

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

GABRIEL DE OLIVEIRA RAMOS

**Redução do espaço de busca de estruturas
de coalizão a partir de informações sobre o
domínio: uma aplicação em *smart grids***

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof^ª. Dra. Ana Lúcia Cetertich Bazzan
Orientadora

Porto Alegre, março de 2013

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Ramos, Gabriel de Oliveira

Redução do espaço de busca de estruturas de coalizão a partir de informações sobre o domínio: uma aplicação em *smart grids* / Gabriel de Oliveira Ramos. – Porto Alegre: PPGC da UFRGS, 2013.

73 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2013. Orientadora: Ana Lúcia Cetertich Bazzan.

1. Inteligência artificial. 2. Teoria dos jogos. 3. Smart grids. I. Bazzan, Ana Lúcia Cetertich. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Pós-Graduação: Prof. Aldo Bolten Lucion

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do PPGC: Prof. Álvaro Freitas Moreira

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“Aprender é a única coisa de que a mente nunca se cansa,
nunca tem medo e nunca se arrepende.”*

— LEONARDO DA VINCI

AGRADECIMENTOS

Início este trabalho agradecendo à minha família. Agradeço aos meus pais, Adilson e Sirlei, que sempre foram um exemplo de vida para mim. Agradeço por me agraciarem com um amor tão legítimo e incondicional, sem o qual eu não poderia viver. Agradeço por nunca terem deixado faltar nada na minha vida, mesmo quando isto lhes trouxe dificuldades. Agradeço à minha namorada, Daiane, por sempre estar ao meu lado, me amando e respeitando, e por compreender minhas escolhas. Agradeço ao meu irmão, que apesar de mais jovem sempre me surpreendeu com seus gestos e pensamentos. Agradeço a estas pessoas por sempre estarem ao meu lado, por acreditarem em mim e nos meus sonhos, por terem me dado força nas horas que precisei, pela compreensão que sempre tiveram nas horas difíceis e pela admiração que sempre demonstraram por mim. Vocês são as pessoas mais importantes da minha vida e realmente eu não sei o que seria de mim sem vocês.

Agradeço profundamente à minha orientadora, professora Ana Bazzan. O fato dela ter aceitado me orientar sob circunstâncias tão adversas, demonstra o quanto ela acredita na minha capacidade e perseverança. Agradeço pelas valiosas discussões e conselhos, que possibilitaram meu crescimento intelectual. Agradeço também pelos “puxões de orelha”, que sempre serviram de incentivo para que eu busque a excelência científica através de meus trabalhos. Finalmente, agradeço pelo exemplo que ela é, cuja dedicação e paixão pelo seu trabalho sempre serviram de incentivo para mim.

Agradeço aos meus amigos de mestrado Anderson Tavares, Cássio Oliveira, Cristiano Galafassi, Daniel Epstein, Diego Noble, Marcelo Hama, Mariana Mendoza, Renato Pereira, Sérgio Montazzolli e outros que, porventura, eu tenha esquecido de mencionar, pela oportunidade de conviver em um grupo tão distinto de indivíduos que em muito contribuiu para meu crescimento pessoal e acadêmico. Tenho certeza de que, sem as discussões, conselhos e companhia destes, eu não teria conseguido chegar aonde cheguei.

Agradeço a todos meus outros amigos e familiares, por terem estado sempre ao meu lado, por terem me apoiado nos momentos difíceis, por terem me animado nos momentos que precisei e, sobretudo, por terem aceitado minha ausência, não deixando, jamais, que isto afetasse nossa amizade.

Agradeço à Universidade Federal do Rio Grande do Sul (UFRGS), em especial ao Instituto de Informática (INF) e ao Programa de Pós-Graduação em Computação (PPGC), pela oportunidade de estudar em um centro de pesquisa tão importante e competente, do qual fazem parte alguns dos mais importantes pesquisadores do Brasil.

Agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela bolsa de mestrado a mim concedida, a qual possibilitou que eu me dedicasse ao curso de mestrado em tempo integral.

Finalmente, agradeço a todas as outras pessoas que me incentivaram e de alguma forma contribuíram para que eu seguisse meus sonhos e atingisse meus objetivos.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE SÍMBOLOS	9
LISTA DE FIGURAS	11
LISTA DE TABELAS	12
LISTA DE ALGORITMOS	13
RESUMO	14
ABSTRACT	15
1 INTRODUÇÃO	16
1.1 Motivação	17
1.2 Objetivos	18
1.3 Publicações relacionadas	18
1.4 Organização da dissertação	19
2 SMART GRIDS	20
2.1 Contextualização	20
2.2 Conceitos	21
2.3 Características e benefícios	22
2.4 Desafios	24
2.5 Tecnologias	25
2.5.1 Geração e armazenamento distribuídos	25
2.5.2 Veículos elétricos	26
2.6 Adoção	26
2.7 Discussão	27
3 FORMAÇÃO DE COALIZÕES	28
3.1 Contextualização	28
3.2 O problema de formação de coalizões	28
3.3 Geração de estruturas de coalizão	29
3.3.1 Algoritmos baseados em heurísticas	31
3.3.2 Algoritmos baseados em programação dinâmica	32
3.3.3 Algoritmos <i>anytime</i>	37
3.4 Discussão	42

4	FORMAÇÃO DE COALIZÕES EM SMART GRIDS	43
4.1	Motivação	43
4.2	Coalizões de aerogeradores	44
4.3	Coalizões de veículos elétricos	45
4.4	Formação dinâmica de coalizões entre DERs	47
4.5	Discussão	49
5	ABORDAGEM	50
5.1	Modelagem do problema	50
5.2	Informação sobre o domínio	51
5.3	Podando o espaço de busca	52
5.4	Algoritmo CPCSG	54
5.5	Discussão sobre a abordagem	58
6	EXPERIMENTOS	59
6.1	Metodologia	59
6.2	Desempenho do CPCSG	61
6.3	Tempo para encontrar a CS^*	63
6.4	Comparação com outros algoritmos para encontrar a CS^*	65
6.5	Discussão	67
7	CONCLUSÕES	68
7.1	Contribuições	68
7.2	Conclusões	68
7.3	Trabalhos futuros	69
	REFERÊNCIAS	70

LISTA DE ABREVIATURAS E SIGLAS

CF	<i>Coalition Formation</i>
CFG	<i>Characteristic Function Game</i>
CPCSG	<i>Constraint-based Pruning of Coalition Structures Graph</i> (algoritmo)
CSG	<i>Coalition Structure Generation</i>
CSS1	<i>Coalition-Structure-Search-1</i> (algoritmo)
CVPP	<i>Cooperative Virtual Power Plant</i>
DCF	<i>Dynamic Coalition Formation</i>
DCF-S	<i>Dynamic Coalition Formation Scheme</i>
DER	<i>Distributed Energy Resource</i>
DP	<i>Dynamic Programming</i> (algoritmo)
EPRI	<i>Electric Power Research Institute</i>
EV	<i>Electric Vehicle</i>
GB	gigabyte
GHz	gigahertz
IA	Inteligência Artificial
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IDP	<i>Improved Dynamic Programming</i> (algoritmo)
IDP-IP	<i>Improved Dynamic Programming for Integer Partition</i> (algoritmo)
IP	<i>Integer Partition</i> (algoritmo)
km	Quilômetro
kW	Quilowatt
kWh	Quilowatt-hora
MME	Ministério de Minas e Energia
MW	Megawatt
NFG	<i>Normal Form Game</i>
PHEV	<i>Plug-in Hybrid Electric Vehicle</i>

RAM *Random Access Memory*
SK Shehory e Krauss (algoritmo)
US\$ Dólar (moeda americana)
V2G *Vehicle-To-Grid*
VPP *Virtual Power Plant*

LISTA DE SÍMBOLOS

α	Distância máxima entre dois agentes de uma mesma coalizão
β	Limite da solução ótima
γ	Tamanho máximo desejado para uma coalizão (função característica)
δ	Quantidade de energia que a rede espera que cada coalizão seja capaz de fornecer (função característica)
ϵ	Incentivo financeiro máximo que a rede pretende dar a uma coalizão (função característica)
A	Conjunto de agentes
a	Quantidade de agentes ($ A $)
avg_s	$v(C)$ médio da lista CL_s (algoritmo IP)
\mathcal{B}	Conjunto dos conjuntos de restrições binárias de todos os agentes
B_i	Conjunto de agentes que possuem uma restrição binária com o agente i
C	Coalizão
C_k	Uma coalizão arbitrária de tamanho k
\mathcal{CL}	Conjunto de todos os CL_s , para $s \in \{1, \dots, a\}$ (algoritmo IP)
CL_s	Conjunto de coalizões de tamanho s , para $s \in \{1, \dots, a\}$ (algoritmo IP)
\mathcal{CS}	Conjunto de todas as estruturas de coalizão possíveis
CS	Estrutura de coalizão
CS^*	Estrutura de coalizão ótima
d_j^i	Distância entre dois agentes i e j
E^*	Conjunto de todas as arestas essenciais para a representação do problema (algoritmo DP)
$E^{s's''}$	Conjunto de arestas resultante da divisão de uma coalizão de tamanho $(s' + s'')$ em duas de tamanho s' e s'' (algoritmo DP)
E_C	Quantidade de energia que a coalizão C tem disponível (função característica)
f_1	Tabela do algoritmo DP que contém a solução de cada coalizão (algoritmo DP)
$f_1[C]$	Solução da coalizão C (algoritmo DP) (algoritmo DP)

f_2	Tabela do algoritmo DP que contém o valor da solução de cada coalizão (algoritmo DP)
$f_2[C]$	Valor da solução da coalizão C (algoritmo DP)
$F(G_n)$	Conjunto de estruturas de coalizão que possuem a configuração G_n (algoritmo IP)
\mathcal{G}_{CS}	Conjunto de todas as configurações possíveis dado CS (algoritmo IP)
G_n	Configuração n , dado $n \in \{1, \dots, CS \}$ (algoritmo IP)
\mathcal{J}	Conjunto dos conjuntos de restrições conjuntas
J_x	Conjunto de restrições conjuntas
J^*	Maior conjunto $J_x \in \mathcal{J}$
max_s	$v(C)$ máximo da lista CL_s (algoritmo IP)
\mathcal{N}	Conjunto de todos os N_n (algoritmo IP)
N_n	Conjunto de estruturas de coalizão que possuem a configuração G_n (algoritmo IP)
p	Preço normal que a rede costuma pagar por uma unidade de energia (função característica)
$V(CS)$	Valor da estrutura de coalizão CS
$v(C)$	Valor da coalizão C

LISTA DE FIGURAS

2.1	Diagrama conceitual de <i>smart grids</i>	22
3.1	Grafo de estruturas de coalizão para quatro agentes	30
3.2	Grafo de estruturas de coalizão para quatro agentes (IP e IDP)	35
3.3	Representação do espaço de busca adotada pelo algoritmo IP	40
5.1	Grafo para quatro agentes	52
5.2	Cenário hipotético utilizado no exemplo 1	57
6.1	Tempo de execução do algoritmo CPCSG, considerando 10 a 20 agentes (séries) e com variação no tamanho de J^* entre 2 e a (eixo x)	61
6.2	Quantidade de estruturas de coalizão que podem ser podadas do espaço de busca, considerando 10 a 20 agentes (séries) e variando o tamanho de J^* de 2 a a (eixo x)	62
6.3	Quantidade de estruturas de coalizão que podem ser podadas do espaço de busca, considerando 30 a 100 agentes (séries), somente dezenas, e variando o tamanho de J^* de 2 a a (eixo x)	62
6.4	Tempo de execução requerido para encontrar a solução ótima, após a realização da poda dos níveis, considerando 10 a 20 agentes (séries) e variando o tamanho de $ J^* $ de 2 a a (eixo x)	64
6.5	Comparação dos tempos de execução dos três algoritmos, IDP, IP e CPCSG-IP (IP com CPCSG), para 10 a 20 agentes, variando o tamanho de J^* de 2 até a	66

LISTA DE TABELAS

3.1	Exemplo de como o algoritmo DP calcula as tabelas f_1 e f_2	34
3.2	Redução do limite β em função da busca adicional	39
5.1	Exemplo de funcionamento do algoritmo CPCSG para o exemplo 1	57

LISTA DE ALGORITMOS

3.1	Algoritmo DP para geração de estruturas de coalizão	33
3.2	Algoritmo IDP para geração de estruturas de coalizão	36
3.3	Algoritmo CSS1 para geração de estruturas de coalizão	38
5.1	Algoritmo CPCSG para encontrar a quantidade de níveis que podem ser podados do grafo de estruturas de coalizão	55

RESUMO

Redes elétricas desempenham um papel fundamental no que tange à distribuição de energia elétrica. Entretanto, as redes elétricas convencionais são muito antigas, apresentando uma série de deficiências e inseguranças. Neste cenário surgem as redes elétricas inteligentes, mais conhecidas como *smart grids*. *Smart grids* são uma evolução para as redes elétricas tradicionais, apresentando como grande diferencial a presença intensiva de tecnologia de ponta para monitorar cada elemento que a compõe. Uma das principais características de *smart grids* é seu fluxo bidirecional de energia e informação, que permite a qualquer elemento tanto consumir quanto fornecer energia para a rede, seja um aerogerador ou mesmo uma residência. Tal característica vai de encontro à necessidade de se distribuir a produção energética, tornando-a mais robusta e tolerante a falhas.

Uma tecnologia que surgiu em meio ao desenvolvimento de *smart grids* denomina-se Veículo-Para-Rede (V2G, do inglês *Vehicle-To-Grid*). Através de sessões V2G, veículos elétricos (EVs, em inglês *electric vehicles*) podem vender a energia de suas baterias para a rede, obtendo lucro com este procedimento. Existem duas vantagens nesta tecnologia. Por um lado, proprietários de EVs podem obter lucro com a venda de energia, reduzindo os custos de se manter seu veículo. Por outro lado, a rede como um todo se beneficia, pois as baterias podem ser utilizadas para aumentar a estabilidade da rede. Entretanto, para que estas vantagens sejam expressivas, é necessário utilizar-se de mecanismos para aumentar a eficiência do processo V2G, uma vez que baterias são muito caras. Uma alternativa que tem sido muito explorada é a formação de coalizões entre os EVs.

A proposta deste trabalho é utilizar informações sobre o domínio de *smart grids* de modo a impor restrições no processo de formação de coalizões de EVs, visando à redução do espaço de busca de estruturas de coalizão. Especificamente, estabelece-se a distância máxima que pode haver entre dois EVs de uma mesma coalizão, através da qual é possível identificar e podar porções inválidas do espaço de busca. Para tanto, é proposto o algoritmo CPCSG, capaz de identificar restrições entre os EVs e de podar o espaço de busca. A abordagem proposta pode ser utilizada em conjunto com algoritmos de geração de estruturas de coalizão para torná-los mais rápidos e eficientes.

Com base em experimentos, percebe-se que a abordagem proposta proporciona um ganho notável de desempenho e uma redução expressiva no uso de memória em relação a outros algoritmos para geração de estruturas de coalizão. Em geral, quanto mais restritiva a rede e quanto maior o número de agentes, maior será o percentual do espaço de busca passível de ser podado. Resultados mostram, ainda, que quando comparada com outros algoritmos de geração de estruturas de coalizão, a técnica proposta chega a superar o tempo dos demais em diversas ordens de magnitude.

Palavras-chave: Inteligência artificial, teoria dos jogos, smart grids.

Reduction of coalition structures' search space based on domain information: an application in smart grids

ABSTRACT

Electric grids play a key role in the energy distribution process. However, conventional grids are very old, which causes the onset of weaknesses and uncertainties. In such a scenario the smart grid concept arises. Smart grids are an evolution to the ageing electric grids, whose major breakthrough is the intensive use of technology to monitor every element that comprises it. One of the main features of smart grids is its bi-directional flow of electricity and information, which allows any element to consume and even supply energy to the grid, regardless of being a wind turbine or even a residence. Such a characteristic meets the need to make the energy production more distributed, making it more robust and fault tolerant.

Amidst the development of smart grids emerged the concept of Vehicle-To-Grid (V2G). Through V2G sessions, electric vehicles (EVs) can sell the surplus energy of their batteries to the grid, making a profit. Two advantages arise from this technology. First, EVs' owners can make a profit from the sale of energy, reducing their vehicles' maintenance cost. Second, the network as a whole is benefited as batteries could be used to increase the network stability. However, in order to benefit from such advantages, it is necessary the use mechanisms to increase the efficiency of the V2G process, since batteries are very expensive. One way that has been explored is the coalition formation among EVs.

The proposal of this work is to use smart grids' domain information to impose constraints on the coalition formation process in order to reduce the coalition structures' search space. Specifically, we define a maximum distance that can exist between two EVs of a given coalition, through which it is possible to identify and prune invalid portions of the search space. To this end, we propose the CPCSG algorithm, which has the capability of identifying constraints among EVs and pruning the search space. The proposed approach can be used together with coalition structure generation algorithms to make them faster and more efficient.

Based on experiments, it can be seen that our approach provides a noticeable performance gain and a significant memory usage reduction compared to other coalition structure generation algorithms. In general, the more restrictive the grid and the greater the number of agents, the greater the percentage of the search space that can be pruned. Results also show that when compared with other coalition structure generation algorithms, the proposed technique is able to overcome the other in time by several orders of magnitude.

Keywords: artificial intelligence, game theory, smart grids.

1 INTRODUÇÃO

Energia elétrica é um recurso fundamental para a sociedade contemporânea. Questões estratégicas para a economia, como telecomunicações, transportes e atividades industriais dependem fortemente de energia elétrica para existir. Uma peça fundamental, que permite o acesso à energia elétrica da forma como conhecemos, é a rede elétrica. Uma rede elétrica é, de forma bem genérica, a infraestrutura que interliga os produtores de energia elétrica aos consumidores.

Apesar de sua importância, as redes elétricas evoluíram muito pouco desde que foram projetadas. Em contrapartida, a demanda por energia cresceu de forma descomunal. Para atender a esta demanda, a maioria dos investimentos do setor foram destinados à construção de novas, e grandes, usinas de produção de energia. Estas políticas de expansão promoveram matrizes energéticas compostas por relativamente poucos produtores, os quais são responsáveis por abastecer uma quantidade expressiva de consumidores. Como consequência, essa infraestrutura tornou-se muito ineficiente, defectível e insegura. Por outro lado, o fornecimento de energia de qualidade, de forma ininterrupta e segura, tornou-se uma necessidade para a economia.

Neste cenário surge o conceito de *smart grids*, ou redes elétricas inteligentes. *Smart grids*, de acordo com o U. S. Department of Energy (2003), são redes elétricas automatizadas que, através do monitoramento e controle intensivos de cada elemento que as compõem, são capazes de fornecer energia de forma eficiente e confiável. Uma das principais características de *smart grids* é o fluxo bidirecional de energia e comunicação entre seus elementos. Isto permite, por exemplo, que um consumidor saiba, em tempo real, o preço que está pagando pela energia que consome. Outra possibilidade interessante é que qualquer elemento pode tanto fornecer quanto consumir energia, de modo que, por exemplo, residências com geração própria de energia possam vender sua produção excedente para a rede.

Smart grids são de natureza intrinsecamente distribuída, com base em engenharia, comunicação, inteligência distribuída, automação e troca de informações (RAMCHURN et al., 2012). Tal característica enseja, entre outros, a distribuição da produção de energia entre diversos elementos na rede. Desta forma, grandes produtores poluentes poderiam ser substituídos por produtores menores baseados em energias renováveis, como aerogeradores e painéis solares. *Smart grids*, portanto, estão fortemente ligados à ideia de uma sociedade mais sustentável, com base na eficiência energética e no uso de fontes energéticas renováveis e menos agressivas ao ambiente.

Tratando-se de um conceito recente e inovador, *smart grids* proporcionam a criação de uma série de tecnologias para auxiliar no aumento de sua eficiência. Um conceito interessante que surgiu é o de Veículo-Para-Rede, ou simplesmente V2G (acrônimo de *Vehicle-To-Grid*). Em sessões V2G, aproveita-se a bateria de veículos elétricos, EVs (acrônimo

de *electric vehicles*), para armazenar e fornecer energia para a rede (KEMPTON; TOMIĆ, 2005). Este mecanismo é importante em situações onde a matriz energética depende diretamente de fontes de energia renováveis intermitentes, como é o caso de aerogeradores e painéis solares. Por intermitência, neste caso, entende-se que estas fontes não são capazes de produzir uma quantidade constante de energia, pois sua produção é condicionada a fatores climáticos. Tal característica, se não for tratada adequadamente, pode gerar grandes transtornos à rede, como desabastecimento. Em tais cenários, a energia armazenada na bateria dos EVs pode ser fornecida à rede prontamente, sempre que a produção não for capaz de atender à demanda. Ambos os lados são beneficiados no fornecimento V2G: a rede pode aumentar sua confiabilidade e os donos dos EVs podem obter lucro com a venda de sua energia.

1.1 Motivação

O mecanismo V2G traz uma série de vantagens para a rede. No entanto, participar de sessões V2G de uma maneira lucrativa e eficiente não é uma tarefa trivial para os EVs. Segundo Pudjianto et al. (2007), ao atuar em sessões V2G, os EVs se comprometem com a rede a fornecer determinada quantidade de energia. Porém, ao atuar individualmente, os EVs se mostram pouco aptos a cumprir o compromisso firmado com a rede, dado que sua disponibilidade é imprevisível (o proprietário pode precisar do veículo a qualquer momento) e que sua capacidade energética (de sua bateria) é pequena.

Diversas abordagens têm sido propostas para solucionar este problema. Uma abordagem particularmente interessante é a formação de coalizões entre EVs, conforme tratado em Chalkiadakis et al. (2011), Vasirani et al. (2011), Kamboj et al. (2010, 2011), Matta et al. (2011) e Mihailescu et al. (2011a, 2011b, 2011c). A vantagem deste tipo de abordagem é que, ao fornecer energia em conjunto, os EVs aumentam a probabilidade de cumprir seu compromisso com a rede. Isto ocorre porque os EVs podem se coordenar para fornecer previsões mais precisas acerca de sua disponibilidade. Além disso, a quantidade de energia a ser fornecida também cresce, o que diminui consideravelmente a quantidade de elementos com quem a rede deve negociar a compra de energia. Tal abordagem, portanto, tem se mostrado como um meio eficiente de aumentar a lucratividade dos EVs (PUDJIANTO et al., 2007; KEMPTON; TOMIĆ, 2005).

Formação de coalizões é um tópico de pesquisa muito importante no campo de sistemas multiagentes. Uma coalizão é um grupo de agentes que decide cooperar entre si para atingir um objetivo em comum. O particionamento do conjunto de agentes em coalizões disjuntas e exaustivas é denominado estrutura de coalizão. De acordo com Sandholm et al. (1999), o processo de formação de coalizões consiste em três etapas: cálculo do valor de cada coalizão possível, geração de estruturas de coalizão e divisão da utilidade entre os agentes. No contexto em discussão, o foco é a geração de estruturas de coalizão. O objetivo desta etapa é encontrar a estrutura de coalizão ótima, ou seja, a que apresenta a maior utilidade global. Tal tarefa foi provada ser NP-completa (SANDHOLM et al., 1999).

A literatura que trata da formação de coalizões em *smart grids*, no entanto, resolve o problema de forma muito ingênua. Nenhum dos trabalhos anteriormente mencionados busca uma solução ótima, nem sequer garantem que sua solução fica próxima do ótimo. A consequência da forma como estes trabalhos tratam o problema é que a solução gerada não tem garantia de qualidade. Tal consequência pode reduzir drasticamente a eficiência da rede.

Desta forma, para investigar com maior precisão os benefícios de se formar coalizões

de EVs em *smart grids*, é necessário utilizar-se de abordagens que encontrem soluções ótimas. Considerando a dinamicidade de *smart grids*, seria necessário um algoritmo rápido, capaz de entregar soluções no menor tempo possível. Neste sentido, pode ser utilizado o IP (RAHWAN et al., 2007), algoritmo *anytime* considerado o estado da arte para geração de estruturas de coalizão. A vantagem deste algoritmo é que ele é capaz de gerar uma solução subótima (dentro de um limite da solução ótima) rapidamente. Em seguida, a solução subótima é aprimorada enquanto houver tempo de processamento disponível ou até que a solução ótima seja encontrada. Entretanto, embora seja o estado da arte, este algoritmo é capaz de encontrar soluções ótimas em um tempo praticável para no máximo 20 agentes. Isto pode explicar o fato de que todas as abordagens de formação de coalizões em *smart grids* têm tratado este problema apenas de forma ingênua ou *ad hoc*.

1.2 Objetivos

A proposta deste trabalho é otimizar o processo de formação de coalizões entre EVs em *smart grids*. Considerando a dificuldade de se gerar soluções ótimas em um curto espaço de tempo, métodos alternativos podem ser utilizados. Uma alternativa viável é utilizar-se de informações sobre o domínio de *smart grids* de forma a viabilizar a poda do espaço de busca de estruturas de coalizão.

O objetivo deste trabalho, portanto, é utilizar informações sobre o domínio de *smart grids* para reduzir o espaço de busca de estruturas de coalizão. Especificamente, busca-se investigar como coalizões podem ser formadas, respeitando-se restrições físicas impostas pela rede. No contexto deste trabalho, através do uso de informações sobre o domínio é possível estabelecer critérios que viabilizem a poda de porções do espaço de busca, sem comprometer, de forma alguma, a solução ótima.

Desta forma, é proposto o algoritmo *constraint-based pruning of coalition structures graph* (CPCSG). Seu funcionamento consiste, basicamente, em identificar restrições conjuntas entre agentes, ou seja, agentes que não podem formar coalizões entre si, para então podar soluções inviáveis do espaço de busca. A restrição, neste caso, refere-se à distância física máxima que pode haver entre dois agentes de uma mesma coalizão, restrição esta imposta diretamente pela rede elétrica. O algoritmo CPCSG é incluído como um pré-processamento do algoritmo IP, reduzindo o espaço de busca antes que a busca propriamente dita comece. Desta forma, a solução ótima pode ser encontrada em um tempo menor.

Nos experimentos realizados, constatou-se que a abordagem proposta é eficiente. De uma forma geral, quanto mais restritiva for a rede elétrica e quanto maior o número de agentes, maior será o percentual do espaço de busca a ser podado. Comparada com outros algoritmos de CSG, nossa técnica se mostra bastante eficiente, chegando a superar os demais no quesito tempo em diversas ordens de magnitude, sobretudo quando o conjunto de restrições entre os agentes é suficientemente grande. Adicionalmente, o algoritmo CPCSG apresenta um desempenho bastante satisfatório, realizando sua tarefa em uma pequena fração de tempo.

1.3 Publicações relacionadas

A abordagem proposta por esta dissertação gerou, até o momento, as seguintes publicações:

- RAMOS, G. de O.; BAZZAN, A. L. C. Reducing the Search Space of Coalition

Structures based on Smart Grids' Domain Information. In: BALSÁ, J.; FUENTES-FERNÁNDEZ, R. (Ed.). *Proceedings of 8th Ibero-American Workshop on Multi-Agent Systems (IBERAGENTS 2012)*. Cartagena de Indias, Colombia: [s.n.], 2012. p. 1-11. Disponível em: <<http://www.inf.ufrgs.br/maslab/pergamus/pubs/Ramos&Bazzan2012iberagents.pdf>>.

- RAMOS, G. de O.; BAZZAN, A. L. C. Reduction of Coalition Structure's Search Space based on Domain Information: an Application in Smart Grids. In: *2012 Third Brazilian Workshop on Social Simulation (BWSS)*. Curitiba, Brasil: [s.n.], 2012. p. 112–119. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6462824&isnumber=6462694>>.

1.4 Organização da dissertação

Esta dissertação está organizada em sete capítulos:

Capítulo 1: Introduce e motiva o tema deste trabalho. Neste capítulo o objetivo é definido e uma visão geral da abordagem proposta é discutida, juntamente dos principais resultados.

Capítulo 2: Apresenta uma visão geral de *smart grids*, descrevendo algumas das inúmeras possibilidades que surgem neste âmbito.

Capítulo 3: Fornece uma revisão bibliográfica detalhada sobre formação de coalizões em sistemas multiagentes. Questões referentes à eficiência destes métodos são discutidas e os principais algoritmos para geração de estruturas de coalizão são apresentados.

Capítulo 4: Descreve os trabalhos relacionados. Neste capítulo, algumas das principais abordagens que utilizam coalizões em *smart grids* são apresentadas e discutidas.

Capítulo 5: Apresenta o algoritmo proposto neste trabalho, capaz de reduzir o espaço de busca de estruturas de coalizão.

Capítulo 6: Realiza uma investigação a respeito do comportamento da abordagem proposta. Trata de questões referentes ao desempenho do algoritmo de poda e à sua influência na busca da solução ótima, apresentando comparativos com outros algoritmos.

Capítulo 7: Apresenta as principais contribuições e conclusões obtidas a partir da elaboração deste trabalho, bem como algumas sugestões de trabalhos futuros.

2 SMART GRIDS

Energia elétrica é um recurso essencial para a sociedade contemporânea. Questões como prosperidade econômica, saúde pública, segurança e transportes dependem fortemente de energia elétrica para existir. Apesar de sua importância, no entanto, a infraestrutura responsável pela distribuição e fornecimento de energia elétrica evoluiu muito pouco desde que foi projetada. Como consequência, essa infraestrutura é muito ineficiente e propensa a falhas. Neste sentido, surge o conceito de *smart grids* como uma solução para estes problemas.

Neste capítulo é apresentada uma visão geral de *smart grids*. Inicialmente, a Seção 2.1 contextualiza o cenário que motivou o surgimento de *smart grids*. O conceito de *smart grids* é detalhado na Seção 2.2. As Seções 2.3 e 2.4 descrevem, respectivamente, as características e os desafios decorrentes do desenvolvimento de *smart grids*. Na Seção 2.5, são apresentadas algumas das principais tecnologias que surgiram com o conceito de *smart grids*. A Seção 2.6, por sua vez, apresenta algumas das principais iniciativas de *smart grids* ao redor do mundo. Finalmente, a Seção 2.7 traz uma discussão geral sobre *smart grids*.

2.1 Contextualização

Redes elétricas existem há mais de um século. Sua existência foi um dos grandes responsáveis pelo rápido desenvolvimento da sociedade, o que permitiu o crescimento da economia, o surgimento de novas tecnologias e a melhoria da qualidade de vida das pessoas. As redes elétricas, no entanto, evoluíram muito pouco desde sua invenção, sendo basicamente as mesmas de 50 anos atrás (U. S. Department of Energy, 2003). Em função disso, as redes elétricas não estão sendo capazes de acompanhar o crescimento dos setores que delas dependem (Ministério de Minas e Energia, 2010b). Consequentemente, tem sido cada vez mais frequente a incidência de desabastecimentos de grandes proporções, como o que ocorreu na costa Oeste dos EUA em 2000 (BORENSTEIN, 2001).

Com base neste cenário, pode-se apontar alguns dos principais riscos decorrentes das deficiências encontradas nas redes elétricas convencionais.

- **Baixa redundância:** as redes atuais geralmente dependem de poucos, mas grandes, produtores. A vantagem é que grandes produtores geram energia a preços mais atraíveis que pequenos produtores. A desvantagem é que existem poucos produtores. Logo, se um produtor falha, o fornecimento de energia pode ser comprometido.
- **Danos ao meio ambiente:** um dos tipos de produtores de energia mais comuns são as usinas termoelétricas. Usinas termoelétricas geralmente dependem de combustíveis fósseis, que por sua vez são altamente poluentes.

- Altos custos operacionais: os custos operacionais para se manter a infraestrutura energética funcionando são muito altos.
- Falta de monitoramento: a ausência de dispositivos de monitoramento dificulta a detecção e reparação de falhas.
- Perda de energia: as redes atuais apresentam um elevado nível de perdas e desperdício energético. Isto é decorrente de diversos fatores, como furtos de energia e sua ineficiência para atender à demanda variável.
- Insegurança: em função da falta de monitoramento, situações perigosas acabam sendo identificadas apenas após já terem gerado avarias.
- Disponibilidade energética: a maioria das fontes de combustíveis fósseis já atingiu o limite de sua produção. Além do mais, algumas das principais fontes de combustíveis fósseis localizam-se em regiões com constantes conflitos políticos e religiosos.

Com base neste cenário, foi proposto o conceito de *smart grids*. *Smart grids* são uma evolução para as redes elétricas convencionais, que se utilizam de tecnologias de ponta para fornecer energia de forma eficiente e com alta confiabilidade.

2.2 Conceitos

Smart grids, de acordo com o U. S. Department of Energy (2003), são:

Redes elétricas automatizadas que monitoram e controlam cada elemento que as compõem, garantindo um fluxo bidirecional tanto de energia quanto de informações entre os produtores, os consumidores e todos os pontos entre eles. Sua inteligência distribuída, associada a sistemas de comunicação e sistemas de controle automatizados, viabiliza transações de mercado em tempo real e estabelece uma *interface* transparente entre pessoas, edifícios, plantas industriais, plantas de geração de energia e a rede elétrica. (U. S. Department of Energy, 2003)

Uma das características mais importantes de *smart grids* é o fluxo bidirecional de energia e de informações. Através desta característica, os elementos da rede podem interagir entre si em tempo real, permitindo, por exemplo, que um consumidor saiba o preço que está sendo cobrado pela energia que ele está consumindo naquele exato momento. Outra possibilidade decorrente desta característica é que qualquer elemento pode atuar tanto como consumidor quanto como fornecedor de energia. Isto permite a criação de um mercado de energia onde todos os elementos podem negociar entre si. As possibilidades são inúmeras, conforme abordado mais a frente no Capítulo 4.

A Figura 2.1 apresenta um exemplo conceitual de *smart grids*. Conforme pode ser observado, todos os elementos da rede estão conectados entre si. Neste cenário, diversas fontes de energia como aerogeradores, células de combustível, microturbinas e painéis solares fornecem energia para a rede, reduzindo a dependência do grande produtor. É possível notar, ainda, a presença de equipamentos de armazenamento de energia, que podem ser utilizados para fornecer energia, quando a demanda é maior que a produção, e para armazenar energia, quando a produção é maior que a demanda. O armazenamento distribuído de energia é um dos grandes trunfos de *smart grids* (IEEE, 2011).

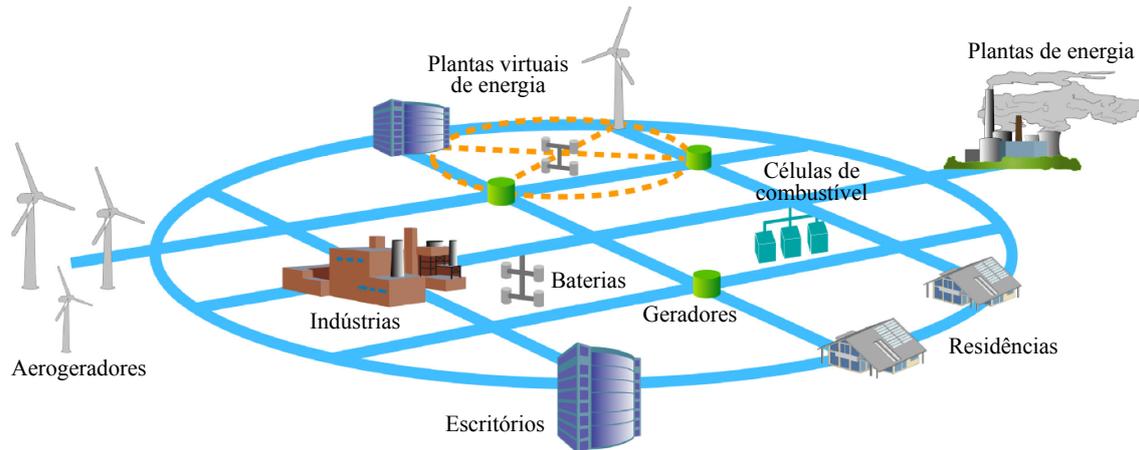


Figura 2.1: Diagrama conceitual de *smart grids* (adaptado de OSI (2009))

Segundo o U. S. Department of Energy (2003) e a IEEE (2011), a substituição das redes elétricas atuais por *smart grids* não implica na construção de toda uma infraestrutura nova. Pelo contrário, *smart grids* dependem das mesmas estruturas que as redes atuais, como subestações e transformadores. A diferença fica por conta de uma série de técnicas, tecnologias e ferramentas que devem ser embutidas na rede, através das quais é possível aumentar a confiabilidade, a eficiência, a segurança e a qualidade dos sistemas já existentes.

2.3 Características e benefícios

De acordo com o estudo realizado pelo Electric Power Research Institute (2009), *smart grids* possuem as seguintes características:

- Participação ativa dos consumidores: em *smart grids*, os consumidores fazem parte da rede. A rede mantém o usuário sempre informado, fornecendo incentivos para que ele adapte seu comportamento para economizar energia.
- Adoção de diversas opções de produção e armazenamento de energia: dada sua natureza distribuída, qualquer tipo de fonte de energia pode fazer parte de *smart grids*. São suportados desde plantas de energia de grande porte até residências dotadas de painéis solares. O mesmo vale para sistemas de armazenamento de energia, que aos poucos têm se mostrado mais eficientes e economicamente viáveis.
- Criação de novos produtos, serviços e mercados: *smart grids* dispõem de mecanismos para que produtores e consumidores interajam entre si. Através destes mecanismos, podem-se estabelecer mercados para diversas finalidades, como leilões de energia. Diversas possibilidades surgem neste sentido.
- Qualidade energética: *smart grids* apresentam um grau elevado de confiabilidade no fornecimento de energia. Isto é possível graças ao alto rigor com que a infraestrutura é monitorada. Desta forma, interrupções são raras e distúrbios são minimizados, características essenciais para abastecer equipamentos eletrônicos.
- Otimização de recursos e eficiência operacional: *smart grids* empregam tecnologias para garantir a otimização da utilização de recursos. Isto maximiza a eficiência

operacional e reduz os custos. Adicionalmente, tecnologias de autorreparação permitem que equipamentos operem por mais tempo sem necessitar de intervenção humana.

- Antecipação a falhas e perturbações: *smart grids* são capazes de identificar e reagir a perturbações, minimizando seu impacto no fornecimento. Para tanto, são empregadas técnicas para isolamento de circuitos e recuperação de falhas. Como resultado, falhas podem ser previstas e medidas podem ser tomadas antes mesmo que as falhas ocorram.
- Tolerância a ataques e desastres naturais: *smart grids* têm a capacidade de resistir a ataques físicos (em subestações e transformadores, por exemplo) e virtuais (comunicações ou informações sigilosas, por exemplo). Para tanto, são utilizados meios de prevenção, para se antecipar a ataques, e técnicas de recuperação, para minimizar o impacto de ataques.

Tendo por base estas características, o Electric Power Research Institute (2009) enumera uma série de benefícios de *smart grids* em relação às redes energéticas tradicionais, os quais podem ser divididos em cinco categorias.

- Fornecimento confiável e de qualidade: graças ao uso intenso de tecnologias de automação, *smart grids* são capazes de se autorreparar, reduzindo a ocorrência de interrupções e tornando o fornecimento mais limpo.
- Segurança física e virtual: através de monitoração constante, *smart grids* são capazes de detectar e evitar situações que possam oferecer riscos à sua operação. Em termos de segurança virtual, *smart grids* proporcionam um elevado grau de privacidade a seus usuários.
- Eficiência energética: *smart grids* são mais eficientes, permitindo a redução do consumo de energia, a redução de picos de energia e a redução de perdas energéticas. Além do mais, *smart grids* incentivam os usuários a economizar energia.
- Sustentabilidade: *smart grids* viabilizam economicamente a utilização fontes de energia renováveis e distribuídas, pois fornecem mecanismos para otimizar a produção e o consumo de energia. Isto reduz a dependência de fontes não renováveis de energia. Considerando, ainda, que *smart grids* realizam um controle inteligente da produção e da demanda, torna-se viável a substituição em massa de veículos movidos à combustão por EVs.
- Economia: os benefícios econômicos proporcionados por *smart grids* são grandes. Custos operacionais são minimizados. Clientes podem controlar seu consumo em tempo real, de acordo com o preço da energia. Adicionalmente, a adoção de novas tecnologias energéticas é acelerada.

Em *smart grids*, o preço da energia é constantemente ajustado para refletir corretamente a relação entre a oferta e a demanda de energia. De acordo com Electric Power Research Institute (2009), este tipo de informação beneficia os consumidores, pois estes podem adaptar seu comportamento para economizar energia e, conseqüentemente, ajudar a rede a se manter estável. Nesse mesmo sentido, *smart grids* incentivam os consumidores a investir em soluções residenciais para o armazenamento de energia. Através destes

dispositivos, o consumidor pode economizar dinheiro, armazenando energia quando ela estiver barata, e pode obter lucros, vendendo a energia armazenada quando seu preço estiver alto. Em contrapartida, ao realizar este procedimento o consumidor ajuda a rede a se manter estável.

2.4 Desafios

A migração das redes energéticas tradicionais para *smart grids* apresenta uma série de desafios (Electric Power Research Institute, 2009). No âmbito deste trabalho, é interessante detalhar alguns dos principais desafios técnicos existentes.

- Dispositivos inteligentes: considerando que *smart grids* representam um conceito tecnológico recente, muitas mudanças podem ocorrer nos próximos anos. Embora o conceito provavelmente permaneça o mesmo, as tecnologias empregadas podem mudar, em especial sua arquitetura. Por exemplo, um medidor de energia poderia ser desenvolvido para trabalhar em uma determinada arquitetura que hoje é utilizada. Mas o que aconteceria se essa a arquitetura fosse substituída? Isto depende de vários fatores, como, por exemplo, a regulamentação do governo. Desta forma, isto deve ser levado em conta durante o desenvolvimento de dispositivos inteligentes para *smart grids*, de forma a evitar grandes prejuízos econômicos. Alguns dos dispositivos que poderiam enfrentar este tipo de problema são: transformadores inteligentes, sensores remotos, medidores de energia, entre outros.
- Sistemas de comunicação: na mesma linha do desafio anterior, sistemas de comunicação devem ser projetados de modo que sejam facilmente adaptáveis a novos padrões que venham a surgir.
- Gerenciamento de dados: considerando que o conceito *smart grids* visa otimizar a eficiência da rede como um todo e levando em conta, ainda, a grande quantidade de sensores disponíveis, fica evidente a grande quantidade de dados que *smart grids* podem disponibilizar. Com base nisto, é necessário encontrar formas eficientes de lidar com este volume de dados e, principalmente, de extrair conhecimento deste universo.
- Segurança virtual: neste sentido, segurança é essencial para garantir confidencialidade de dados, integridade e disponibilidade de sistemas, entre outros.
- Privacidade: privacidade torna-se uma preocupação muito grande em *smart grids* porque, dado o nível de controle existente, dispõe-se de uma quantidade muito grande de informações de todos os elementos que compõem a rede. Isto gera desconfortos por parte dos envolvidos.
- Softwares: softwares são um dos principais requerimentos de *smart grids*. É graças ao uso de softwares que *smart grids* apresentam um comportamento inteligente. Neste sentido, devem-se estudar as diversas possibilidades de algoritmos para lidar com a realidade de *smart grids*. Neste ponto entra o uso de inteligência artificial. Técnicas de inteligência artificial podem ser utilizadas para diversas finalidades. A título de exemplo, pode-se utilizar IA para otimizar a produção de fontes de energia intermitente de modo a evitar o desabastecimento. Outro exemplo interessante do uso de IA é para coordenar o processo de recarga de EVs, de modo a evitar sobrecarga na rede.

Portanto, grande parte dos desafios existentes é decorrente da indefinição de padrões de *smart grids*. Isto não chega a ser um problema, pois o conceito de *smart grids* é relativamente novo, sendo comum a presença de incertezas quanto a padrões. Desta forma, é interessante utilizar-se de mecanismos abstratos para testar novas abordagens em *smart grids*, como o uso de simulações.

2.5 Tecnologias

Nesta seção são apresentadas algumas das tecnologias existentes em *smart grids*, em especial aquelas que têm ligação com o tema deste trabalho. Desta forma, atém-se a tecnologias voltadas a duas finalidades: (i) geração e armazenamento distribuídos de energia elétrica e (ii) veículos elétricos.

2.5.1 Geração e armazenamento distribuídos

Conforme mencionado anteriormente, uma das visões de *smart grids* é o fornecimento distribuído de energia. Neste sentido, surge o conceito de DER (fonte de energia distribuída, do inglês *Distributed Energy Resource*). DERs são dispositivos capazes de armazenar ou produzir energia, cuja capacidade varia entre 2kW e 2MW (CHALKIADAKIS et al., 2011; Electric Power Research Institute, 2009). Qualquer dispositivo que se enquadre neste conceito pode atuar como um DER. Alguns exemplos de dispositivos que podem atuar como DERs são: aerogeradores, painéis solares, baterias e mesmo EVs.

Neste sentido, surge também o conceito de planta virtual de energia, ou simplesmente VPP (em inglês, *Virtual Power Plant*). Segundo (PUDJANTO et al., 2007), uma VPP é um agrupamento de DERs. Entretanto uma VPP não apenas agrupa DERs, mas também agrega a produção destes de forma a atuar no mercado como um elemento único. Em outras palavras, pode-se dizer que VPPs são uma alternativa distribuída para as plantas de energia centralizadas convencionais (usinas hidrelétricas, por exemplo).

O armazenamento de energia é outro aspecto muito importante para *smart grids*. Através de dispositivos de armazenamento é possível responder de forma eficiente ao excesso e à falta de energia no mercado. Um caso típico que justifica esta necessidade seria, por exemplo, uma bateria funcionando como um DER, que poderia armazenar energia, sempre que a produção de energia for maior que a demanda, e fornecer energia, sempre que a produção não for capaz de atender à demanda. Através deste tipo de mecanismo, é possível evitar instabilidades na rede. A desvantagem em relação a esta funcionalidade refere-se ao alto custo de baterias aliado à sua vida útil relativamente curta. Entretanto, a demanda cada vez maior por este tipo de equipamento aliado aos altos investimentos realizados em pesquisas tem amenizado estas desvantagens (Ministério de Minas e Energia, 2010b).

No que se refere à geração distribuída de energia, e importante falar também de *microgrids*, ou microrredes. De acordo com Falcão (2009), uma *microgrid* é uma divisão de uma *smart grid* que atua de forma independente. Os elementos de uma *microgrid* não são abastecidos diretamente pela rede, mas pela *microgrid*. Desta forma, durante perturbações a *microgrid* pode desconectar-se da rede e continuar alimentando parcial ou totalmente seus elementos. Após a normalização da rede, a *microgrid* conecta-se novamente a esta. Este tipo de abordagem é vantajosa porque é mais eficiente e segura quando se deseja conectar um grande número de DERs à rede (FALCÃO, 2009).

2.5.2 Veículos elétricos

Em meio a um cenário onde se busca sustentabilidade através de uma dependência cada vez menor de combustíveis fósseis, surgem os EVs. Dentre os diversos tipos de EVs, uma variação é particularmente interessante para este trabalho: os veículos elétricos híbridos *plug-in*, os PHEVs (acrônimo de *Plug-in Hybrid Electric Vehicles*). Considerando que baterias são caras e que não possuem autonomia muito grande, PHEVs têm chamado bastante atenção dos consumidores. PHEVs possuem um motor elétrico principal, alimentado por uma bateria, e um motor à combustão auxiliar¹. Os PHEVs recebem este nome porque podem ser ligados à rede elétrica para recarregar suas baterias. Embora PHEVs sejam mais poluentes que EVs puramente elétricos, sua vantagem está na sua autonomia quase que ilimitada, pois quando a bateria acaba pode-se utilizar o motor auxiliar para gerar energia.

Considerando que baterias podem ser utilizadas como DERs, surgiu o conceito de fornecimento V2G. De acordo com Kempton e Tomić (2005), este mecanismo pode ser utilizado para transformar PHEVs em DERs sempre que eles não estiverem sendo utilizados. Através de sessões V2G, é possível obter lucros com a compra e venda de energia, servindo como uma alternativa para reduzir o custo associado à bateria de PHEVs.

2.6 Adoção

Diversas iniciativas têm promovido o desenvolvimento de *smart grids* ao redor do mundo. Considerando que a instalação de medidores inteligentes é um ponto fundamental na migração das redes convencionais para *smart grids*, diversas iniciativas têm se focado nesta transição, sendo que algumas destas encontram-se em andamento na Alemanha (Ministério de Minas e Energia, 2010b), nos EUA (NAPHADE et al., 2011) e em Portugal (FERREIRA et al., 2010; INOVGRID, 2008). Em linhas gerais, o objetivo destes projetos têm sido basicamente fornecer aos consumidores um conjunto detalhado de informações sobre seu padrão de consumo, permitindo que os mesmos descubram como mudar seus hábitos para economizar energia, reduzindo seus custos e contribuindo para uma sociedade mais sustentável. Alguns destes projetos permitem, inclusive, que consumidores vendam a energia que eventualmente venham a produzir.

A Dinamarca é outro país que tem investido muito no desenvolvimento de *smart grids*, sendo atualmente um dos mais adiantados neste processo. Uma de suas ilhas, Bornholm, depende quase que exclusivamente de aerogeradores para manter-se energeticamente. Isto motivou a cooperação entre o governo e a iniciativa privada na elaboração de um projeto para a criação de tecnologias que possibilitem o gerenciamento do processo de carga de EVs, controlando ainda o balanço de cargas da rede e o valor das tarifas (GERDING et al., 2011).

Vale ressaltar, no entanto, que todos estes projetos são apenas pilotos, ou seja, foram idealizados com o intuito de desenvolver e testar novas tecnologias. Isto explica o fato de todos estarem sendo conduzidos apenas em pequena ou média escala. Atualmente, ainda não existem redes elétricas comerciais que contenham todos os conceitos de *smart grids*. No caso de Portugal, porém, aparentemente este projeto encontra-se mais adiantado, visto que se espera uma adoção em massa da tecnologia em toda sua porção continental até 2017 (INOVGRID, 2008).

¹O motor à combustão de PHEVs é utilizado apenas para recarregar a bateria, não para movimentar o veículo.

No Brasil a adoção de *smart grids* ainda se encontra em fase inicial. Até o momento, o maior passo dado foi a criação de um grupo de trabalho para “analisar e identificar as ações necessárias para subsidiar o estabelecimento de políticas públicas para a implantação de um Programa Brasileiro de Rede Elétrica Inteligente” (Ministério de Minas e Energia, 2010a), que gerou um relatório técnico com estudos sobre a implantação de *smart grids* no Brasil (Ministério de Minas e Energia, 2010b).

2.7 Discussão

Smart grids apresentam um grande avanço em termos de eficiência energética em relação às redes elétricas convencionais. As redes elétricas convencionais são muito ineficientes e geram grandes prejuízos à economia. Com a adoção de *smart grids*, uma série de vantagens é obtida, como o aumento da eficiência e confiabilidade da rede.

O fluxo bidirecional de energia e informações é uma das grandes inovações de *smart grids*. Desta forma, todos os elementos da rede podem interagir entre si, sejam eles produtores, consumidores ou qualquer outro elemento entre eles. Através desta característica, surgem inúmeras possibilidades. Uma das mais abordadas é a criação do mercado de energia, onde os elementos podem fechar negócios entre si e obter lucros.

A adoção de *smart grids* apresenta uma série de desafios. Um dos mais importantes é o desenvolvimento de softwares capazes de aumentar a eficiência e confiabilidade de *smart grids*. Tal objetivo pode ser atingido com o uso de inteligência artificial. Um cenário em que o uso de IA é particularmente apropriado diz respeito ao balanceamento entre a oferta e a demanda, onde DERs podem armazenar energia, quando a oferta excede a demanda, e fornecer energia, quando a oferta não atende à demanda.

Neste capítulo foram apresentados, ainda, alguns projetos piloto de *smart grids* que estão sendo executados em diversos lugares do mundo. Dos projetos apresentados, até o momento nenhum foi capaz de implementar uma *smart grid* em sua totalidade. Entretanto, considerando que este conceito é relativamente novo e que sua adoção representa custos muito altos, percebe-se que este processo está ocorrendo de forma relativamente rápida. Isto é reflexo dos problemas que estão sendo enfrentados pelas redes convencionais. Portanto, fica claro que a adoção de *smart grids* é inevitável a médio e longo prazo.

No que se refere a coalizões, o cenário de *smart grids* apresenta inúmeras possibilidades para sua aplicação. Prova disto é a grande quantidade de publicações que tem abordado o melhor dos dois temas. Mais detalhes sobre este assunto são discutidos no Capítulo 4.

3 FORMAÇÃO DE COALIZÕES

Neste capítulo é apresentada uma revisão bibliográfica detalhada sobre formação de coalizões em sistemas multiagentes. Questões referentes à eficiência destes métodos são discutidas e os principais algoritmos para geração de estruturas de coalizão são apresentados.

Este capítulo encontra-se organizado da seguinte forma. A Seção 3.1 contextualiza o problema de formação de coalizões dentro de sistemas multiagentes. Na Seção 3.2 o problema é definido formalmente. A Seção 3.3 trata, em especial, do problema de formação de estruturas de coalizão. Finalmente, a Seção 3.4 apresenta uma discussão sobre formação de coalizões e a direção almejada por este trabalho.

3.1 Contextualização

A organização de agentes de uma forma eficiente é um dos problemas mais desafiadores para a comunidade de sistemas multiagente. De acordo com Horling e Lesser (2004), para agir de maneira coordenada um sistema multiagente deve ser projetado de tal forma que os papéis, as relações e as estruturas de autoridade, que determinam a dinâmica comportamental dos agentes, estejam claramente definidos. Neste sentido, existem diferentes paradigmas organizacionais, tais quais coalizões, times, hierarquias, federações, congregações, entre outros. Segundo Rahwan (2007), o paradigma baseado em coalizões é especialmente relevante, pois é capaz de modelar cenários onde pode não existir uma autoridade central para resolver possíveis conflitos entre as decisões dos agentes.

De acordo com Horling e Lesser (2004), duas características distinguem coalizões de outros paradigmas organizacionais. Primeiramente, elas são formadas especificamente para cumprir um objetivo, ou seja, elas só existem para cumprir um objetivo em comum entre diversos agentes. Segundo, coalizões têm vida curta, ou seja, elas são desfeitas quando seu objetivo for cumprido ou mesmo quando elas deixarem de ser lucrativas. Outro ponto importante é que coalizões costumam ser planas, ou seja, não existe uma hierarquia formal.

Considerando estes aspectos, o uso de coalizões tem atraído grande atenção da comunidade científica. Sobretudo, recentemente tem-se percebido um grande crescimento no número de publicações que tratam do uso de coalizões em *smart grids*.

3.2 O problema de formação de coalizões

A formação de coalizões é uma forma de interação que permite agrupar agentes de maneira coerente, de modo que eles tenham maiores condições de atingir objetivos indi-

viduais ou coletivos de maneira eficiente. Uma coalizão, portanto, é um grupo de agentes que decide cooperar para atingir um objetivo em comum. O particionamento de um conjunto de agentes A em coalizões disjuntas e exaustivas recebe o nome estrutura de coalizão.

De acordo com Sandholm et al. (1999), a formação de coalizões envolve uma sequência de três atividades:

- Cálculo do valor das coalizões: consiste em calcular o valor de todas as possíveis coalizões.
- Geração de estruturas de coalizão: onde o conjunto de agentes deve ser particionado em coalizões disjuntas exaustivamente (de modo que cada agente pertença a exatamente uma coalizão). A este particionamento dá-se o nome estrutura de coalizão.
- Divisão da utilidade: onde é determinada a recompensa de cada agente.

O objetivo deste trabalho, como já mencionado no Capítulo 1, é utilizar informações sobre o domínio de *smart grids* para melhorar a eficiência de algoritmos estado da arte para geração de estruturas de coalizão. Desta forma, este trabalho foca-se apenas na segunda atividade descrita por Sandholm et al. (1999). O restante deste capítulo foca-se exclusivamente nesta atividade, descrevendo-a com mais profundidade e apresentando o estado da arte para solução deste tipo de problema.

3.3 Geração de estruturas de coalizão

O problema de geração de estruturas de coalizão, ou simplesmente CSG (do inglês, *coalition structure generation*), consiste em gerar um particionamento do conjunto de agentes em coalizões disjuntas e exaustivas de modo a maximizar a utilidade global do sistema. O objetivo do problema de CSG não é simplesmente gerar uma solução válida, mas gerar a solução ótima, ou seja, trata-se de um problema de otimização. Desta forma, algumas definições básicas são necessárias. Seja A o conjunto de agentes, com $a = |A|$. Uma coalizão é um subconjunto de agentes $C \subseteq A$, cujo valor é representado por $v(C)$. Uma estrutura de coalizão $CS \in \mathcal{CS}$ é uma partição do conjunto de agentes A em coalizões disjuntas e exaustivas, cujo valor pode ser calculado por $V(CS) = \sum_{C \in CS} v(C)$. O objetivo é maximizar a utilidade global. Logo, seja CS^* a estrutura de coalizão ótima, cujo valor é dado por $CS^* = \arg \max_{CS \in \mathcal{CS}} V(CS)$.

Este tipo de problema geralmente é tratado na forma de jogos de função característica, ou simplesmente CFG (em inglês, *characteristic function games*) (SCHMEIDLER, 1969; SANDHOLM et al., 1999). Nesta classe de jogos, assume-se que o valor de uma coalizão C , dado por $v(C)$, não sofre influência de agentes não membros da coalizão¹. De uma forma mais formal, $v : 2^A \mapsto \mathbb{R}$ mapeia uma coalizão para um valor real de utilidade. É importante observar ainda duas características que devem ser evitadas na modelagem de uma função característica: superaditividade e subaditividade. Um jogo é superaditivo se $v(S \cup T) \geq v(S) + v(T)$ para quaisquer duas coalizões $S, T \subseteq A$ com $S \cap T = \emptyset$. Contrariamente, um jogo é subaditivo se $v(S \cup T) \leq v(S) + v(T)$ para quaisquer duas

¹Alternativamente, existem cenários onde o valor de uma coalizão é afetado por não-membros. São os chamados jogos de função de partição (PFG, do inglês *partition function games*) (THRALL; LUCAS, 1963). Entretanto, neste trabalho é dado foco apenas em CFGs.

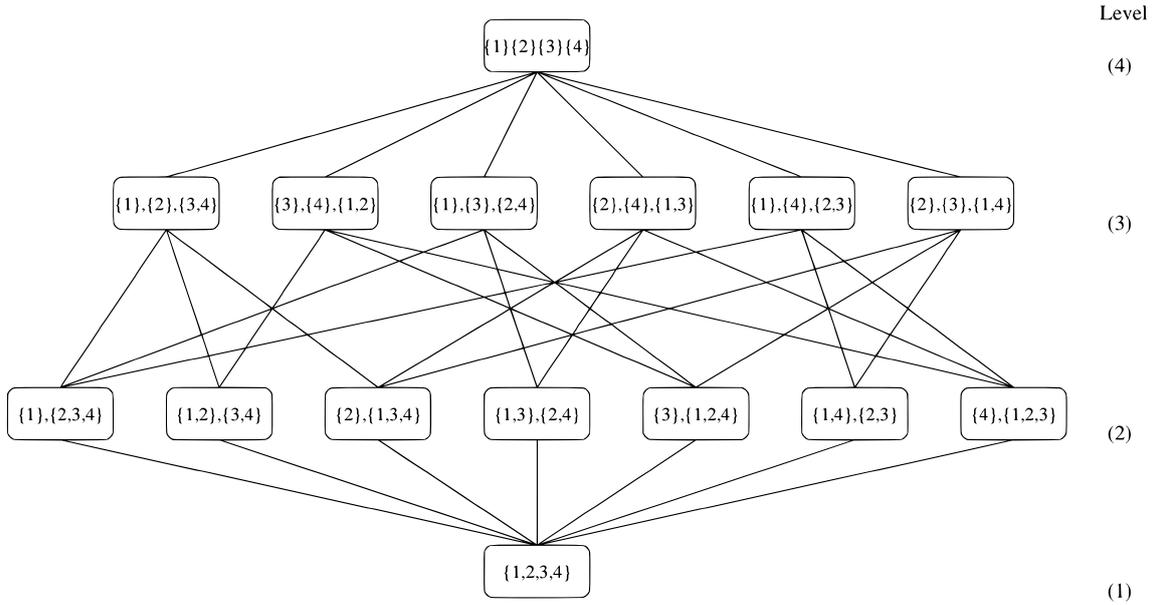


Figura 3.1: Grafo de estruturas de coalizão para quatro agentes (extraído de Sandholm et al. (1999))

coalizões $S, T \subseteq A$ com $S \cap T = \emptyset$. O problema em relação a estas duas características é que elas tornam a solução do problema trivial, de forma que a solução gerada seja sempre com todos os agentes em uma coalizão única (superaditividade) ou com todos os agentes em coalizões separadas (subaditividade) (SANDHOLM et al., 1999). Desta forma, tanto superaditividade quanto subaditividade são evitadas neste trabalho.

O processo de CSG é muito complexo, pois encontrar a estrutura de coalizão ótima frequentemente implica enumerar todas as possibilidades. Conseqüentemente, o espaço de busca cresce exponencialmente em relação ao número de agentes. De um modo mais específico, de acordo com Sandholm et al. (1999), para a agentes existem $2^a - 1$ coalizões e entre $O(a^a)$ e $\omega(a^{\frac{a}{2}})$ estruturas de coalizão. Adicionalmente, Sandholm et al. (1999) provaram que este tipo de problema é NP-completo.

Na literatura, o espaço de busca das estruturas de coalizão costuma ser representado na forma de um grafo, conforme mostra a Figura 3.1. No grafo, cada nodo representa uma estrutura de coalizão. As arestas representam a divisão de uma coalizão em outras duas, quando segue para cima, ou a junção de duas coalizões em uma só, quando segue para baixo. As estruturas de coalizão são agrupadas em níveis de acordo com a quantidade de coalizões que contém, ou seja, a estrutura de coalizão com uma coalizão fica no nível 1, as estruturas de coalizão compostas por duas coalizões ficam no nível 2 e assim por diante.

É possível calcular a quantidade de estruturas de coalizão presentes em um dado nível l do grafo, para a agentes, a partir dos números de Stirling do segundo tipo, conforme a recorrência:

$$Z(a, l) = lZ(a - 1, l) + Z(a - 1, l - 1) , \quad (3.1)$$

onde $Z(a, a) = Z(a, 1) = 1$. Adicionalmente, o número total de estruturas de coalizão pode ser obtido através da soma dos números de Stirling de cada nível do grafo:

$$\sum_{l=1}^a Z(a, l) . \quad (3.2)$$

Segundo Rahwan e Jennings (2008b), o problema de CSG pode ser resolvido através de três classes diferentes de algoritmos:

- Algoritmos baseados em heurísticas: algoritmos deste tipo geralmente estabelecem restrições na criação de coalizões. Sua vantagem está na velocidade com que geram uma solução válida. Porém, algoritmos deste tipo não são capazes de garantir a geração de soluções ótimas e, mesmo que uma solução ótima venha a ser encontrada, eles não são capazes de verificar este fato.
- Algoritmos baseados em programação dinâmica: originalmente criados para o problema de particionamento de conjuntos, o qual é tido como análogo ao de CSG. Sua grande vantagem está na garantia de encontrar uma solução ótima em $O(3^a)$. Entretanto fazem um uso muito intenso de memória.
- Algoritmos *anytime*: estes algoritmos unem características de ambas as abordagens anteriores. Funcionam gerando soluções iniciais dentro de um limite do ótimo, que então são melhoradas enquanto houver tempo disponível. Embora não gerem a solução ótima de imediato, geram soluções subótimas em pouco tempo.

Nesta seção são apresentados os principais algoritmos de cada uma das classes apresentadas acima. Inicialmente, a Seção 3.3.1 descreve um importante algoritmo baseado em heurísticas. Posteriormente, na Seção 3.3.2 são detalhados os algoritmos estado da arte baseados em programação dinâmica. Finalmente, na Seção 3.3.3 são apresentados os mais importantes algoritmos *anytime*.

3.3.1 Algoritmos baseados em heurísticas

Métodos heurísticos compreendem algoritmos capazes de fornecer soluções para um problema, porém sem definir um limite na sua qualidade. Algoritmos heurísticos também não são capazes de garantir que a solução ótima será encontrada. No entanto, estes métodos podem retornar soluções razoavelmente boas em um tempo bem menor que outras abordagens. Desta forma, este tipo de algoritmo costuma ser mais aplicável a cenários com um número maior de agentes.

Este trabalho não é focado em algoritmos baseados em heurísticas. No entanto, optou-se por abordar ao menos um algoritmo deste tipo de modo a ampliar o entendimento sobre o tema. Na subseção a seguir é apresentado um algoritmo proposto por Shehory e Kraus (1998) que utiliza esta abordagem. O algoritmo não possui um nome, portanto, aqui ele é chamado algoritmo SK, remetendo às iniciais do sobrenome de seus autores.

3.3.1.1 Algoritmo SK

O algoritmo SK é voltado para a geração de estruturas de coalizão em ambientes descentralizados (SHEHORY; KRAUS, 1998). O SK consiste de um algoritmo guloso *anytime*² para a formação de estruturas de coalizão em domínios orientados a tarefas. Neste cenário, cada tarefa é atribuída a uma coalizão com recursos suficientes para realizá-la. Um algoritmo guloso realiza escolhas localmente ótimas na expectativa de que estas escolhas conduzam à solução ótima global (CORMEN et al., 2001).

Shehory e Kraus (1998) assumem em seu algoritmo que as coalizões possuem um limite máximo de k agentes. Desta forma, nem todas as coalizões precisam ser avaliadas, o que determina um limite polinomial para o número de estruturas de coalizão em

²Embora o algoritmo SK seja *anytime*, sua principal característica refere-se a algoritmos baseados em heurísticas.

$O(a^k)$. A justificativa para esta escolha é que, de acordo com os autores, a quantidade de comunicação necessária seria reduzida e a gestão dos recursos por parte dos agentes seria facilitada. O problema, no entanto, permanece sendo NP-completo.

O algoritmo proposto por Shehory e Kraus (1998) consiste em duas etapas distintas. A primeira etapa refere-se à distribuição do cálculo dos valores das coalizões. Nesta etapa, o cálculo dos valores é repetido iterativamente, pois estes mudam conforme o ambiente sofre variações. Na segunda etapa, cada agente possui uma lista de coalizões potenciais, as quais ele se comprometeu a calcular o valor. Nesta etapa, cada agente escolhe a melhor coalizão potencial de sua lista e anuncia esta coalizão aos agentes referidos. Realizados os anúncios, cada agente ingressa na coalizão que apresenta o melhor valor. Este comportamento implica a necessidade de se repetir o cálculo dos valores das coalizões iterativamente.

A restrição assumida pelo algoritmo SK faz com que a solução encontrada possa estar a um limite arbitrário do ótimo. Isto ocorre porque o espaço de soluções consideradas restringe-se a coalizões de tamanho k . No entanto, encontrar a solução ótima só seria possível através da busca em todas as coalizões, sem que houvesse a imposição da restrição k (SHEHORY; KRAUS, 1998). Porém, apesar desta dificuldade, os autores provam que esta restrição é capaz de reduzir a complexidade do processo de exponencial para polinomial.

3.3.2 Algoritmos baseados em programação dinâmica

Programação dinâmica, segundo Cormen et al. (2001), é um método tabular que permite a resolução de um problema através da combinação das soluções de seus subproblemas. Esta classe de métodos geralmente é aplicada a problemas de otimização, onde pode haver um grande espaço de soluções possíveis mas deseja-se encontrar apenas a solução ótima.

De acordo com Cormen et al. (apud RAHWAN, 2007), problemas solucionáveis com programação dinâmica apresentam as seguintes propriedades:

- Subestrutura ótima: onde, para resolver um problema é necessário quebrá-lo em subproblemas, os quais podem ser resolvidos recursivamente, para então recombinar as soluções e resolver o problema original;
- Subproblemas superpostos: onde subproblemas não são independentes, ou seja, subproblemas compartilham subsubproblemas.

Algoritmos de programação dinâmica resolvem cada subproblema apenas uma vez, armazenando as soluções em uma tabela para posterior consulta. Embora esta abordagem gere um consumo excessivo de memória, elimina-se a necessidade de se recalcular a solução de um subproblema toda vez que ele é encontrado (CORMEN et al., 2001).

Programação dinâmica pode ser aplicada para resolver o problema de geração de estruturas de coalizão. Neste caso, o problema é encontrar a estrutura de coalizão ótima e os subproblemas referem-se a encontrar a partição ótima de cada coalizão (RAHWAN, 2007). O algoritmo considerado o estado da arte para resolver o problema de geração de coalizões através de programação dinâmica é o IDP (do inglês, *Improved Dynamic Programming*), proposto por Rahwan e Jennings (2008b). O algoritmo IDP é baseado no algoritmo DP (do inglês, *Dynamic Programming*), proposto por Yeh (1986). Ambos os algoritmos são apresentados nesta seção.

3.3.2.1 Algoritmo DP

O algoritmo DP, proposto inicialmente por Yeh (1986), foi projetado para resolver o problema de particionamento de conjuntos. De acordo com Sandholm et al. (1999), o problema de particionamento de conjuntos é análogo ao de CSG, pois ambos buscam particionar um conjunto de elementos em subconjuntos de modo a maximizar os valores a eles associados. Neste sentido, o problema a ser resolvido é particionar o conjunto de agentes A de modo a maximizar seu valor. A solução deste problema é a estrutura de coalizão ótima CS^* . Um subproblema, por sua vez, representa uma única coalizão $C \subseteq A$. A solução de um subproblema C corresponde à sua melhor partição (considerando no máximo duas divisões), conforme a Equação 3.3.

$$\text{Solução de } C = \arg \max_{C' \subset C, C'' = C \setminus C'} v(C') + v(C'') . \quad (3.3)$$

O algoritmo DP, portanto, consiste de duas etapas: solucionar os subproblemas para então poder solucionar o problema principal. Inicialmente, cada subproblema deve ser solucionado individualmente. Desta forma, duas tabelas são mantidas em memória: f_1 , que contém a solução de cada coalizão; e f_2 , que contém o valor da solução de cada coalizão. O processo começa a partir das coalizões de tamanho 1. Como uma coalizão com apenas um elemento não pode ser dividida, sua solução é ela mesma, ou seja, $f_1[C] = C$ e, conseqüentemente, $f_2[C] = v(C)$. Computadas as coalizões de tamanho 1, parte-se para as de tamanho 2, depois as de tamanho 3, e assim sucessivamente, até a coalizão de

Algoritmo 3.1: Algoritmo DP para geração de estruturas de coalizão

Entrada: $v(C)$ para todo $C \subseteq A$. Se $v(C)$ não for especificado, então $v(C) = 0$

Saída: CS^*

```

1  para todo  $i \in \{1, \dots, a\}$  faça
2  |    $f_1[\{a_i\}] \leftarrow \{a_i\}$  ;
3  |    $f_2[\{a_i\}] \leftarrow v(\{a_i\})$  ;
4  fim para todo
5  para todo  $s \in \{2, \dots, a\}$  faça
6  |   para todo  $C \subseteq A$ , com  $|C| = s$  faça
7  |   |    $f_2[C] \leftarrow \max\{f_2[C'] + f_2[C''] : C' \subset C \text{ e } C'' = C \setminus C' \text{ e } 1 \leq |C'| \leq \frac{|C|}{2}\}$  ;
8  |   |   se  $f_2[C] \geq v(C)$  então
9  |   |   |    $f_1[C] \leftarrow C^*$ , onde  $C^*$  é o maior  $C'$  do passo anterior ;
10 |   |   senão
11 |   |   |    $f_1[C] \leftarrow C$  ;
12 |   |   fim se
13 |   fim para todo
14 fim para todo
15  $CS^* \leftarrow \{A\}$ ;
16 para todo  $C \subseteq CS^*$  faça
17 |   se  $f_1[C] \neq C$  então
18 |   |    $CS^* \leftarrow (CS^* \setminus \{C\}) \cup \{f_1[C]\}$  ;
19 |   |   volta à linha 16 e recomeça com a nova  $CS^*$  ;
20 |   fim se
21 fim para todo

```

tamanho a . A partir das coalizões de tamanho 2, sempre há ao menos uma divisão possível. Supondo uma coalizão de três elementos $C = \{1, 2, 3\}$, por exemplo, ela poderia ser dividida de três formas diferentes: $\{\{1\}, \{2, 3\}\}$, $\{\{2\}, \{1, 3\}\}$ ou $\{\{3\}, \{1, 2\}\}$. Supondo ainda que a última delas tivesse o maior valor, então $f_1[\{1, 2, 3\}] = \{\{3\}, \{1, 2\}\}$ e, conseqüentemente, $f_2[\{1, 2, 3\}] = v(\{3\}) + v(\{1, 2\})$.

É importante observar que os subproblemas devem, necessariamente, ser solucionados de maneira ascendente, ou seja, começando com as coalizões de tamanho 1 até a . Isto se deve ao fato de que, para avaliar as partições de uma coalizão C_k de tamanho k , já se deve ter a solução das de tamanho $\{C_1, \dots, C_{k-1}\}$, já que as partições possíveis são $\{C_1, C_{k-1}\}, \{C_2, C_{k-2}\}, \dots, \{C_{\frac{k}{2}}, C_{\frac{k}{2}}\}$.

Finalmente, após ter preenchido as duas tabelas com a solução de todos os subproblemas, o algoritmo inicia a solução do problema original. Para tanto, assume-se, provisoriamente, que a coalizão de tamanho a é a CS^* . Então, substitui-se recursivamente cada coalizão $C \subseteq CS^*$ por sua solução f_1 . O processo termina quando $C = f_1[C]$ para todo $C \subseteq CS^*$. Desta forma, o algoritmo DP é capaz de encontrar uma estrutura de coalizão com uma complexidade de $O(3^a)$ (maiores detalhes da análise de complexidade no trabalho de Yeh (1986)). O Algoritmo 3.1 apresenta o funcionamento do algoritmo DP.

Conforme pode ser observado no Algoritmo 3.1, a primeira etapa do DP, onde ocorre a solução dos subproblemas, é compreendida entre as linhas 1 e 14. Nas linhas 1 a 4, o procedimento é realizado apenas para as coalizões de tamanho 1. Já nas linhas 5 a 14, o procedimento é realizado para todas as demais coalizões. Na linha 7, especificamente, a coalizão é particionada em todas as possibilidades, gerando sua solução. Finalmente, a

Tabela 3.1: Exemplo de como o algoritmo DP calcula as tabelas f_1 e f_2 para 4 agentes

Tamanho	Coalizão	Avaliações realizadas para encontrar f_1 e f_2	f_1	f_2
1	{1}	$v(\{1\}) = 30$	{1}	30
	{2}	$v(\{2\}) = 40$	{2}	40
	{3}	$v(\{3\}) = 25$	{3}	25
	{4}	$v(\{4\}) = 45$	{4}	45
2	{1, 2}	$v(\{1, 2\}) = 50$ $f_2[\{1\}] + f_2[\{2\}] = 70$	{1}{2}	70
	{1, 3}	$v(\{1, 3\}) = 60$ $f_2[\{1\}] + f_2[\{3\}] = 55$	{1, 3}	60
	{1, 4}	$v(\{1, 4\}) = 80$ $f_2[\{1\}] + f_2[\{4\}] = 75$	{1, 4}	80
	{2, 3}	$v(\{2, 3\}) = 55$ $f_2[\{2\}] + f_2[\{3\}] = 65$	{2}{3}	65
	{2, 4}	$v(\{2, 4\}) = 70$ $f_2[\{2\}] + f_2[\{4\}] = 85$	{2}{4}	85
	{3, 4}	$v(\{3, 4\}) = 80$ $f_2[\{3\}] + f_2[\{4\}] = 70$	{3, 4}	80
3	{1, 2, 3}	$v(\{1, 2, 3\}) = 90$ $f_2[\{1\}] + f_2[\{2, 3\}] = 95$ $f_2[\{2\}] + f_2[\{1, 3\}] = 100$ $f_2[\{3\}] + f_2[\{1, 2\}] = 95$	{2}{1, 3}	100
	{1, 2, 4}	$v(\{1, 2, 4\}) = 120$ $f_2[\{1\}] + f_2[\{2, 4\}] = 115$ $f_2[\{2\}] + f_2[\{1, 4\}] = 120$ $f_2[\{4\}] + f_2[\{1, 2\}] = 115$	{1, 2, 4}	120
	{1, 3, 4}	$v(\{1, 3, 4\}) = 100$ $f_2[\{1\}] + f_2[\{3, 4\}] = 110$ $f_2[\{3\}] + f_2[\{1, 4\}] = 105$ $f_2[\{4\}] + f_2[\{1, 3\}] = 105$	{1}{3, 4}	110
	{2, 3, 4}	$v(\{2, 3, 4\}) = 115$ $f_2[\{2\}] + f_2[\{3, 4\}] = 120$ $f_2[\{3\}] + f_2[\{2, 4\}] = 110$ $f_2[\{4\}] + f_2[\{2, 3\}] = 110$	{2}{3, 4}	120
4	{1, 2, 3, 4}	$v(\{1, 2, 3, 4\}) = 140$ $f_2[\{1\}] + f_2[\{2, 3, 4\}] = 150$ $f_2[\{2\}] + f_2[\{1, 3, 4\}] = 150$ $f_2[\{3\}] + f_2[\{1, 2, 4\}] = 145$ $f_2[\{4\}] + f_2[\{1, 2, 3\}] = 145$ $f_2[\{1, 2\}] + f_2[\{3, 4\}] = 150$ $f_2[\{1, 3\}] + f_2[\{2, 4\}] = 145$ $f_2[\{1, 4\}] + f_2[\{2, 3\}] = 145$	{1, 2}{3, 4}	150

Adaptado de Rahwan et al. (2007).

etapa final do algoritmo é apresentada nas linhas 15 a 21, onde as soluções dos subproblemas são agrupadas para gerar a solução do problema principal.

De modo a embasar melhor os conceitos do algoritmo DP, propõe-se um exemplo onde $A = \{1, 2, 3, 4\}$, conforme a Tabela 3.1. Inicialmente, o algoritmo percorre as coalizões de tamanho 1. Conforme mencionado anteriormente, para toda coalizão com apenas um agente $f_1[C] = C$ e $f_2[C] = v(C)$. Em seguida, o algoritmo inicia o processamento das coalizões de tamanho 2. Como estas coalizões são divisíveis, o algoritmo avalia se é melhor mantê-las inteiras ou dividi-las. No caso da coalizão $\{1, 2\}$, pode-se observar que seu valor é 50, ao passo que sua divisão em $\{1\}$ e $\{2\}$ possui o valor 70. Portanto, a solução para a coalizão $\{1, 2\}$ é sua divisão, ou seja, $f_1[\{1, 2\}] = \{1\}\{2\}$ e $f_2[\{1, 2\}] = f_2[\{1\}] + f_2[\{2\}] = 70$. Seguindo em frente, pode-se observar que a coalizão $\{1, 3\}$ possui um valor maior quando inteira do que ao ser dividida, o que significa que ela deve permanecer inteira. Neste caso, $f_1[\{1, 3\}] = \{1, 3\}$ e $f_2[\{1, 3\}] = 60$. O processo continua até que todos os subproblemas tenham sido solucionados.

Na etapa final, o algoritmo assume provisoriamente que $CS^* = \{\{1, 2, 3, 4\}\}$. A partir disso, cada coalizão interna é comparada com sua solução, iterativamente. Assim sendo, a coalizão $\{1, 2, 3, 4\}$ é substituída por $\{1, 2\}\{3, 4\}$, pois $f_1[\{1, 2, 3, 4\}] = \{1, 2\}\{3, 4\}$. Tem-se, então, $CS^* = \{\{1, 2\}, \{3, 4\}\}$. Seguindo em frente, tem-se que $f_1[\{1, 2\}] = \{1\}\{2\}$. Logo, tem-se $CS^* = \{\{1\}, \{2\}, \{3, 4\}\}$. A partir deste momento, todas as coalizões em CS^* possuem $C = f_1[C]$. Portanto, $CS^* = \{\{1\}, \{2\}, \{3, 4\}\}$ é a solução ótima, cujo valor é 150. É importante salientar que, ao iniciar a etapa final, embora a estrutura de coalizão ótima ainda não seja conhecida, seu valor já é. Isto é uma consequência da primeira etapa, que acumula os valores dos subproblemas ao longo de sua execução.

A etapa final do algoritmo DP é ilustrada na Figura 3.2. Como pode ser observado, a partir do nodo inferior existem 3 nodos que levam para a solução ótima. As arestas destacadas indicam os caminhos que o DP poderia percorrer para chegar a esta solução. Embora no exemplo dado tenha sido escolhido o nodo $\{1, 2\}\{3, 4\}$, os outros dois também poderiam ser escolhidos, pois possuem o mesmo valor e também levam à solução ótima.

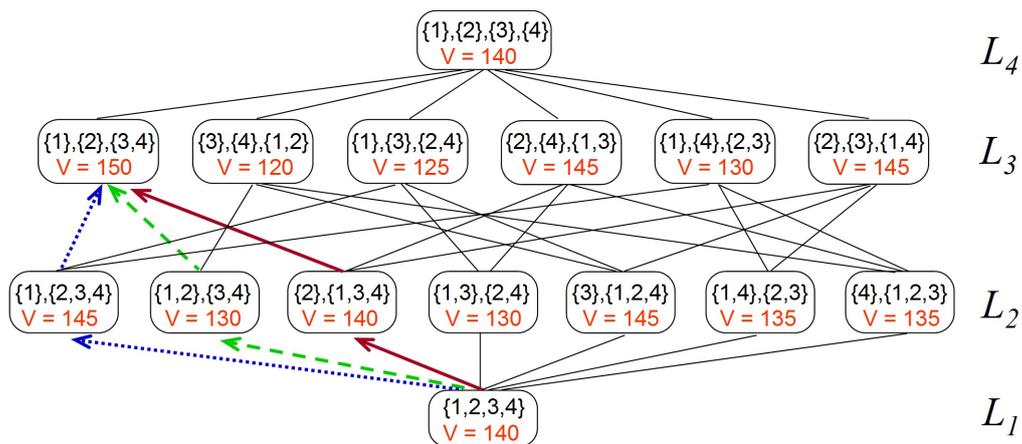


Figura 3.2: Grafo de estruturas de coalizão para quatro agentes (extraído de Rahwan e Jennings (2008b)). No exemplo, as arestas destacadas mostram todos os caminhos que o DP poderia percorrer para encontrar a solução ótima. O IDP, por outro lado, elimina a maioria destas arestas, de forma que cada nodo contenha apenas uma aresta direcionada para ele (no sentido de baixo para cima).

3.3.2.2 Algoritmo IDP

O algoritmo IDP foi proposto por Rahwan e Jennings (2008b) como uma versão otimizada do algoritmo DP. O objetivo do IDP é reduzir a quantidade de divisões avaliadas sem que isto comprometa a busca pela solução ótima. Desta forma, Rahwan e Jennings se fizeram valer de uma observação importante no grafo de estruturas de coalizão, estabelecendo que cada nodo precisa de apenas uma aresta direcionada para si para garantir que a solução ótima pode ser encontrada. Esta observação é válida porque o algoritmo DP é capaz de percorrer qualquer caminho para encontrar a solução ótima, mesmo que haja apenas um. Logo, não há a necessidade de se manter vários caminhos para o mesmo nodo. Portanto, ao eliminar arestas desnecessárias, pode-se reduzir bastante a quantidade de divisões de uma coalizão a serem avaliadas. A Figura 3.2 ilustra este conceito, onde as arestas destacadas apresentam os diferentes caminhos que poderiam ter sido percorridos para chegar à solução ótima. Como pode ser observado, se houvesse apenas um dos caminhos já seria possível encontrar a solução ótima.

Com base neste conceito, Rahwan e Jennings (2008b) definem E^* como o conjunto de todas as arestas essenciais para a representação do problema, desconsiderando caminhos redundantes. Em outras palavras, com base em E^* , cada nodo tem exatamente uma aresta direcionada para si. O conjunto E^* é formado com base na Equação 3.4.

Algoritmo 3.2: Algoritmo IDP para geração de estruturas de coalizão

Entrada: $v(C)$ para todo $C \subseteq A$. Se $v(C)$ não for especificado, então $v(C) = 0$

Saída: CS^*

```

1  para todo  $i \in \{1, \dots, a\}$  faça
2  |    $f_1[\{a_i\}] \leftarrow \{a_i\}$ ;
3  |    $f_2[\{a_i\}] \leftarrow v(\{a_i\})$ ;
4  fim para todo
5  para todo  $s \in \{2, \dots, a\}$  faça
6  |   para todo  $C \subseteq A$ , com  $|C| = s$  faça
7  |   |    $f_2[C] \leftarrow \max\{f_2[C'] + f_2[C''] : C' \subset C \text{ e } C'' = C \setminus C' \text{ e } 1 \leq |C'| \leq \frac{|C|}{2}$ 
8  |   |   |   e  $(|C''| \leq n - s \text{ ou } |C'| + |C''| = n)\}$ ;
9  |   |   se  $f_2[C] \geq v(C)$  então
10 |   |   |    $f_1[C] \leftarrow C^*$ , onde  $C^*$  é o maior  $C'$  do passo anterior;
11 |   |   senão
12 |   |   |    $f_1[C] \leftarrow C$ ;
13 |   |   fim se
14 |   fim para todo
15  $CS^* \leftarrow \{A\}$ ;
16 para todo  $C \subseteq CS^*$  faça
17 |   se  $f_1[C] \neq C$  então
18 |   |    $CS^* \leftarrow (CS^* \setminus \{C\}) \cup \{f_1[C]\}$ ;
19 |   |   volta à linha 16 e recomeça com a nova  $CS^*$ ;
20 |   fim se
21 fim para todo

```

$$E^* = \left(\bigcup_{s'' \leq n - (s' + s'')} E^{s' s''} \right) \cup \left(\bigcup_{s' + s'' = n} E^{s' s''} \right), \quad (3.4)$$

onde $E^{s' s''}$ representa todas as arestas que surgem com a divisão de uma coalizão de tamanho $(s' + s'')$ em duas de tamanho s' e s'' .

A partir desta equação, Rahwan e Jennings (2008b) introduzem o Teorema 1.

Teorema 1. Para cada estrutura de coalizão CS contendo s coalizões ($s \in \{2, \dots, a\}$), existe uma aresta em E^* que conecta CS com outra estrutura de coalizão contendo $s - 1$ coalizões.

Através deste teorema, Rahwan e Jennings (2008b) provam que a porção $\bigcup_{s' + s'' = n} E^{s' s''}$ da Equação 3.4 engloba todas as arestas da coalizão de tamanho n até as coalizões de tamanho $n - 1$. A porção $\bigcup_{s'' \leq n - (s' + s'')} E^{s' s''}$ da Equação 3.4, por sua vez, garante que apenas uma aresta chega a cada nodo. A prova, com maiores detalhes deste teorema, pode ser encontrada diretamente no trabalho de Rahwan e Jennings (2008b).

A estrutura do algoritmo IDP é idêntica à do DP, conforme mostrado no Algoritmo 3.2. A única diferença fica por conta da linha 7, que sofre uma pequena modificação para considerar apenas as arestas em E^* . A mudança corresponde à validação ($|C''| \leq n - s$ ou $|C'| + |C''| = n$), que adapta a Equação 3.4 ao algoritmo DP.

Embora a diferença entre os algoritmos DP e IDP seja muito pequena, os experimentos demonstram que o desempenho do IDP é bem superior ao do DP. De acordo com Rahwan e Jennings (2008b), no caso de 25 agentes há uma redução entre 33,3% e 66,6% no uso de memória. Adicionalmente, o IDP executa apenas 38,7% das operações realizadas pelo DP, no cenário com 25 agentes. Portanto, o IDP representa o estado da arte na geração de estruturas de coalizão através de programação dinâmica.

3.3.3 Algoritmos *anytime*

Nas seções anteriores foram descritas duas abordagens diferentes para gerar estruturas de coalizão. De um lado têm-se algoritmos baseados em heurísticas, que geram soluções rapidamente e que são apropriados para um grande número de agentes. Entretanto, estes algoritmos não geram soluções ótimas nem sequer fornecem um limite em relação a elas. Do outro lado, têm-se algoritmos baseados em programação dinâmica, que garantem a geração de soluções ótimas. Porém, estes algoritmos necessitam percorrer todo o espaço de busca antes de fornecer a solução.

Algoritmos *anytime* apresentam-se como um meio termo entre algoritmos que geram soluções ótimas e algoritmos que geram soluções válidas em tempo reduzido (RAHWAN, 2007). Seu conceito envolve gerar uma solução inicial, dentro de um limite do ótimo, que pode ser melhorada progressivamente tão logo haja tempo disponível. Esta solução é denominada melhor solução até o momento. A vantagem deste tipo de abordagem está na sua capacidade de gerar soluções de boa qualidade rapidamente (equiparando-se a algoritmos baseados em heurísticas) e, mais importante, na sua habilidade de melhorar a melhor solução até o momento enquanto houver tempo (podendo inclusive encontrar uma solução ótima).

O algoritmo estado da arte para este tipo de abordagem é o IP (do inglês, *Integer Partition*), proposto por (RAHWAN et al., 2007). Nesta seção é apresentado o algoritmo IP bem como outros dos principais algoritmos que utilizam este tipo de abordagem.

3.3.3.1 Algoritmo CSS1

Sandholm et al. (1999) foram os primeiros a introduzir um algoritmo *anytime* para geração de estruturas de coalizão que estabelece um limite na qualidade da melhor solução até o momento (RAHWAN, 2007). A proposta deste algoritmo é representar o problema de formação de estruturas de coalizão como uma busca no grafo de estruturas de coalizão (Figura 3.1).

Segundo Sandholm et al. (1999), é possível estabelecer um limite $\beta = a$ da solução ótima através da busca nos dois níveis inferiores do grafo de estruturas de coalizão, conforme mostra o Teorema 2 (SANDHOLM et al., 1999)³. A explicação é que para estabelecer tal limite é necessário percorrer todas as coalizões possíveis. Esta busca pode ser efetuada nos dois níveis inferiores do grafo, pois nesta porção do espaço de busca cada coalizão aparece exatamente uma vez. Logo, este limite é firme, ou seja, nenhum outro limite pode ser melhor para esta porção de busca. Portanto, provou-se que nenhum algoritmo pode estabelecer um limite sem efetuar uma busca em pelo menos 2^{a-1} estruturas de coalizão.

Teorema 2. Busca mínima para estabelecer um limite. Para limitar β , é suficiente realizar uma busca nos dois níveis inferiores do grafo de estrutura da coalizão. Desta maneira, o limite é $\beta = a$ e o número de nodos visitados é 2^{a-1} .

No trabalho de Sandholm et al. (1999), é definido ainda que o limite pode ser diminuído cada vez que um novo nível do grafo é pesquisado. Esta busca adicional, deve ser iniciada a partir dos níveis superiores e executada de forma descendente. A razão de melhora do limite é definida pelo Teorema 3 (SANDHOLM et al., 1999). Com base neste teorema, ao efetuar a busca no nível superior o limite é reduzido pela metade, ou seja, $\beta = \frac{a}{2}$. Ainda, a cada dois novos níveis percorridos, o divisor no limite é acrescido de uma unidade. A Tabela 3.2 apresenta a ordem de redução do limite β em função da busca adicional.

Teorema 3. Diminuindo o limite com busca adicional. Após realizar uma busca no nível l , o limite β é $\lceil \frac{a}{h} \rceil$ se $a \equiv (h-1) \pmod{h}$ e $a \equiv l \pmod{2}$. Caso contrário o limite β é $\lfloor \frac{a}{h} \rfloor$, onde $h = \lfloor \frac{a-l}{2} \rfloor + 2$.

Com base nestes teoremas, Sandholm et al. (1999) propuseram o algoritmo CSS1 (acrônimo para *Coalition-Structure-Search-1*). A ideia do algoritmo CSS1 é, basicamente, definir uma solução inicial dentro do limite $\beta = a$ e, enquanto houver tempo disponível, utilizar busca em largura para melhorar esta solução. Desta forma, o algoritmo pode ser interrompido a qualquer momento, provendo uma solução com um limite preciso e confiável dentro do ótimo. O CSS1 é apresentado no Algoritmo 3.3.

Algoritmo 3.3: Algoritmo CSS1 para geração de estruturas de coalizão

- 1 Pesquisar os dois níveis inferiores do grafo de estruturas de coalizão;
 - 2 Efetuar busca em largura a partir do níveis superior do grafo de estruturas de coalizão enquanto houver tempo ou até que o grafo inteiro tenha sido percorrido;
 - 3 Retornar a estrutura de coalizão com maior valor encontrada;
-

Sandholm et al. (1999) apresentaram resultados teóricos sobre seus estudos, dando um foco na análise das garantias de desempenho no pior caso. Através de seu trabalho,

³Maiores detalhes sobre o assunto, bem como a prova dos Teoremas 2 e 3, podem ser encontrados diretamente no trabalho de Sandholm et al. (1999).

Tabela 3.2: Redução do limite β em função da busca adicional, conforme Teorema 3

Nível	Limite
a	$a/2$
$a - 1$	$a/2$
$a - 2$	$a/3$
$a - 3$	$a/3$
$a - 4$	$a/4$
$a - 5$	$a/4$
\vdots	\vdots
2	a
1	Nenhum

Sandholm et al. foram os primeiros a provar que um limite inferior pode ser estabelecido após a busca em um número mínimo de estruturas de coalizão. Sua abordagem, no entanto, acaba necessitando uma busca muito grande no grafo de estruturas de coalizão. De acordo com Rahwan (2007), considerando 24 agentes, após efetuar a busca nos níveis 1, 2 e 24, o que corresponde a 8×10^6 estruturas de coalizão, teria-se $\beta = 12$. Isto significa que, mesmo esta busca tendo sido grande, a solução encontrada seria, no pior caso, 12 vezes pior que a solução ótima, ou seja, a garantia é que a solução encontrada não seja pior que 8,33% da solução ótima. Portanto, basear-se apenas nestas garantias pode não ser uma boa alternativa para cenários reais (RAHWAN, 2007).

3.3.3.2 Algoritmo IP

O algoritmo IP, proposto por (RAHWAN et al., 2007), representa o estado da arte no que se refere a algoritmos *anytime* para geração de estruturas de coalizão. Este algoritmo é baseado em uma nova e concisa forma de representação do espaço de estruturas de coalizão. Através desta representação, o algoritmo é capaz de realizar uma busca direcionada e eficiente, conferindo-lhe reduzidos tempo de execução e complexidade em relação a outras abordagens.

A representação do espaço de busca é baseada no agrupamento de estruturas de coalizão em *configurações*. A configuração de uma estrutura de coalizão determina o tamanho das coalizões que a compõem. Por exemplo, ambas as estruturas de coalizão $\{\{1\}, \{2, 3\}\}$ e $\{\{3\}, \{1, 2\}\}$ são compostas por uma coalizão com um elemento e por uma coalizão com dois elementos; logo, ambas possuem a configuração $[1, 2]$. Adicionalmente, as coalizões são agrupadas por tamanho.

A seguir são apresentadas as definições básicas utilizadas pelo algoritmo IP. Seja $CL_s \in \mathcal{CL}$ o conjunto de coalizões de tamanho $s \in \{1, 2, \dots, a\}$. Seja $G_1, G_2, \dots, G_{|\mathcal{G}_{CS}|} \in \mathcal{G}_{CS}$ o conjunto de todas as configurações possíveis. Ainda, seja $F : \mathcal{G}_{CS} \rightarrow 2^{\mathcal{CL}_s}$ uma função que recebe uma configuração e retorna todas as estruturas de coalizão daquela configuração, tal que $\mathcal{N} = \{F(G_1), F(G_2), \dots, F(G_{|\mathcal{G}_{CS}|})\}$ representa uma lista onde cada elemento é um conjunto de estruturas de coalizão de mesma configuração. Neste caso, $N_1, N_2, \dots, N_{\mathcal{G}_{CS}} \in \mathcal{N}$ denota os elementos de \mathcal{N} . A Figura 3.3 demonstra a representação do espaço de busca utilizado pelo IP para um cenário com quatro agentes. São apresentadas na figura as listas \mathcal{CL} e \mathcal{N} .

$$\mathcal{CL} =$$

CL_1	CL_2	CL_3	CL_4
$\{a_4\}$	$\{a_3, a_4\}$	$\{a_2, a_3, a_4\}$	$\{a_1, a_2, a_3, a_4\}$
$\{a_3\}$	$\{a_2, a_4\}$	$\{a_1, a_3, a_4\}$	
$\{a_2\}$	$\{a_2, a_3\}$	$\{a_1, a_2, a_4\}$	
$\{a_1\}$	$\{a_1, a_4\}$	$\{a_1, a_2, a_3\}$	
	$\{a_1, a_3\}$		
	$\{a_1, a_2\}$		

$$G_1 = [4], F(G_1) = N_1 = \{ \{a_1, a_2, a_3, a_4\} \}$$

$$G_2 = [2, 2], F(G_2) = N_2 = \left\{ \begin{array}{l} \{ \{a_1, a_2\}, \{a_3, a_4\} \} \\ \{ \{a_1, a_3\}, \{a_2, a_4\} \} \\ \{ \{a_1, a_4\}, \{a_2, a_3\} \} \end{array} \right.$$

$$G_3 = [1, 3], F(G_3) = N_3 = \left\{ \begin{array}{l} \{ \{a_4\}, \{a_1, a_2, a_3\} \} \\ \{ \{a_3\}, \{a_1, a_2, a_4\} \} \\ \{ \{a_2\}, \{a_1, a_3, a_4\} \} \\ \{ \{a_1\}, \{a_2, a_3, a_4\} \} \end{array} \right.$$

$$G_4 = [1, 1, 2], F(G_4) = N_4 = \left\{ \begin{array}{l} \{ \{a_3\}, \{a_4\}, \{a_1, a_2\} \} \\ \{ \{a_2\}, \{a_4\}, \{a_1, a_3\} \} \\ \{ \{a_2\}, \{a_3\}, \{a_1, a_4\} \} \\ \{ \{a_1\}, \{a_4\}, \{a_2, a_3\} \} \\ \{ \{a_1\}, \{a_3\}, \{a_2, a_4\} \} \\ \{ \{a_1\}, \{a_2\}, \{a_3, a_4\} \} \end{array} \right.$$

$$G_5 = [1, 1, 1, 1], F(G_5) = N_5 = \{ \{ \{a_1\}, \{a_2\}, \{a_3\}, \{a_4\} \} \}$$

$$\mathcal{G}_{CS} = \{G_1, G_2, G_3, G_4, G_5\}$$

$$\mathcal{N} = \{N_1, N_2, N_3, N_4, N_5\}$$

Figura 3.3: Representação do espaço de busca adotada pelo algoritmo IP. No exemplo, $A = \{a_1, a_2, a_3, a_4\}$ e os elementos de \mathcal{N} foram expandidos. Adaptado de Rahwan et al. (2007).

De maneira simplificada, o algoritmo IP aplica técnicas de *branch-and-bound* ao espaço de busca, definindo limites para as configurações G e podando as que possuem baixa probabilidade de conter a solução ótima. Estes limites são refinados sempre que a busca é aprofundada, o que permite uma poda cada vez maior do espaço de busca. Através desta abordagem, a busca é então direcionada para um espaço de busca relativamente pequeno. Embora este algoritmo seja capaz de encontrar uma solução ótima, ele apresenta o melhor do seu desempenho quando a busca é direcionada a soluções com um limite pré-definido em relação ao ótimo. Este limite é ajustado pelo parâmetro β , onde $\beta = 1$ significa que se deseja encontrar uma solução ótima. O algoritmo é composto de três etapas:

1. Pré-processamento: percorre todos $CL_s \in \mathcal{CL}$, obtendo os valores $v(C)$ máximo e

médio de cada lista. Evidentemente, esta etapa envolve a busca em todas as coalizões possíveis.

2. Escolha da configuração ótima: utiliza os valores obtidos na etapa anterior para gerar a lista $G \in \mathcal{G}_{CS}$ e selecionar o elemento $F(G)$ que tem a maior probabilidade de conter a estrutura de coalizão ótima CS^* . Apenas uma configuração é selecionada nesta etapa, a partir da qual é possível analisar todas as CS que a compõem na etapa seguinte.
3. Busca da CS^* : onde são determinados os valores $V(CS) \in F(G)$ para encontrar a CS^* . Esta é a etapa com maior custo computacional.

O estágio de pré-processamento calcula os valores máximo e médio de cada lista $CL_s \in \mathcal{CL}$. Para tanto, para cada lista CL_s , analisa-se todas suas coalizões C , guardando os valores máximos ($max_s = \max_{C \in CL_s} v(C)$) e médios ($avg_s = \frac{1}{|CL_s|} \sum_{C \in CL_s} v(C)$). Durante estes cálculos, o algoritmo é capaz de analisar ainda uma pequena fração das estruturas de coalizão, encontrando uma solução provisória com garantia de estar a um limite de $\frac{\alpha}{2}$ da solução ótima.

O objetivo da segunda etapa é encontrar a configuração G com a maior probabilidade de conter a CS^* . Este procedimento é realizado com base nos limites inferior e superior de cada configuração. Estes limites são constantemente atualizados, de modo que, na terceira etapa, seja possível podar as configurações que não têm a possibilidade de conter a estrutura de coalizão ótima.

Na terceira etapa do algoritmo IP é realizada a busca pela estrutura de coalizão ótima, propriamente dita. Desta forma, o algoritmo gera cada uma das estruturas de coalizão possíveis contidas no $F(G)$ selecionado pela etapa anterior. O algoritmo é interrompido ao encontrar uma estrutura de coalizão com $V(CS) \geq \beta$ ou após percorrer todas as configurações. Esta etapa é a que possui custo mais elevado. Portanto, o algoritmo foi projetado de modo a evitar redundância e sobreposição de coalizões. Tais mecanismos melhoram consideravelmente o desempenho do algoritmo. Esta etapa é executada iterativamente em conjunto com a segunda. Desta forma, cada vez que uma configuração é percorrida, seus limites são atualizados, o que permite a poda das configurações cujo limite superior esteja abaixo do limite inferior da melhor solução encontrada até o momento. Através deste outro mecanismo, grande parte das configurações tende a ser podada do espaço de busca antes mesmo de ser analisada. Com base nestas técnicas, o algoritmo geralmente é capaz de encontrar uma solução ótima após percorrer apenas $3 \times 2^{a-1}$ estruturas de coalizão.

Com base em experimentos realizados, Rahwan et al. (2007) demonstram empiricamente a eficácia de seu algoritmo em comparação com o algoritmo DP. Os resultados comprovam que, em um cenário com 20 agentes, o algoritmo IP é capaz de obter soluções que estão, no pior caso, dentro do limite de 1% do ótimo, utilizando 0,0043% do tempo e 77% da memória gastos pelo algoritmo DP.

O algoritmo *anytime* IP, embora seja o estado da arte para geração de estruturas de coalizão, possui uma complexidade de $O(a^a)$, que é bem maior que o IDP ($O(3^a)$), estado da arte com base em programação dinâmica. Posteriormente, Rahwan e Jennings (2008a) estenderam o algoritmo IDP, incorporando a ele elementos do IP. Este novo algoritmo foi denominado IDP-IP. O IDP-IP poda um número maior de arestas do grafo de estruturas de coalizão que o IDP. Desta forma o grafo fica desconexo, formando diversos subespaços. O algoritmo IP é então utilizado para aplicar técnicas de *branch-and-bound*, podando subespaços e encontrando a solução ótima. Em determinados casos, o IDP-IP é capaz de

encontrar a solução ótima em menos tempo que o IP e consumindo menos memória que o IDP. Entretanto, para os cenários abordados neste trabalho, IDP-IP apresenta resultados muito próximos aos obtidos por seus antecessores. Além do mais, implementar este algoritmo é uma tarefa muito complexa. Desta forma, percebeu-se que os benefícios de se comparar a abordagem deste trabalho com o IDP-IP não justificam o tempo necessário para seu desenvolvimento. Deste modo, a comparação com o IDP-IP não entrou no escopo deste trabalho.

3.4 Discussão

Este capítulo forneceu uma visão geral de formação de coalizões, sobretudo no que diz respeito ao processo de geração de estruturas de coalizão. Ao longo deste, foi possível perceber a relevância que o tema tem na área de sistemas multiagente.

Entretanto, embora existam diversas alternativas para o problema de CSG, nenhuma delas é apropriada para cenários reais. Em cenários reais, não existiriam apenas 20 agentes, como nos cenários onde os algoritmos foram experimentados. Haveria centenas ou até mesmo milhares. Por outro lado têm-se os métodos heurísticos que, embora apresentem um bom desempenho para um número maior de agentes, não são capazes de encontrar a solução ótima. Isto inviabilizaria a utilização das abordagens aqui estudadas.

Outro ponto recorrente é que nenhuma das abordagens foi experimentada com dados reais. Em todos os casos, são utilizadas distribuições sintéticas, propostas em trabalhos como o de Larson e Sandholm (2000). Se por um lado a utilização deste tipo de dados para experimentações facilita a comparação de diferentes algoritmos, por outro deixa-se de experimentar estas técnicas em cenários realistas, o que poderia gerar ótimas discussões e avanços para a área.

Desta forma, abordagens capazes de preencher estas lacunas tendem a ser interessantes e a gerar contribuições interessantes para a área. Em função destes assuntos em aberto, faz-se relevante o tema deste trabalho, que visa possibilitar a utilização de algoritmos de CSG em cenários reais, utilizando informação sobre o domínio para reduzir sua complexidade e viabilizar a busca pela solução ótima para um número maior de agentes.

4 FORMAÇÃO DE COALIZÕES EM SMART GRIDS

Neste capítulo é apresentada uma visão geral do uso de coalizões em *smart grids*. Inicialmente, a Seção 4.1 apresenta, de uma forma geral, as vantagens de se utilizar coalizões em *smart grids*. Na Seção 4.2 é apresentada uma abordagem em que são formadas coalizões de aerogeradores. Na Seção 4.3 é detalhada outra abordagem, onde as coalizões são entre EVs. Já na Seção 4.4, são apresentadas algumas abordagens que utilizam formação dinâmica de coalizões. Finalmente, a Seção 4.5 apresenta uma discussão sobre as abordagens existentes, traçando um panorama da solução almejada por este trabalho.

4.1 Motivação

O uso de coalizões em *smart grids* tem sido amplamente discutido pela comunidade de sistemas multiagente, conforme demonstram estudos realizados por Ramchurn et al. (2012) e McArthur et al. (2007). Um dos principais interesses deste tipo de abordagem tem sido aumentar a confiabilidade de matrizes energéticas renováveis. A motivação deste interesse, por sua vez, é consequência de um dos principais objetivos de *smart grids*: tornar a produção e distribuição de energia mais sustentável e eficiente.

A substituição de fontes não renováveis (como usinas termoelétricas) por fontes renováveis de energia (como aerogeradores e painéis solares) implica em uma série de riscos à rede elétrica. Fontes renováveis costumam gerar energia de forma intermitente, ou seja, a produção depende diretamente de fatores climáticos. Como consequência o fornecimento de energia pode deixar de ser constante, o que representa um risco alto demais se tratando de um recurso tão importante para a economia moderna.

Portanto, a ideia de formação de coalizões em *smart grids* visa justamente utilizar-se de medidas que tornem possível aumentar a confiabilidade em fontes intermitentes de energia. Neste sentido, dois tipos de abordagem têm sido bastante explorados:

- Agregação da produção de aerogeradores. Neste tipo de abordagem costuma-se agregar a previsão de produção de aerogeradores, de modo que suas previsões sejam mais precisas em termos estatísticos. Um trabalho que utiliza esta abordagem é o de Chalkiadakis et al. (2011).
- Agrupamento de veículos elétricos. Em tais abordagens, busca-se agrupar EVs de modo a torná-los mais competitivos e representativos na rede, aumentando seu lucro com a venda de energia. Exemplos deste tipo de abordagem incluem Kamboj et al. (2010, 2011) e Ramos e Bazzan (2012a).
- Agrupamento de aerogeradores e dispositivos auxiliares. Neste tipo de abordagem costuma-se utilizar dispositivos para armazenar e fornecer energia para amenizar os

impactos da intermitência na produção dos aerogeradores. Um exemplo de trabalho que utiliza esta abordagem é o de Mihailescu et al. (2011c).

Através da formação de coalizões em *smart grids*, inúmeras vantagens podem ser obtidas. Entre elas destacam-se: aumento da confiabilidade de fontes renováveis, aumento da lucratividade de DERs, aumento da precisão de previsões de produção, produção de energia mais distribuída, maior diversidade na matriz energética, menos poluição, entre outros. As seções a seguir apresentam alguns dos principais trabalhos que estudam a formação de coalizões em *smart grids* sob diferentes perspectivas.

4.2 Coalizões de aerogeradores

O trabalho de Chalkiadakis et al. (2011) estuda a formação de coalizões entre aerogeradores para aumentar a confiabilidade de matrizes energéticas que deles dependem. Nos cenários almejados para *smart grids*, a matriz energética é composta essencialmente por fontes de energia renováveis, como aerogeradores. Entretanto, a produção energética de aerogeradores é intermitente devido a fatores climáticos. Esta intermitência dificulta a realização de previsões precisas acerca da produção energética. Consequentemente, para evitar o desabastecimento de energia, é recorrente o uso de fontes de energia não renováveis alternativas, como geradores movidos a combustíveis fósseis. Desta forma, é extremamente importante dispor de previsões precisas da produção para reduzir a necessidade de se utilizar fontes poluentes de energia. Com base nisso, Chalkiadakis et al. (2011) propõem a formação de coalizões entre aerogeradores, agregando sua produção energética e buscando reduzir a margem de erro da previsão de sua produção. Do ponto de vista econômico, a rede tende a favorecer grandes produtores, pois estes costumam ter uma produção mais estável e confiável (PUDJIANTO et al., 2007).

Na abordagem de Chalkiadakis et al. (2011), sugere-se o aprimoramento de VPPs sob uma perspectiva de teoria dos jogos. Desta forma, é criado o conceito de VPPs cooperativas, ou simplesmente CVPPs (do inglês, *Cooperative Virtual Power Plants*), onde DERs colaboram entre si visando um fornecimento energético estável e confiável para a planta como um todo. O que difere uma VPP de uma CVPP é que, basicamente, enquanto a VPP simplesmente agrega a produção de seus DERs, a CVPP cria mecanismos para incentivar os DERs a cooperar para atingir um objetivo em comum. Desta forma, Chalkiadakis et al. (2011) determinam que a CVPP deve fornecer incentivos monetários para que os DERs gerem previsões confiáveis sobre sua produção. Vale salientar que, diferentemente de uma VPP, uma CVPP é formada de forma dinâmica e a operação de suas tarefas pode ser desempenhada de forma distribuída entre os DERs que a compõem. Nessa abordagem, no entanto, não são detalhados aspectos referentes a quem é responsável pelo gerenciamento do CVPP e a como os DERs são distribuídos inicialmente entre CVPPs.

O fluxo de funcionamento da abordagem proposta por Chalkiadakis et al. (2011) é o seguinte. Inicialmente, cada DER faz uma estimativa da sua produção para as 24 horas seguintes e repassa esta informação à CVPP a qual pertence. Em seguida, a CVPP estima sua produção de energia conjunta, com base na estimativa dos DERs que a compõem, repassando esta informação à rede. Logo após, a rede utiliza as estimativas do CVPPs para garantir o suprimento de energia aos consumidores para as próximas 24 horas. Nesta etapa, é definido o valor da energia ao longo deste período, de modo que o valor seja mais alto quando a produção prevista for menor que a demanda prevista, e vice-versa. Quando a energia é consumida, a CVPP recebe seu pagamento. O pagamento, então, é distribuído entre os DERs, de acordo com a produção individual de cada.

Com base neste cenário, é definido um mecanismo de precificação, através do qual é estabelecido o valor que cada DER recebe como pagamento pela energia produzida. O mecanismo de precificação baseia-se nas estimativas fornecidas pelos DERs, incentivando estimativas precisas e punindo as incoerentes. Desta forma, um DER que fizer estimativas incorretas recebe um valor menor, por unidade de energia, do que um DER que fizer estimativas precisas. Este tipo de mecanismo incentiva os DERs a fazer estimativas precisas de sua produção, de modo que quanto mais precisa for a estimativa do DER, maior será o valor pago pela sua energia. Adicionalmente, a rede fornece incentivos para CVPPs maiores, ou seja, quanto maior a produção de energia de uma CVPP, maior o valor pago pela rede.

Através desta política de precificação são estabelecidos CVPPs mais confiáveis e eficientes. Os DERs, por sua vez, são incentivados a participar de cooperativas porque ao atuar de maneira individual eles não são capazes de gerar energia de forma eficiente e confiável, recebendo um pagamento muito inferior pela energia produzida. Este mecanismo, portanto, incentiva os DERs a ingressar e se manter em CVPPs. As CVPPs, por outro lado, podem controlar os DERs que lhe compõem, visando maximizar seu ganho. Este controle é feito com base no desempenho de cada DER. Um DER que apresenta um histórico de estimativas precárias pode ser removido da cooperativa para não afetar o resultado dos demais.

Para validar esta abordagem, foram efetuados experimentos com base em dados obtidos a partir do Parque Eólico Experimental de Sotavento, localizado em Galicia, Espanha. Este parque possui 24 aerogeradores, que produzem energia suficiente para abastecer 12 mil residências. O conjunto de dados utilizado contém informações referentes ao período de um ano. Com base nos experimentos, foi possível constatar que os DERs apresentaram uma forte tendência a participar de CVPPs para obter resultados melhores. Constatou-se ainda que DERs atuando individualmente receberam pagamentos muito menores, mesmo quando suas estimativas de produção eram precisas. Entretanto, quando em uma CVPP, todas os DERs têm um ganho maior, independentemente se suas estimativas foram precisas ou não. Uma constatação interessante é que, mesmo com todos DERs de uma CVPP fazendo estimativas medíocres, a CVPP se mostra capaz de gerar estimativas precisas de sua produção conjunta, obtendo bons resultados para os DERs que a compõem.

Esta abordagem, entretanto, é mais voltada a *mechanism design* do que à formação de coalizões em si. Embora alguns aspectos de formação de coalizões tenham sido abordados, a técnica não é capaz de garantir que as coalizões formadas gerem a estrutura de coalizão ótima.

4.3 Coalizões de veículos elétricos

No trabalho de Kamboj et al. (2010, 2011) é proposta a formação de coalizões entre EVs para que estes atuem no mercado regulatório de energia. Em cenários típicos idealizados para *smart grids*, onde a matriz energética é composta exclusivamente por fontes não poluidoras, a produção fica condicionada a fatores climáticos. Entretanto, a rede deve ser capaz de abastecer o mercado sem interrupções. É neste cenário que entra o mercado regulatório de energia. O objetivo do mercado regulatório é conferir maior estabilidade à rede, fornecendo energia quando a produção não atende à demanda e armazenando energia quando a produção excede a demanda. No desempenho desta tarefa são utilizados dispositivos de armazenamento de energia (baterias) e dispositivos para geração de energia (geradores), ambos acoplados diretamente à rede de distribuição de energia.

O mercado regulatório de energia, no entanto, apresenta algumas desvantagens. Primeiramente, geradores geralmente são alimentados com combustíveis fósseis. Isto implicaria em substituir um grande produtor poluente por inúmeros menores também poluentes. Em segundo lugar, baterias são muito caras e têm uma vida útil relativamente curta. Considerando que as baterias viriam a ser utilizadas com muita frequência, elas precisariam ser substituídas com frequência, gerando altos custos operacionais. Em terceiro lugar, para manter esta infraestrutura funcionando seria necessário conceder mais espaço físico às operadoras de energia (para posicionar os dispositivos) e depender de mais profissionais (para manutenção e monitoramento). Aliados, estes problemas inviabilizariam economicamente a existência de uma *smart grid* que dependa exclusivamente de fontes intermitentes de energia.

Com base neste cenário, Kamboj et al. (2010, 2011) estudam a utilização de EVs no mercado regulatório de energia, eliminando a necessidade de dispositivos dedicados a esta tarefa e, conseqüentemente, criando uma solução distribuída para o armazenamento e o fornecimento de energia. Através desta abordagem, os custos para manter o mercado regulatório seriam muito menores e parte da economia obtida poderia ser revertida para os EVs. Além do mais, esta abordagem é bastante realista, pois, conforme Kempton e Tomić (2005), veículos permanecem estacionados cerca de 96% do tempo. Portanto, os EVs poderiam armazenar ou fornecer energia à rede sempre que estiverem conectados a ela. Naturalmente, foi assumida a hipótese de que os EVs só participam deste mercado quando não estiverem sendo utilizados, garantindo que sempre haja carga suficiente para a próxima viagem de seu proprietário.

Nesta abordagem, cada EV é controlado por um agente. O agente é responsável por garantir que o EV sempre possua a quantidade mínima de energia necessária para sua próxima viagem. Para garantir isto, o agente estima a próxima viagem do EV com base nas suas viagens anteriores. A partir desta informação, é possível determinar a quantidade de energia que deve ser mantida intacta na bateria e que não pode ser fornecida para a rede. Respeitada esta restrição, o agente é livre para operar a bateria do EV no mercado regulatório, comprando ou vendendo energia.

As coalizões podem ser formadas de três maneiras: (i) grande coalizão, ou seja, todos agentes em uma única coalizão; (ii) duas coalizões, sendo uma para os agentes fornecedores e outra para armazenadores; e (iii) duas coalizões mistas, parecido com o anterior, porém cada agente possuindo uma pequena probabilidade de pertencer às duas coalizões ao mesmo tempo. Após as coalizões terem sido formadas, as trocas de energia com a rede podem ser efetuadas. O pagamento é repassado às coalizões sempre que a energia é consumida. Com base neste mecanismo, a rede é capaz de identificar, em tempo real, o nível de participação que as coalizões podem ter no mercado regulatório.

Os experimentos deste trabalho foram conduzidos em um ambiente simulado e em um ambiente real. No ambiente simulado foram utilizados 300 EVs, cada um com uma bateria de 35kWh e com um trajeto diário de menos de 160km. Neste cenário, os resultados demonstram que os EVs atuaram no mercado com maior intensidade durante a noite e no início do horário comercial. Isto indica que, após atingir o limite mínimo que deve ser mantido em suas baterias, os EVs, de fato, deixam de atuar no mercado como fornecedores, atuando apenas no armazenamento de energia. Nos experimentos conduzidos em ambiente real, foram utilizados 5 EVs reais, cuja capacidade de energia não foi divulgada. Os experimentos foram conduzidos ao longo de 9 meses. Ao final, constatou-se que a participação no mercado foi bastante lucrativa, de modo que, ao atuar em média 15 horas por dia no mercado, foi obtido um faturamento médio entre US\$ 100,00 e US\$ 200,00 por

mês, o que poderia ser utilizado para minimizar o custo para se manter um EV.

Entretanto, esta abordagem não fornece qualquer formalismo de formação de coalizões, ou seja, as coalizões não são formadas de forma a encontrar a solução ótima. Desta forma, as coalizões representam basicamente o agrupamento dos EVs de acordo com sua estratégia, seja ela voltada para o fornecimento ou para o armazenamento de energia.

4.4 Formação dinâmica de coalizões entre DERs

Na literatura existem diversos trabalhos que abordam a formação dinâmica de coalizões (DCF, do inglês *Dynamic Coalition Formation*) em *smart grids*. De acordo com Klusch e Gerber (2002), DCF refere-se à formação de coalizões em ambientes estocásticos, abertos, distribuídos e heterogêneos. Em tais cenários, agentes cooperativos entram e saem de coalizões frequentemente e as tarefas e recursos disponíveis mudam constantemente. É importante salientar a diferença entre CF e DCF. Enquanto abordagens de CF assumem um ambiente estático e geram soluções ótimas, abordagens de DCF assumem ambientes dinâmicos e geram soluções válidas, porém não ótimas, as quais são constantemente adaptadas a variações no ambiente.

Smart grids apresentam um cenário particularmente interessante para o uso de abordagens de DCF. Tomando como exemplo o cenário proposto por Kamboj et al. (apresentado na Seção 4.3), EVs entram e saem da rede constantemente, conforme a necessidade de seus proprietários. Neste mesmo cenário, EVs podem adotar estratégias diferentes (como armazenar ou fornecer energia) dependendo da quantidade de energia disponível em suas baterias. Portanto, a abordagem de DCF mostra-se uma alternativa apropriada para lidar com formação de coalizões neste tipo de cenário.

Gerber e Klusch (2003) propuseram uma abordagem genérica para tratar o problema de DCF, denominada DCF-S (*Dynamic Coalition Formation Scheme*). Nesta abordagem, cada coalizão elege um agente líder, o qual simula, avalia, escolhe e negocia a formação de coalizões, as quais devem ser capazes de realizar determinadas tarefas. O funcionamento do DCF-S é dividido em quatro etapas: (i) preparação, onde o agente líder define as tarefas que deverão ser realizadas; (ii) simulação, onde o líder simula a formação de diferentes coalizões para executar uma dada tarefa (limitando sua busca a coalizões melhores que sua atual, apenas); (iii) negociação, onde o líder negocia com os demais agentes a formação das coalizões simuladas na etapa anterior (embora o líder negocie a formação de todas as coalizões possíveis ao mesmo tempo, a única que será formada, de fato, é a com maior utilidade); e (iv) avaliação, onde o agente líder avalia o resultado e faz a distribuição dos valores obtidos (esta etapa, naturalmente, é executada apenas após o término da coalizão).

Existem diversos trabalhos que se utilizam de DCF em *smart grids*. Embora nenhum deles use o DCF-S explicitamente, traços deste estão presentes na maioria. Nos trabalhos de Mihailescu et al. (2011a, 2011b, 2011c) é proposta a formação de coalizões entre fornecedores e consumidores, sob o ponto de vista de DCF. Neste trabalho existem dois tipos de agentes, os fornecedores (produzem ou armazenam energia) e os consumidores de energia. Esta abordagem estrutura-se em três etapas: (i) inicialização das coalizões, (ii) agregação de fornecedores e (iii) agregação de consumidores. Na primeira etapa, cada fornecedor, cuja disponibilidade de energia é superior a um limite pré-estabelecido, cria sua própria coalizão. Cada um destes fornecedores assume o papel de coordenador da coalizão que criou. Na segunda etapa, os coordenadores convidam os demais fornecedores a se juntarem às suas coalizões. Os convites constituem-se do perfil de fornecimento que cada coordenador estabelece para sua coalizão, ou seja, a quantidade e a curva de

demanda que a coalizão pretende suprir. Cada fornecedor ingressa na coalizão que tiver maior ganho com sua participação. Finalmente, na terceira etapa, os consumidores enviam sua previsão de consumo aos coordenadores das coalizões mais próximas geograficamente. Os coordenadores, por sua vez, calculam a utilidade que cada consumidor representa para sua coalizão. Os consumidores, então, ingressam na coalizão que obtiver o maior ganho com sua participação. Ao final, após as coalizões cumprirem suas tarefas, elas podem ser desfeitas.

Este mecanismo permite que as coalizões sejam rearranjadas sempre que ocorrer uma mudança no ambiente. Através desta capacidade adaptativa, as coalizões tendem a maximizar a utilidade global. Entretanto, é importante ressaltar que esta abordagem é focada na utilidade global, não individual, dos agentes, ou seja, um agente pode concordar em reduzir sua utilidade caso isto aumente o bem estar social. Nesta abordagem, todos os elementos do DCF-S estão presentes. A diferença fica por conta da simulação, que não analisa todas as coalizões possíveis. Embora analisar um número menor de coalizões reduza a complexidade do algoritmo, isto também reduz a possibilidade de se encontrar a solução ótima.

A abordagem de Mihailescu et al. (2011a, 2011b, 2011c) foi testada em diferentes cenários. No primeiro cenário, foram utilizados 100 agentes, distribuídos em três topologias de rede distintas: aleatória¹, *small-world*² e *scale-free*³. Os fornecedores possuem capacidade entre 50kW e 250kW e assume-se que as coalizões tenham, em média, 1MW de capacidade. A métrica de eficiência adotada para avaliar uma solução baseia-se na quantidade de agentes dispostos a participar de coalizões (por assim ser mais lucrativo). Neste cenário, pôde-se observar que esta abordagem apresentou bons resultados para os três tipos de redes. Entretanto, a maior eficiência se deu na rede *small-world* (93%), seguida da *scale-free* (85%) e por último a aleatória (78%).

No segundo cenário, o experimento foi modelado de forma que fossem formadas no máximo 1000 coalizões. O número de agentes e a topologia não foram divulgados. Entretanto, foi mencionado que cada agente possui no máximo 6 vizinhos. As coalizões, por sua vez, poderiam ter no máximo 500 DERs, atingindo até 6MW de capacidade energética. Assumiu-se, ainda, que os fornecedores possuem no máximo 25kW. O desempenho desta abordagem foi comparado com uma abordagem egoísta⁴. Com base nos experimentos, pôde-se observar que o bem-estar social ficou em torno de 75% do ótimo⁵ com a abordagem proposta. Por outro lado, a abordagem egoísta atingiu um bem-estar social de apenas 30% do ótimo.

Outro trabalho que aborda a formação de coalizões em *smart grids* do ponto de vista de DCF é proposto por Vasirani et al. (2011), cujo foco é formar coalizões entre aerogeradores e EVs. O objetivo desta abordagem é utilizar as baterias dos EVs para amenizar a intermitência produtiva dos aerogeradores, aumentando, desta forma, a confiabilidade dos aerogeradores. Esta abordagem é viável graças ao fato de que EVs permanecem esta-

¹Em uma rede aleatória, dois agentes quaisquer possuem uma aresta entre si com uma dada probabilidade.

²Em uma rede *small-world*, a maioria dos agentes não possui arestas entre si, porém a maioria dos agentes pode acessar qualquer outro agente com uma pequena quantidade de passos.

³Em uma rede *scale-free*, a probabilidade de um agente se conectar com outro agente é proporcional ao número de conexões que ele possui.

⁴Por estratégia egoísta, os autores assumiram um ótimo de Pareto, onde um DER só participa de uma coalizão caso seu ganho imediato seja maior que o atual (desconsiderando o bem-estar social).

⁵O ótimo, neste caso, não se refere necessariamente à estrutura de coalizão ótima, mas à alocação ótima dos recursos energéticos entre os agentes.

cionados em média 96% do tempo, conforme mencionado anteriormente na Seção 4.3.

A abordagem de Vasirani et al. (2011) funciona através de quatro etapas. Na primeira etapa, os DERs mais representativos tornam-se líderes e ficam responsáveis pela formação de uma coalizão cada. Nesta etapa, os líderes de coalizão utilizam-se de mecanismos de confiança para escolher seus parceiros. Na segunda etapa, cada membro reporta a previsão de sua produção no dia seguinte ($n + 1$) ao líder de sua coalizão. Os líderes utilizam esta informação para balancear a produção de sua coalizão através do uso das baterias dos EVs. Com base nestes cálculos, os líderes leiloam a disponibilidade energética de suas coalizões para o dia $n + 1$. Na terceira etapa, já no dia $n + 1$, a energia é fornecida à rede elétrica e o pagamento é repassado às coalizões. Finalmente, na quarta etapa, o valor obtido é dividido entre os membros das coalizões, garantindo a equidade entre os agentes. Ao final, as coalizões podem ser dissolvidas. Esta abordagem é interessante e utiliza-se de diversos mecanismos para otimizar o ganho dos agentes. Entretanto, por se tratar de um trabalho em andamento, não foram apresentados resultados.

Portanto, abordagens de formação dinâmica de coalizões representam uma alternativa interessante para o problema de formação de coalizões em *smart grids*. Entretanto este tipo de mecanismo é totalmente *ad hoc*, ou seja, não garante que uma solução ótima será encontrada, nem sequer estabelece um limite em relação à solução ótima. Sua vantagem está justamente na sua capacidade em lidar com cenários onde ocorrem mudanças constantemente. Por outro lado, fica evidente que algoritmos *anytime* de CSG podem ser facilmente adaptados para este cenário, bastando que sejam reexecutados cada vez que houver alguma mudança no ambiente.

4.5 Discussão

A formação de coalizões é, indubitavelmente, uma ótima alternativa para melhorar o desempenho de *smart grids* sob diversas perspectivas. Entre as melhorias, pode-se citar uma produção energética mais distribuída, matrizes energéticas renováveis mais robustas e confiáveis, entre outros. Neste capítulo foram apresentados diversos trabalhos para esta finalidade, empregando abordagens que vão de *mechanism design* até a formação dinâmica de coalizões.

Na literatura não foram encontradas quaisquer alternativas para a geração de estruturas de coalizão ótimas em *smart grids*. Uma explicação plausível para esta tendência pode ser encontrada nos trabalhos que abordam problemas de DCF. Nestes trabalhos, sugere-se que *smart grids* são cenários dinâmicos demais, o que inviabilizaria a geração de soluções ótimas para um grande número de agentes em um tempo viável. Porém, neste capítulo foi discutido que algoritmos *anytime* de CSG podem ser facilmente adaptados para cenários dinâmicos, invalidando qualquer hipótese de que apenas abordagens de DCF são apropriadas para cenários dinâmicos.

Portanto, com base nesta análise bibliográfica, fica evidente a carência de abordagens que busquem solucionar o problema de formação de coalizões em cenários realistas. As abordagens existentes situam-se em dois extremos: de um lado encontram-se abordagens genéricas capazes de gerar de estruturas de coalizão ótimas porém para um número limitado de agentes (abordadas no Capítulo 3) e de outro lado encontram-se abordagens voltadas para cenários realistas capazes de lidar com um número maior de agentes (abordadas neste capítulo). O foco deste trabalho é unir estes dois extremos, fornecendo uma alternativa capaz de gerar estruturas de coalizão ótimas, em tempo aceitável, para um número maior de agentes e que funcione no cenário de *smart grids*.

5 ABORDAGEM

Conforme discutido nos capítulos anteriores, a formação de coalizões é um meio eficiente de aumentar a lucratividade de EVs que participam de sessões V2G. No entanto, como demonstrado no Capítulo 3, o processo de geração de estruturas de coalizão é muito complexo. A hipótese deste trabalho é que é possível estabelecer restrições no processo de geração de estruturas de coalizão através do uso de informações sobre o domínio de *smart grids*, permitindo a poda de níveis do grafo de estruturas de coalizão e, conseqüentemente, possibilitando que soluções ótimas para um número maior de agentes sejam encontradas em tempo viável. Desta forma, é proposto o algoritmo *constraint-based pruning of coalition structures graph* (CPCSG) que, como o próprio nome diz, é capaz de podar o grafo de estruturas de coalizão com base em restrições existentes entre os agentes.

Neste capítulo é apresentada a abordagem desenvolvida neste trabalho. Inicialmente, a Seção 5.1 apresenta a modelagem básica do problema. Em seguida, o uso de informações sobre o domínio de *smart grids* é exposto na Seção 5.2 e sua aplicabilidade na poda do espaço de busca é discutida na Seção 5.3. A Seção 5.4 apresenta em detalhes o algoritmo CPCSG. Finalmente, a Seção 5.5 realiza uma discussão sobre a abordagem apresentada.

5.1 Modelagem do problema

O cenário definido para este trabalho consiste em uma *smart grid* onde EVs participam de sessões V2G. Conforme discutido nos Capítulos 2 e 4, EVs não são capazes de obter lucros ao atuar individualmente no mercado de energia. Neste sentido, a formação de coalizões entre EVs surge como uma alternativa capaz de solucionar este problema. Na modelagem deste cenário, assume-se que a rede fornece um incentivo financeiro para a formação de coalizões entre os EVs. Este incentivo é proporcional ao tamanho da coalizão, de modo que quanto maior a coalizão, maior o incentivo.

Neste trabalho, assume-se que a rede sempre está disposta a comprar a energia fornecida pelos EVs. Em um cenário real, a compra de energia de EVs estaria condicionada às curvas de oferta e demanda. Entretanto, para simplificar o problema e centrar o foco do trabalho na formação de coalizões, tais aspectos são abstraídos da modelagem. Especificamente, sempre que um EV quiser vender sua energia, a rede a comprará.

O problema proposto pode ser representado na forma de jogos de função característica. O valor da função característica representa o quanto de incentivo a rede fornece para uma determinada coalizão. O incentivo é calculado em função da quantidade de energia que a coalizão tem disponível (quanto mais, melhor) e do tamanho da coalizão (subaditividade e superaditividade são evitados). Dadas estas características de uma coalizão, a rede define um incentivo financeiro para beneficiar as coalizões que se enquadrem nos requisitos desejados. Este incentivo é somado ao valor normal de uma unidade de ener-

gia. Em termos práticos, o valor da coalizão corresponde à quantia que a rede vai pagar a mais por unidade de energia. Por exemplo, se o preço normal de uma unidade de energia fosse \$1,00 por kWh e a rede comprasse cada unidade de energia de uma dada coalizão C por \$1,05, então $v(C)$ seria igual a 0,05. Seguindo este mesmo raciocínio, um agente atuando sozinho no mercado, ou seja, que não participa de uma coalizão, não receberia qualquer incentivo da rede, logo seu valor seria zero.

Neste trabalho foi modelado apenas um agente, o EV. O objetivo do agente EV, é vender a energia disponível em sua bateria para a rede, obtendo o maior lucro possível. No entanto, um EV não pode vender toda a energia contida em sua bateria para a rede, pois isto impediria que seu proprietário o utilizasse para sua próxima viagem. Desta forma, o veículo sempre precisa manter uma quantidade reserva na sua bateria, suficiente para suprir ao menos as necessidades mais urgentes de seu proprietário. Para simplificar a modelagem, assume-se que o veículo sabe exatamente a quantidade mínima de energia que ele precisa manter em sua bateria para esta finalidade. Especificamente, considera-se que todos agentes da simulação estejam sempre aptos a participar de sessões V2G.

O problema de CSG é naturalmente assumido como um sistema fechado, de tal forma que o número de agentes permaneça o mesmo ao longo do tempo. Isto significa que o conjunto de agentes permanece intacto do início ao fim do problema. Tal característica facilita a modelagem do problema e torna este tipo de solução mais genérica. Por outro lado, adaptar este problema para um cenário dinâmico, onde o conjunto de agentes sofre variações ao longo do tempo, é simples: basta reexecutar o algoritmo de CSG sempre que ocorrer uma mudança no sistema.

5.2 Informação sobre o domínio

O uso de informações sobre o domínio permite que sejam estabelecidos critérios para a execução de podas no espaço de busca de estruturas de coalizão. Neste sentido, é necessário identificar alguma característica do domínio que restrinja a formação de determinadas coalizões, como alguma restrição física ou lógica.

No que se refere a *smart grids*, a distância entre os EVs pode ser utilizada com uma restrição física da rede (RAMOS; BAZZAN, 2012b). Segundo Ramchurn et al. (2012), isto vai de encontro ao fato de que as linhas de distribuição possuem capacidades limitadas para o fluxo de energia. Levando em conta que para fornecer energia para algum consumidor diversas linhas de distribuição podem ser percorridas, longas distâncias podem acarretar sobrecarga na rede.

Para explicar melhor o motivo da restrição de distância, considere como exemplo uma *smart grid* no formato de um quadrado. Considere quatro EVs que desejam vender energia para a rede, cada um localizado em uma de suas extremidades. Suponha ainda que a energia fornecida por eles seja destinada a um único consumidor, localizado no centro do quadrado. Assumindo que a rede utiliza constantemente a capacidade máxima para a qual foi projetada para suportar, o que aconteceria quando os quatro EVs fornecessem energia para o consumidor? A resposta é simples: haveria uma sobrecarga. Sobrecarga, como visto no Capítulo 2, é um dos grandes problemas que se busca evitar em *smart grids*.

Neste sentido dois aspectos devem ser controlados. Em primeiro lugar, os EVs devem fornecer energia apenas para consumidores que estão próximos de si. Isto pode ser controlado diretamente pela rede. Em segundo lugar, os EVs devem formar coalizões apenas com EVs que estejam próximos. Este segundo aspecto, por sua vez, pode ser tra-

tado através de algoritmos de geração de estruturas de coalizão. É importante observar que a distância entre dois EVs leva em consideração a localização geográfica de ambos. Entretanto, medidas de distância alternativas podem ser utilizadas, como uma distância topológica¹, por exemplo. A distância entre dois agentes i e j é denotada por d_{ij}^t .

Portanto, a distância se mostra como um bom critério para definir restrições na formação de coalizões entre EVs. A partir desta constatação, deve ser estabelecida a distância máxima que deve haver entre dois EVs para que eles possam formar uma coalizão. A distância máxima é representada por α e seu valor deve ser estabelecido pela própria rede de modo a refletir a capacidade de suas linhas de distribuição.

A representação deste problema pode ser feita através de um grafo, onde os nodos representam os agentes e as arestas representam as distâncias entre eles. A Figura 5.1 apresenta o exemplo de um grafo para quatro agentes.

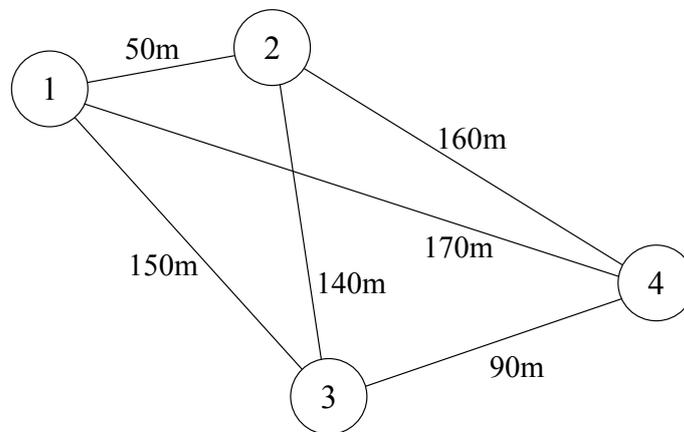


Figura 5.1: Grafo para quatro agentes. Neste grafo, os nodos representam os agentes e as arestas representam as distâncias entre eles

5.3 Podando o espaço de busca

Conforme visto na seção anterior, a distância entre os EVs é um fator que deve ser considerado na formação de coalizões em *smart grids*. Entretanto, o simples fato de identificar coalizões que não podem ser formadas não ajuda a tornar o processo de CSG mais ágil. Isto possibilita, apenas, o descarte de estruturas de coalizão inválidas já identificadas durante a busca. Em outras palavras, o espaço de busca continua sendo percorrido por inteiro. Ao invés de identificar apenas coalizões inválidas, é necessário identificar porções inválidas do espaço de busca. Desta forma, estas porções inválidas podem ser descartadas antes mesmo que a busca seja feita, agilizando o processo de CSG.

Neste trabalho o foco é na poda de níveis inteiros do grafo de estruturas de coalizão. Ao identificar conjuntos de agentes que não podem formar coalizões entre si, é possível podar níveis inteiros do grafo. Entretanto, antes de se aprofundar na lógica por trás desta ideia, dois conceitos se mostram necessários: o de restrições binárias e o de restrições conjuntas.

¹A distância topológica leva em consideração a localização dos EVs na topologia da rede. Neste caso, a distância poderia medir a quantidade de linhas entre os elementos. Por exemplo, do EV1 até o EV2 seria necessário percorrer 3 linhas da distribuição: uma do EV1 até seu transformador, outra do transformador do EV1 até o transformador do EV2, e outra do transformador do EV2 até o EV2.

Uma restrição binária, como o próprio nome diz, é uma restrição que ocorre entre dois agentes. A existência de uma restrição binária entre dois EVs implica que estes não podem fazer parte da mesma coalizão. No caso do grafo da Figura 5.1, supondo que $\alpha = 80m$, por exemplo, existiria uma restrição binária entre os agentes 1 e 3, pois a distância entre eles é maior que α . Da mesma forma, haveria uma restrição binária entre os agentes 1 e 4, entre 2 e 3, entre 2 e 4 e, finalmente, entre 3 e 4. Considerando o grande número de restrições que pode haver, torna-se conveniente representar restrições binárias como restrições de cada agente. Desta forma, seja $B_i \subseteq A$ o conjunto de restrições do agente i , ou seja, todos os agentes com os quais i não pode formar coalizão. Considerando que cada agente possui seu conjunto de restrições, seja $\mathcal{B} = \{B_1, B_2, \dots, B_a\}$ o conjunto de conjuntos de restrições dos agentes. No caso do exemplo, tem-se $B_1 = \{3, 4\}$, $B_2 = \{3, 4\}$, $B_3 = \{1, 2, 4\}$ e $B_4 = \{1, 2, 3\}$, que formam $\mathcal{B} = \{\{3, 4\}, \{3, 4\}, \{1, 2, 4\}, \{1, 2, 3\}\}$. A composição do conjunto B_i de um agente $i \in A$ é dada pela Equação 5.1:

$$B_i = \{j \in A \setminus i, \text{ se } d_j^i > \alpha\} . \quad (5.1)$$

Uma restrição conjunta, por sua vez, representa uma restrição existente entre diversos agentes ao mesmo tempo. Todos os agentes que compartilham uma restrição conjunta possuem restrições binárias entre si. Voltando ao grafo da Figura 5.1 com $\alpha = 80m$, pode-se dizer que os agentes 1, 3 e 4 possuem uma restrição conjunta entre si. Outra restrição conjunta existe entre os agentes 2, 3 e 4. No entanto, não se pode dizer que há uma restrição conjunta entre 1, 2, 3 e 4 (todos agentes), pois os agentes 1 e 2 não possuem uma restrição binária entre si. Logo, uma representação adequada para as restrições conjuntas também se mostra necessária. Assim sendo, seja $J_x \subseteq A$ um conjunto de agentes que possuem restrições binárias entre si, de tal forma que $\forall i \in J_x, \forall j \in J_x \setminus i$, tenha-se $d_j^i > \alpha$. Seja, ainda, $\mathcal{J} = \{J_1, J_2, \dots, J_{|\mathcal{J}|}\}$ o conjunto de todas as restrições conjuntas possíveis. Desta forma, considerando o exemplo anterior, tem-se $J_1 = \{1, 3, 4\}$ e $J_2 = \{2, 3, 4\}$, que compõem $\mathcal{J} = \{\{1, 3, 4\}, \{2, 3, 4\}\}$. É importante observar que, enquanto cada agente possui seu próprio conjunto de restrições B_i , os conjuntos de restrições conjuntas são globais, ou seja, não estão associados a um agente específico.

Tendo por base estes conceitos, surge a seguinte pergunta: como utilizar estas restrições para podar o espaço de busca de estruturas de coalizão? A resposta para esta pergunta está inerentemente ligada às restrições conjuntas. Suponha novamente o exemplo da Figura 5.1, considerando especificamente o conjunto $J_1 = \{1, 3, 4\}$. Com base nesta restrição conjunta, que regras uma estrutura de coalizão deve respeitar para ser válida? Obviamente, é necessário que haja ao menos uma coalizão para cada um dos agentes que faz parte do conjunto de restrições conjuntas J_1 , de modo que cada agente esteja em uma coalizão diferente. Caso contrário, dois ou mais agentes do conjunto de restrições conjuntas J_1 estariam, inevitavelmente, em uma mesma coalizão, violando as restrições existentes. Com base neste raciocínio, pode-se dizer que, no caso deste exemplo, nenhuma estrutura de coalizão com menos de três partições é válida. Isto vai de encontro ao fato de que para estes agentes participarem de coalizões diferentes, ao menos três coalizões são necessárias. Neste sentido, pode-se formular a seguinte proposição.

Proposição 1. Dado um conjunto de restrições conjuntas J_x , nenhuma estrutura de coalizão com menos de $|J_x|$ coalizões é válida. Logo, os níveis $\{1, \dots, |J_x| - 1\}$ do grafo de estruturas de coalizão podem ser podados sem afetar a busca pela solução ótima.

Prova. Considerando que nenhum agente de J_x pode participar da mesma coalizão que os demais agentes deste conjunto, uma estrutura de coalizão precisa de pelo menos $|J_x|$

coalizões para que todos os agentes fiquem separados. Isto implica que qualquer estrutura de coalizão menor com um número de partições menor que $|J_x|$ viola ao menos uma restrição binária entre os agentes de J_x . Como o grafo de estruturas de coalizão é organizado em níveis e em cada nível l as estruturas de coalizão possuem l partições, qualquer nível $l < |J_x|$ é inválido. Portanto, os níveis $\{1, \dots, |J_x| - 1\}$ do grafo podem ser podados sem que isso afete a solução ótima. \square

Tendo por base a Proposição 1, a abordagem proposta neste trabalho busca podar o espaço de busca de estruturas de coalizão a partir das restrições conjuntas entre os agentes. Levando em conta que o objetivo é podar a maior porção possível do espaço de busca, a ideia é identificar o maior conjunto de restrições conjuntas entre os agentes, denotado por J^* , conforme a Equação 5.2. Através desta abordagem, grande parte dos níveis pode ser eliminada antes que a busca comece, de fato, a ser executada.

$$J^* = \arg \max_{J_x \in \mathcal{J}} |J_x| . \quad (5.2)$$

Vale salientar, no entanto, que podar $|J^*| - 1$ níveis não implica em remover todas as estruturas de coalizão inválidas. O foco desta abordagem é podar uma parte considerável do grafo em uma quantidade de tempo pequena, sem que isso envolva uma análise mais extensa no espaço de busca.

5.4 Algoritmo CPCSG

Com base na modelagem proposta nas seções anteriores foi projetado o algoritmo CPCSG (acrônimo de *constraint-based pruning of coalition structures graph*). O algoritmo CPCSG é capaz de identificar restrições conjuntas entre os agentes. Através de restrições conjuntas, é possível identificar níveis podáveis do grafo de estruturas de coalizão, ou seja, níveis que podem ser podados por garantidamente não possuírem a solução ótima. Para desempenhar esta tarefa, o algoritmo precisa receber o conjunto de agentes A e o conjunto de restrições binárias \mathcal{B} . O conjunto de agentes é uma informação prontamente disponível. Já o conjunto de restrições binárias \mathcal{B} deve ser criado a partir da Equação 5.1, utilizando-se o parâmetro α . Este procedimento é trivial, pois requer apenas que cada agente seja testado contra todos os demais. Logo, a complexidade envolvida neste processo é apenas $O(a^2)$.

Identificadas as restrições binárias e criado o conjunto \mathcal{B} , o próximo passo é buscar o maior conjunto de restrições conjuntas possíveis, o J^* . Para esta finalidade, é necessário percorrer cada conjunto de restrições B_i , buscando restrições entre cada um dos agentes. O CPCSG é apresentado no Algoritmo 5.1.

O funcionamento do CPCSG se dá da seguinte forma. Para cada agente é criada uma lista potencial de conjuntos J_x , denominada $lista_i$ (linha 2). O objetivo não é encontrar todos os conjuntos J_x , apenas o maior. Logo, para cada agente i é criado um conjunto J_x contendo, inicialmente, apenas i (adicionado à $lista_i$, na linha 2). Em seguida, tenta-se colocar cada um dos agentes $j \in B_i$ no conjunto J_x , respeitando a regra de que todos os agentes em J_x devem ter restrições entre si (linhas 6 a 12). Cada vez que um agente j não tiver restrições com algum dos elementos do conjunto J_x (linhas 6 a 10), deve-se criar um novo conjunto J_x para o agente j , contendo todos os elementos de J_x exceto os que não têm restrições com j (linhas 14 e 15). Este novo conjunto J_x também é adicionado à $lista_i$ (linha 20). É por este motivo que cada agente i possui uma $lista_i$. A $lista_i$ armazena diversos conjuntos J_x , de modo que cada elemento de B_i esteja em

Algoritmo 5.1: Algoritmo CPCSG para encontrar a quantidade de níveis que podem ser podados do grafo de estruturas de coalizão

Requer : A : o conjunto de agentes
: B : o conjunto de restrições binárias dos agentes
Retorna: a quantidade de níveis que podem ser podados do grafo

```

1 para cada  $i \in A$  faça
2    $lista_i \leftarrow \{\{i\}\}$ ;
3   para cada  $j \in B_i : j > i$  faça
4      $toAdd \leftarrow \{\}$ ;
5     para cada  $li \notin lista_i$  faça
6       para cada  $k \in li$  faça
7         se  $j \notin B_k$  então
8            $restr_{li}^j \leftarrow restr_{li}^j \cup k$ ;
9         fim se
10      fim para cada
11      se  $restr_{li}^j = \emptyset$  então
12         $li \leftarrow li \cup j$ ;
13      senão
14         $novo \leftarrow li \setminus restr_{li}^j$ ;
15         $novo \leftarrow novo \cup j$ ;
16         $toAdd \leftarrow toAdd \cup novo$ ;
17      fim se
18      fim para cada
19      se  $toAdd \neq \emptyset$  então
20         $lista_i \leftarrow lista_i \cup toAdd$ ;
21      fim se
22      fim para cada
23       $\mathcal{J} \leftarrow \mathcal{J} \cup lista_i$ ;
24 fim para cada
25 retorna  $\max_{J_x \in \mathcal{J}} |J_x|$ ;

```

pelo menos um deles. Deve ficar claro que o algoritmo adiciona os agentes j a cada um dos $J_x \in lista_i$, não apenas ao inicial (por este motivo é executado o laço da linha 5). Desta forma, garante-se que todas as maiores combinações de restrições conjuntas serão testadas. Após ter percorrido todo o conjunto de restrições B_i do agente i , o algoritmo adiciona as restrições conjuntas encontradas ao conjunto \mathcal{J} (linha 23) e repete o processo para os demais agentes $\{i + 1, \dots, a\}$ (volta à linha 1). Ao final, basta verificar qual o maior J_x encontrado e retornar o seu tamanho (linha 25).

Conforme pode ser observado na linha 3, existe a condição $j > i$. O objetivo desta condição é evitar que conjuntos J_x redundantes sejam calculados. Tal condição é válida porque, considerando que o agente i analisa as restrições do agente j e encontra o maior conjunto J_x do qual ambos fazem parte, se o agente j fizesse o mesmo processo para o agente i ele obteria o mesmo resultado. Logo, não faz sentido ter executar estes cálculos redundantes. A complexidade do algoritmo é demonstrada através da proposição a seguir.

Proposição 2. A complexidade do algoritmo CPCSG é $O(a^4)$.

Prova. O primeiro laço do algoritmo (linha 1) sempre será executado a vezes. O segundo

laço (linha 3), no pior caso (ou seja, quando todos agentes possuem restrições com todos), será executado $a - 1$ vezes.

O terceiro laço (linha 5) percorre a lista de conjuntos $lista_i$. Esta lista é formada através do quarto laço (linha 6). Neste sentido, duas possibilidades podem ocorrer. Na primeira, existe um único conjunto na $lista_i$, com $a - 2$ elementos. Logo, no pior caso, o quarto laço (linha 6) percorreria $a - 2$ elementos deste conjunto. Na segunda possibilidade, existem $a - 2$ conjuntos na $lista_i$, cada um com 2 elementos. Assim, no pior caso, o quarto laço (linha 6) percorreria apenas 2 elementos de cada conjunto. Logo, o terceiro e o quarto laço executam juntos, no pior caso, $a - 2$ iterações.

Finalmente, deve-se analisar as demais operações além dos laços. Dentre estas operações, há inicialização de variáveis (linhas 2 e 4), união (linhas 8, 12, 15, 16, 20 e 23) e diferença (linha 14) de conjuntos. Com exceção da diferença de conjuntos, todas estas operações são executadas em tempo constante. No caso da diferença de conjuntos, o tempo é linear em $|li|$. Já que $|li|$ se dá em função dos laços 3 e 4, seu tempo no pior caso é $a - 2$.

Desta forma, agregando-se a análise das diversas porções do algoritmo, tem-se as seguintes operações: laço 1 (a); laço 2 ($a - 1$); laços 3 e 4 ($a - 2$); diferença de conjuntos da linha 14 ($a - 2$). Logo, tem-se $a \times a - 1 \times a - 2 \times a - 2$. Removendo-se as constantes, tem-se $a \times a \times a \times a = a^4$. Portanto, a complexidade do algoritmo, no pior caso, é de $O(a^4)$. \square

É interessante observar que, idealmente, quanto maior o conjunto J_x encontrado nas restrições binárias de um agente i , melhor. Em outras palavras, na melhor das hipóteses todos os agentes do conjunto B_i de um agente terão restrições entre si e apenas um conjunto J_x será criado para ele. Por outro lado, na pior das hipóteses, nenhum dos agentes de B_i possui restrições entre si, resultando em um conjunto J_x para cada restrição binária. Obviamente, o primeiro caso é preferível, pois é o que permite a maior poda no grafo. Caso nenhuma restrição conjunta seja encontrada, o grafo não pode ser podado.

Vale lembrar que o algoritmo CPCSG é voltado especificamente para a poda do espaço de busca, não para a busca em si. A proposta deste trabalho é que a técnica desenvolvida seja aplicada ao algoritmo IP (Seção 3.3.3.2). Especificamente, o algoritmo CPCSG pode ser executado logo após a etapa de pré-processamento do algoritmo IP. Desta forma, a poda é feita antes que a busca pela estrutura de coalizão ótima inicie, evitando a busca em níveis desnecessários. O algoritmo CPCSG também pode ser facilmente empregado em outros algoritmos que se baseiam no grafo de estruturas de coalizão, como o CSS1 (Seção 3.3.3.1). Já com os algoritmos DP e IDP (Seções 3.3.2.2 e 3.3.2.2) a técnica não se mostra compatível, pois o espaço de busca destes é estruturado de uma maneira diferente.

De forma a facilitar o entendimento do funcionamento do algoritmo CPCSG, propõe-se o seguinte exemplo.

Exemplo 1. Seja $A = \{1, 2, 3, 4, 5\}$, $\alpha = 100m$ e o cenário apresentado na Figura 5.2.

Inicialmente, é necessário compor o conjunto de restrições binárias \mathcal{B} com base na Equação 5.1. O agente 1 possui restrições binárias com os agentes 4 e 5, logo $B_1 = \{4, 5\}$. O agente 2 não pode formar coalizões com os agentes 3 e 5, resultando em $B_2 = \{3, 5\}$. Para o agente 3, tem-se $B_3 = \{2, 4\}$. O agente 4 tem $B_4 = \{1, 3, 5\}$. E, finalmente, o agente 5 tem $B_5 = \{1, 2, 4\}$. Assim pode ser montado o conjunto de restrições binárias $\mathcal{B} = \{\{4, 5\}, \{3, 5\}, \{2, 4\}, \{1, 3, 5\}, \{1, 2, 4\}\}$. Em seguida, deve-se executar o algoritmo CPCSG (Algoritmo 5.1) para identificar quantos níveis do grafo podem ser podados.

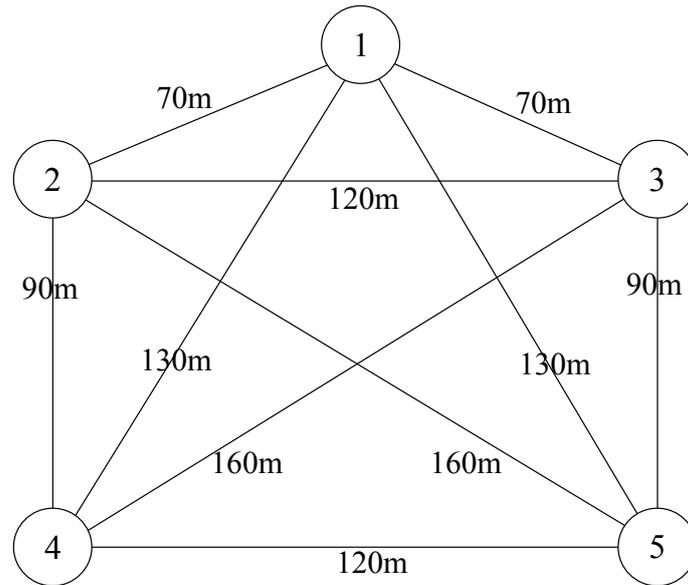


Figura 5.2: Cenário hipotético utilizado no exemplo 1

A Tabela 5.1 mostra passo a passo o funcionamento do algoritmo CPCSG. O processo começa com $i = 1$. No primeiro passo, é criada a $lista_1 = \{\{1\}\}$. Em seguida, começa o teste com cada um dos elementos em $B_1 = \{4, 5\}$. Primeiro $j = 4$, que é adicionado diretamente ao único elemento de $lista_1$, resultando $\{1, 4\}$. Logo após, para $j = 5$, verifica-se se j tem restrições com os outros elementos de $\{4, 5\}$. Como o 5 tem restrições também com o 4, ele pode ser adicionado ao conjunto existente, sem a necessidade de criar outro. Isto resulta na $lista_1 = \{\{1, 4, 5\}\}$, ou seja, um único conjunto de restrições conjuntas em que nenhum agente pode formar coalizões com os outros. Este elemento é adicionado a \mathcal{J} .

O procedimento continua com $i = 2$, criando a $lista_2 = \{\{2\}\}$. É escolhido $j = 3$, que pode ser adicionado ao único conjunto da lista. No passo seguinte, $j = 5$, verifica-se que os elementos 3 e 5 não possuem uma restrição binária entre si. Desta forma, é neces-

Tabela 5.1: Exemplo de funcionamento do algoritmo CPCSG para o exemplo 1

i	j	$list_i$	\mathcal{J}
1		$\{\{1\}\}$	
	4	$\{\{1, 4\}\}$	
	5	$\{\{1, 4, 5\}\}$	$\{\{1, 4, 5\}\}$
2		$\{\{2\}\}$	
	3	$\{\{2, 3\}\}$	
	5	$\{\{2, 3\}, \{2, 5\}\}$	$\{\{1, 4, 5\}, \{2, 3\}, \{2, 5\}\}$
3		$\{\{3\}\}$	
	4	$\{\{3, 4\}\}$	$\{\{1, 4, 5\}, \{2, 3\}, \{2, 5\}, \{3, 4\}\}$
4		$\{\{4\}\}$	
	5	$\{\{4, 5\}\}$	$\{\{1, 4, 5\}, \{2, 3\}, \{2, 5\}, \{3, 4\}, \{4, 5\}\}$

sário criar outro conjunto na lista sem o elemento 3, mas com o elemento 5, resultando em $lista_2 = \{\{2, 3\}, \{2, 5\}\}$. Ao final, os elementos da $lista_2$ também são adicionados a \mathcal{J} .

O procedimento para os demais i 's ocorre da mesma forma que os anteriores. Resta comentar o funcionamento do mecanismo para evitar redundância. Como pode ser observado na Tabela 5.1, são considerados apenas os i 's e j 's que respeitem $i < j$. Por este motivo, o elemento 5 nem aparece na tabela, pois não existe outro agente com um número maior que o dele.

No final, o algoritmo identifica que o maior conjunto de restrições conjuntas tem tamanho 3, implicando que os 2 níveis inferiores do grafo podem ser podados. De acordo com a Equação 3.2, para cinco agentes existem 52 estruturas de coalizão possíveis. Similarmente, através da Equação 3.1 é possível calcular a porção equivalente aos dois níveis inferiores do grafo, que corresponde a 16 estruturas de coalizão. Isto representa uma poda de aproximadamente 30,77% do espaço de busca.

5.5 Discussão sobre a abordagem

No início deste capítulo foi definida a hipótese deste trabalho. A hipótese é que é possível estabelecer restrições no processo de geração de estruturas de coalizão, através do uso de informações sobre o domínio de *smart grids*, permitindo a poda de níveis do grafo de estruturas de coalizão e, conseqüentemente, possibilitando que soluções ótimas para um número maior de agentes sejam encontradas em tempo viável. Seguindo esta ideia foi apresentado o algoritmo CPCSG, capaz de fazer justamente isto: usar informações sobre o domínio para podar o grafo.

Inicialmente, foi realizada uma discussão sobre algumas das restrições físicas encontradas em *smart grids*. Mostrou-se que estas restrições, quando desrespeitadas, podem gerar impactos muito grandes na rede. Com base neste cenário, foi possível criar um formalismo que permite a utilização destas restrições para melhorar o desempenho de algoritmos de CSG. Especificamente, o tratamento adequado de tais restrições viabiliza a poda no espaço de busca de estruturas de coalizão.

O algoritmo CPCSG é capaz de reduzir consideravelmente o espaço de busca de estruturas de coalizão. Ficou evidente que, quanto maior o número de restrições, maior o espaço que pode ser podado. O resultado disto é a redução do tempo para encontrar uma solução ótima. Conseqüentemente, tal modelagem tem uma aplicabilidade mais relevante em cenários reais, como *smart grids*, do que abordagens genéricas convencionais.

Uma das vantagens do CPCSG é que ele pode ser utilizado em conjunto com diversos algoritmos de geração de estruturas de coalizão. Considerando que sua poda é baseada no grafo de estruturas de coalizão, esta abordagem pode ser adaptada para trabalhar com qualquer outro algoritmo que se baseie na mesma representação do espaço de busca.

6 EXPERIMENTOS

Este capítulo apresenta uma investigação a respeito do comportamento da abordagem proposta, tratando de seus aspectos próprios, bem como sua influência no processo de geração de estruturas de coalizão. Neste capítulo, questões referentes ao desempenho do algoritmo CPCSG e à sua influência na busca da solução ótima são abordadas e comparativos com outros algoritmos são realizados.

Este capítulo encontra-se organizado em 5 seções, começando pela Seção 6.1, que apresenta a metodologia adotada na realização dos experimentos. A Seção 6.2 avalia o desempenho do algoritmo CPCSG no cálculo da quantidade de níveis que podem ser podados do grafo. Na Seção 6.3 é analisado o desempenho da abordagem proposta para encontrar a solução ótima. Em seguida, a Seção 6.4 efetua uma comparação da abordagem proposta com outros algoritmos de CSG. Finalmente, a Seção 6.5 realiza uma discussão sobre os resultados obtidos.

6.1 Metodologia

No que diz respeito à avaliação do método proposto neste trabalho, diversas questões devem ser investigadas. Inicialmente, surge a questão: qual o custo envolvido no cálculo do maior conjunto de restrições conjuntas entre os agentes, J^* (conforme definido na Seção 5.3)? Através desta questão, busca-se entender a possibilidade do algoritmo CPCSG onerar o processo de geração de estruturas de coalizão. Como o objetivo do trabalho é melhorar o desempenho do algoritmo IP, a solução final deve, no mínimo, apresentar desempenho equivalente ao algoritmo IP original.

Em um segundo momento, uma questão ainda mais importante deve ser levantada: qual o impacto da técnica proposta no algoritmo IP? Em outras palavras, o algoritmo CPCSG é capaz de melhorar o desempenho do algoritmo IP? Esta questão chama atenção para o desempenho da abordagem como um todo, o qual deve ser utilizado, posteriormente, para comparação com outros algoritmos. Ao investigar esta questão, espera-se que a abordagem proposta possua um desempenho superior ao do algoritmo IP na maioria dos cenários.

Finalmente, surge a necessidade de se comparar a abordagem proposta com outras abordagens existentes. A questão a ser respondida é: a técnica proposta é mais eficiente que outros algoritmos existentes? Com esta investigação, não se espera concluir, necessariamente, que a abordagem proposta seja melhor que as demais em todos os casos. Para realizar esta investigação, podem ser utilizados os mesmos métodos utilizados na investigação da questão anterior.

No intuito de responder estas questões, alguns pontos devem ser considerados. Em primeiro lugar, considerando que a poda de níveis do grafo é o foco central desta aborda-

gem, na realização de experimentos faz-se necessário a existência de um parâmetro que permita definir a quantidade de níveis que serão podados. A alternativa óbvia é o parâmetro α (conforme definido na Seção 5.2). No entanto, é preciso compreender o impacto do parâmetro α no desempenho do sistema para saber se ele é ou não apropriado para a finalidade desejada. O parâmetro α é um elemento chave nesta abordagem, pois é ele que impõe restrições entre os agentes, viabilizando a poda do espaço de busca. Porém, o α é ligado apenas indiretamente à quantidade de níveis que serão podados. De maneira geral, quanto mais restritivo o α , maior a probabilidade de haver restrições entre os agentes e, conseqüentemente, maior a chance de haver restrições conjuntas maiores. Entretanto, esta relação nem sempre é tão estreita, pois ela é afetada pela disposição dos agentes no ambiente. Logo, o parâmetro α não é apropriado para definir com precisão a quantidade de níveis que serão podados do grafo. Conseqüentemente, o α não pode ser utilizado na avaliação do desempenho desta abordagem.

Uma alternativa ao parâmetro α é justamente o conjunto J^* . Embora o conjunto J^* seja calculado a partir das restrições impostas pelo α , para fins de experimentação pode-se definir o tamanho desejado para o parâmetro J^* e definir arbitrariamente $|J^*|$ agentes para compor este conjunto. Estes $|J^*|$ agentes seriam então os únicos a possuir restrições binárias entre si. Desta forma, é possível definir exatamente a quantidade de níveis que serão podados do grafo, facilitando a avaliação da relação entre o tamanho de J^* e o desempenho do algoritmo.

No que se refere ao número de agentes (a), pôde-se perceber que os algoritmos não são capazes de entregar soluções em um tempo viável para mais de 20 agentes. Desta forma, optou-se por realizar os experimentos com no máximo 20 agentes. Optou-se ainda por restringir os experimentos a no mínimo 10 agentes, pois uma quantidade menor que essa não representa um cenário suficientemente interessante para a formação de coalizões.

Considerando que o enfoque deste trabalho é na formação de coalizões em jogos de função característica, é importante falar da modelagem adotada para os experimentos. Neste trabalho, foi criada uma função característica mais realista e apropriada para *smart grids*, a qual é formulada através da Equação (6.1), onde γ representa o tamanho máximo desejado para uma coalizão, δ define a quantidade de energia que a rede espera que cada coalizão seja capaz de fornecer, ϵ é o incentivo financeiro máximo que a rede pretende dar a uma coalizão, E_C é a quantidade de energia que a coalizão C tem disponível e p é o preço normal que a rede costuma pagar por uma unidade de energia.

$$v(C) = \begin{cases} 0, & \text{se } |C| < 2 \text{ ou } |C| > \gamma \\ 0, & \text{se } \exists i \in C, \exists j \in C \setminus i : d_j^i > \alpha \\ \min\left\{\left(\frac{E_C}{\delta}\right)^2 \times \epsilon, \epsilon\right\} * p, & \text{caso contrário} \end{cases} \quad (6.1)$$

Na primeira linha da Equação (6.1), a função característica evita duas características indesejáveis, a subaditividade e a superaditividade, já que coalizões muito pequenas ou muito grandes têm seu valor zerado. Através da segunda linha da Equação (6.1), qualquer coalizão que possua agentes com restrições binárias entre si recebe o valor zero, de modo que a restrição física da rede (abordada na Seção 5.2) seja respeitada. Desta forma, as duas primeiras linhas da Equação (6.1) são responsáveis por atribuir valor zero a todas as coalizões inválidas. Finalmente, a última linha da Equação (6.1) é a que determina o valor das coalizões válidas, ou seja, as que não se enquadram nos critérios da primeira e da segunda linha. O valor de uma coalizão C é calculado com base na quantidade de energia E_C que a mesma possui, de forma que quanto maior a quantidade de energia maior o valor da coalizão. No entanto, considerando que o valor da coalizão representa o quanto

a rede vai pagar pela sua energia além do preço normal, é estabelecido um limite através do parâmetro ϵ , o qual impede que coalizões com muita energia recebam um valor muito alto. O parâmetro ϵ , além de limitar o valor que a rede pagará por unidade de energia, também incentiva que as coalizões se formem visando disponibilizar o mais próximo de δ unidades de energia possível.

É importante salientar que empiricamente pôde-se verificar que variações nos parâmetros da Equação (6.1) (γ , δ , ϵ e p) não influenciam de forma expressiva no resultado final dos algoritmos. Desta forma, o valor atribuído a cada um destes parâmetros não é uma questão relevante para o escopo dos experimentos aqui realizados. De qualquer forma, os valores adotados para os parâmetros em todos os experimentos foram: $\gamma = 7$ (coalizões de no máximo 7 agentes), $\delta = 15$ (a soma da energia disponível em todos os EVs de uma coalizão deve resultar em 15kWh no máximo), $p = 1$ (o preço normal que a rede paga por 1kWh de energia é \$1,00) e $\epsilon = 0.9$ (a rede paga no máximo \$1,90 por kWh de energia, ou seja, o valor de uma coalizão é no máximo 0,9). Finalmente, resta definir a quantidade de energia, E_C , que cada agente possui. Neste sentido, estes valores foram extraídos aleatoriamente a partir de uma distribuição uniforme no intervalo de 1 a 3kWh, considerando que os EVs produzidos atualmente possuem por volta desta capacidade.

Os algoritmos utilizados neste trabalho foram implementados com a linguagem Java versão 1.6. Os experimentos foram executados em um computador com processador Intel(R) Core(TM) i7-2600 3.40GHz, com 16GB de RAM e com sistema operacional Ubuntu 12.04 64 bits. Experimentos cujo tempo de execução foi superior a 24 horas foram extrapolados, mesma lógica adotada no trabalho de Rahwan (2007).

6.2 Desempenho do CPCSG

De modo a avaliar o desempenho do algoritmo CPCSG, é necessário verificar seu tempo de execução para diferentes quantidades de agentes ($a = \{10, \dots, 20\}$). Desta

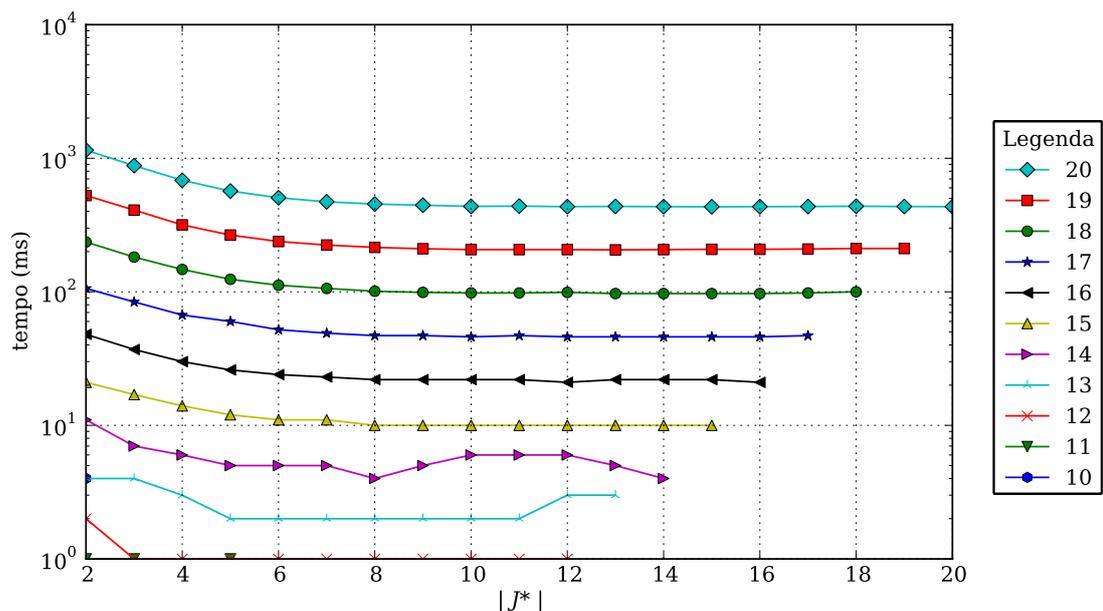


Figura 6.1: Tempo de execução do algoritmo CPCSG, considerando 10 a 20 agentes (séries) e com variação no tamanho de J^* entre 2 e a (eixo x)

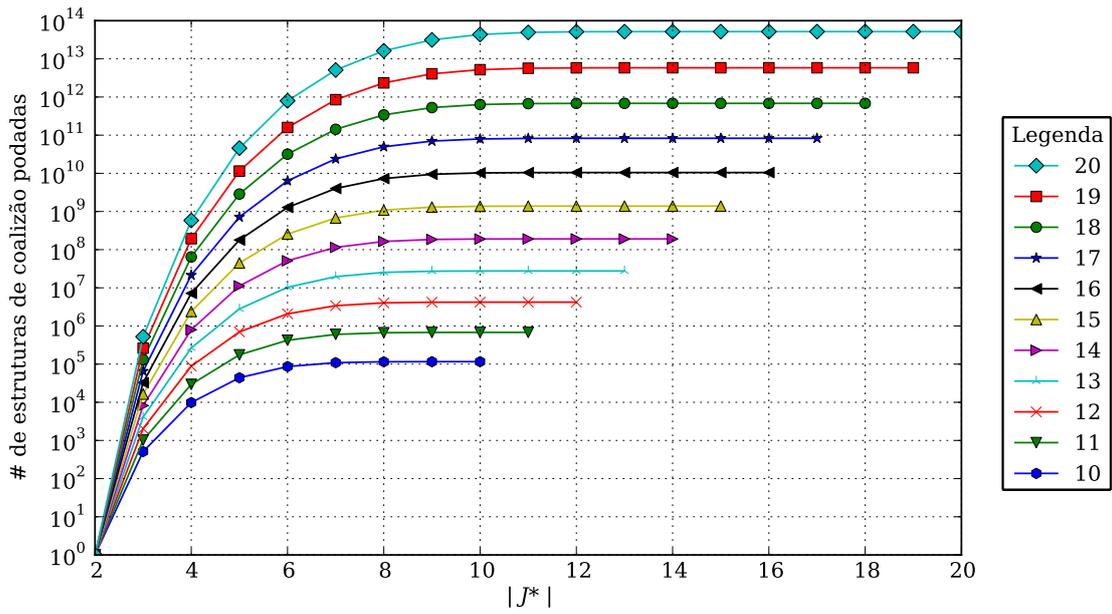


Figura 6.2: Quantidade de estruturas de coalizão que podem ser podadas do espaço de busca, considerando 10 a 20 agentes (séries) e variando o tamanho de J^* de 2 a a (eixo x)

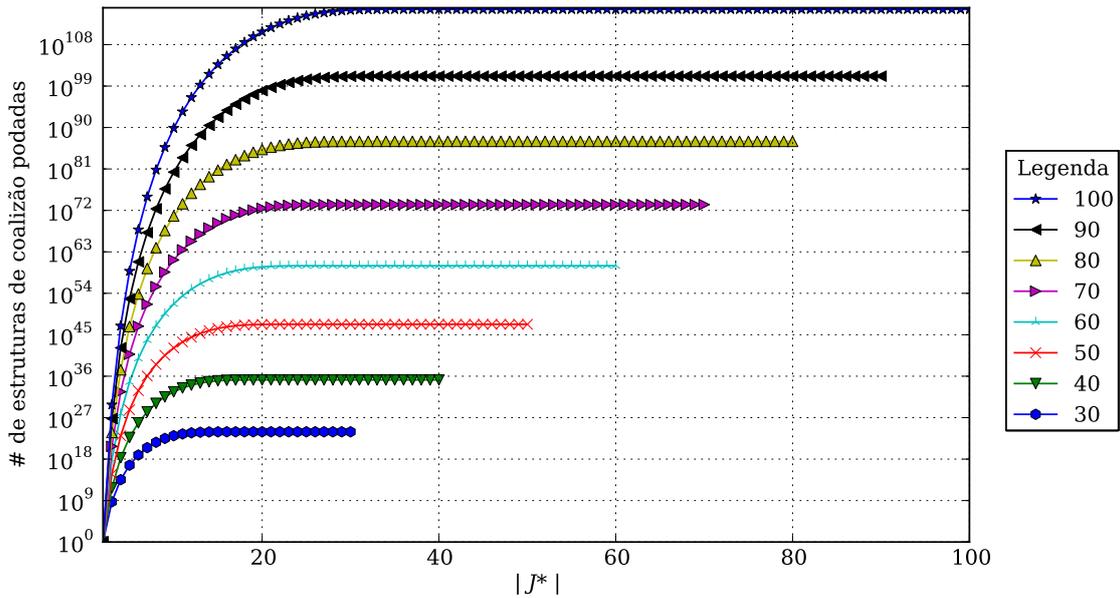


Figura 6.3: Quantidade de estruturas de coalizão que podem ser podadas do espaço de busca, considerando 30 a 100 agentes (séries), somente dezenas, e variando o tamanho de J^* de 2 a a (eixo x)

forma, foram conduzidos experimentos para 10 a 20 agentes, variando o tamanho de J^* de 2 até a (considerando que para $|J^*| = 2$ apenas um nível é podado e que para $|J^*| = a$ são podados $a - 1$ níveis). Os resultados são apresentados na Figura 6.1. No gráfico, o tempo é dado em milissegundos em escala logarítmica, de modo a melhorar a legibilidade.

Surpreendentemente, ao contrário do que se poderia se imaginar, o algoritmo CPCSG demora mais tempo para executar quando J^* é pequeno do que quando ele é grande. Isto se deve ao fato de que, geralmente quando J^* é pequeno, ao invés do algoritmo encontrar

poucos e grandes conjuntos J_x , tem-se muitos e pequenos J_x . Como consequência, um número maior de operações acaba sendo realizado, acarretando em um desempenho ligeiramente pior. Como esperado, no entanto, o algoritmo CPCSG se mostrou muito rápido, desempenhando sua tarefa em uma quantia ínfima de tempo e não impondo custo extra no restante do processo de CSG, o qual é executado pelo algoritmo IP.

Outro ponto importante na análise do desempenho da abordagem proposta neste trabalho refere-se a mensurar o quanto pode ser podado do espaço de busca. A Figura 6.2 detalha o quanto do espaço de busca é podado para diferentes quantidades de agentes (de 10 a 20), variando o tamanho de J^* (de 2 a a , o que significa podar do nível 1 ao $a - 1$). Conforme pode ser observado, os níveis mais baixos do grafo de estruturas de coalizão representam a maior porção do espaço de busca. Tal característica enfatiza que a abordagem proposta é capaz de realizar podas significativas no espaço de busca mesmo quando o conjunto J^* é pequeno. Tomando por base o caso de 20 agentes, por exemplo, a existência de uma restrição conjunta entre três agentes é o suficiente para podar dois níveis do grafo de estruturas de coalizão, o que corresponde a cerca de 5×10^5 estruturas de coalizão. No caso de uma restrição conjunta entre quatro agentes, aproximadamente 5×10^8 estruturas de coalizão podem ser podadas.

Em suma, fica claro que o espaço de busca a ser podado cresce muito rapidamente à medida que o número de agentes e o tamanho de J^* aumentam. Por outro lado, fica evidente que o espaço de busca para um número grande de agentes é imenso. Para se ter uma noção da ordem de grandeza do espaço de busca e do quanto pode ser podado, a Figura 6.3 apresenta os mesmos experimentos da Figura 6.2, porém para 30 a 100 agentes (apenas as dezenas). É importante ressaltar, no entanto, que embora a poda aproxime-se rapidamente de 100% do espaço de busca, sob hipótese alguma é possível podar 100% do espaço de busca. No melhor caso, onde todos agentes possuem restrições binárias entre si, pode-se podar $a - 1$ níveis, de forma que seja mantido apenas o nível cuja única estrutura de coalizão possui todos agentes em coalizões separadas.

6.3 Tempo para encontrar a CS^*

Nesta seção o impacto da poda no processo de geração de estruturas de coalizão é discutido e investigado. Conforme detalhado no Capítulo 5, a abordagem proposta neste trabalho consiste em podar níveis do grafo de estruturas de coalizão antes que a busca pela solução ótima, desempenhada pelo algoritmo IP, seja iniciada. Este mecanismo, é executado logo após o pré-processamento do algoritmo IP, podando todas as configurações pertencentes aos níveis menores que $|J^*|$. Com base na forma como esta abordagem foi modelada, deve-se investigar o impacto das podas no tempo de execução do algoritmo IP. Em outras palavras, o quanto a poda melhora o desempenho do processo de geração de estruturas de coalizão.

Na avaliação do tempo de execução do algoritmo IP em conjunto com o CPCSG, foram conduzidos experimentos com 10 a 20 agentes, variando, para cada experimento, o tamanho de $|J^*|$ de 2 até a . Os resultados obtidos com estes experimentos são apresentados na Figura 6.4, onde o tempo é dado em milissegundos com escala logarítmica, de modo a facilitar a leitura.

Conforme pode ser observado no gráfico, através da poda do grafo de estruturas de coalizão é possível reduzir o tempo de execução do algoritmo em diversas ordens de grandeza. Considerando como exemplo o caso de 17 agentes, ao podar apenas um nível ($|J^*| = 2$) o algoritmo demora cerca de 10^5 ms para encontrar a solução ótima. Aumen-

tando a poda para 9 níveis ($|J^*| = 10$), o tempo é reduzido para aproximadamente 10^4 ms. Aumentando a poda ainda mais, para 12 níveis ($|J^*| = 13$), o tempo de execução cai para cerca de 10^2 ms apenas.

Vale ressaltar que, se por um lado a poda permite uma redução considerável no tempo de execução do algoritmo, por outro percebe-se que este ganho só ocorre de forma expressiva a partir do momento que o tamanho de J^* atinge cerca 6 a 8 agentes. Pode-se observar ainda que quando J^* tem menos de 6 agentes, em geral há uma ligeira piora no desempenho do algoritmo. Isto ocorre porque o algoritmo IP tende a começar sua busca nas configurações pertencentes aos níveis inferiores do grafo (que possuem menos estruturas de coalizão), permitindo a poda de outras configurações em menos tempo. Entretanto, como o CPCSG poda os níveis mais baixos do grafo, o IP acaba iniciando sua busca nas configurações de níveis maiores (que possuem mais estruturas de coalizão), consequentemente levando mais tempo para podar outras configurações. Porém, quando o tamanho de J^* é grande o suficiente (a partir de 6 a 8 agentes), configurações maiores também acabam sendo podadas. Consequentemente, a melhora no desempenho do algoritmo passa a ser mais significativa.

É importante observar que, como a melhora no desempenho ocorre sempre a partir de quando J^* tem entre 6 e 8 agentes, reduz-se o percentual de agentes necessários em J^* para reduzir o tempo do algoritmo. Por exemplo, no caso de 15 agentes, quando $|J^*| = 8$ (53% dos agentes no J^*) o tempo de execução do algoritmo cai de aproximadamente 10^3 ms (quando $|J^*| = 2$) para aproximadamente 6×10^2 , o que representa uma redução de aproximadamente 40%. Já no caso de 20 agentes, quando $|J^*| = 8$ (40% dos agentes em J^*) o tempo de execução cai de aproximadamente 6×10^7 (quando $|J^*| = 2$) para aproximadamente $1,6 \times 10^7$, representando uma redução de mais de 70%. Isto mostra que, na abordagem proposta, quanto maior o tamanho de a , menor o percentual de agentes que devem fazer parte de J^* para tornar o desempenho do algoritmo expressivamente melhor.

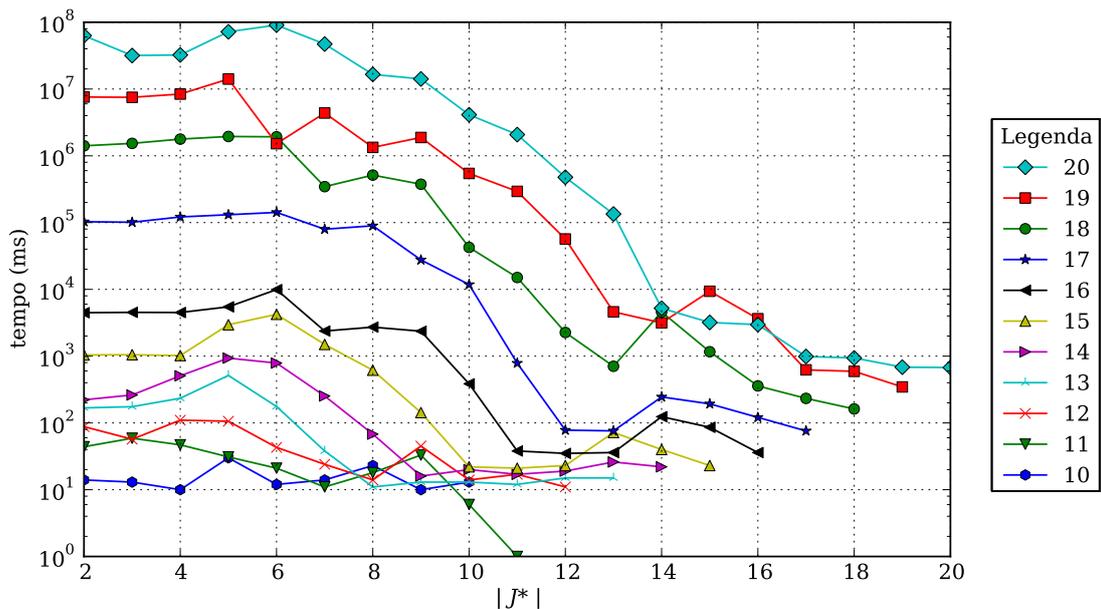


Figura 6.4: Tempo de execução requerido para encontrar a solução ótima, após a realização da poda dos níveis, considerando 10 a 20 agentes (séries) e variando o tamanho de $|J^*|$ de 2 a a (eixo x)

6.4 Comparação com outros algoritmos para encontrar a CS^*

Finalmente, nesta seção, a abordagem proposta por este trabalho é comparada com alguns dos principais algoritmos para geração de estruturas de coalizão, o IDP (RAHWAN; JENNINGS, 2008b) e o IP (RAHWAN et al., 2007).

Inicialmente, antes de dar início às comparações, é necessário enfatizar algumas características dos algoritmos. O algoritmo IP, em termos de função característica, apresenta um desempenho satisfatório sobretudo quando o valor das coalizões é gerado a partir de distribuições uniformes no intervalo de 0 a até $|C|$. Entretanto, como esta classe de problemas não é muito realista, optou-se por utilizar uma função característica própria que traduz mais coerentemente a realidade de *smart grids*. Esta função característica difere-se da utilizada na literatura porque ela não considera apenas o tamanho da coalizão, mas também a quantidade de energia das coalizões e a distância entre os agentes. Com a distribuição adotada, espera-se que os problemas a serem resolvidos pelos algoritmos sejam mais desafiantes. O algoritmo IDP, por outro lado, apresenta a vantagem de não ser sensível à distribuição de valores da função característica, ou seja, seu desempenho não varia de uma distribuição para outra.

A Figura 6.5 apresenta a comparação dos tempos de execução dos três algoritmos, IDP, IP e CPCSG-IP (referindo-se ao IP com o CPCSG), para 10 a 20 agentes. De modo a melhorar a legibilidade, foi criado um gráfico para cada tamanho de população: Figura 6.5a para 10 agentes, Figura 6.5b para 11 agentes, até Figura 6.5k para 20 agentes. Em cada gráfico, o tamanho de J^* foi variado de 2 até a . Desta forma, pode-se comparar facilmente os algoritmos e o impacto das restrições no seu desempenho para cada tamanho de população. Nos experimentos realizados, a função característica descrita anteriormente foi utilizada para os três algoritmos. Vale ressaltar, porém, que a técnica de poda proposta neste trabalho é desempenhada apenas no algoritmo referido como CPCSG-IP.

Conforme pode ser observado nos gráficos da Figura 6.5, o algoritmo CPCSG-IP tende a reduzir em muito seu tempo de execução conforme aumenta o tamanho de J^* . Esta tendência se mostra válida em todos os cenários da Figura 6.5. O algoritmo IDP, por sua vez, não sofre variações no seu tempo de execução em função das restrições de agentes. Isto ocorre porque o IDP sempre percorre uma quantidade linear do espaço de busca ($O(3^a)$), independente do número de agentes em J^* . Comparando-se o algoritmo CPCSG-IP com o IDP, pode-se notar uma leve tendência do CPCSG-IP se sobressair ao IDP antes mesmo que metade dos níveis do grafo sejam podados. Vale ressaltar que, embora o IDP apresente resultados mais interessantes que o CPCSG-IP quando há uma quantidade menor de podas, o IDP continua tendo que percorrer todo seu espaço de busca para encontrar a solução ótima. Em contrapartida, a abordagem deste trabalho é *anytime* e consome menos memória que o IDP, sendo de mais fácil implantação em cenários reais.

Em relação ao IP, o algoritmo CPCSG-IP apresentou um desempenho superior na maioria dos experimentos. Chama atenção o fato de que o IP sofre uma grande piora no seu tempo conforme aumenta a quantidade de agentes no conjunto J^* . Isto ocorre porque, como o IP foi projetado com base em técnicas de *branch-and-bound*, seu bom desempenho se deve à poda de configurações com base em seus limites. Nas distribuições comumente utilizadas na literatura (onde os valores das coalizões são gerados a partir de uma distribuição uniforme no intervalo entre 0 e $|C|$), a discrepância entre os valores das estruturas de coalizão é maior, facilitando a poda de configurações que apresentem um limite baixo. Por outro lado, em distribuições mais realistas (como a utilizada neste trabalho), a discrepância entre os valores das estruturas de coalizão é menor, dificultando o estabelecimento de garantias nos limites das mesmas e minimizando a chance de podas.

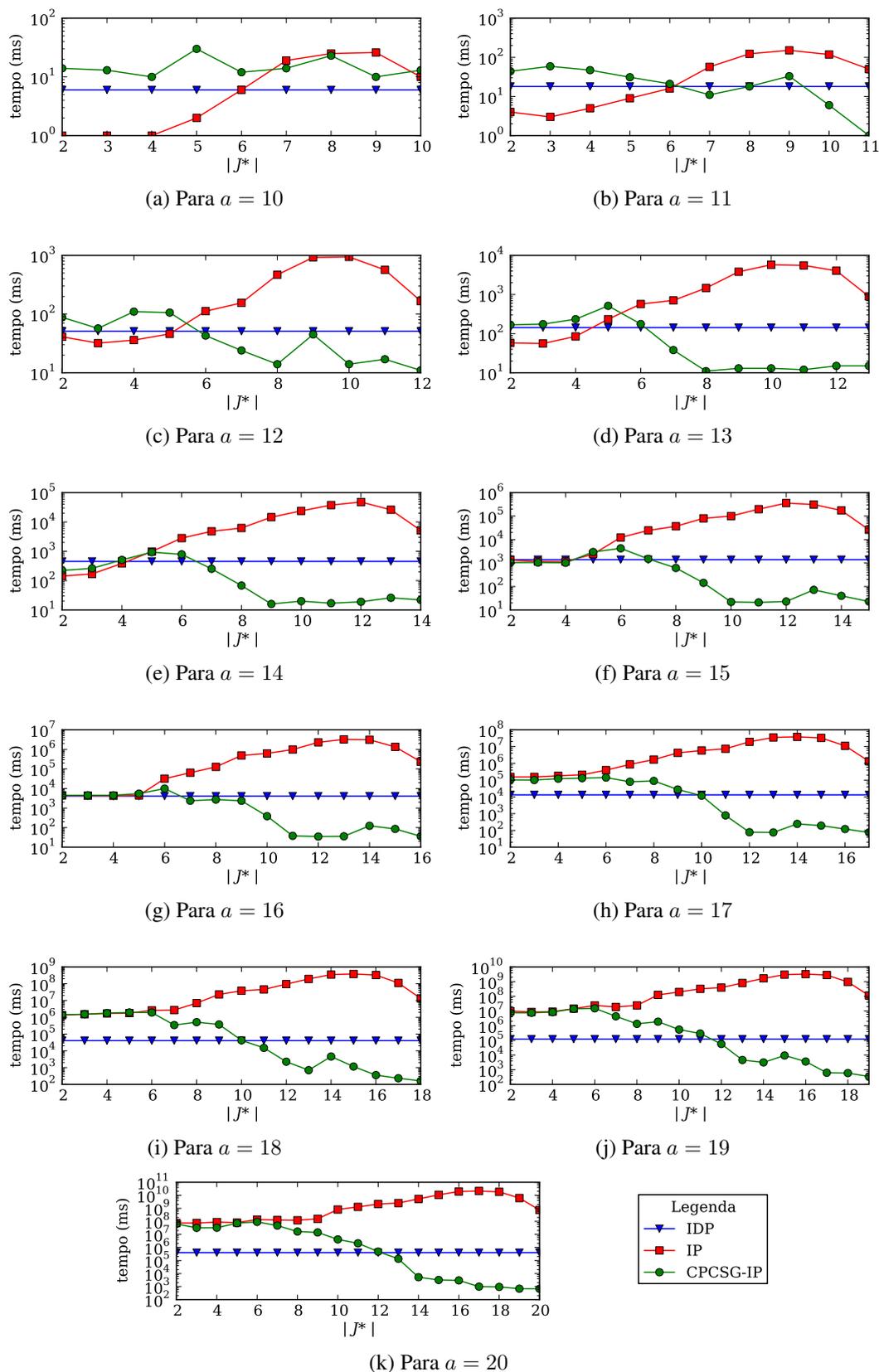


Figura 6.5: Comparação dos tempos de execução dos três algoritmos, IDP, IP e CPCSG-IP (IP com CPCSG), para 10 a 20 agentes, variando o tamanho de J^* de 2 até a

6.5 Discussão

Neste capítulo foi apresentada uma investigação do comportamento da abordagem proposta neste trabalho. Foram levantadas questões referentes ao custo envolvido no cálculo da poda e ao impacto da poda no desempenho do algoritmo IP. Além disso, foram realizadas comparações com outros algoritmos existentes. Como parte da metodologia adotada, foi definida uma função característica mais realista e apropriada para o cenário de *smart grids*.

Através dos experimentos constatou-se que o cálculo da poda é simples e consome uma porção ínfima de tempo. No que se refere à busca da estrutura de coalizão ótima, o algoritmo CPCSG desempenha um papel determinante na redução do tempo de execução do algoritmo IP. Mostrou-se ainda que as podas, por menores que sejam, são capazes de reduzir bastante o espaço de busca, além de reduzir consideravelmente o tempo de execução.

Experimentos demonstram, ainda, que a abordagem proposta tem um desempenho muito bom se comparado com os algoritmos IP e IDP, sobretudo para um número maior de agentes. Desta forma, a abordagem proposta se mostra superior ao IP (pois é mais rápido na maioria das vezes e percorre um espaço de busca menor) e ao IDP (ocupando menos memória, mostrando-se mais rápido em diversos cenários e sendo capaz de gerar soluções *anytime*) em diversos cenários.

Em contrapartida, a abordagem proposta possui algumas deficiências. Em primeiro lugar, o algoritmo CPCSG é capaz de podar apenas níveis inteiros do grafo de estruturas de coalizão. Se por um lado isto torna o algoritmo CPCSG simples e rápido, por outro o algoritmo não é capaz de eliminar todas as soluções inválidas. Outra questão importante é que o algoritmo CPCSG necessita de um certo número de restrições para apresentar um desempenho satisfatório. Em cenários onde a rede elétrica é pouco restritiva (parâmetro α com valor alto), por exemplo, o algoritmo tende a apresentar um desempenho não muito satisfatório.

7 CONCLUSÕES

Neste capítulo são apresentadas as principais contribuições e conclusões obtidas a partir da elaboração deste trabalho, bem como algumas sugestões de trabalhos futuros.

7.1 Contribuições

A principal contribuição deste trabalho para a área de sistemas multiagentes foi a introdução de um método novo e eficiente para a poda do espaço de busca de estruturas de coalizão com base em informações sobre o domínio de *smart grids*. Para desempenhar esta tarefa foi proposto o algoritmo CPCSG (definido na Seção 5.4), o qual é utilizado conjuntamente com o IP (apresentado na Seção 3.3.3.2), algoritmo *anytime* considerado o estado da arte para geração de estruturas de coalizão. Embora tenha sido projetado para trabalhar em conjunto com o IP no domínio de *smart grids*, o algoritmo CPCSG pode ser facilmente adaptado para outros domínios (contanto que haja uma restrição semelhante à da distância, adotada neste trabalho) e, inclusive, para trabalhar em conjunto com outros algoritmos de CSG (contanto que utilizem uma representação do espaço de busca baseada no grafo de estruturas de coalizão).

Outra contribuição importante refere-se à aplicação da técnica proposta no domínio de *smart grids*. O tema *smart grids* apresenta um grande potencial para o uso de sistemas multiagente, o que tem chamado atenção da comunidade científica. Embora o assunto *smart grids* esteja em alta, este é um dos primeiros trabalhos que se tem notícia a tratar o problema formação de coalizões em *smart grids* seguindo o formalismo de geração de estruturas de coalizão. A maioria dos trabalhos neste sentido aborda o processo de formação de coalizão de maneira *ad hoc*.

7.2 Conclusões

Este trabalho foi proposto com o objetivo de otimizar o processo de geração de estruturas de coalizão com base em informações sobre o domínio de *smart grids*. Na abordagem modelada, EVs que desejam participar de sessões V2G tentam ingressar em coalizões de modo a obter lucros maiores com a venda de energia. Considerando que este é um processo muito complexo, optou-se por utilizar informações sobre o domínio para estabelecer critérios que viabilizem a poda de porções inválidas do espaço de busca. Neste sentido foi proposto o algoritmo CPCSG, que identifica restrições conjuntas entre EVs, ou seja, agentes que não podem formar coalizões entre si devido à distância entre eles, para então realizar a poda de soluções inviáveis do espaço de busca. Após a realização da poda, é utilizado o algoritmo IP para buscar a solução ótima nos níveis restantes do grafo.

Resultados demonstram que o algoritmo CPCSG é rápido e eficiente, sendo capaz de calcular a quantidade de níveis a ser podada de forma quase instantânea. Adicionalmente, através dos experimentos realizados, constatou-se que quanto maior a poda realizada, maior o ganho de desempenho do algoritmo CPCSG. Comparado com outros algoritmos para CSG (IP e IDP), o método proposto tem um desempenho superior em diversos aspectos. Especificamente, em relação ao IP a abordagem proposta é mais rápida e percorre um espaço de busca menor. Já na comparação com o IDP, o método proposto consome menos memória e é *anytime*.

Portanto, o objetivo do trabalho foi alcançado conforme o esperado. Foi encontrada uma motivação consistente para a imposição de restrições na formação de coalizões. Foi criado um algoritmo de complexidade $O(a^4)$ capaz de calcular a quantidade de níveis que devem ser podados. O algoritmo proposto foi incorporado ao IP, podando o espaço de busca antes mesmo que a busca comece. Finalmente, os resultados obtidos foram satisfatórios.

Contudo, a abordagem proposta neste trabalho apresenta certas deficiências. Primeiramente, o algoritmo CPCSG é capaz de podar apenas níveis inteiros do grafo de estruturas de coalizão. Como consequência, muitas soluções inválidas podem permanecer nos níveis restantes, demandando mais tempo para encontrar a solução ótima. Adicionalmente, a abordagem proposta depende de um certo número de restrições entre os agentes para apresentar um desempenho satisfatório. Tal característica pode afetar o desempenho do algoritmo, sobretudo quando a rede elétrica for pouco restritiva.

7.3 Trabalhos futuros

A técnica desenvolvida neste trabalho é capaz de podar níveis inteiros do espaço de busca. Entretanto, o algoritmo CPCSG não é capaz de identificar e podar porções inválidas que se encontram dentro de níveis não podados. Desta forma, sugere-se como trabalho futuro uma remodelagem do algoritmo CPCSG, de modo a possibilitar a poda não apenas de níveis inteiros, mas também de porções específicas dentro deles. Uma possibilidade é basear-se na representação do espaço de busca proposto por Rahwan et al. (2007) para efetuar a poda de configurações específicas de níveis válidos, ao invés de podar apenas as configurações dos níveis inválidos. Outra possibilidade é a definição de maneiras alternativas de se representar o espaço de busca.

Outra abordagem que pode ser investigada é a proposta por Bazzan e Dahmen (2007), onde é criada uma heurística capaz de encontrar o nível exato onde se encontra a estrutura de coalizão ótima para uma classe específica de jogos de função característica. Neste trabalho, é abordado o conceito de jogadores de maior valor (MVPs, do inglês *more valuable players*), agentes que agregam um valor adicional à coalizão que eles ingressam. Esta modelagem mostra-se interessante para *smart grids*, uma vez que existem DERs de capacidade maior, como aerogeradores, que podem ser considerados como MVPs.

Conforme abordado no Capítulo 3, a formação de coalizões é uma tarefa que costuma ser abordada de maneira centralizada. Entretanto, esta não é uma prática muito condizente com a área de sistemas multiagentes. Neste sentido, propõe-se ainda o estudo de mecanismos descentralizados para formação de coalizões. Em um mecanismo desta natureza, os próprios agentes poderiam se coordenar para formar coalizões com seus vizinhos. Este tipo de mecanismo foge um pouco ao formalismo de geração de estruturas de coalizão, porém ainda poderiam ser investigados métodos capazes de estabelecer um limite de qualidade nas soluções encontradas.

REFERÊNCIAS

BAZZAN, A. L. C.; DAHMEN, S. R. Targeted search in coalition structures. In: *Proceedings of the 5th European Workshop on Multi-agent Systems*. Hammamet, Tunisia: [s.n.], 2007. p. 180–192. Disponível em: <www.inf.ufrgs.br/maslab/pergamus/pubs/BazzanDahmen2007eumas.pdf.gz>.

BORENSTEIN, S. *The trouble with electricity markets (and some solutions)*. [S.l.]: University of California Energy Institute, 2001.

CHALKIADAKIS, G.; ROBY, V.; KOTA, R.; ROGERS, A.; JENNINGS, N. R. Cooperatives of distributed energy resources for efficient virtual power plants. In: *Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS'11)*. Taipei, Taiwan: [s.n.], 2011. p. 787–794.

CORMEN, T.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. *Introduction To Algorithms*. [S.l.]: MIT Press, 2001. ISBN 9780262032933.

Electric Power Research Institute. *Report to NIST on the Smart Grid Interoperability Standards Roadmap*. [S.l.], Agosto 2009.

FALCÃO, D. M. Smart grids e microredes: o futuro já é presente. In: *Proceedings of VIII Simpósio de Automação de Sistemas Elétricos*. [S.l.: s.n.], 2009. v. 8, p. 1–11.

FERREIRA, H.; FULLI, G.; KLING, W. L.; L'ABBATE, A.; FAAS, H.; LOPES, J. P. Distributed generation in europe: the european regulatory framework and the evolution of the distribution grids towards smart grids. In: *Fifth IEEE Young Researchers Symposium in Electrical Power Engineering Proceedings*. [S.l.: s.n.], 2010. p. 1–6.

GERBER, A.; KLUSCH, M. Forming dynamic coalitions of rational agents by use of the dcf-s scheme. In: *The Second International Conference on Autonomous Agents and Multiagent Systems (AAMAS'03)*. New York, NY, USA: ACM, 2003. p. 994–995.

GERDING, E. H.; ROBU, V.; STEIN, S.; PARKES, D. C.; ROGERS, A.; JENNINGS, N. R. Online mechanism design for electric vehicle charging. In: *Proceedings of 10th International Conference on Autonomous Agents and Multiagent Systems*. Taipei, Taiwan: IFAAMAS, 2011. (AAMAS '11), p. 811–818.

HORLING, B.; LESSER, V. A survey of multi-agent organizational paradigms. *Knowledge Engineering Review*, v. 19, n. 4, p. 281–316, 2004.

IEEE. Ieee guide for smart grid interoperability of energy technology and information technology operation with the electric power system (eps), end-use applications, and loads. *IEEE Std 2030-2011*, p. 1–126, Outubro 2011.

INOVGRID. Inovgrid project - distribution network evolution as a decisive answer to new electrical sector challenges. In: *SmartGrids for Distribution, 2008. IET-CIRED. CIRED Seminar*. [S.l.: s.n.], 2008. p. 1–2.

KAMBOJ, S.; DECKER, K.; TRNKA, K.; PEARRE, N.; KERN, C.; KEMPTON, W. Exploring the formation of electric vehicle coalitions for vehicle-to-grid power regulation. In: *First International Workshop on Agent Technology for Energy Systems (ATES 2010)*. [S.l.: s.n.], 2010. p. 1–8.

KAMBOJ, S.; KEMPTON, W.; DECKER, K. S. Deploying power grid-integrated electric vehicles as a multi-agent system. In: *The Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS'11)*. Taipei, Taiwan: [s.n.], 2011. p. 13–20.

KEMPTON, W.; TOMIĆ, J. Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy. *Journal of Power Sources*, Elsevier, v. 144, n. 1, p. 280–294, 2005.

KLUSCH, M.; GERBER, A. Dynamic coalition formation among rational agents. *IEEE Intelligent Systems*, v. 17, n. 3, p. 42–47, May/June 2002.

LARSON, K.; SANDHOLM, T. Anytime coalition structure generation: An average case study. *Journal of Experimental & Theoretical Artificial Intelligence*, Taylor & Francis, v. 12, n. 1, p. 23–42, 2000.

MATTA, N.; RAHIM-AMOUD, R.; MERGHEM-BOULAHIA, L.; JRAD, A. A cooperative aggregation-based architecture for vehicle-to-grid communications. In: IEEE. *Global Information Infrastructure Symposium (GIIS), 2011*. [S.l.], 2011. p. 1–6.

MCARTHUR, S.; DAVIDSON E. M.; CATTERSON, V. M.; DIMEAS, A. L.; HATZARGYRIOU, N. D.; PONCI, F.; FUNABASHI, T. Multi-agent systems for power engineering applications – part i: Concepts, approaches, and technical challenges. *IEEE Transactions on Power Systems*, v. 22, n. 4, p. 1743–1752, Nov 2007.

MIHAILESCU, R.-C.; VASIRANI, M.; OSSOWSKI, S. Dynamic coalition adaptation for efficient agent-based virtual power plants. In: KLÜGL, F.; OSSOWSKI, S. (Ed.). *Multiagent System Technologies*. [S.l.]: Springer Berlin Heidelberg, 2011, (Lecture Notes in Computer Science, v. 6973). p. 101–112.

MIHAILESCU, R.-C.; VASIRANI, M.; OSSOWSKI, S. Dynamic coalition formation and adaptation for virtual power stations in smart grids. In: *Proceedings of the Second International Workshop on Agent Technology for Energy Systems (ATES 2011)*. [S.l.: s.n.], 2011. p. 1–4.

MIHAILESCU, R.-C.; VASIRANI, M.; OSSOWSKI, S. An organizational approach to agent-based virtual power stations via coalitional games. In: PÉREZ, J. et al. (Ed.). *Highlights in Practical Applications of Agents and Multiagent Systems*. [S.l.]: Springer Berlin Heidelberg, 2011, (Advances in Intelligent and Soft Computing, v. 89). p. 125–134.

Ministério de Minas e Energia. *Portaria número 440, de 15 de abril de 2010*. Abril 2010. Diário Oficial da União (DOU) de 16/04/2010.

Ministério de Minas e Energia. *Relatório do Grupo de Trabalho de Redes Elétricas Inteligentes*. [S.l.], 2010. Disponível em: <http://www.mme.gov.br/mme/galerias/arquivos/acoes/Energia/Relatxrio_GT_Smart_Grid_Portaria_440-2010.pdf>.

NAPHADE, M.; BANAVAR, G.; HARRISON, C.; PARASZCZAK, J.; MORRIS, J. Smarter cities and their innovation challenges. *Computer*, v. 44, n. 6, p. 32–39, 2011.

OSI. *Smart Grid Initiatives White Paper*. [S.l.], 2009. Disponível em: <<http://www.osii.com/pt/files/smartgridinitiativeswhitepaper.pdf>>.

PUDJIANTO, D.; RAMSAY, C.; STRBAC, G. Virtual power plant and system integration of distributed energy resources. *Renewable Power Generation, IET*, v. 1, n. 1, p. 10–16, March 2007.

RAHWAN, T. *Algorithms for Coalition Formation in Multi-Agent Systems*. Tese (Doutorado) — University of Southampton, August 2007.

RAHWAN, T.; JENNINGS, N. R. Coalition structure generation: Dynamic programming meets anytime optimisation. In: *Proceedings of the 23rd Conference on Artificial Intelligence (AAAI 2008)*. Estoril, Portugal: [s.n.], 2008. p. 156–161.

RAHWAN, T.; JENNINGS, N. R. An improved dynamic programming algorithm for coalition structure generation. In: *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS'08)*. Estoril, Portugal: [s.n.], 2008. p. 1417–1420.

RAHWAN, T.; RAMCHURN, S. D.; DANG, V. D.; JENNINGS, N. R. Near-optimal anytime coalition structure generation. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*. [S.l.: s.n.], 2007. p. 2365–2371.

RAMCHURN, S.; VYTELINGUM, P.; ROGERS, A.; JENNINGS, N. R. Putting the “smarts” into the smart grid: A grand challenge for artificial intelligence. *Communications of the ACM*, v. 55, n. 4, p. 86–97, 2012.

RAMOS, G. de O.; BAZZAN, A. L. C. Reducing the search space of coalition structures based on smart grids' domain information. In: BALSÁ, J.; FUENTES-FERNÁNDEZ, R. (Ed.). *Proceedings of 8th Ibero-American Workshop on Multi-Agent Systems (IBERAGENTS 2012)*. Cartagena de Indias, Colombia: [s.n.], 2012. p. 1–11. Disponível em: <<http://www.inf.ufrgs.br/maslab/pergamus/pubs/Ramos&Bazzan2012iberagents.pdf>>.

RAMOS, G. de O.; BAZZAN, A. L. C. Reduction of coalition structures' search space based on domain information: an application in smart grids. In: *2012 Third Brazilian Workshop on Social Simulation (BWSS)*. Curitiba, Brasil: [s.n.], 2012. p. 112–119. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6462824&isnumber=6462694>>.

SANDHOLM, T.; LARSON, K.; ANDERSSON, M.; SHEHORY, O.; TOHMÉ, F. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, v. 111, n. 1–2, p. 209–238, 1999.

SCHMEIDLER, D. The nucleolus of a characteristic function game. *SIAM Journal on applied mathematics*, v. 17, n. 6, p. 1163–1170, 1969.

SHEHORY, O.; KRAUS, S. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, v. 101, n. 1–2, p. 165–200, 1998.

THRALL, R. M.; LUCAS, W. F. N-person games in partition function form. *Naval Research Logistics Quarterly*, v. 10, n. 1, p. 281–298, 1963.

U. S. Department of Energy. *Grid 2030: A National Vision for Electricity's Second 100 Years*. July 2003.

VASIRANI, M.; KOTA, R.; CAVALCANTE, R.; OSSOWSKI, S.; JENNINGS, N. R. Using coalitions of wind generators and electric vehicles for effective energy market participation. In: *The Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS'11)*. [S.l.: s.n.], 2011. p. 1099–1100.

YEH, D. Y. A dynamic programming approach to the complete set partitioning problem. *BIT Numerical Mathematics*, Springer, v. 26, n. 4, p. 467–474, 1986.