviados de cada agente entre duas mensagens consecutivas. Esse tempo t deve ser um intervalo suficiente para receber as mensagens corretamente. Caso algum agente demore mais tempo que t para enviar uma mensagem, o feederAgent poderá enviar um alerta para a equipe de manutenção da empresa, apontando que há algo anômalo no setor onde reside o agente possivelmente defeituoso.

As operações do feederAgent, de acordo com as funcionalidades já comentadas nesta seção, são apresentadas no Algoritmo 5.4.

```
Algoritmo 5.4: Algoritmo do feederAgent
```

```
1: Definir t
 2: Carrega topologia do sistema
 3: Carrega alocação de loaderAgents
 4: N \leftarrow \text{número de } loaderAgents
 5: M \leftarrow 0
 6: enquanto 1 = 1 faça
      enquanto ((M \neq N)) ou (\neg t) faça
 7:
         se Recebeu mensagem então
 8:
            Trata mensagem recebida
 9:
            Identifica autor da mensagem
10:
11:
           M \leftarrow M + 1
            Reinicia contagem de t
12:
         fim se
13:
14:
      fim enquanto
      se t então
15:
         Verifica agente que não enviou mensagem
16:
         Emite um alerta sobre a anomalia
17:
18:
      fim se
      M \leftarrow 0
20: fim enquanto
21: retorne
```

No Algoritmo 5.4, a estrutura "enquanto" da linha 6 significa que o feederAgent executará suas operações indefinidamente, pois o mesmo está em um laço infinito. N é o número de loaderAgents e M o número de mensagens recebidas. Na estrutura "enquanto" da linha 7, o critério de parada ocorrerá quando o *feederAgent* receber todas as mensagens de todos os loaderAgents, ou quando ele atingir o tempo limite t de espera de mensagens. Caso o feederAgent tenha atingido o tempo limite de espera, significa que algum ou alguns loaderAgents não enviaram mensagens.

Para as mensagens recebidas, o feederAgent faz o tratamento necessário para a funcionalidade ao qual foi implementado, identifica o autor da mensagem, incrementa número de mensagens recebidas M e reinicia a contagem de t.

Caso a estrutura "enquanto" seja finalizada por conta do tempo limite t ter sido atingido, a estrutura "se" da linha 15 faz o tratamento necessário, emitindo alertas sobre quais loaderAgents estão com problemas. Se a finalização for motivada pelo número de mensagens ter completado um ciclo (todos os loaderAgents enviaram mensagens), a variável M recebe o valor 0 novamente (linha 19) e o laço de operações do feederAgent se repete.

Agora, sobre os agentes nas cargas, eles devem fazer as medições que são responsáveis e enviá-las para o agente na central de distribuição.

Os *loaderAgent*s são inicializados sabendo qual o nome do agente na central de distribuição responsável por receber as informações. Já em execução, cada *loaderAgent* faz a medição pela qual é responsável a cada intervalo de tempo *e*, e envia para a central. A leitura recorrente dentro de um intervalo é interessante pois possibilita o acompanhamento do estado do sistema. O Algoritmo 5.5 apresenta os passos a serem executados pelos agentes nas cargas.

Algoritmo 5.5: Algoritmo dos *loaderAgents*

- 1: Definir e
- 2: Carrega nome do agente na central
- 3: $i \leftarrow \text{Medição do tempo}$
- 4: $u \leftarrow 0$
- 5: enquanto 1 = 1 faça
- 6: $u \leftarrow \text{Medição do tempo}$
- 7: se $u i \ge e$ então
- 8: Faz medições elétricas
- 9: Envia mensagem com medições para central
- 10: $i \leftarrow u$
- 11: **fim se**
- 12: fim enquanto
- 13: retorne

No Algoritmo 5.5, linha 3, a variável *i* recebe o valor da medição do tempo no momento em que o agente é inicializado. Na estrutura "enquanto" da linha 5, existe um *loop* infinito que significa que o *loaderAgent* sempre executará as instruções nesta estrutura, de forma cíclica, indefinidamente.

Na estrutura "enquanto", na linha 6 do Algoritmo 5.5, a variável *u* recebe a medição do tempo naquele momento, que em seguida será subtraído por *i* e comparado ao intervalo de ações

do agente *e* na linha 7. Se a diferença entre *u* e *i* for maior ou igual a *e*, o agente executa as ações de realizar as medições elétricas correspondentes a sua função e enviá-las ao agente na central de distribuição. Em seguida, a variável *i* recebe o valor de *u*, o que zera a contagem de tempo do intervalo para executar novamente as funções de medição elétrica e envio de mensagens.

Os algoritmos apresentados nesta seção, quando implementados utilizando ferramentas próprias para o desenvolvimento de sistemas multiagentes (BORDINI et al., 2006), conseguem simular a troca de informações entre sistemas computacionais distribuídos, algo que se espera de trabalhos de simulação de redes elétricas inteligentes.

O que fazer com os dados compartilhados entre os agentes é algo que o projetista deverá ponderar, a partir da funcionalidade de *smart grid* que pretende-se simular. Dependendo da funcionalidade, métodos de inteligência artificial podem ser aplicados e, a partir dos resultados obtidos, novos algoritmos de comunicação no sistema deverão ser implementados para proceder a reconfiguração do mesmo. No exemplo aqui apresentado, o *feederAgent* avisa uma equipe técnica de manutenção quando algum *loaderAgent* não envia mensagens no tempo adequado.

Esta abordagem, assim como a justificativa teórica e relação entre sistemas distribuídos, sistemas multiagentes e *smart grids*, foram descritas em artigo submetido ao IV Simpósio Brasileiro de Sistemas Elétricos (IV SBSE), organizado pela Sociedade Brasileira de Automática, e encontra-se em fase de avaliação pelo comitê técnico-científico do evento.

5.4 Reconfiguração de Sistemas de Distribuição tipo *Smart Grids* através de Sistemas Multiagentes

Com base nos trabalhos preliminares apresentados anteriormente, esta seção abordará a implementação de agentes em um sistema multiagente capazes de prover um controle distribuído sobre o sistema elétrico de distribuição. São elencados dois objetivos: primeiro, verificar que a troca de informações entre os agentes é possível, pois a mesma é essencial para a implementação das funcionalidades prometidas pelos *smart grids*; segundo, modelar e implementar um algoritmo que permite aos agentes reconfigurarem a topologia do sistema de distribuição.

5.4.1 Modelagem do Sistema Multiagente

No sistema há 4 tipos de agentes: os *feederAgents*, responsáveis por monitorarem os alimentadores do sistema; os *switcherAgents*, que verificam as chaves do sistema e seus estados; os *loaderAgents*, responsáveis por verificar as demandas das cargas do sistema e também por realizarem alguns cálculos; e finalmente o *demandChanged*, agente que serve apenas par enviar mensagens aos *loaderAgents* que terão sua demanda alterada no decorrer da simulação computacional, conferindo uma característica dinâmica ao sistema. Destes quatro tipos de agentes, os três primeiros são implementados baseados num agente básico que tem as principais funcionalidades de um agente para sistemas elétricos, o *electricalAgent*.

Por conta do *demandChanged* ser um agente que apenas envia mensagens, e não emula qualquer equipamento do sistema elétrico, ele não será detalhado nesta seção.

A Figura 5.3 apresenta um rascunho do diagrama de classes (BOOCH; RUMBAUGH; JACOBSON, 2006; PÁDUA, 2003) para os três agentes do sistema que herdam da classe *electricalAgent*.

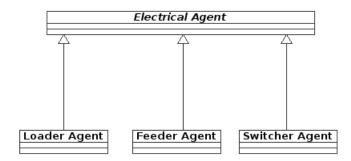


Figura 5.3: Diagrama de classe dos principais agentes do sistema

Já a Figura 5.4 apresenta o diagrama de classe um pouco mais específico do *electricalA-gent*. Este agente nunca será instanciado - ele serve apenas como base para os demais agentes já comentados. As funcionalidades (*comportamentos*) implementadas no *electricalAgent* são funcionalidades compartilhadas pelos demais agentes do sistema, como funções para trabalhar a representação do grafo do sistema elétrico de distribuição, as definições de coalizão e caminho elétrico, uma classe para escrever *logs* da execução do agente durante seu tempo de vida, além de enumerações que definem os tipos de agente, os protocolos de mensagem e os estados de um *switcherAgent*.

Os protocolos de mensagens são de fundamental importância para o projeto de sistemas multiagentes. É necessário que os diferentes agentes que compõem o sistema consigam "se en-

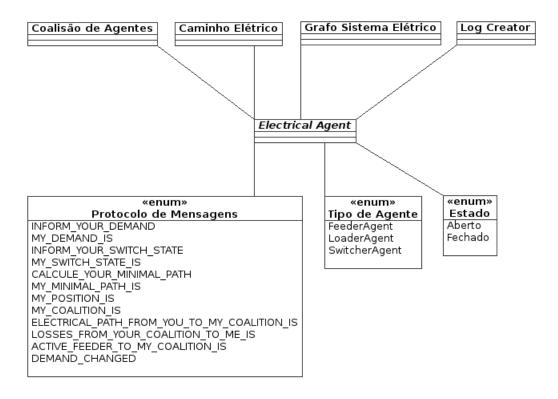


Figura 5.4: Diagrama de classe para o Electrical Agent

tender" entre si, ou seja, uma mensagem enviada por um agente deve ser passível de tratamento pelo agente que a recebe.

A definição de mensagens FIPA (*Foundation for Intelligent Physical Agents*⁵), utilizado na implementação deste trabalho, especifica que cada mensagem é composta por um campo de protocolo, que servirá pra identificar sobre o que se trata aquela mensagem (WOOLDRIDGE, 2009; BORDINI et al., 2006). Com o protocolo configurado, um agente poderá dar o tratamento adequado às informações daquela mensagem, a partir das suas funcionalidades.

A Tabela 5.2 identifica os protocolos definidos para a comunicação dos agentes implementados neste trabalho, já apresentados na enumeração Protocolo de Mensagens da Figura 5.4, bem como apresenta as funcionalidades às quais eles se referem.

As classes Coalizão de Agentes e Caminho Elétrico, que aparecem na Figura 5.4, são utilizadas pelos agentes do sistema para o cálculo e tomada de decisão da reconfiguração do sistema elétrico. Coalizão é o conjunto de agentes em cargas delimitados pelo mesmo conjunto de chaves. Essa delimitação significa que, para qualquer das chaves que fechem o circuito que atinge uma coalizão, todos os agentes pertencentes a este grupo necessariamente receberão energia que

⁵O site da FIPA está localizado em http://fipa.org/.

Tabela de Protocolo de Mensagens e Ações				
Protocolo da Mensagem	Ações esperadas			
INFORM_YOUR_DEMAND	Requisita informação sobre demanda			
MY_DEMAND_IS	Envia informação sobre demanda			
INFORM_YOUR_SWITCH_STATE	Requisita informação sobre estado do switch			
MY_SWITCH_STATE_IS	Informa estado do switch			
CALCULE_YOUR_MINIMAL_PATH	Requisita cálculo do caminho mínimo			
MY_MINIMAL_PATH_IS	Informa o caminho mínimo			
MY_POSITION_IS	Informa posição no grafo do sistema elétrico			
MY_COALITION_IS	Informa coalizão			
ELECTRICAL_PATH_FROM_YOU_TO_MY_COALITION_IS	Informa caminho elétrico entre um feeder e uma coalizão			
LOSSES_FROM_YOUR_COALITION_TO_ME_IS	Perdas elétricas entre o feeder e a coalizão			
ACTIVE_FEEDER_TO_MY_COALITION_IS	Informa qual o feeder ativo para aquela coalizão			
DEMAND_CHANGED	Informa mudança na leitura da demanda			

Tabela 5.2: Protocolos das Mensagens

trafegará pelo trecho monitorado pela chave. Os principais conjuntos de dados desta classe são o subgrafo do sistema elétrico completo que representa as ligações entre os agentes da coalizão e as demandas de cada agente que a compõe.

Para exemplificar o conceito de coalizão, a Figura 5.5 apresenta à direita um sistema elétrico baseado em (CIVANLAR et al., 1988), que será utilizado mais a frente neste trabalho. A imagem a esquerda da Figura 5.5 apresenta apenas o subgrafo que representa a coalizão composta pelos agentes nas cargas 3, 4, 5 e 6.

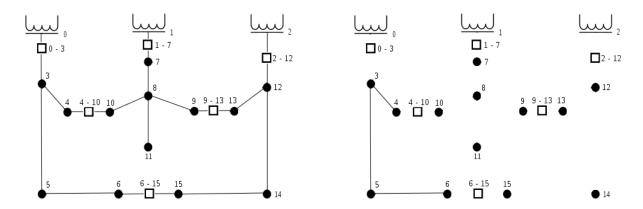


Figura 5.5: À esquerda, exemplo de sistema elétrico completo - grafo completo; a direita, subgrafo que forma a coalizão dos agentes nas cargas 3, 4, 5 e 6

Caminho Elétrico é uma classe que indica qual o menor caminho, em termos apenas da resistência elétrica, entre um determinado alimentador e um agente componente de alguma coalizão. Esta classe auxilia na formação da configuração radial do sistema elétrico para o cálculo do fluxo de carga. A exemplo da classe Coalizão de Agentes, o principal conjunto de

dados da classe Caminho Elétrico é um subgrafo do sistema completo com o caminho entre um alimentador e algum loader agent componente da coalizão.

Para ilustrar, a Figura 5.6 apresenta um sistema de distribuição completo, novamente à esquerda. Ao lado direito da figura, apresenta-se o caminho elétrico entre o alimentador 1 e a coalizão, que é a mesma coalizão apresentada na Figura 5.5.

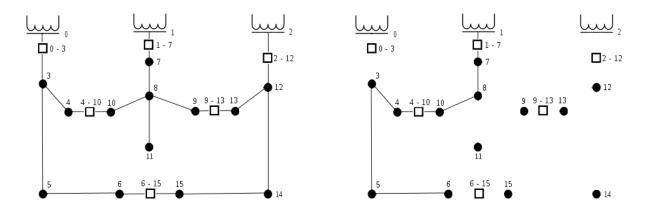


Figura 5.6: À esquerda, exemplo de sistema elétrico completo - grafo completo; a direita, o subgrafo que forma o caminho elétrico para coalizão da Figura 5.5

Agora, será detalhado o *feederAgent*, responsável por monitorar os alimentadores do sistema. Seu diagrama de classes parcial é dado na Figura 5.7:

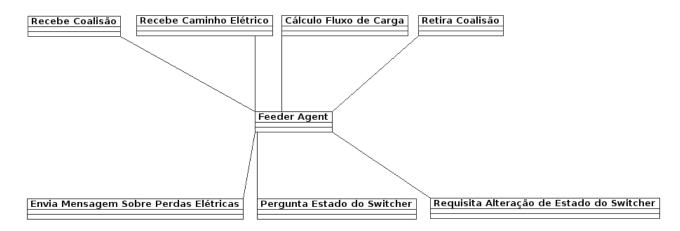


Figura 5.7: Diagrama de classe para o Feeder Agent

No momento de sua execução, um *feederAgent*, primeiramente, envia mensagens aos *swit-cherAgents* perguntando qual o estado em que as chaves se encontram, colhendo a resposta na sequência. Em seguida, o *feederAgent* fica na espera do envio de mensagens dos *loaderAgents* líderes de coalizões. Estas mensagens dizem respeito à coalizão liderada e ao caminho elétrico entre o *feederAgent* e a coalizão.

Enquanto o *feederAgent* recebe mensagens sobre coalizões e caminhos elétricos, ele vai "empilhando" os respectivos subgrafos que estão contidos nas mensagens em uma representação de grafo que combina todas estas informações. Ao final, teremos tanto uma representação radial do sistema elétrico quanto informações sobre demanda dos agentes. Estas informações permitirão ao *feederAgent* ter uma ideia do estado do sistema de distribuição naquele momento e, a partir delas, realizar suas operações.

Com estas informações obtidas, o *feederAgent* procede a execução de um algoritmo de fluxo de carga, utilizando neste trabalho o método da varredura (KERSTING, 2006). Isto permitirá ao *feederAgent* verificar se a topologia construída é viável e, também, definir quais as perdas elétricas para cada coalizão do sistema.

Caso a topologia seja inviável, o *feederAgent* faz um procedimento de retirar do cálculo a coalizão com maior caminho elétrico, em termos de somatória da resistência. Isso torna o sistema a ser trabalhado menor, tanto em tamanho quanto em demanda total, e permite ao *feederAgent* re-executar o fluxo de carga.

Após o método convergir, o *feederAgent* calcula as perdas elétricas para cada coalizão, medidas por $\sum_{i\in C}^i r_i I_i^2$, onde i é um trecho qualquer que pertence ao conjunto de trechos da coalizão e seu respectivo caminho elétrico, identificado por C; r_i e I_i são respectivamente a resistência e a corrente encontrada no trecho i.

Em seguida, o *feederAgent* envia mensagens para os líderes das coalizões com o valor calculado de suas perdas elétricas. Após a tomada de decisão dos *loaderAgents*, os *feederAgents* recebem mensagens avisando qual deles foi escolhido para prover energia àquela coalizão. Com esta informação, o *feederAgent* verifica quais *switcherAgents* devem estar fechados para que a energia que sai do alimentador por ele monitorado chegue às coalizões. Então, o *feederAgent* envia mensagens para estes *switcherAgents* pedindo que fechem o circuito. Com todas estas ações terminadas, o *feederAgent* volta ao comportamento de esperar mensagens dos *loaderAgents*, retornando o início da iteração de seu processamento.

As principais atividades executadas pelo *feederAgent* são apresentadas em forma de fluxograma na Figura 5.8.

Tratar-se-á agora do processamento realizado pelos agentes localizados nas cargas, os *loa-derAgents*. A Figura 5.9 apresenta o diagrama de classes para este agente.

Quando os loaderAgents são inicializados, eles recebem o valor da carga a qual estão mo-

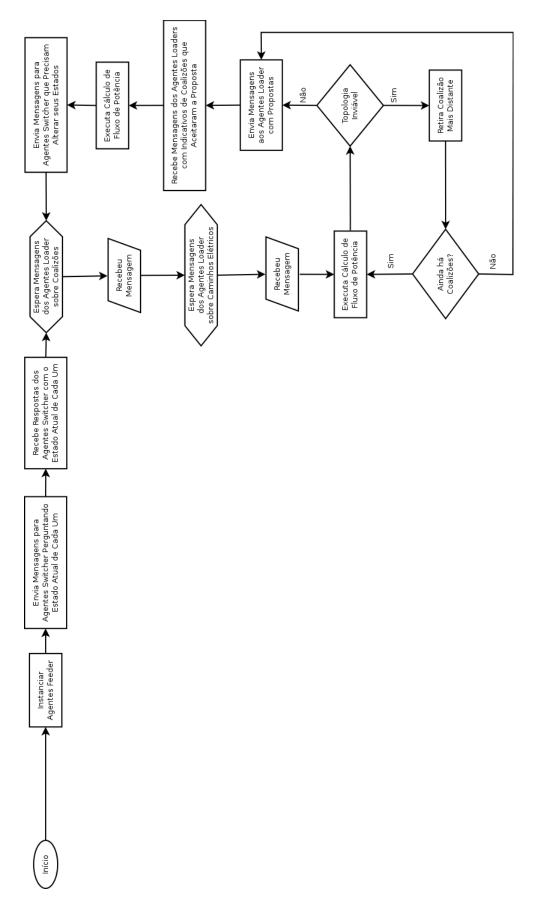


Figura 5.8: Fluxograma de execução dos agentes feeder

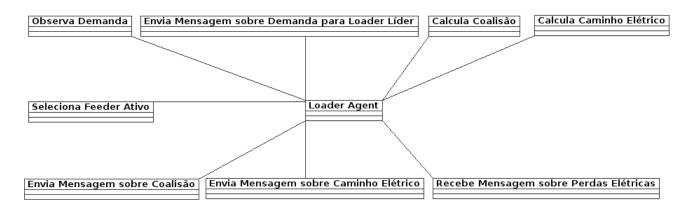


Figura 5.9: Diagrama de classe para o Loader Agent

nitorando. Em seguida, executam um algoritmo de busca em profundidade (CORMEN et al., 2009; RABUSKE, 1992) com modificações para o problema aqui trabalhado, para identificarem quais são seus vizinhos, qual coalizão eles pertencem e qual o líder desta coalizão. O Algoritmo 5.6 apresenta a busca empregada.

```
Algoritmo 5.6: Busca em profundidade - encontra coalizão

Entrada: G, v, Sw, Z

1: para todo z \in Vizinhos(v) faça

2: z = Seleciona-Vizinho(v)

3: se(z \notin Z) e(v, z) \notin Sw então

4: Z = Z \bigcup z)

5: Z = Busca-Profundidade(G, z, Sw, Z)

6: fim se

7: fim para

8: retorne Z
```

Dado um grafo G, um nó v que representa a posição do *loaderAgent*, a definição das arestas que contém chaves representadas por Sw e o conjunto com os nós da coalizão inicialmente vazio Z, o algoritmo 5.6 inicia varrendo os vizinhos de v, conforme a linha 1.

Com um vizinho selecionado, a condição se da linha 3 verifica, primeiramente, se z já foi avaliado. Caso não, verifica se a aresta (v, z) contém uma chave. Caso não, o vértice z é marcado como fazendo parte da coalizão, e agora ele terá a sua vizinhança avaliada, sendo passado recursivamente para a mesma função.

Ao final da execução do Algoritmo 5.6, Z será o conjunto de vértices do subgrafo do sistema de distribuição, representado pelo grafo completo, que contém a coalizão liderada pelo agente *loaderAgent*.

A seleção do líder da coalizão é bem simples: cada *loaderAgent* recebe um nome que inicia com *loaderAgent* seguido de um número de identificação. Após o algoritmo de busca em profundidade listar todos os agentes de uma coalizão, o agente que possuir o maior número de identificação será o líder do grupo.

Após a busca em profundidade, os agentes já conseguiram construir o subgrafo que indica a localização da coalizão. Os *loaderAgents* da coalizão que não são o líder, neste momento, enviam para o líder suas respectivas demandas, que são adicionadas à classe Coalizão de Agentes pelo líder. Após receber todas as demandas, o agente líder envia os dados da coalizão para todos os *feedergents* do sistema.

Em seguida, o *loaderAgent* líder executa um algoritmo de busca no grafo para encontrar o menor caminho, em termos de resistência elétrica, entre cada *feederAgent* do sistema e algum agente que pertença a sua coalizão. Para tanto, é utilizado um algoritmo de Dijkstra modificado, onde existem vários pontos finais e o objetivo é que o algoritmo encontre o primeiro destes pontos. O Algoritmo 5.7 apresenta esta versão do algoritmo de Dijkstra.

Algoritmo 5.7: Algoritmo de Dijkstra com múltiplos pontos finais

```
Entrada: G, v_i
 1: Q_i = \text{Inicializar-Custos-Relativos}(G), \forall i
 2: A_i = \text{Inicializar-no-Anterior}(), \forall i
 3: F = \text{Define-Pontos-Finais}()
 4: S = \emptyset
 5: R = \emptyset
 6: D = \emptyset
 7: temp = 0
 8: noAvaliado = No-Agent-Feeder-Avaliado()
 9: enquanto noAvaliado \notin F faça
       Q_i = \text{Calc-Custo-Relativo}(G, noAvaliado), \forall i
10:
       S = S \cup JnoAvaliado
11:
12:
       temp = noAvaliado
       noAvaliado = Minimo((V - S), Q)
13:
14:
       A_{noAvaliado} = temp
15: fim enquanto
16: para todo v \in V faça
       D = D \bigcup (v, A_v)
18: fim para
19: R = (V, D)
20: retorne R
```

No Algoritmo 5.7, grande parte do procedimento é idêntico ao descrito no Algoritmo 5.1. Dado o grafo $G = \{V, E\}$, onde V é o conjunto dos vértices e E o conjunto das arestas; v_i que será o vértice-fonte, que será a barra onde se encontra o *loaderAgent* líder. Na linha 3, a chamada

para a função *Define-Pontos-Finais()* retorna o conjunto de pontos finais do algoritmo, ao qual, quando encontrar pelo menos um deles, terá seu objetivo atingido - representando a condição do laço *enquanto* da linha 9.

Com os caminhos elétricos calculados, o líder envia os dados de cada caminho ao *feederA-gent* relacionado. Este é o momento em que os *feederAgents* executam o algoritmo de fluxo de carga, como descrito anteriormente.

A próxima ação executada pelos *loaderAgents* é receber o valor das perdas elétricas enviadas por cada *feederAgent* e selecionar qual deles será o *feederAgent* ativo. Essa seleção é feita a partir do *feederAgent* que indica a menor perda para a coalizão. Apesar do critério de escolha, isso não garante que a configuração do sistema elétrico, globalmente, terá a menor perda possível. Esse critério apenas tornará a topologia final radial.

Depois de selecionar o *feederAgent* ativo para a coalizão, os *loaderAgents* voltam sua atenção para a variação do seu consumo. Caso haja alguma modificação, é repetido todo o processo de troca de mensagens descrito anteriormente.

O fluxo de execução dos loaderAgents é apresentado na Figura 5.10.

Por último, dos agentes que compõem a modelagem aqui apresentada, temos o *switcherA-gent*, que é puramente reativo (WOOLDRIDGE, 2009) e também o mais simples. A Figura 5.11 apresenta seu diagrama de classes.

Basicamente, o *switcherAgent* apenas "reage" a mensagens que são enviadas pelos *feede-rAgents*: ele informa seu estado atual (aberto ou fechado, conforme a enumeração Estado da classe Electrical Agent, apresentada na Figura 5.4) ou o altera. Quando aberto, significa que no trecho que o *switcherAgent* monitora não está passando corrente; quando fechado, significa que ali está passando corrente.

O fluxograma das atividades do *switcherAgent* é apresentado na Figura 5.12.

Do exposto, pode-se perceber que há um considerável número de mensagens e uma certa ordem entre elas. A Figura 5.13 apresenta o diagrama de sequência (PÁDUA, 2003) com a ordem da troca de mensagens entre os tipos de agentes citados neste trabalho.

Já a Figura 5.14 apresenta a visão completa do fluxograma de todos os agentes aqui apresentados que modelam funções dos equipamentos do *smart grid* que interessam ao nível de gerenciamento do sistema elétrico trabalhado neste texto.

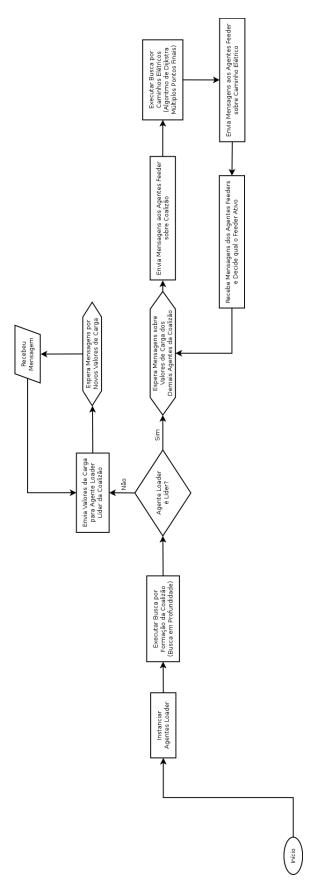


Figura 5.10: Fluxograma de execução dos agentes *loader*

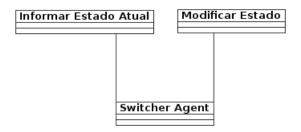


Figura 5.11: Diagrama de classe para o Switcher Agent

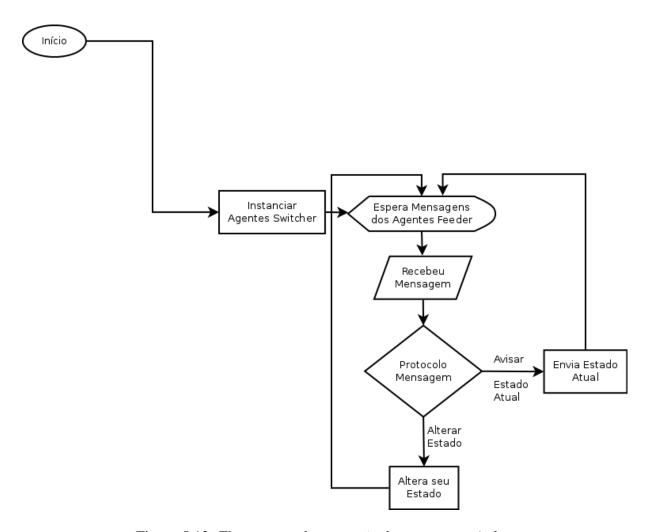


Figura 5.12: Fluxograma de execução dos agentes switcher

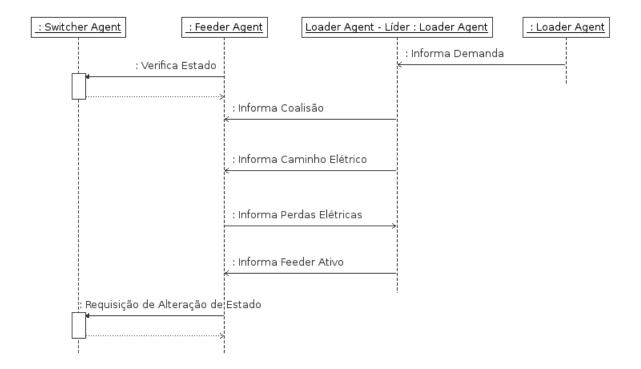


Figura 5.13: Diagrama de sequência com a ordem de trocas de mensagens entre os agentes

Com a parte conceitual exposta, o próximo passo da pesquisa foi implementar a modelagem e aplicá-la a alguns exemplos teóricos de sistemas de distribuição para avaliar se os agentes conseguem gerenciar a topologia da rede. A próxima subseção apresentará as tecnologias utilizadas no desenvolvimento da implementação computacional.

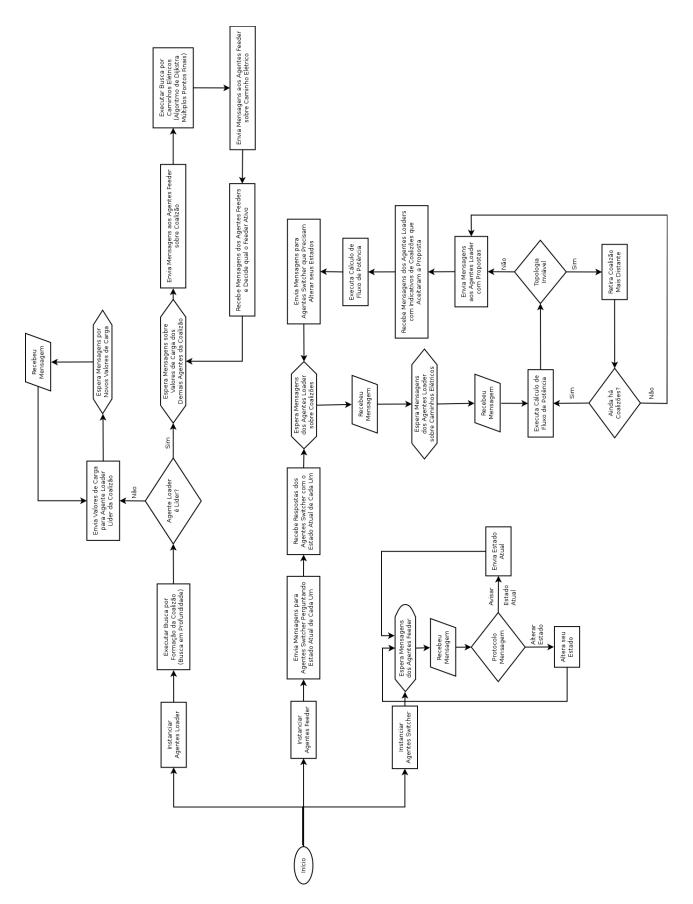


Figura 5.14: Fluxograma de execução dos agentes

5.4.2 Implementação do Sistema Multiagente

A implementação foi realizada usando apenas softwares livres, todos disponíveis na internet. A linguagem de programação escolhida foi a orientada a objetos (SANTOS, 2003) Java⁶ (DEITEL; DEITEL, 2003), com o *framework* de programação multiagente JADE⁷ (*Java Agent Development*) (BELLIFEMINE; CAIRE; GREENWOOD, 2007; BORDINI et al., 2006). O critério de escolha desta tecnologia foi motivada pelo bom suporte e documentação do *framework*, bem como por ele já ser aplicado e ter boa aceitação na comunidade de pesquisadores que desenvolvem sistemas multiagentes aplicados a sistemas elétricos de potência (MCARTHUR et al., 2007b). A máquina virtual Java utilizada foi a OpenJDK⁸.

Os agentes trocam informações entre si através do envio de mensagens. O conteúdo destas mensagens, normalmente, é uma cadeia de caracteres conhecidas como *string*. Para fazer a interface entre objetos Java e *strings*, foi utilizado uma biblioteca livre criada e mantida pelo Google chamada GSON⁹. Esta biblioteca converte um objeto Java para um objeto JavaScript¹⁰ (também conhecido como JSON¹¹ (CROCKFORD, 2006)), que é representado por uma cadeia de caracteres que preserva a estrutura da classe Java da qual ele foi gerado. Gson também consegue fazer a conversão inversa.

Como a aplicação aqui trabalhada exige manipulações de números complexos, e este tipo de representação numérica não tem uma implementação padrão em Java, foi utilizada a classe Complex da biblioteca livre Apache Commons Math, mantida pela Apache Software Foundation¹².

O ambiente de desenvolvimento utilizado foi o Eclipse¹³, e o sistema para controle de versão empregado foi o Git¹⁴. O software foi executado em um ambiente computacional com um processador Intel i5 de quatro núcleos e 4 GB de memória RAM, utilizando softwares KDE¹⁵ e GNU¹⁶ com o sistema operacional Linux¹⁷, kernel 3.2.6-server de 32 bits, empacotado

```
GJava-http://java.net/

JADE-http://jade.tilab.com/

OpenJDK-http://openjdk.java.net/

GSON-http://code.google.com/p/google-gson/

JavaScript-http://mozilla.org/js/

JSON-http://json.org/

Apache Commons Math-http://commons.apache.org/math/

Belipse-http://eclipse.org/

HGit-http://git-scm.com/

KDE-http://kde.org/

GNU-http://gnu.org/
```

17Linux - http://kernel.org/

pela distribuição Mageia¹⁸.

Apesar do código do software ter sido escrito em Java, foi criado um script em Python¹⁹ (BORGES, 2009) para fazer o tratamento do arquivo de entrada e gerar o comando Java que lança a plataforma dos agentes. Isso foi necessário pois, usando o método convencional de interação por linha de comando, seria muito complicado instanciar dezenas de agentes sem incorrer em algum erro.

Com a modelagem e a tecnologia utilizada na implementação computacional apresentadas, a próxima subseção descreverá alguns sistemas de distribuição teóricos, onde o sistema multiagente desenvolvido foi aplicado.

5.4.3 Simulações Computacionais

Os sistemas de testes em que foram aplicadas a implementação computacional da proposta aqui apresentada foram selecionados especificamente para testar o desempenho do software em atender as funcionalidades imaginadas. Em todos os sistemas, a tensão nominal dos alimentadores é igual a 23 kV, a tolerância utilizada no método da varredura (KERSTING, 2006) é igual ao produto da tensão nominal por 10^{-3} e o número máximo de iterações executadas é igual a 10^2 .

Todos os dados utilizados tem como intuito prover um ambiente em que seja possível verificar se o método computacional implementado consegue, de fato, desempenhar as funcionalidades de gerenciamento da rede aqui trabalhadas, de forma adequada.

Sistema com 10 barras

O primeiro exemplo, apresentado graficamente na Figura 5.15 e cujos dados se encontram na Tabela 5.3, testa a comunicação entre os agentes e a respectiva tomada de decisão da forma mais simples possível. Trata-se de um sistema sem ramificações e sem chaves, com apenas um alimentador. Este sistema foi retirado de (GRAINGER; LEE, 1982), e também servirá com base para os próximos dois exemplos de sistemas elétricos testados.

O objetivo é verificar se a comunicação entre os agentes, de fato, funcionará. Que esta comunicação permitirá que apareça uma liderança nos *loaderAgents* e, por consequência, uma coalizão; que essa liderança consiga executar os cálculos de caminho mínimo e reunir as men-

¹⁸Mageia - http://mageia.org/

¹⁹Python - http://python.org/

sagens sobre as demandas nos demais agentes; que o agente *feederAgent* faça o cálculo de fluxo de carga a partir das informações enviadas pelo líder da coalizão; que o *feederAgent* realize a proposta para o líder e, ao final, que o líder aceite esta proposta e eleja o *feederAgent* como o alimentador ativo para aquela coalizão.



Figura 5.15: Sistema de testes - 10 barras

Tabela 5.3: Sistema de 10 barras

Sistema 10 barras				
Carga	Potência Ativa (kW) Potência Reativa (kW)			
1	1840	460		
2	980	340		
3	1790	446		
4	1598	1840		
5	1610	600		
6	780	110		
7	1150	60		
8	980	130		
9	1640	200		
Trechos	Resistência (Ω)	Reatância (Ω)		
0-1	0,1233	0,4127		
1-2	0,0140	0,6051		
2-3	0,7463	1,2050		
3-4	0,6984	0,6084		
4-5	1,9831	1,7276		
5-6	0,9053	0,7886		
6-7	2,0552	1,1640		
7-8	4,7953	2,7160		
8-9	5,3434	3,0264		

Este sistema tem apenas uma coalizão, que será liderada pelo *loaderAgent* 9. Após descobrirse líder, este agente recebe mensagens com os valores das demandas dos demais *loaderAgents*, formando a coalizão. Ele então envia estes dados para o único *feederAgent* do sistema. Em seguida, o líder executa o algoritmo de Dijksta modificado para encontrar o menor caminho entre o *feederAgent* e a coalizão. Encontrando (este caminho é apenas o trecho 0 - 1), ele envia esta informação ao *feederAgent*.

Com todos estes dados obtidos, o feeder Agent executa o algoritmo de fluxo de carga (método da varredura) e encontra uma solução em 12 iterações, indicando que a topologia é viável. Quando o feederAgent envia a resposta para o líder da coalizão, este seleciona-o como ativo para seu grupo.

A Tabela 5.4 trás o resultado do fluxo de carga para esta instância.

Tensões Finais - Sistema de 10 barras			
Barra	Barra Tensão (kV)		
0	23,0000		
1	22,8357		
2	22,7041		
3	22,1400		
4	21,7810		
5	21,0504		
6	20,8105		
7	20,3796		
8	19,6622		
9	19,1574		
Número de iterações: 12			

Tabela 5.4: Resultado final para o sistema de 10 barras

Sistema com 11 barras

O próximo exemplo, cujo desenho esquemático pode ser visto na Figura 5.16 e os dados encontram-se na Tabela 5.5, é o exemplo anterior da Figura 5.15 com a adição de mais um alimentador e duas chaves, uma entre as barras 0 e 2 e outra entre as barras 1 e 2.

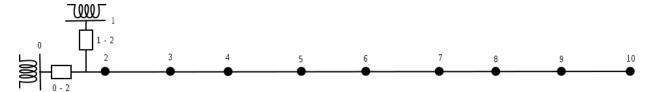


Figura 5.16: Sistema de testes - 11 barras e 2 chaves

A finalidade deste exemplo é, além de atingir os objetivos já listados no primeiro exemplo, verificar se a única coalizão do sistema escolherá apenas um alimentador, e que essa escolha será pelo alimentador que conduz a uma menor perda - por conta da diferença entre as resistências

Tabela 5.5: Sistema de 11 barras e 2 chaves

	Sistema 11 barras 2 chaves						
Carga							
2	1840	460					
3	980	340					
4	1790	446					
5	1598	1840					
6	1610	600					
7	780	110					
8	1150	60					
9	980	130					
10	1640	200					
Trechos	Resistência (Ω)	Reatância (Ω)					
0-2	0,1233	0,4127					
1-2	0,0123	0,0412					
2-3	0,0140	0,6051					
3-4	0,7463	1,2050					
4-5	0,6984	0,6084					
5-6	1,9831	1,7276					
6-7	0,9053	0,7886					
7-8	2,0552	1,1640					
8-9	4,7953	2,7160					
9-10	5,3434	3,0264					
Chaves: Trechos 0 - 2 e 1 - 2							

nos trechos 0 - 2 e 1 - 2. A opção escolhida também implicará em uma configuração específica para os *switcherAgents* do sistema.

Após as trocas de mensagens entre os *loaderAgents* e o envio de mensagens sobre coalizão e caminho elétrico para os *feederAgents*, a execução do fluxo de carga chega a resultados diferentes: o *feederAgent* 0 encontra uma perda de 78,3778 kW e o *feederAgent* 1 uma perda de 72,9378 kW.

Quando os *feederAgents* enviam mensagem para o líder dos *loaderAgents*, ele seleciona o *feederAgent* 1 como alimentador ativo da coalizão. Então, o *feederAgent* 1 envia uma mensagem para o *switcherAgent* em 1 - 2 requisitando que ele assuma o estado fechado, permitindo então que a energia percorra a rede elétrica e alimente as cargas do sistema.

A Tabela 5.6 apresenta o resultado para os dois *feederAgents*; entretanto, o *feederAgent* escolhido é apenas o 1.

Tensões Finais Barra Feeder 0 Tensão (kV) Feeder 1 Tensão (kV) 0 23,0000 1 23,0000 2 22,8357 22,9839 3 22,7041 22,8559 4 22,1400 22,2998 5 21.7810 21.9447 6 21,0504 21,2235 7 20,8105 20,9869 8 20,3796 20,5612 9 19,6622 19,8523 10 19,3536 19,1574 12 12 Número de Iterações **Perdas** 78,3778 kW 72,9378 kW

Tabela 5.6: Resultado final para o sistema de 11 barras e 2 chaves

Sistema com 11 barras

O terceiro exemplo, representado na Figura 5.17 e na Tabela 5.7, é também composto por dois alimentadores, mas cada um está em uma extremidade do sistema. Há também uma chave que divide o sistema elétrico entre as barras 5 e 6.

O comportamento a ser verificado neste exemplo, em adição aos anteriores, é observar se,

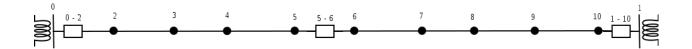


Figura 5.17: Sistema de testes - 11 barras e 3 chaves

Tabela 5.7: Sistema de 11 barras e 3 chaves

Sistema 11 barras 3 chaves							
Carga							
2	1840	460					
3	980	340					
4	1790	446					
5	1598	1840					
6	1610	600					
7	780	110					
8	1150	60					
9	980	130					
10	1640	200					
Trechos	Resistência (Ω)	Reatância (Ω)					
0-2	0,1233	0,4127					
2-3	0,0140	0,6051					
3-4	0,7463	1,2050					
4-5	0,6984	0,6084					
5-6	∞	∞					
7-6	1,9831	1,7276					
8-7	0,9053	0,7886					
9-8	2,0552	1,1640					
10-9	4,7953	2,7160					
1-10	5,3434	3,0264					
Chaves: Trechos 0 - 2, 5 - 6 e 1 - 10							

de fato, serão formadas duas coalizões e se estas coalizões ativarão, cada uma, o alimentador que lhe está mais próximo. O trecho 5 - 6, monitorado por um *switcherAgent*, tem uma resistência muito alta. Isso implicará que os *feederAgents* não conseguirão convergir o método de cálculo do fluxo de carga para uma topologia onde cada um tenta alimentar as duas coalizões juntas.

Assim, os *feederAgents* iniciarão o método de retirar as coalizões que tem o caminho elétrico maior e, sem estas, executarão novamente o algoritmo de fluxo de carga tentando verificar se a nova topologia é viável.

De fato, este comportamento pôde ser observado. Na Tabela 5.8 pode-se perceber que cada *feederAgent* está alimentando a coalizão que lhe é mais próxima. Isso implica uma configuração de chaves com os trechos 0 - 2 e 1 - 10 configuradas como fechadas e a chave no trecho 5 - 6 aberta.

Tabela 5.8: Resultado final para o sistema de 11 barras e 3 chaves

Tensões Finais				
Barra	Feeder 0 Tensão (kV)	Feeder 1 Tensão (kV)		
0	23,0000	_		
1	_	23,0000		
2	22,9090	_		
3	22,8344	_		
4	22,5986	_		
5	22,4994	_		
6	_	19,0806		
7	_	19,3064		
8	_	19,8824		
9	_	21,1452		
10	-	21,7890		
Número de Iterações	5	12		
Perdas	44,9422 kW	904,7419 kW		

Sistema com 16 barras

O próximo e último exemplo é baseado no clássico sistema elétrico de Cinvalar de 16 barras (CIVANLAR et al., 1988), mas aqui foi modificado para ter apenas 6 chaves. A Figura 5.18 e a Tabela 5.9 trazem, respectivamente, o desenho e o conjunto de dados desta instância. Outra diferença em relação ao sistema original foi a adição de limites máximos de correntes no trechos, cujos valores são dados na Tabela 5.10. A adição desta restrição visa apresentar a versatilidade do método aqui trabalhado, que consegue lidar tanto com sistemas que contém esse

tipo de restrição quanto os que não a tem.

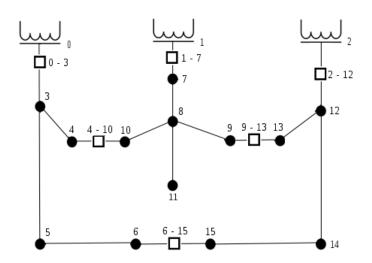


Figura 5.18: Sistema de testes - 16 barras e 6 chaves

Este exemplo é interessante pois será apreciada a capacidade do sistema de, dinamicamente, reconfigurar a topologia do sistema elétrico a partir de alterações nas cargas do mesmo, respeitando a restrição de correntes máximas nos trechos. Após a decisão dos *loaderAgents* sobre qual *feederAgent* é o ativo para sua coalizão, os *loaderAgents* entram em um comportamento de esperar por alterações em suas demandas. Por sua vez, os *feederAgents* retornam ao comportamento de esperar mensagens sobre coalizões dos *loaderAgents* líderes.

Para simular a alteração da demanda do *loaderAgent*, introduz-se um novo tipo de agente ao sistema, chamado de *demandChanged*. Este agente tem o funcionamento bem simples: apenas lê um arquivo que contém uma lista com nome de um *loaderAgent* e suas novas cargas. O agente *demandChanged*, então, prepara uma mensagem tendo o nome do *loaderAgent* como destinatário e sua nova carga como conteúdo desta mensagem.

Ao receber a mensagem, o *loaderAgent* atualiza o valor da sua carga e repassa a mensagem ao líder da sua coalizão. Após receber as novas cargas de todos os demais agentes sob sua liderança, o *loaderAgent* líder envia mensagens com os dados atualizados sobre sua coalizão aos *feederAgents* alocados no sistema elétrico.

As variações das cargas foram divididas em doze "momentos": iterações onde as mudanças nas cargas acontecem. Estas mudanças seguem a ordem em que são apresentadas neste texto e as características de cada estão na Tabela 5.11.

A mudança nos valores das cargas são dadas em agentes alocados em uma mesma coalizão

Tabela 5.9: Sistema de 16 barras e 6 chaves

Sistema 16 barras 6 chaves								
Carga								
3	2,0	1,6						
4	3,0	1,5						
5	2,0	0,8						
6	1,5	1,2						
7	4,0	2,7						
8	5,0	3,0						
9	1,0	0,9						
10	0,6	0,1						
11	4,5	2,0						
12	1,0	0,9						
13	1,0	0,7						
14	1,0	0,9						
15	2,1	1,0						
Trechos	Resistência (Ω)	Reatância (Ω)						
0-3	0,3967	0,5290						
3-4	0,4232	0,5819						
3-5	0,4761	0,9522						
5-6	0,2116	0,2116						
1-7	0,5819	0,5819						
7-8	0,4232	0,5819						
7-9	0,5819	0,5819						
8-10	0,5819	0,5819						
8-11	0,4232	0,5819						
2-12	0,5819	0,5819						
12-13	0,4761	0,6348						
12-14	0,4232	0,5819						
14-15	0,2116	0,2116						
4-10	0,2116	0,2116						
9-13	0,2116	0,2116						
6-15	0,6348	0,6348						
Chaves: Trechos 0 - 3, 1 - 7, 2 - 12, 4 - 10, 6 - 15, 9 - 13								

Tabela 5.10: Limite máximo de corrente nos trechos

Limites de Correntes			
Trechos Correntes (A)			
0-3	1700		
3-4	1500		
3-5	1500		
5-6	1500		
1-7	1200		
7-8	1500		
7-9	1500		
8-10	1000		
8-11	1000		
2-12	2500		
12-13	1500		
12-14	1500		
14-15	1500		
4-10	1500		
9-13	1500		
6-15	1500		

Tabela 5.11: Características dos Momentos

Momentos e Características			
Momentos	entos Características		
Momento 1	Estado Inicial		
Momento 2	Agentes 12, 13 e 14 com 20% a mais de energia		
Momento 3	Agentes 12, 13 e 14 com 50% a mais de energia		
Momento 4	Agentes 7, 8, 9, 10, 12, 13 e 14 com 20% a mais de energia		
Momento 5	Agentes 7, 8, 9, 10, 12, 13 e 14 com 50% a mais de energia		
Momento 6	Agentes 7, 8, 9 e 10 com 20% a mais de energia		
Momento 7	Agentes 7, 8, 9 e 10 com 50% a mais de energia		
Momento 8	Agentes 3, 4 e 5 com 50% a mais de energia		
Momento 9	Agentes 3, 4, 5, 7, 8, 9 e 10 com 50% a mais de energia		
Momento 10	Agentes 3, 4, 5, 7, 8, 9, 10, 11, 12, 13 e 14 com 50% mais de energia		
Momento 11	Agentes 3, 4, 5, 7, 8, 9, 10, 11, 12, 13 e 14 com 20% mais de energia		
Momento 12	Agentes 3, 4, 5, 11, 12, 13 e 14 com 50% mais de energia, 7, 8, 9 e 10 com 20% mais		

ou conjunto de coalizões - esse aumento tenta simular o comportamento em que, em um horário definido do dia em uma determinada região, esta passa a consumir mais energia que em outro horário. Poderia exemplificar o perfil de consumo de cargas, onde há um horário bem definido em que normalmente o consumo é maior ou menor (KAGAN; OLIVEIRA; ROBBA, 2005).

A seguir, da Tabela 5.12 à 5.23, são apresentadas as tabelas com a configuração final da topologia para cada momento, quais alimentadores estão enviando energia para cada coalizão e quais os valores finais de tensão nas barras consumidoras. Estes são os valores medidos pelos agentes alocados em suas respectivas barras.

Já a Tabela 5.24 apresenta a configuração das chaves fechadas para cada topologia final do sistema de distribuição em cada momento.

Os exemplos aqui apresentados mostraram que a reconfiguração topológica do sistema de distribuição foi alterada a partir da relação entre os agentes alocados ao sistema elétrico.

Tabela 5.12: Estado Inicial

Estado Inicial (Momento 1)					
Cargas	Pot. Ativa (kW)	Pot. Reativa (kVar)	Alim. 0 (kV)	Alim. 1 (kV)	Alim. 2 (kV)
3	2	1,6	22,7309	-	-
4	3	1,5	22,6359	-	-
5	2	0,8	22,5717	-	-
6	1,5	1,2	22,5463	-	-
7	4	2,7	-	22,1161	-
8	5	3	-	21,5280	-
9	1	0,9	-	21,2357	-
10	0,6	0,1	-	21,5088	-
11	4,5	2	-	21,3826	-
12	1	0,9	-	20,9680	-
13	1	0,7	-	21,1483	-
14	1	0,9	-	20,8507	-
15	2,1	1	-	20,8189	-

Tabela 5.13: Agentes 12, 13 e 14 com 20% a mais de energia

	Momento 2					
Cargas	Pot. Ativa (kW)	Pot. Reativa (kVar)	Alim. 0 (kV)	Alim. 1 (kV)	Alim. 2 (kV)	
3	2	1,6	22,7309	-	-	
4	3	1,5	22,6359	-	-	
5	2	0,8	22,5717	-	-	
6	1,5	1,2	22,5463	-	-	
7	4	2,7	-	22,0821	-	
8	5	3	-	21,4650	-	
9	1	0,9	-	21,1402	-	
10	0,6	0,1	-	21,4458	-	
11	4,5	2	-	21,3192	-	
12	1,2	1,08	-	20,8393	-	
13	1,2	0,84	-	21,0411	-	
14	1,2	1,08	-	20,7120	-	
15	2,1	1	-	20,6799	-	

Tabela 5.14: Agentes 12, 13 e 14 com 50% a mais de energia

	Momento 3				
Cargas	Pot. Ativa (kW)	Pot. Reativa (kVar)	Alim. 0 (kV)	Alim. 1 (kV)	Alim. 2 (kV)
3	2	1,6	22,1972	-	-
4	3	1,5	21,5301	-	-
5	2	0,8	22,0330	-	-
6	1,5	1,2	22,0070	-	-
7	4	2,7	20,4607	-	-
8	5	3	20,6223	-	-
9	1	0,9	20,5685	-	-
10	0,6	0,1	21,2828	-	-
11	4,5	2	20,4690	-	-
12	1,5	1,35	-	-	22,7078
13	1,5	1,05	-	-	22,6467
14	1,5	1,35	-	-	22,5794
15	2,1	1	-	-	22,5503

Tabela 5.15: Agentes 7, 8, 9, 10, 12, 13 e 14 com 20% a mais de energia

	Momento 4				
Cargas	Pot. Ativa (kW)	Pot. Reativa (kVar)	Alim. 0 (kV)	Alim. 1 (kV)	Alim. 2 (kV)
3	2	1,6	22,1080	-	-
4	3	1,5	21,3452	-	-
5	2	0,8	21,9431	-	-
6	1,5	1,2	21,9169	-	-
7	4,8	3,24	20,0906	-	-
8	6	3,6	20,2884	-	-
9	1,2	1,08	20,2227	-	-
10	0,72	0,12	21,0570	-	-
11	4,5	2	20,1322	-	-
12	1,2	1,08	-	-	22,7508
13	1,2	0,84	-	-	22,7021
14	1,2	1,08	-	-	22,6353
15	2,1	1	-	-	22,6063

Tabela 5.16: Agentes 7, 8, 9, 10, 12, 13 e 14 com 50% a mais de energia

	Momento 5				
Cargas	Pot. Ativa (kW)	Pot. Reativa (kVar)	Alim. 0 (kV)	Alim. 1 (kV)	Alim. 2 (kV)
3	2	1,6	-	-	20,9185
4	3	1,5	-	-	20,8140
5	2	0,8	-	-	21,1788
6	1,5	1,2	-	-	21,2897
7	6	4,05	-	22,1275	-
8	7,5	4,5	-	21,6245	-
9	1,5	1,35	-	21,5475	-
10	0,9	0,15	-	21,5958	-
11	4,5	2	-	21,4798	-
12	1,5	1,35	-	-	22,3205
13	1,5	1,05	-	-	22,2583
14	1,5	1,35	-	-	21,8713
15	2,1	1	-	-	21,7031

Tabela 5.17: Agentes 7, 8, 9 e 10 com 20% a mais de energia

	Momento 6				
Cargas	Pot. Ativa (kW)	Pot. Reativa (kVar)	Alim. 0 (kV)	Alim. 1 (kV)	Alim. 2 (kV)
3	2	1,6	22,7309	-	-
4	3	1,5	22,6359	-	-
5	2	0,8	22,5717	-	-
6	1,5	1,2	22,5463	-	-
7	4,8	3,24	-	22,2737	-
8	6	3,6	-	21,8473	-
9	1,2	1,08	-	21,7864	-
10	0,72	0,12	-	21,8247	-
11	4,5	2	-	21,7043	-
12	1	0,9	-	-	22,7793
13	1	0,7	-	-	22,7388
14	1	0,9	-	-	22,6724
15	2,1	1	-	-	22,6434

Tabela 5.18: Agentes 7, 8, 9 e 10 com 50% a mais de energia

	Momento 7				
Cargas	Pot. Ativa (kW)	Pot. Reativa (kVar)	Alim. 0 (kV)	Alim. 1 (kV)	Alim. 2 (kV)
3	2	1,6	-	-	21,0225
4	3	1,5	-	-	20,9185
5	2	0,8	-	-	21,2813
6	1,5	1,2	-	-	21,3917
7	6	4,05	-	22,1275	-
8	7,5	4,5	-	21,6245	-
9	1,5	1,35	-	21,5475	-
10	0,9	0,15	-	21,5958	-
11	4,5	2	-	21,4798	-
12	1	0,9	-	-	22,3957
13	1	0,7	-	-	22,3545
14	1	0,9	-	-	21,9703
15	2,1	1	-	-	21,8030

Tabela 5.19: Agentes 3, 4 e 5 com 50% a mais de energia

	Momento 8				
Cargas	Pot. Ativa (kW)	Pot. Reativa (kVar)	Alim. 0 (kV)	Alim. 1 (kV)	Alim. 2 (kV)
3	3	2,4	22,6209	-	-
4	4,5	2,25	22,4772	-	-
5	3	1,2	22,4215	-	-
6	1,5	1,2	22,3960	-	-
7	4	2,7	-	22,3696	-
8	5	3	-	21,9933	-
9	1	0,9	-	21,9429	-
10	0,6	0,1	-	21,9746	-
11	4,5	2	-	21,8514	-
12	1	0,9	-	-	22,7793
13	1	0,7	-	-	22,7388
14	1	0,9	-	-	22,6724
15	2,1	1	-	-	22,6434

Tabela 5.20: Agentes 3, 4, 5, 7, 8, 9 e 10 com 50% a mais de energia

	Momento 9				
Cargas	Pot. Ativa (kW)	Pot. Reativa (kVar)	Alim. 0 (kV)	Alim. 1 (kV)	Alim. 2 (kV)
3	3	2,4	-	-	20,2401
4	4,5	2,25	-	-	20,0766
5	3	1,2	-	-	20,6470
6	1,5	1,2	-	-	20,8193
7	6	4,05	-	22,1275	-
8	7,5	4,5	-	21,6245	-
9	1,5	1,35	-	21,5475	-
10	0,9	0,15	-	21,5958	-
11	4,5	2	-	21,4798	-
12	1	0,9	-	-	22,2203
13	1	0,7	-	-	22,1787
14	1	0,9	-	-	21,6491
15	2,1	1	-	-	21,4185

Tabela 5.21: Agentes 3, 4, 5, 7, 8, 9, 10, 11, 12, 13 e 14 com 50% mais de energia

	Momento 10				
Cargas	Pot. Ativa (kW)	Pot. Reativa (kVar)	Alim. 0 (kV)	Alim. 1 (kV)	Alim. 2 (kV)
3	3	2,4	-	-	20,1297
4	4,5	2,25	-	-	19,9652
5	3	1,2	-	-	20,5390
6	1,5	1,2	-	-	20,7122
7	6	4,05	-	22,1275	-
8	7,5	4,5	-	21,6245	-
9	1,5	1,35	-	21,5475	-
10	0,9	0,15	-	21,5958	-
11	4,5	2	-	21,4798	-
12	1,5	1,35	-	-	22,1431
13	1,5	1,05	-	-	22,0804
14	1,5	1,35	-	-	21,5467
15	2,1	1	-	-	21,3148

Tabela 5.22: Agentes 3, 4, 5, 7, 8, 9, 10, 11, 12, 13 e 14 com 20% mais de energia

	Momento 11				
Cargas	Pot. Ativa (kW)	Pot. Reativa (kVar)	Alim. 0 (kV)	Alim. 1 (kV)	Alim. 2 (kV)
3	2,4	1,92	22,6871	-	-
4	3,6	1,8	22,5727	-	-
5	2,4	0,96	22,5119	-	-
6	1,5	1,2	22,4865	-	-
7	4,8	3,24	-	22,2737	-
8	6	3,6	-	21,8473	-
9	1,2	1,08	-	21,7864	-
10	0,72	0,12	-	21,8247	-
11	4,5	2	-	21,7043	-
12	1,2	1,08	-	-	22,7508
13	1,2	0,84	-	-	22,7021
14	1,2	1,08	-	-	22,6353
15	2,1	1	-	-	22,6063

Tabela 5.23: Agentes 3, 4, 5, 11, 12, 13 e 14 com 50% mais de energia; 7, 8, 9 e 10 com 20% mais de energia

	Momento 12				
Cargas	Pot. Ativa (kW)	Pot. Reativa (kVar)	Alim. 0 (kV)	Alim. 1 (kV)	Alim. 2 (kV)
3	3	2,4	22,6209	-	-
4	4,5	2,25	22,4772	-	-
5	3	1,2	22,4215	-	-
6	1,5	1,2	22,3960	-	-
7	4,8	3,24	-	22,2737	-
8	6	3,6	-	21,8473	-
9	1,2	1,08	-	21,7864	-
10	0,72	0,12	-	21,8247	-
11	4,5	2	-	21,7043	-
12	1,5	1,35	-	-	22,7078
13	1,5	1,05	-	-	22,6467
14	1,5	1,35	-	-	22,5794
15	2,1	1	-	-	22,5503

Tabela 5.24: Chaves fechadas por Momentos

Chaves fecha	das por Momentos
Momentos	Chaves fechadas
Momento 1	0 - 3, 1 - 7, 9 - 13
Momento 2	0 - 3, 1 - 7, 9 - 13
Momento 3	0 - 3, 4 - 10, 2 - 12
Momento 4	0 - 3, 4 - 10, 2 - 12
Momento 5	6 - 15, 1 - 7, 2 - 12
Momento 6	0 - 3, 1 - 7, 2 - 12
Momento 7	6 - 15, 1 - 7, 2 - 12
Momento 8	0 - 3, 1 - 7, 2 - 12
Momento 9	6 - 15, 1 - 7, 2 - 12
Momento 10	6 - 15, 1 - 7, 2 - 12
Momento 11	0 - 3, 1 - 7, 2 - 12
Momento 12	0 - 3, 1 - 7, 2 - 12

Análise do Método Computacional

O exemplo com dez barras mostrou que o método computacional desenvolvido consegue prover subsídios para o cálculo do fluxo de potência a partir do envio de mensagens de pontos dispersos do sistema elétrico para o agente localizado no alimentador. Este exemplo serviu para introduzir o conceito de troca de informações distribuídas para dar subsídios à tomada de decisão, algo que deverá estar presente nos futuros *smart grids* (BROWN, 2008; WEI; YU-HUI; JIE-LIN, 2009).

Os exemplos de sistemas com onze barras serviram para apontar o sucesso do método sobre a tomada de decisão distribuída dos agentes para encontrar uma topologia viável do sistema elétrico, segundo o estado do mesmo. Também mostrou que os agentes conseguiram decidir qual alimentador seria utilizado para prover energia a sua coalizão quando havia mais de uma possibilidade de escolha.

Do exemplo com dezesseis barras percebe-se que o sistema multiagente cumpriu o objetivo de gerenciar o sistema elétrico de distribuição, através da reconfiguração topológica da rede, dada mudanças em em "tempo real" nas cargas do sistema. Em adição, com o uso de restrições de fluxo máximo de correntes nos trechos, o sistema teve a oportunidade de mostrar que consegue se adaptar a este tipo de ambiente, contornando as restrições e mantendo a topologia no espaço viável de soluções.

A Tabela 5.25 apresenta o tempo, dado em milissegundos, para cada execução de um feederAgent durante o processamento de cada momento. O cálculo do tempo dos agentes no
alimentador foi contabilizado pela diferença entre o tempo de chegada da primeira mensagem
de um loaderAgent e o tempo do cálculo do fluxo de potência após mensagens de decisão do
alimentador ativo de cada loaderAgent líder. Este intervalo de operações é interessante pois ele
dá uma estimativa do tempo que o sistema leva para gerenciar as informações necessárias para
a tomada de decisão, conforme apresentado anteriormente no fluxograma do feederAgent na
Figura 5.8.

Vale também ressaltar que o tempo poderia ser menor se o sistema não fizesse acessos ao disco rígido do computador para salvar as mensagens de *log* do processamento, pois este tipo de operação tem um alto custo computacional e consome um tempo considerável da execução do sistema (TANENBAUM, 2005).

Por outro lado, caso o sistema multiagente tivesse sido executado em máquinas diferentes (subconjuntos de agentes hospedados em máquinas diferentes), certamente o tempo de execução

Tabela 5.25: Tempos de execução por momento

	Tempos de	Execução
Momentos	Agente no Alimentador	
	0	8642
Momento 1	1	7542
	2	279
	0	2538
Momento 2	1	2279
	2	2367
	0	2341
Momento 3	1	2597
	2	116
	0	1317
Momento 4	1	760
	2	1820
	0	430
Momento 5	1	330
	2	1130
	0	1146
Momento 6	1	420
	2	1153
	0	180
Momento 7	1	210
	2	1440
	0	1870
Momento 8	1	1142
	2	320
	0	210
Momento 9	1	190
	2	1290
	0	640
Momento 10	1	190
	2	1290
	0	1231
Momento 11	1	320
	2	1520
	0	1139
Momento 12	1	310
	2	1127

seria maior, dado a necessidade de tráfego das mensagens via a rede de telecomunicação utilizada. Avaliar o desempenho do método neste tipo de cenário teria que levar em conta muitas características das redes, por exemplo, qual tecnologia utilizada, distância entre as máquinas, qual a arquitetura implementada pela rede, entre outras (KUROSE et al., 2007; COULOURIS; DOLLIMORE; KINDBERG, 2005).

O número de mensagens enviadas pelos agentes neste método computacional segue a equação:

$$4(numF \ x \ numS) + (numLo - numLi) + 4(numF \ x \ numLi)$$

Onde *numF* representa o número de *feederAgents*, *numS* o número de *switcherAgents*, *numLo* o número de *loaderAgents* que não são líderes de coalizão e finalmente, *numLi* o número de *loaderAgents* líderes.

Assim, o número de mensagens trocadas por momento para o exemplo de dezesseis barras é igual a 124 mensagens, sem levar em consideração as mensagens enviadas pelo agente *demandChanged* para alteração das cargas, pois a mesma varia para cada momento.

Finalizando, o sistema multiagente modelado serviu para o propósito do trabalho. Ele é um método distribuído, que se utiliza de inteligência computacional a partir da tomada de decisão distribuída dos agentes e serve como um sistema de controle também distribuído sobre o sistema elétrico, fazendo medidas e informando os demais agentes sobre o estado da rede elétrica.

Alterações dinâmicas do estado, representadas aqui pela mudança nos valores das cargas, forçou o sistema multiagente a reconfigurar a topologia da rede elétrica afim de cumprir restrições que mantém a configuração no espaço de soluções viáveis do problema. As restrições modeladas foram delimitadas pela convergência do método de varredura, utilizado para calcular o fluxo de carga, e por limites máximos de fluxo de corrente nos trechos da rede.

Quando em um primeiro momento a restrição era apenas a convergência do método de cálculo do fluxo de carga, e em seguida foram adicionados limites de fluxos de correntes nos trechos, o método computacional apresentado mostrou-se versátil e adaptável para a adição de novas restrições, sendo necessário para tanto algumas modificações pontuais em seu código fonte.

A técnica também apresenta um interessante potencial para ser estudada também a aplicações de otimização de *smart grids*, na busca por solucionar problemas reais típicos, tais como minimização das perdas, alocação de bancos de capacitores, qual o momento propício para

utilização de geração ou armazenamento distribuído de energia, entre outras.

Logo, sistemas multiagentes aplicados ao gerenciamento de sistemas elétricos de distribuição do tipo *smart grid* tem potencial para ser uma interessante ferramenta computacional a ser estudada para este tipo de cenário das redes elétricas do futuro.

6 Conclusões e Trabalhos Futuros

6.1 Conclusões

O futuro dos sistemas elétricos em geral, e das redes elétricas de distribuição em específico, combina avanços em sistemas distribuídos, inteligência artificial, controle e outras áreas, para prover uma rede de distribuição inteligente e com um nível de autonomia suficiente para se auto-gerenciar, provendo funcionalidades como reconfiguração para minimização das perdas, proteção, restauração do sistema, acompanhamento inteligente do consumo e demanda, dentre outras. Esta tecnologia convencionou-se chamar de *smart grids*, a próxima geração dos sistemas elétricos de potência.

A implementação destas funcionalidades em um ambiente de produção - uma rede elétrica real - passará, necessariamente, por uma série de testes e simulações, desenvolvimentos de protótipos, etc. Isso visa o desenvolvimento e aperfeiçoamento da melhor solução para determinada funcionalidade, avaliando questões referentes ao desempenho, custo, e outras, o que é necessário para selecionar a melhor proposta para a funcionalidade estudada.

Entre os tipos de simulação, uma das que demandam menor custo são as simulações computacionais, realizadas a partir de programas de computador que emulam as características e comportamentos do objeto simulado, com alguma margem de proximidade, e avaliam a qualidade das propostas testadas.

Este trabalho tratou da implementação de um sistema multiagente para simulação computacional de um sistema de distribuição elétrico do tipo *smart grid*, buscando executar a reconfiguração topológica do sistema a partir de dados de carga capturados de forma distribuída pelos agentes dispersos na rede elétrica.

A escolha pela utilização de sistemas multiagentes foi motivada pela interessante proximidade conceitual sobre como se espera que se dê o funcionamento de um *smart grid* e como 6.2 Trabalhos Futuros 108

um sistema multiagente é modelado e executado. Ambas as tecnologias compartilham características relacionadas com sistemas distribuídos e inteligência artificial, o que permite que aplicações de multiagentes para a simulação das redes elétricas inteligentes seja interessante e tenha um embasamento teórico-metodológico claro.

O sistema multiagente aqui proposto e apresentado foi constituído a partir da modelagem e implementação de agentes responsáveis pelo monitoramento de alimentadores, cargas e chaves de um sistema elétrico. Cada agente tem a sua funcionalidade definida e a comunicação entre eles se dá através da troca de mensagens. O objetivo para o qual eles foram criados foi realizar a reconfiguração topológica do sistema elétrico.

A implementação computacional contou com aporte teórico sobre sistemas multiagentes, sistemas distribuídos, inteligência artificial, sistemas elétricos de distribuição e *smart grids*. Diversas tecnologias foram utilizadas neste processo, desde linguagens de programação orientadas a objetos a *parsers* entre objetos de linguagens diferentes, *frameworks* para desenvolvimento e mais. A própria articulação destas várias tecnologias constituiu, em si, um trabalho interessante.

Os testes realizados nesta dissertação procuraram atingir alguns comportamentos esperados para o objetivo proposto: trocar informações sobre as cargas e permitir o cálculo do fluxo de carga com os dados coletados naquele momento; selecionar qual alimentador enviará energia para um conjunto de cargas; configurar a topologia do sistema elétrico, balanceando-o entre os alimentadores presentes; encontrar topologias viáveis para o sistema, inclusive saindo de um estado de inviabilidade para um factível; reconfiguração topológica *online*, com a mudança nos valores de potência nas cargas também sendo alterados de forma *online*, em qualquer momento da simulação.

Todos estes comportamentos puderam ser observados nas simulações aplicadas aos sistemas elétricos de testes, o que cumpre, por agora, os objetivos para o presente trabalho.

6.2 Trabalhos Futuros

O desenvolvimento das redes elétricas inteligentes abre uma gama de possibilidades para estudos de vários tipos, entre os quais pode-se citar as simulações computacionais, simulações físicas, testes com redes de comunicação, testes com tomada de decisão, desenvolvimento de novos algoritmos para alguma funcionalidade do *smart grid*, criação de novos equipamentos, entre outras.

6.2 Trabalhos Futuros 109

O trabalho desenvolvido nesta dissertação focou-se no estudo do relacionamento entre sistemas multiagentes, *smart grids*, as características comuns entre eles, e como isso embasa a proposta de utilização daquele para simulação computacional deste. Então, foi criada uma modelagem computacional para um sistema multiagente que executa a configuração topológica de um sistema elétrico de distribuição do tipo *smart grid* que posteriormente foi implementada.

A partir dos resultados positivos deste estudo, é possível agora extender o modelo para aplicação a alguma funcionalidade que tem a reconfiguração topológica como ferramenta para sua execução. Por exemplo, encontrar uma topologia que reduz as perdas do sistema elétrico; determinar o ajuste ótimo em equipamentos como tap de transformadores, bancos de capacitores ou reatores, e outros.

Outro foco de estudo também importante diz respeito à avaliação de como a comunicação distribuída e em "tempo-real" impactará nas funcionalidades dos sistemas elétricos. Uma possibilidade de trabalhos futuros seria hibridizar métodos convencionais de inteligência artificial utilizadas para otimização de sistemas elétricos, como busca tabu, algoritmos genéticos ou outra metaheurística, com sistemas multiagentes. As possibilidades deste foco partem do princípio de que os sistemas multiagentes poderiam alimentar um programa que executasse uma metaheurística de forma *online*, afim de obter boas configurações para aquela rede naquele estado.

Referências Bibliográficas

BARAN, M.; WU, F. Network reconfiguration in distribution systems for loss reduction and load balancing. *IEEE Transactions on Power Delivery*, v. 4, n. 2, p. 1401–1407, 1989. ISSN 0885-8977.

BELLIFEMINE, F. L.; CAIRE, G.; GREENWOOD, D. Developing Multi-Agent Systems with *JADE*. [S.l.]: Wiley, 2007. ISBN 9780470057476.

BI, T. et al. The implication and implementation of smart grid in China. In: IEEE. *Power and Energy Society General Meeting*. [S.l.], 2010. p. 1–5. ISSN 1944-9925.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. [S.l.]: Livraria Tempo Real Inform, 2006.

BORDINI, R. et al. A survey of programming languages and platforms for multi-agent systems. *Special Issue: Hot Topics in European Agent Research II Guest Editors: Andrea Omicini*, Citeseer, v. 30, p. 33–44, 2006.

BORGES, L. Python para desenvolvedores. Rio de Janeiro, ed, v. 1, 2009.

BRAY, T. et al. Extensible markup language (XML) 1.0. [S.l.]: W3C recommendation, 2000.

BROWN, R. Impact of Smart Grid on distribution system design. In: IEEE. *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century*, 2008 IEEE. [S.l.], 2008. p. 1–4. ISSN 1932-5517.

CASTRO, J.; ALENCAR, F.; SILVA, C. Engenharia de software orientada a agentes. *Capítulo*, v. 5, p. 245–242, 2006.

CIVANLAR, S. et al. Distribution feeder reconfiguration for loss reduction. *IEEE Transactions on Power Delivery*, v. 3, n. 3, p. 1217–1223, 1988. ISSN 0885-8977.

COMMISSION, E. European SmartGrids Technology Platform: Vision and Strategy for Europe's Electricity Networks of the Future. Belgium: European Communities, 2006. ISBN 92-79-01414-5.

CORMEN, T. H. et al. *Introduction to Algorithms, Second Edition*. [S.l.]: The MIT Press, 2009. ISBN 0262032937.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. *Distributed systems: concepts and design*. [S.l.]: Addison-Wesley Longman, 2005. ISBN 0321263545.

CROCKFORD, D. Json: The fat-free alternative to xml. In: *Proc. of XML*. [S.l.: s.n.], 2006. v. 2006.

DEITEL, H.; DEITEL, P. Java como programar; 4^a edição. *Porto Alegre*, 2003.

DELBEM, A.; CARVALHO, A. de; BRETAS, N. Main chain representation for evolutionary algorithms applied to distribution system reconfiguration. *Power Systems, IEEE Transactions on*, IEEE, v. 20, n. 1, p. 425–436, 2005. ISSN 0885-8950.

DIJKSTRA, E. W. A Note on Two Problems in Connection with Graphs. *Numerical Mathematics*, v. 1, p. 269–271, 1959.

DUAN, R.; DECONINCK, G. Multi-agent model and interoperability of a market mechanism of the Smart Grids. In: IEEE. *Network Operations and Management Symposium Workshops* (NOMS Wksps), 2010 IEEE/IFIP. [S.1.], 2010. p. 312–315.

ENERGY, E. C. *Energy 2020: A Strategy for Competitive, Sustainable and Secure Energy.* Belgium: Publications Office of the European Union, 2011. ISBN 978-92-79-18869-5.

FANNING, R. Distribution vision 2010: planning for automation. In: IEEE. *Power Systems Conference and Exposition*, 2004. *IEEE PES*. [S.l.], 2005. p. 1692–1693. ISBN 078038718X.

GARCIA, A. C. B.; SICHMAN, J. S. Agentes e Sistemas Multiagentes. In: REZENDE, S.O. *Sistemas Inteligentes: fundamentos e aplicações*. S?o Paulo: Editora Manole Ltda, 2003. ISBN 85-204-1683-7.

GLOVER, F. Tabu search: A tutorial. *Interfaces*, INFORMS, v. 20, n. 4, p. 74–94, 1990.

GRAINGER, J. J.; LEE, S. H. Capacity Release by Shunt Capacitor Placement on Distribution Feeders: A New Voltage-Dependent Model. *IEEE Transactions on Power Apparatus and Systems*, v. 101, n. 05, p. 1236–1244, 1982.

GUERRA, W.; UNSIHUAY-VILA, C.; KAGAN, N. Impacto da Geração Distribuída na Reconfiguração de Sistemas de Distribuío de Energia Elétrica Utilizando Busca em Vizinhança Variável. In: *Anais do XLIII Simpósio Brasileiro de Pesquisa Operacional*. S?o Paulo: SOBRAPO, 2011.

HORLING, B.; LESSER, V. A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, Cambridge Univ Press, v. 19, n. 04, p. 281–316, 2004. ISSN 0269-8889.

JIANG, Z. et al. A vision of smart transmission grids. In: IEEE. *Power & Energy Society General Meeting*, 2009. *PES'09. IEEE.* [S.l.], 2009. p. 1–10. ISSN 1944-9925.

KAGAN, N.; OLIVEIRA, C. de; ROBBA, E. *Introdução aos sistemas de distribuição de energia elétrica*. [S.l.]: Edgard Blucher, 2005. ISBN 8521203551.

KEMENES, A.; FORSBERG, B.; MELACK, J. Methane release below a tropical hydroelectric dam. *Geophysical Research Letters*, American Geophysical Union, v. 34, n. 12, p. L12809, 2007. ISSN 0094-8276.

KERSTING, W. Distribution system modeling and analysis. [S.1.]: CRC, 2006.

KOESTLER, A. O fantasma da máquina. [S.l.]: Zahar Editôres, 1969.

KUROSE, J. et al. *Redes de computadores ea Internet: uma abordagem top-down.* [S.l.]: Pearson Addison Wesley, 2007.

KWON, S. et al. Current development and future plan for smart distribution grid in Korea. In: IET. *SmartGrids for Distribution*, 2008. IET-CIRED. CIRED Seminar. [S.1.], 2008. p. 1–4. ISSN 0537-9989.

LU, J.; AI, Q. Research on smart grid in China. In: IEEE. *Transmission & Distribution Conference & Exposition: Asia and Pacific*, 2009. [S.l.], 2009. p. 1–4.

MANTOVANI, J.; CASARI, F.; ROMERO, R. Reconfiguração de sistemas de distribuição radiais utilizando o critério de queda de tensão. *Revista Controle e Automação*, *Sociedade Brasileira de Automática*, *SBA*, SciELO Brasil, v. 11, n. 03, 2000.

MCARTHUR, S. et al. Multi-agent systems for power engineering applications - Part I: concepts, approaches, and technical challenges. *Power Systems, IEEE Transactions on*, IEEE, v. 22, n. 4, p. 1743–1752, 2007. ISSN 0885-8950.

MCARTHUR, S. et al. Multi-agent systems for power engineering applications - Part II: technologies, standards, and tools for building multi-agent systems. *Power Systems, IEEE Transactions on*, IEEE, v. 22, n. 4, p. 1753–1759, 2007. ISSN 0885-8950.

MICHALEWICZ, Z. Genetic algorithms+ data structures. [S.l.]: Springer, 1996.

PÁDUA, W. P. F. Engenharia de software: fundamentos, métodos e padrões. [S.l.]: Livros Técnicos e Científicos, 2003.

RABUSKE, M. Introdução à teoria dos grafos. [S.l.]: Editora da UFSC, 1992.

RAYWARD-SMITH, V. et al. *Modern heuristic search methods*. [S.l.]: Wiley, 1996. ISBN 0471962805.

REZENDE, S. *Sistemas Inteligentes: fundamentos e aplicações*. S?o Paulo: Editora Manole Ltda, 2003. ISBN 85-204-1683-7.

RODRIGUES, F. F. C.; BORGES, C. L.; FALCÃO, D. M. Programação da contratação de energia considerando geração distribuída e incertezas na previsão de demanda. *Controle e Automação*, scielo, v. 18, p. 361 – 371, 09 2007. ISSN 0103-1759.

RUGTHAICHAROENCHEEP, N.; SIRISUMRANNUKUL, S. Feeder reconfiguration for loss reduction in distribution system with distributed Generators by Tabu search. Asian Institute of Technology, 2008.

RUMLEY, S. et al. Multi-agent approach to electrical distribution networks control. In: IEEE. *Computer Software and Applications, 2008. COMPSAC'08. 32nd Annual IEEE International.* [S.l.], 2008. p. 575–580. ISSN 0730-3157.

RUSSEL, S.; NORVIG, P. Inteligência Artificial. Brasil: Ed. Campus, 2004.

SANTOS, R. *Introdução à programação orientada a objetos usando Java*. [S.l.]: Elsevier, 2003.

SARAIVA, F. d. O.; SOUSA, A.; ASADA, E. Implementação distribuída do algoritmo de dijkstra através de sistemas multiagentes. *Anais do X Simpósio Brasileiro de Automação Inteligente - X SBAI*, Sociedade Brasileira de Automática - SBA, X, p. 51–56, 2011. ISSN 2175-8905.

SCHMIDT, H. et al. Fast reconfiguration of distribution systems considering loss minimization. *Power Systems, IEEE Transactions on*, IEEE, v. 20, n. 3, p. 1311–1319, 2005. ISSN 0885-8950.

SHOHAM, Y.; LEYTON-BROWN, K. Multiagent systems: Algorithmic, game-theoretic, and logical foundations. [S.l.]: Cambridge Univ Pr, 2009.

SILVA, B. da. *Utilização de um Sistema Multi-Agentes em Redes de Comunicação para a Proteção Digital de Distância Adaptativa*. Tese (Doutorado) — Universidade de São Paulo, 2009.

TANENBAUM, A. S. Sistemas operacionais modernos. [S.l.]: São Paulo:, Prentice-Hall, 2005.

WEI, X.; YU-HUI, Z.; JIE-LIN, Z. Energy-efficient distribution in smart grid. In: IEEE. Sustainable Power Generation and Supply, 2009. SUPERGEN'09. International Conference on. [S.1.], 2009. p. 1–6.

WEISS, G. *Multiagent systems: a modern approach to distributed artificial intelligence*. [S.l.]: The MIT press, 1999. ISBN 0262731312.

WOOLDRIDGE, M. An introduction to multiagent systems. England: Wiley, 2009. ISBN 04-70-51946-0.