

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

DANIEL EPSTEIN

**Um algoritmo distribuído para resolução do
problema de geração de estruturas de
coalizão com presença de externalidades**

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

Prof^ª. Dr^ª. Ana L. C. Bazzan
Orientadora

Porto Alegre, Maio de 2013

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Epstein, Daniel

Um algoritmo distribuído para resolução do problema de geração de estruturas de coalizão com presença de externalidades / Daniel Epstein. – Porto Alegre: PPGC da UFRGS, 2013.

75 f.: il.

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

1. Inteligência artificial. 2. Sistemas multiagente. 3. Teoria dos jogos. 4. Algoritmos distribuídos. 5. Estruturas de coalizão. 6. Externalidade. I. Bazzan, Ana L. C.. 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. Luigi Carro

Bibliotecário-chefe do Instituto de Informática: Alexander Borges Ribeiro

*"Quem vive estudando, mas nunca repete o que aprendeu
se parece com quem vive semeando, mas nunca ceifa."*

— TALMUD

AGRADECIMENTOS

Agradeço à minha família e a minha namorada pelo apoio incondicional dado a mim por todos esses anos. Agradeço também à ajuda valorosa dos meus colegas de mestrado, em especial ao Anderson Tavares, Cristiano Galafassi, Gabriel Ramos e Mariana Mendoza pela ajuda e amizade durante esses anos. Finalmente, agradeço à professora Ana Bazzan, minha orientadora e quem, sem dúvida, mais contribuiu para meu amadurecimento como cientista e profissional. Tenho certeza que tudo que aprendi durante esses anos servirão de base para aquilo que ainda tenho por aprender.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE SÍMBOLOS	8
LISTA DE TABELAS	9
RESUMO	10
ABSTRACT	12
1 INTRODUÇÃO	14
2 CONCEITOS RELACIONADOS	18
2.1 Teoria dos jogos	18
2.2 Jogos de função característica	19
2.3 Jogos de partição	20
3 TRABALHOS RELACIONADOS À GERAÇÃO DE ESTRUTURAS DE COALIZÃO DE AGENTES	24
3.1 Algoritmo distribuído sem garantia de desempenho	26
3.2 Garantia de limite ótimo	27
3.3 Melhoria do limite ótimo	30
3.4 Uma nova representação	31
3.5 Algoritmo IP	34
3.6 O primeiro algoritmo distribuído com garantia de desempenho	35
3.7 Algoritmos para problemas com presença de externalidade (IP^+)	38
3.8 Discussão e revisão dos algoritmos existentes	41
4 ABORDAGEM	43
4.1 Motivação	43
4.2 Calculando o valor de estruturas de coalizão em ambientes descentralizados com presença de externalidades	45
4.2.1 Primeira etapa: cálculo distribuído dos valores das coalizões	45
4.2.2 Segunda etapa: busca pela CS^* dentro das configurações	46
4.3 Análise do Algoritmo	50
4.3.1 Garantia de desempenho	50
4.3.2 Comunicação	51
4.4 Principais contribuições do algoritmo	52

5	EXPERIMENTOS E RESULTADOS	53
5.1	Metodologia	53
5.2	Como a formação de uma coalizão influencia nas demais?	54
5.3	Experimentos	54
5.4	Resultados	55
5.4.1	Experimentos em condições ideais	55
5.4.2	Experimentos com falhas de comunicação	60
6	CONCLUSÕES	64
6.1	Conclusões	64
6.2	Trabalhos Futuros	65
	REFERÊNCIAS	66
	APÊNDICEA	68
A.1	Resultados completos considerando $\beta = 1$	68
A.1.1	Porcentagem de configurações removidas	68
A.1.2	Número de estruturas de coalizão avaliadas em relação ao número total de estruturas de coalizão do problema	70
A.2	Resultados completos considerando $\beta = 0.98$	72
A.2.1	Porcentagem de configurações removidas	72
A.2.2	Número de estruturas de coalizão avaliadas em relação ao número total de estruturas de coalizão do problema	74

LISTA DE ABREVIATURAS E SIGLAS

CSG	Geração da Estrutura de Coalizão
PFG	Jogos de Partição
CFG	Jogos em Função Característica
SMA	Sistema(s) Multiagente

LISTA DE SÍMBOLOS

A	conjunto de agentes
S	Subconjunto de agentes de A
$ A , n$	número de agentes
a_i	i -ésimo agente em A
CS	uma estrutura de coalizão
CS^*	estrutura de coalizão ótima
S^A	conjunto de todas as estruturas de coalizão
C_i	i -ésima coalizão da estrutura de coalizão CS
k	número máximo de agentes em qualquer coalizão
$v(C)$	valor da coalizão C
$V(CS)$	valor da estrutura de coalizão CS
β	relação limite entre a melhor CS encontrada e a CS^*

LISTA DE TABELAS

Tabela 1.1:	Comparação entre o número de coalizões e o número de estruturas de coalizões versus número de agentes.	15
Tabela 3.1:	Redução gradual do limite da solução ótima garantida pelo Teorema 2.	29

RESUMO

Uma importante parte de um sistema multiagente é o seu mecanismo de coordenação que permite que os agentes possam agir de maneira coesa em direção aos seus objetivos, sejam eles individuais ou coletivos. Um agente pode optar por cooperar para atingir um determinado objetivo que seria inalcançável através de ações individuais, para realizar uma tarefa de maneira mais eficiente ou simplesmente porque ele foi projetado para tal. Em todos os casos, a formação de coalizões (grupos de agentes que concordam em coordenar suas ações em torno de um objetivo comum) é uma questão fundamental.

O problema de geração de estruturas de coalizão entre agentes (conjunto de todas as combinações de coalizões) é um tópico de pesquisa que recebeu muita atenção principalmente na resolução do problema quando considerado como um jogo de função característica, onde o valor das coalizões independe dos agentes que não estão presentes nela. Essa abordagem, apesar de ser indicada para muitos tipos de problema, não cobre toda a área de pesquisa do assunto, visto que em muitos casos a criação de uma coalizão irá afetar os demais agentes do sistema. Quando o sistema possui agentes com objetivos sobrepostos ou contrários, uma coalizão cujos recursos são destinados a completar tais objetivos irá influenciar as demais coalizões desse sistema. Essa influência se chama externalidade e, nesses casos, o problema de formação de estruturas de coalizão deve ser tratado como um jogo de partição. Apesar das pesquisas na área de jogos de partição serem recentes, elas trazem resultados promissores e há alguns poucos algoritmos já desenvolvidos para buscar soluções a esse problema.

A busca pela melhor estrutura de coalizão geralmente demanda que seja calculado o valor de todas possíveis coalizões, a fim de se encontrar aquele conjunto cuja soma dos valores das coalizões forneça o melhor resultado. Esse processo requer um alto número de computações e de memória, devido à natureza exponencial do problema. Assim, ao invés de apenas um agente central realizar todas as operações, é mais eficiente do ponto de vista do uso de recursos computacionais distribuir essas operações entre os diversos agentes presentes no sistema. Além dos benefícios computacionais, distribuir o processo de busca pela melhor estrutura de coalizão permitiria trabalhar com questões como privacidade e tolerância a falhas, tendo em vista que as informações não estão concentradas em um único agente. Apesar disso, não há na literatura qualquer algoritmo capaz de solucionar o problema de geração de estrutura de coalizão em ambientes distribuídos e que sejam modelados como jogos de partição.

A proposta desse trabalho é utilizar a fundamentação teórica existente acerca do problema de formação de estruturas de coalizão (modelados tanto como jogos de função característica quanto como jogos de partição) para criar um algoritmo distribuído capaz de encontrar a estrutura de coalizão ótima em ambientes que possuam externalidade. Esse algoritmo utiliza como base a ordenação das coalizões e dos agentes para permitir a dis-

tribuição do cálculo dos limites superiores e inferiores de cada coalizão. Após, esses valores são utilizados para se encontrar o subespaço mais provável de conter a estrutura de coalizão ótima.

Com base nos experimentos, percebe-se que o algoritmo encontrou a estrutura de coalizão ótima buscando em apenas uma pequena parte do espaço de busca. Para os experimentos com 16 agentes, o algoritmo foi capaz de encontrar a solução ótima procurando em apenas 0,01% do espaço de busca. Também, é demonstrado que em cenários com externalidade negativa os agentes necessitaram investigar um espaço de busca menor para encontrar a estrutura de coalizão ótima que em cenários com externalidade positiva. Experimentos também demonstram que o algoritmo não consegue encontrar a estrutura de coalizão ótima quando há falhas na comunicação entre os agentes.

Palavras-chave: Inteligência artificial, sistemas multiagente, teoria dos jogos, algoritmos distribuídos, estruturas de coalizão, externalidade.

A distributed algorithm for solving coalition structure generation problem with externalities

ABSTRACT

An important part of a multi-agent system is its coordination mechanism that allows the agents to act cohesively towards their goals, whether individual or collective. An agent can choose to cooperate to achieve a certain goal that would be unattainable through individual actions, to perform a task more efficiently or simply because it was designed to do so. In all cases, the formation of coalitions (group of agents that agree to coordinate their actions around a common goal) is a key issue.

The problem of generating coalition structures between agents (set of all combinations of coalitions) is a research topic that has received much attention mostly on solving the problem when considered as a characteristic function game, where the value of coalitions is independent of agents that are not part of it. This approach, although suitable for many types of problem, does not cover the whole area of research on the subject, since in many cases the creation of a coalition will affect the other agents of the system. When the system has agents with overlapping goals or opposing goals, a coalition whose resources are devoted to completing these objectives will influence the other coalitions of that system. This influence is called externality, and in these cases, the problem of formation of coalition structures should be treated as a partition function game. Although research in the area of partition games is recent, it brings promising results and there are few algorithms already developed to find solutions to this problem.

The search for the best coalition structure generally requires computation of the value of all possible coalitions in order to find the set that the sum of the values of the coalitions provides the best result. This process requires a large number of computations and memory due to the exponential nature of the problem. Hence, instead of just one central agent performing all operations, it is more efficient to distribute those operations among several agents. Besides the computational benefits, distributing the search process for the best coalition structure would address issues such as privacy and fault tolerance, given that the information is not concentrated in a single agent. Nevertheless, in the literature there is not algorithm capable of solving the problem of coalition structure generation in decentralized environments and modeled as partition function game.

The purpose of this work is to use the existing theoretical foundations for solving the coalition structure generation problem (modeled both as a characteristic function game and as a partition function game) to create a distributed algorithm capable of finding the optimal coalition structure in environments that have externality. This algorithm uses as a base the ordering of coalitions and agents to distribute the calculation of the upper and lower limits for each coalition. Afterwards, these values are used to find the subspace more likely to contain the optimal coalition structure.

Based on experiments, the algorithm found the optimal coalition structure searching only a small part of the search space. For the experiments with 16 agents, the algorithm

was able to find the solution looking at just 0.0001 % of the search space. Also, it is shown that in scenarios with negative externality agents need to investigate a smaller search space to find the optimal coalition structure than in scenarios with positive externality. Experiments also show that the algorithm can not find the optimal coalition structure when there are failures in the communication among the agents.

Keywords: artificial intelligence, multi-agent system, game theory, distributed algorithms, coalition structure generation, externality.

1 INTRODUÇÃO

O processo pelo qual um grupo de agentes se une ou concorda em coordenar suas ações e cooperar entre si para realizar um conjunto de tarefas é denominado "formação de coalizão". Esta forma de interação tem particular importância em sistemas multiagente, pois estas coalizões podem melhorar o desempenho individual dos agentes e do sistema como um todo. Quando uma determinada tarefa necessita de mais recursos do que aqueles que um agente dispõe, a união desse agente com outros torna a realização da tarefa mais eficiente ou mesmo possível. (HORLING; LESSER, 2005) afirmam que coalizões diferem de outros modelos de organização por serem apenas necessárias enquanto há um objetivo em comum para ser realizado. Coalizões são formadas tendo como base fundamental um objetivo comum entre os participantes da coalizão em questão e são dissolvidas quando o objetivo já não mais existe ou é impossível de ser realizado. Além disso, a estrutura da coalizão é plana, ou seja, não há uma hierarquia definida dentro da coalizão. Não existem líderes ou agentes que possam decidir as ações dos demais quando há um impasse.

Considerando a existência de um conjunto de agentes A em um sistema, todo o subconjunto de agentes de A é uma possível coalizão. O número total de coalizões possível é $2^{|A|} - 1$. Cada coalizão c formada possui um valor, chamado de valor da coalizão ($v(c)$), que é utilizado para se decidir quais coalizões devem ser formadas. Esse valor pode ser alterado pela presença de outras coalizões, podendo ser esta alteração positiva ou negativa. Em todos os casos, calcular o valor de cada coalizão é uma tarefa cuja complexidade é exponencial, o que pode impossibilitar que seja realizada. Entre as abordagens propostas para resolver esse problema estão a de buscar apenas um subconjunto das coalizões possíveis e a de propor algoritmos distribuídos, cujo fardo de calcular o valor das coalizões e armazenar seus resultados possa ser dividido entre diversos agentes.

Alguns aspectos são desejáveis ao se propor algoritmos distribuídos para a solução do problema (RAHWAN; JENNINGS, 2007):

- O processo de distribuição deve ser descentralizado, ou seja, nenhum agente deve ser responsável por distribuir as coalizões aos demais agentes nem por centralizar seus valores. Caso isso aconteça, esse agente seria diretamente responsável pelo desempenho do algoritmo e haveria a possibilidade do algoritmo falhar caso aconteça algum problema com o agente.
- A comunicação entre os agentes deve ser minimizada.
- A computação redundante dos valores de coalizão e quaisquer outros cálculos efetuados pelos agentes devem ser minimizados.
- O problema deve ser balanceado de tal forma que os agentes recebam uma quantidade de subproblemas para resolver proporcional aos seus recursos. Ou seja, consi-

derando que todos os agentes tem a mesma capacidade, o balanceamento de carga deve prover uma quantidade similar de problemas para cada agente resolver.

- A quantidade de memória requerida para resolver o problema deve ser minimizada.

De maneira geral, a formação de coalizões em sistemas multiagente pode ser vista como um processo composto por três atividades entrelaçadas, sendo elas:

- A geração da estrutura da coalizão, que estuda como dividir os agentes em conjuntos ou coalizões de maneira a maximizar a utilidade do sistema multiagente;
- O cálculo do valor de cada coalizão;
- A divisão do ganho das coalizões entre seus membros.

As duas primeiras atividades têm sido mais amplamente pesquisadas dentro da área de ciência da computação e SMA do que a terceira (SANDHOLM et al., 1999). A parte correspondente à geração de estruturas de coalizões é particularmente complexa e tem sido largamente estudada nos últimos anos, já que esta parte representa o maior gargalo no processo como um todo (RAHWAN et al., 2007b). Em particular, (SANDHOLM et al., 1999) provaram que o problema de encontrar a estrutura de coalizão que maximiza a utilidade do sistema (denominada de estrutura de coalizão ótima, ou CS^*) é um problema *NP - completo*.

Estruturas de coalizão (CS) são divisões dos agentes em coalizões exaustivas e disjuntas. O problema de gerar estruturas de coalizão tem complexidade muito maior que o problema de dividir agentes em coalizões. Mais precisamente, enquanto o problema de dividir os agentes em coalizões tem complexidade de $O(2^{|A|})$, o problema de gerar todas as estruturas de coalizão possíveis tem complexidade de $O(|A|^{|A|})$. Pela Tabela 1.1 podemos observar que a diferença entre o número de coalizões e de estruturas de coalizões cresce de forma exponencial com o aumento do número de agentes.

Tabela 1.1: Comparação entre o número de coalizões e o número de estruturas de coalizões versus número de agentes.

$ A $	$ C $	$ CS $
1	1	1
3	7	5
5	31	52
7	127	877
9	511	21.147
11	2047	678.570
13	8195	27.644.437
15	32783	1.382.958.545

A definição formal de estrutura de coalizão é a que segue:

Definição 1. Uma estrutura de coalizão $CS = \{C_1, C_2, \dots, C_{|CS|}\}$ é uma partição exaustiva de $|A|$ agentes em coalizões disjuntas.

A teoria dos jogos é uma teoria matemática que descreve a interação entre agentes racionais, conhecidos como jogadores. É muito explorada no campo da economia, pois modela de forma precisa a interação entre entidades racionais, tentando estabelecer o que seria a melhor ação para cada jogador em um jogo. Dentro da teoria dos jogos há o estudo da teoria dos jogos de coalizão, denominado de teoria dos jogos cooperativos. A literatura desta área descreve ainda quais coalizões serão formadas dadas diferentes situações e como dividir o ganho do trabalho cooperativo para garantir certos tipos de estabilidade, onde estabilidade significa prover pouco ou nenhum incentivo para os agentes abandonarem a coalizão a qual pertencem em busca de ganhos superiores.

Até recentemente, a grande maioria das pesquisas voltadas ao assunto de geração de estruturas de coalizão por parte da ciência da computação tratavam o problema de encontrar a CS^* como um *jogo de função característica* (CFG, do inglês *characteristic function game*). CFG é um dos modelos mais simples para representar jogos de coalizão ou jogos cooperativos, onde o valor de uma coalizão não é afetado pelos membros que não participam da coalizão. Porém, há muitos contextos onde esta hipótese não é verdadeira. Tais configurações são modeladas como *Jogos de Partição* (PFG, do inglês *partition function games*), onde o valor de uma coalizão pode ser influenciado pela formação de outras coalizões no sistema. A influência que a formação de uma coalizão apresenta sobre as demais é chamada de *externalidade*. Jogos com externalidade foram bastante estudados principalmente pela área de economia. Exemplo dessas pesquisas podem ser encontradas nas áreas de conluio entre empresas (CATILINA; FEINBERG, 2006), redução da emissão de carbono (FINUS; RUNDSHAGEN, 2003), coordenação de políticas entre países, entre outros.

Uma das principais razões para não haver uma extensa pesquisa da área de ciência da computação acerca de PFG é devido ao tamanho da entrada do problema crescer de $O(2^{|A|})$ para $O(|A|^{|A|})$. Ou seja, o tamanho da entrada do problema se torna equivalente a ter de enumerar todas as possíveis estruturas de coalizão. Uma vez que o valor de cada coalizão é influenciado pela formação de outras coalizões, não há um valor único e específico para uma determinada coalizão, sendo necessário enumerar todas as CS possíveis para se avaliar o valor de cada coalizão. Apenas em 2008 um artigo de (MICHALAK et al., 2008) abordou o problema de encontrar a CS^* para PFG. Em (RAHWAN et al., 2012) um estudo computacional foi proposto para jogos com externalidades em um sistema multiagente. Também foi apresentado um algoritmo capaz de encontrar a CS^* e os parâmetros computacionais necessários para realização de simulações nesse cenário. Porém, todos os estudos computacionais e algoritmos existentes para encontrar a estrutura de coalizão ótima em PFG são centralizados, não havendo nem pesquisas publicadas nem algoritmos para tratar o problema de geração de estrutura de coalizão de forma distribuída nesses cenários.

Este trabalho tem por objetivo desenvolver um algoritmo distribuído capaz de encontrar a solução para o problema de formação de estruturas de coalizão em PFG. Para isso, é necessário levar em consideração as particularidades de ambientes modelados como PFG e as vantagens que se deseja obter de sistemas descentralizados. O algoritmo proposto, chamado de *DCSGE* (Distributed Coalition Structure Generation with Externality), distribui o cálculo dos valores das coalizões entre os agentes e propõe um modelo de comunicação que permite que apenas as coalizões consideradas mais importantes sejam transmitidas. Após, cada agente se torna responsável por procurar em um subespaço do problema a estrutura de coalizão ótima. O DCSGE é capaz de encontrar soluções em problemas abordados tanto como CFG quanto PFG, além de ser possível utilizá-lo tanto em ambientes centralizados quanto descentralizados.

O restante do texto está organizado como segue:

- **Capítulo 2:** Apresenta uma visão geral sobre teoria dos jogos e informações relevantes acerca de CFG e PFG.
- **Capítulo 3:** Apresenta a fundamentação teórica acerca de formação de estruturas de coalizão. Apresenta também diversos trabalhos da área, bem como os algoritmos mais eficientes e suas principais características.
- **Capítulo 4:** Descreve o *DCSGE* e sua área de aplicação.
- **Capítulo 5:** Apresenta uma série de experimentos realizados com o algoritmo, bem como seus resultados e análises.
- **Capítulo 6:** Apresenta uma discussão sobre o assunto tratado nessa dissertação, o algoritmo proposto e sugestões de trabalhos futuros.

2 CONCEITOS RELACIONADOS

Este capítulo contém os conceitos preliminares importantes para a compreensão do restante do texto. Muitas das ideias acerca de coalizões de agentes e noções básicas sobre jogos de coalizão serão descritas nessa seção a fim de permitir uma completa compreensão do restante do texto.

Este capítulo está dividido em quatro seções. A Seção 2.1 descreve os conceitos básicos de teoria dos jogos; a Seção 2.2 foca em jogos de função característica; por fim, a Seção 2.3 descreve os jogos de partição.

2.1 Teoria dos jogos

A teoria dos jogos diz respeito à matemática da interação entre entidades racionais, fornecendo ferramentas para podermos analisar essa interação do ponto de vista matemático. As entidades envolvidas nessas interações são denominadas jogadores ou agentes e é assumido que esses agentes são capazes de escolher (dentro de um conjunto de possibilidades) a ação que garanta a eles o maior ganho, sendo este ganho definido por uma função de utilidade. Para cada conjunto de ações ou movimentos, é associado um vetor de utilidade que define o ganho correspondente a cada agente naquele momento.

A teoria dos jogos é relacionada à teoria da decisão, que tenta encontrar a melhor decisão que um agente pode tomar. No caso da teoria dos jogos, a questão que se procura responder é qual ação um agente deve tomar considerando que suas ações sofrem efeito das decisões de outros agentes.

A aplicação inicial para a teoria dos jogos foi a análise de decisão humana. Essa abordagem é alvo de muitas críticas e possui diversas limitações, uma vez que as ações não são sempre racionais ou a preferência entre alternativas não pode ser expressa de maneira consistente entre os participantes. A teoria dos jogos é formulada baseada na premissa de que os jogadores são puramente racionais e sempre tentam maximizar seus ganhos. Variáveis emocionais ou mesmo preferências individuais não são levadas em consideração. Apesar disso, há exemplos de sucesso evidente da teoria dos jogos como, por exemplo, nas áreas da economia, teoria política e psicologia.

Podemos dividir o estudo dos jogos em três principais modelos: jogos em forma estratégica (ou normal), jogos em forma extensiva e jogos em forma de coalizão. A diferença entre as três formas está na quantidade de informação que há no modelo em relação à interação entre os agentes.

Na forma estratégica de um jogo, cada jogador escolhe uma estratégia própria baseada em um conjunto de estratégias possíveis e adiciona a estratégia em seu conjunto de estratégias, ou espaço de ações. Este modelo assume que todas as escolhas são realizadas simultaneamente e, apenas quando todos os jogadores tiverem escolhido suas ações, estas

serão reveladas e então será associado um ganho para cada agente. O cenário de interação descrito acima pode ser formalizado com o uso de uma notação matriz de ganho ou matriz de utilidade. O jogo em forma extensa assume que as decisões de ação são tomadas de maneira alternada entre os agentes. Isto adiciona as noções de posição e movimento. A representação usual deste modelo de jogo é através de uma árvore denominada de árvore de jogo. O jogo em forma de coalizão é abordado dentro da área de pesquisa de jogos cooperativos da teoria dos jogos e será mais amplamente abordado ao longo do restante do texto.

2.2 Jogos de função característica

Um dos modelos mais simples para representar jogos de coalizão ou jogos cooperativos é o formalismo dos *jogos de função característica* (CFG). Neste formalismo cada coalizão C possui um valor $v(C)$ que depende exclusivamente de seus membros. Por não haver influência externa no valor da coalizão, podemos analisar apenas o resultado da coalizão em questão, sem nos preocuparmos com as demais partições ou coalizões. Existem outros modelos que consideram que o valor da coalizão está sujeito a fatores externos a coalizão (chamados de "*externalidades*"), que serão discutidos na Seção 2.3. A seguir são apresentadas algumas definições necessárias para o formalismo de CFG.

Definição 2. Seja $A = \{a_1, a_2, \dots, a_n\}$ o conjunto dos agentes no jogo. Uma *coalizão* é um subconjunto de A , ($S \subseteq A$) e o conjunto de todas as coalizões é denotado por 2^A .

Definição 3. A coalizão correspondente ao conjunto de todos os agentes A é denominada de *grande coalizão*.

Definição 4. Um *jogo de coalizão* em forma característica é um par (A, v) , onde

- A é o conjunto de agentes; e
- $v : 2^A \mapsto \mathfrak{R}$ é uma função que mapeia cada grupo de agentes $S \subseteq A$ para um valor real de utilidade. Esta função é denominada *função característica* do jogo.

Em algumas situações é possível assumir que a união de duas coalizões permite que esta tenha um valor no mínimo igual a soma dos valores destas duas coalizões iniciais. Neste caso, o jogo é denominado de *super-aditivo*.

Definição 5. Um jogo em forma característica é *super-aditivo* se $v(S \cup T) \geq v(S) + v(T)$, para quaisquer duas coalizões disjuntas $S, T \subseteq A$ ($S \cap T = \emptyset$).

Também existe a classe dos jogos *sub-aditivos*. Neste caso, o valor da coalizão resultante da união de duas coalizões é igual ou inferior a soma dos valores das coalizões individuais.

Definição 6. Um jogo em forma característica é *sub-aditivo* se $v(S \cup T) \leq v(S) + v(T)$, para quaisquer duas coalizões disjuntas $S, T \subseteq A$ ($S \cap T = \emptyset$).

A solução de um jogo de coalizão é definida como um vetor de utilidade, ou de ganho, que define a quantidade que deve ser alocada para cada agente da coalizão.

Definição 7. O vetor de solução de um jogo é $\vec{x} = (x_1, x_2, \dots, x_{|A|}) \in \mathfrak{R}^{|A|}$. Ainda, a utilidade do agente i na solução \vec{x} é denotada por x_i .

Muitas vezes estamos interessados apenas em soluções "válidas", ou seja, soluções que não distribuem um \vec{x} maior do que o vetor de ganho é capaz de produzir (por exemplo, se o vetor tiver um valor Y , a soma de \vec{x} não pode ser maior que Y).

Definição 8. Um vetor de ganho \vec{x} é válida se existe um conjunto de coalizões $S = \{c_1, \dots, c_k\}$ onde:

$$\bigcup_{c \in S} v(c) \geq \sum_{i \in A} \vec{x}_i. \quad (2.1)$$

Um vetor \vec{x} é válido se podemos encontrar um conjunto de coalizões cujos valores são tanto quanto em \vec{x} , de maneira que se pode pagar os agentes de acordo com \vec{x} . Este conjunto de coalizões é muitas vezes referido como estrutura de coalizão.

Para avaliar se uma determinada coalizão é melhor que alguma outra qualquer, podemos utilizar diversos métodos que nos permitem ordenar as coalizões em uma ordem de preferência. Muitas vezes o método utilizado é a utilidade do sistema, calculado pela soma das utilidades recebidas pelos agentes. Essa soma nos fornece um valor global. Utilizamos esse critério de avaliação quando estamos interessados em avaliar o sistema como um todo e não o valor obtido por agentes específicos.

2.3 Jogos de partição

Muitos são os cenários onde é possível (e até mesmo indicado) modelar o problema de formação de estruturas de coalizão como um *CFG*. Este é o caso de sistemas onde não há interação entre coalizões ou mesmo onde as interações são pequenas o suficiente para serem desconsideradas. Porém, há um número considerável de sistemas onde há interação entre as diferentes coalizões, ou seja, a formação de uma determinada coalizão influencia na atuação de uma ou mais coalizões. Nesses casos, é dito que há efeitos externos referentes à formação de uma estrutura de coalizão (chamado de externalidades, do inglês *externalities*). Esses cenários, onde há uma interdependência entre as coalizões, são comumente encontrados em aplicação reais onde os agentes têm interesses conflitantes ou sobrepostos. Exemplo desses sistemas é a divisão do uso de recursos computacionais ou de uma transmissão de rede, onde alocar recursos para uma coalizão significa retirar recursos de outra. Nestes casos, o uso de *CFG* não traz uma representação real do sistema, sendo inadequado para simular essas situações ou sistemas.

Para representar sistemas onde há externalidades é necessário utilizar outra representação: jogos de partição (*PFG*). *PFG* (no âmbito de formação de estruturas de coalizão) consiste de um grupo de agentes A , todas as estruturas de coalizão possíveis e do valor de cada coalizão em cada possível estrutura de coalizão. Ou seja, não é mais possível saber o valor de uma estrutura de coalizão antes de computá-la. Isso representa um enorme aumento do espaço de busca pela estrutura de coalizão ótima, uma vez que é necessário expandir todas as estruturas de coalizão possíveis para avaliar cada valor de coalizão dentro delas.

Em *PFG*, o valor de uma coalizão pode ser influenciado pela formação de outras coalizões dentro do sistema. Por exemplo, dadas duas estruturas de coalizão $CS' = \{C_1, C_2, C_3\}$ e $CS'' = \{C_1, C_2 \cup C_3\}$, o valor de C_1 pode ser diferente nas estruturas de coalizão CS' e CS'' , dado que a junção das coalizões C_2 e C_3 pode ter influência sobre o valor de C_1 . Essa influência é chamada de *externalidade produzida pela formação de uma coalizão* e, nesse caso, a externalidade é induzida sobre a coalizão C_1 pela união das outras coalizões. A externalidade produzida pela formação de uma coalizão pode ser definida como:

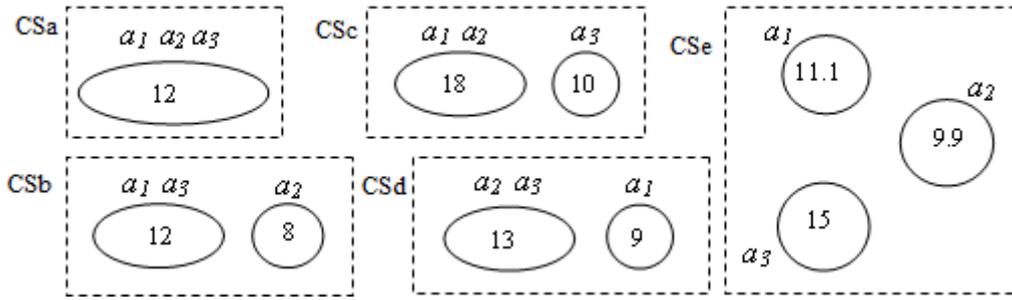


Figura 2.1: Exemplo de um PFG para $A = \{a_1, a_2, a_3\}$.

Definição 9. Sejam CS e CS' estruturas de coalizão tal que $|CS'| \geq 3$ e existem duas coalizões $C', C'' \in CS'$ tal que $CS = \{\{C' \cup C''\} \cup CS' \setminus \{C', C''\}\}$. Então, a formação de $(C' \cup C'', CS)$ proveniente de (C', CS') e (C'', CS') produz uma externalidade em toda outra coalizão (C, CS) tal que $C \neq C' \cup C''$, que pode ser definida como:

$$\epsilon(C, CS, CS') = \omega(C, CS) - \omega(C, CS'). \quad (2.2)$$

$\omega(C, CS)$ define o valor da coalizão C quando dentro da estrutura de coalizão CS . Assim, toda a externalidade pode ser definida de forma única pela tupla (C, CS, CS') .

Com base na Definição 9 podemos definir jogos de partição com externalidade positiva (negativa) como:

Definição 10. Um jogo de partição é caracterizado como sendo de externalidade positiva (negativa) se e somente se $\forall(C, CS, CS')$ pertencentes à um jogo, $\epsilon(C, CS, CS') \geq (\leq) 0$.

É importante ressaltar que essa definição indica que CFG podem ser vistos como casos particulares dos PFG onde a externalidade é exatamente zero. Sendo assim, os algoritmos desenvolvidos para CFG não podem ser diretamente aplicados aos PFG, enquanto o contrário é possível.

As figuras 2.1 e 2.2 demonstram como a externalidade afeta o valor das coalizões e, por conseguinte, o valor das estruturas de coalizões. Na Figura 2.1, o valor das coalizões não é único e depende de como os demais agentes estão agrupados. Na Figura 2.2, os dois tipos possíveis de externalidades são demonstrados:

- na transformação da CS de CSa para CSc , há uma externalidade positiva e a coalizão $\{a_1, a_2\}$ tem um aumento em seu valor;
- na transformação da CS de CSb para CSc , há uma externalidade negativa e a coalizão $\{a_1, a_2\}$ tem um decréscimo em seu valor.

Um PFG (a exemplo do CFG) pode ter propriedades super-aditivas (se a união de quaisquer coalizões produz um valor maior para a estrutura de coalizão que a soma dos valores das coalizões individuais) ou sub-aditivas (quando ocorre o efeito contrário, ou seja, a união produz um valor menor). Ainda, a PFG pode exibir um efeito negativo (positivo) em externalidades quando a união de uma ou mais coalizões torna o valor geral da estrutura de coalizão menor (maior), através da alteração nos valores das demais coalizões presentes nessa CS.

A seguir é demonstrado um exemplo retirado de (MICHALAK et al., 2008), onde há um PFG com $A = \{1, 2, 3\}$. Considere:

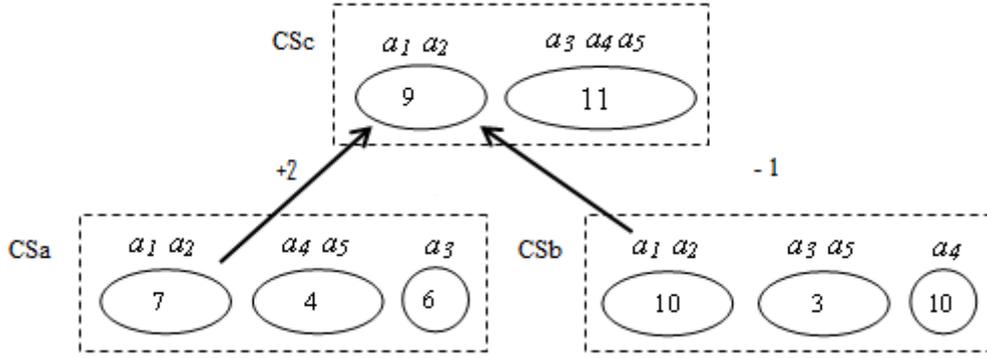


Figura 2.2: Exemplo de dois tipos de externalidades provenientes da formação da coalizão $\{a_3, a_4, a_5\}$ sobre a coalizão $\{a_1, a_2\}$

- $v((i); \{(1), (2), (3)\}) = 4, \forall i = 1, 2, 3;$
- $v((j, k); \{(i), (j, k)\}) = 9, \forall i, j, k \in A | i \neq j \neq k;$
- $v((i); \{(i), (j, k)\}) = 1, \forall i, j, k \in A | i \neq j \neq k;$
- $v(A; \{A\}) = 8.$

Nesse caso, o PFG é super-aditivo, mas a grande coalizão não é o melhor valor possível para as estruturas de coalizão (o melhor valor é quando todos os agentes estão em coalizões individuais), pois há um efeito de externalidade negativa no exemplo. O exemplo oposto pode ser visto a seguir:

- $v((i); \{(1), (2), (3)\}) = 3, \forall i = 1, 2, 3;$
- $v((j, k); \{(i), (j, k)\}) = 2, \forall i, j, k \in A | i \neq j \neq k;$
- $v((i); \{(i), (j, k)\}) = 8, \forall i, j, k \in A | i \neq j \neq k;$
- $v(A; \{A\}) = 4.$

Neste caso, o PFG é sub-aditivo, mas ainda assim a CS contendo apenas coalizões individuais não é a CS^* (nesse caso, a CS^* é composta por $\{(i), (j, k)\}$).

Isso demonstra que, ao contrario dos cenários descritos por CFG, a propriedade de ser super-aditivo ou sub-aditivo não garante que as melhores CS serão a grande coalizão ou a soma das coalizões individuais, respectivamente. É necessário dividir os cenários de PFG em quatro classes diferentes (MICHALAK et al., 2008):

1. Super-aditiva com externalidade positiva (PF_{sup}^+);
2. Super-aditiva com externalidade negativa (PF_{sup}^-);
3. Sub-aditiva com externalidade positiva (PF_{sub}^+);
4. Sub-aditiva com externalidade negativa (PF_{sub}^-);

No caso (PF_{sup}^+) e (PF_{sub}^-) é possível garantir que a CS contendo a grande coalizão e aquela contendo apenas coalizões individuais, respectivamente, são as CS^* . Sendo assim, por questão de simplificação, jogos PF_{sub}^+ serão referenciados no restante do texto simplesmente como PF^+ , enquanto jogos PF_{sup}^- serão referenciados como PF^- .

Um algoritmo para encontrar a CS^* em PFG foi proposto por (MICHALAK et al., 2008) e se baseia fortemente em algoritmos desenvolvidos para CFG (tanto o algoritmo de (SANDHOLM et al., 1999) quanto o de (RAHWAN et al., 2007b)). Em (RAHWAN et al., 2012) foi elaborado um algoritmo *anytime* que é o estado da arte e referência para esse tipo de problema (um algoritmo *anytime* devolve uma solução válida mesmo se interrompido antes do seu término e melhora a sua solução quanto mais tempo for executado). Também em (RAHWAN et al., 2012) foi desenvolvido um estudo computacional do problema de encontrar a CS^* em PFG, além de desenvolver uma função para alocar valores às coalizões baseada na possível influência de externalidade. Esta função é de particular importância por ser a única função que pode ser utilizada para validação dos algoritmos de PFG e por permitir limitar a influência da externalidade sobre as coalizões. Ela será abordada com mais detalhes na Seção 5.2.

Jogos com externalidade foram amplamente estudados em economia e outras ciências sociais, onde a interdependência de coalizões é um fator importante. Exemplos podem ser encontrados em conluíus de empresas que procuram retirar espaço do mercado de demais companhias. Quando duas empresas se unem para produzir um produto de melhor qualidade, elas buscam diminuir a competitividade das demais companhias do ramo, impondo assim uma externalidade negativa à elas (RAHWAN et al., 2012). Por outro lado, quando um conjunto de países se une para reduzir a emissão de gás carbônico, todos os demais países são beneficiados dessa coalizão, impondo uma externalidade positiva a eles (FINUS; RUNDSHAGEN, 2003). Outros exemplos na área de cooperação entre empresas podem ser encontrados em (CATILINA; FEINBERG, 2006). Na área matemática, (KOCZY, 2006) e (DO; NORDE, 2007) apresentaram grandes resultados na busca do *valor de Shapley* e na solução *core* de PFG.

De forma mais geral, políticas ambientais de países e sistemas multiagente com objetivos sobrepostos são todos jogos com externalidade puramente positiva, estando dentro da classe de jogos PF^+ . Do outro lado, conluio entre companhias e sistemas multiagente com recursos compartilhados ou objetivos conflitantes tem externalidade negativa e pertencem à classe de jogos PF^- (RAHWAN et al., 2012).

3 TRABALHOS RELACIONADOS À GERAÇÃO DE ESTRUTURAS DE COALIZÃO DE AGENTES

Em inteligência artificial e na ciência da computação de maneira geral, o procedimento que determina em qual ordem deve-se investigar as possíveis soluções de um dado problema é usualmente referido como um procedimento de busca. O procedimento de busca é executado sobre uma representação abstrata e estruturada do universo de soluções, denominada espaço de estados. Este modelo é geralmente implementado no algoritmo de busca através de uma estrutura de dados.

O principal enfoque deste procedimento de busca na área de jogos de coalizão é encontrar a estrutura de coalizão que maximize alguma métrica de avaliação do sistema. Em domínios que não são necessariamente super-aditivos ou sub-aditivos, algumas coalizões possuem maior valor trabalhando em separado enquanto outras ganham valor ao se unirem. Em tais situações, a estrutura de coalizão que maximiza o bem-estar social varia e a geração da estrutura de coalizão torna-se não trivial.

Denota-se o conjunto de todas as estruturas de coalizão como S^A . O objetivo é maximizar o bem-estar social dos agentes no sistema, encontrando a estrutura de coalizão ótima:

$$CS^* = \arg \max_{CS \in S^A} V(CS) \quad (3.1)$$

onde

$$V(CS) = \sum_{S \in CS} v(S). \quad (3.2)$$

Sandholm et al. (1999) definem uma estrutura de coalizão como uma partição exaustiva que contém somente coalizões disjuntas. Ou seja, cada um dos agentes deve pertencer a exatamente uma coalizão. Outros autores, como Shehory e Kraus (1998), relaxam esta restrição e consideram casos em que as coalizões podem se sobrepor. Para coalizões disjuntas, a complexidade assintótica do espaço de estados (número de estruturas de coalizão) é $O(|A|^{|A|})$ (SANDHOLM et al., 1999).

(SANDHOLM et al., 1999) descreve o processo de formação de estruturas de coalizão como uma sequência de três atividades que podem, em alguns casos, serem independentes umas das outras. A formação de estruturas de coalizão procede da seguinte maneira:

- Geração da estrutura de coalizão. Esta etapa consiste em particionar exaustivamente um dado conjunto de agentes do sistema em coalizões disjuntas. A partição resultante é denominada estrutura de coalizão. Por exemplo, em um sistema com três agentes $\{\{a_1, a_2, a_3\}\}$, existem sete possíveis coalizões: $\{\{a_1\}, \{a_2\}, \{a_3\},$

$\{a_1, a_2\}, \{a_1, a_3\}, \{a_2, a_3\}$ e $\{a_1, a_2, a_3\}$. Estas coalizões podem ser combinadas em cinco possíveis estruturas de coalizão: $\{\{\{a_1, a_2, a_3\}\}; \{\{a_1\}, \{a_2, a_3\}\}; \{\{a_2\}, \{a_1, a_3\}\}; \{\{a_3\}, \{a_1, a_2\}\}$ e $\{\{a_1\}, \{a_2\}, \{a_3\}\}$.

- Otimização do valor da coalizão. Nesta etapa os agentes devem ser divididos em coalizões que otimizem a utilidade total recebida.
- Divisão da utilidade entre os integrantes da coalizão. O valor, ou ganho, de cada coalizão deve ser dividido entre os agentes pertencentes de acordo com algum esquema. Usualmente, este esquema deve ser projetado de forma a garantir equidade ou estabilidade.

O número exato de coalizões que podem ser formadas com $|A|$ agentes é $2^{|A|} - 1$ e o número de estruturas de coalizão é

$$\sum_{i=1}^{|A|} Z(|A|, i), \quad (3.3)$$

onde $Z(|A|, i)$ é o número de estruturas de coalizão contendo i coalizões. A quantidade $Z(|A|, i)$, conhecida como o número de Stirling do segundo tipo, é definida pela recorrência:

$$Z(|A|, i) = iZ(|A| - 1, i) + Z(|A| - 1, i - 1), \quad (3.4)$$

considerando $Z(|A|, |A|) = Z(|A|, 1) = 1$ como limitantes. O valor da recorrência pode ser calculado pela seguinte fórmula fechada:

$$Z(|A|, i) = \frac{1}{i!} \sum_{j=0}^{i-1} (-1)^j \binom{i}{j} (i - j)^{|A|}. \quad (3.5)$$

De acordo com Sanholm et al. (1999), existe uma distinção primária que pode ser feita entre métodos desenvolvidos para encontrar uma estrutura de coalizão. Para o caso em que se possui capacidade de observar diretamente o valor de cada coalizão $v(S)$, o problema torna-se equivalente a uma classe de problemas de otimização como o empacotamento e cobertura de conjuntos. É equivalente também ao problema da determinação do vencedor em leilões combinatoriais (SANDHOLM et al., 1999). Em alguns domínios pode não ser possível observar $v(S)$ diretamente, pois este valor depende, por exemplo, de cálculos complexos ou negociação entre os agentes. Ou seja, existe um custo associado à informação.

Considerando que o espaço de busca das estruturas de coalizão é muito grande para ser exaustivamente enumerado, a busca pela estrutura de coalizão deve ser realizada em um subconjunto $N \subset S^A$ das estruturas de coalizão:

$$CS_N^* = \arg \max_{CS \in N} V(CS) \quad (3.6)$$

com a garantia desejável de que a estrutura de coalizão encontrada está dentro de um limite da solução ótima, ou seja, que existe um β finito e tão pequeno quanto possível onde:

$$\beta \geq \frac{V(CS^*)}{V(CS_N^*)} \quad (3.7)$$

O objetivo deste capítulo é fornecer uma base para o leitor compreender o problema relacionado à formação de estruturas de coalizão e como esse problema é abordado na literatura. A maioria dos algoritmos que são apresentados são do tipo *anytime*, ou seja, eles apresentam uma solução válida para o problema mesmo que interrompidos antes do seu final. Isso é desejável porque o problema de encontrar a estrutura de coalizão ótima é *NP-completo* e pode levar muito tempo para executar o algoritmo até seu término. Também é esperado que algoritmos *anytime* melhorem a sua solução quanto mais tempo forem executados.

O primeiro algoritmo a ser apresentado (Seção 3.1) é um algoritmo guloso utilizado para formação de estrutura de coalizão em ambientes descentralizados, proposto por Shehory e Kraus (1998). A seguir, na Seção 3.2 é apresentado o algoritmo proposto por Sandholm et al. (1999), que foi pioneiro em garantir um desempenho de pior caso. O trabalho mostrou ainda que nenhum outro método publicado até aquele momento era capaz de garantir um limite do ótimo. Na sequência (Seção 3.3), aborda-se a contribuição de Dang e Jennings (2004), que melhora os resultados de Sandholm e colegas. Após, o método de Rahwan et al. (2007) e Rahwan et al. (2009) (Seções 3.4 e 3.5, respectivamente) garantem um desempenho melhor com a vantagem de serem algoritmos do tipo *anytime*. Um algoritmo distribuído com garantias de qualidade é apresentado na Seção 3.6, sendo este considerado o melhor algoritmo distribuído existente para solucionar o problema de geração de estruturas de coalizão. Por fim é descrito um algoritmo desenvolvido por Rahwan/Michalak et al. (2012) na Seção 3.7, que permite localizar a CS^* em sistemas onde há externalidades.

3.1 Algoritmo distribuído sem garantia de desempenho

Alguns algoritmos para formação de estruturas de coalizão foram desenvolvidos para sistemas descentralizados. Um exemplo deles é o de (SHEHORY; KRAUS, 1998), que propuseram um algoritmo guloso para a formação de CS em domínios orientados à tarefas. Para cada tarefa no ambiente é atribuída uma coalizão com recursos suficientes para realizá-la. Um algoritmo (ou heurística) guloso é um procedimento que realiza escolhas localmente ótimas na expectativa de que esta escolha leve ao ótimo global. O método é baseado em um algoritmo guloso publicado por (CHVÁTAL, 1979) para o problema da cobertura de conjuntos.

Uma característica da heurística proposta por (SHEHORY; KRAUS, 1998) deve ser bem evidenciada: as coalizões possuem um *limite máximo de K agentes por coalizão*. Ou seja, não são absolutamente todas as coalizões que são avaliadas. Esta restrição determina um limite polinomial para o número de coalizões em $O(|A|^k)$. Contudo, o problema permanece *NP-completo*. Os autores argumentam que, em um sistema descentralizado, coalizões pequenas serviriam para evitar a necessidade de um uso abusivo de comunicação e que os recursos poderiam ser mais facilmente geridos pelos agentes.

O algoritmo proposto por (SHEHORY; KRAUS, 1998) consiste em duas fases distintas. A primeira fase diz respeito à distribuição do cálculo dos valores das coalizões. Após esta etapa preliminar, cada agente tem uma lista das coalizões potenciais com as quais ele se comprometeu em calcular os valores. Também possuem informação sobre as competências dos outros agentes membros destas coalizões.

A primeira sub-etapa do algoritmo é responsável pelo cálculo repetido dos valores das coalizões. Este cálculo precisa ser feito de forma iterada, pois os valores das coalizões podem variar continuamente conforme outras coalizões vão sendo formadas, o que significa que tarefas são realizadas e agentes ficam indisponíveis ou com menos recursos. Após o cálculo dos valores das coalizões, os agentes devem continuar com a segunda sub-etapa onde são decididas quais coalizões são melhores e devem ser formadas. Os passos do algoritmo são demonstrados abaixo.

1. Na fase preliminar todas as coalizões possíveis são calculadas de maneira distribuída. Cada agente a_i calcula as coalizões das quais faz parte e as coloca em uma lista L_i .
2. No estágio principal, um procedimento guloso subdividido em duas etapas é iterado por cada agente:
 - Para cada tarefa, os valores de todas as coalizões capazes de realizá-la são calculados. A distribuição do cálculo entre os agentes é feita da seguinte maneira. Cada agente a_i aborda os outros agentes pertencentes à coalizão sob análise, da lista L_i , e compromete-se com o cálculo dos valores das coalizões nas quais ambos agentes são membros. Assim, os compromissos de cálculo de cada valor de coalizão serão assumidos somente uma vez. Qual agente ficará responsável depende da ordem em que a comunicação ocorre. Durante a abordagem aos outros agentes da sua coalizão, o agente iniciador pode atualizar a sua informação sobre as capacidades atuais de seus parceiros potenciais.
 - Os agentes optam pela coalizão que possui o maior valor relacionado à tarefa a ser realizada.

Esse algoritmo também pode ser aplicado para considerar tarefas que devam ser realizadas em uma determinada ordem. Além disso, (SHEHORY; KRAUS, 1998) estabelecem um limite do ótimo para o seu algoritmo. Este limite, porém, é válido somente para o caso onde não há precedência entre tarefas e cresce logaritmicamente com o tamanho da coalizão. Mais especificamente, sendo CS a estrutura de coalizão e seu custo total $c_{\text{tot}} = \sum_{C_j \in CS} c_j$ o limite é dado por:

$$\rho = \frac{c_{\text{tot}}}{c_{\text{tot}}^*} \leq \sum_{i=1}^{\max(|C_j|)} \frac{1}{i}. \quad (3.8)$$

Contudo, no caso geral, a restrição imposta no tamanho máximo das coalizões faz com que a solução gerada pelo método possa estar a um limite arbitrário do ótimo, pois o limite apresentado acima considera apenas as coalizões válidas. Ou seja, o ótimo *verdadeiro* seria obtido pela busca completa em todas as coalizões em um espaço de estados gerado *sem* a imposição da restrição de tamanho máximo.

3.2 Garantia de limite ótimo

De acordo com (SANDHOLM et al., 1999), o espaço de busca das estruturas de coalizões pode ser visto como um reticulado parcialmente ordenado. Neste reticulado, o elemento que se encontra no nível mais elevado (nível $|A|$) é uma estrutura de coalizão onde cada coalizão consiste de apenas um agente e o elemento que se encontra no nível menos

elevado (nível 1) é a grande coalizão contendo todos os agentes. A Figura 3.1 mostra um exemplo deste reticulado, denominado de grafo de estruturas de coalizão, para um sistema com quatro agentes.

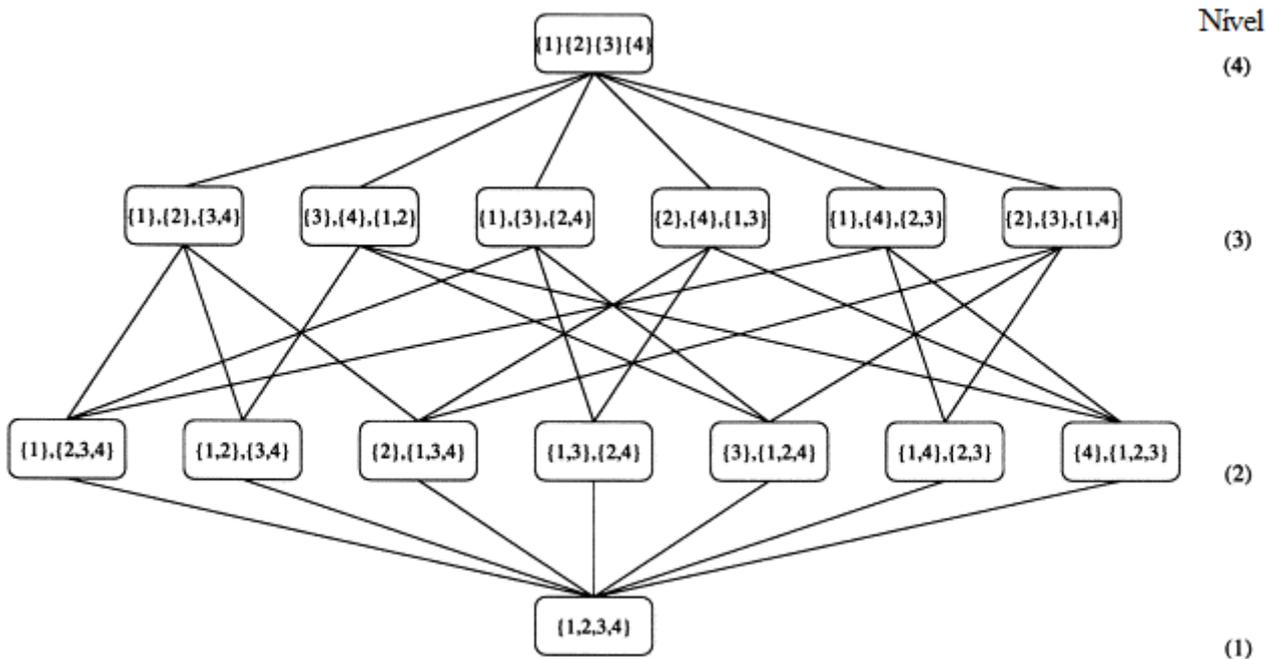


Figura 3.1: Reticulado da estrutura de coalizão para um jogo de quatro agentes, com nós representando estruturas de coalizão. Figura retirada de (SANDHOLM et al., 1999).

Quando o grafo é percorrido da base em direção ao topo, temos que se uma estrutura de coalizão CS_1 está acima de uma estrutura de coalizão CS_2 , a estrutura de coalizão CS_1 pode ser obtida a partir de CS_2 ao particionar uma de suas coalizões em duas coalizões. Então CS_1 tem exatamente uma coalizão a mais que CS_2 . Esta ordem parcial impõe uma estrutura no reticulado de maneira que podemos descrever o reticulado como tendo n níveis, onde n é o número de agentes. Ademais, cada estrutura no nível i contém exatamente i coalizões. Algo similar ocorre quando o grafo é percorrido do topo em direção a base, porém neste caso a operação não é uma partição e sim uma união.

Através do Teorema 1, Sandholm e colegas determinam como um limite β da solução ótima pode ser estabelecido realizando a busca na menor porção possível do espaço de estados representado pelo grafo de estruturas de coalizão. Adicionalmente, os autores provam outros dois teoremas que garantem que este limite é forte e que nenhum algoritmo diferente de um algoritmo que visite inicialmente os dois níveis inferiores consegue estabelecer um limite visitando apenas $n = 2^{|A|-1}$ estruturas ou menos.

Teorema 1. (SANDHOLM et al., 1999) **Busca mínima para estabelecer um limite.** Para se obter um limite β do ótimo, 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 nós visitados é $n = 2^{|A|-1}$.

Segundo os autores, estes resultados podem ser interpretados de maneira positiva ou como um resultado de impossibilidade. Na interpretação positiva do fato tem-se que um limite de desempenho pode ser garantido sem a busca exaustiva do espaço de estados.

Tabela 3.1: Redução gradual do limite da solução ótima garantida pelo Teorema 2.

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

Ademais, com o aumento do número de agentes a fração do espaço de busca que necessita ser visitada tende a zero ($\frac{N_{min}}{|M|} \rightarrow 0$ quando $|A| \rightarrow \infty$). Visto como resultado de impossibilidade estabelece-se que um número exponencial de estruturas de coalizão $n = 2^{|A|-1}$ deve ser buscado antes de se obter alguma garantia de desempenho.

Partindo dos resultados acima mencionados, os autores ainda mostram como o limite pode ser diminuído com a realização de busca adicional. Para tanto, propõem o algoritmo cujos passos são:

1. Realizar uma busca nos dois níveis inferiores do grafo de estrutura da coalizão.
2. Continuar realizando uma busca em largura a partir do nível superior do grafo enquanto houver tempo suficiente ou até visitar todo o grafo.
3. Retornar a estrutura de coalizão que possui a maior utilidade entre as vistas até então.

Este algoritmo estabelece o limite $\beta = |A|$ realizando inicialmente a busca nos dois níveis inferiores (conforme o Teorema 1), e depois segue com a execução iterada de um passo anytime que reduz o limite enquanto visita os espaços dos outros níveis. O Teorema 2 provado pelos autores estabelece como o limite é reduzido com esta busca adicional. A Tabela 3.1 mostra os limites que podem ser obtidos através da execução do algoritmo de (SANDHOLM et al., 1999) e que são garantidos pelo Teorema 2.

Teorema 2. (SANDHOLM et al., 1999) **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 $h = \lfloor \frac{|A|-l}{2} \rfloor + 2$.

Sandholm e colegas apresentaram resultados teóricos de seus estudos e focam na análise das garantias de desempenho no pior caso. Foram os primeiros a provar que há um número mínimo de estruturas de coalizão a serem buscadas antes que se possa estabelecer um limite inferior. Também foram os primeiros a demonstrar que é possível estabelecer esse limite.

3.3 Melhoria do limite ótimo

O trabalho de (DANG; JENNINGS, 2004) introduz um novo algoritmo anytime para o problema da geração da estrutura de coalizão baseado no algoritmo proposto por (SANDHOLM et al., 1999). Os autores comparam seu algoritmo com o de Sandholm et al. e mostram que, quando são desejados limites do ótimo que são pequenos (baixos), o método proposto é 107 vezes mais rápido (para sistemas com 50 agentes), 1023 vezes mais rápido (para sistemas com 100 agentes) e 10379 vezes mais rápido (para sistemas com 1000 agentes). Ainda, para maiores limites do ótimo, os autores mostram que não há diferença significativa entre o seu algoritmo e o de Sandholm e colegas.

Algumas definições feitas pelos autores são necessárias para a compreensão do algoritmo e, como a intenção deste trabalho não é explorar esse algoritmo, serão apresentadas apenas as definições mais importantes para a compreensão do mesmo. Seja E_k o conjunto de todas as estruturas de coalizão com k coalizões. Assim, tem-se o conjunto E de todas as estruturas de coalizão conforme a Equação 3.9, onde n é o número de agentes que é igual ao número máximo de coalizões (todos os agentes disjuntos).

$$E = \bigcup_{k=1}^n E_k, \quad (3.9)$$

Definição 11. Seja $SE(n, k, c)$ o conjunto de todas as estruturas de coalizão que possuem k coalizões e no mínimo uma coalizão com cardinalidade não menor que c .

Definição 12. Seja $SE(n, c)$ o conjunto de todas as estruturas de coalizão com cardinalidade entre 3 e $n - 1$ e que possuem ao menos uma coalizão com cardinalidade não menor que c . Ou seja:

$$SE(n, c) = \bigcup_{k=3}^{n-1} SE(n, k, c). \quad (3.10)$$

Dadas as Definições 11 e 12, o algoritmo de Dang e Jennings pode ser explicado da seguinte maneira. Inicialmente, uma busca é efetuada em todas as estruturas de coalizão que possuem uma, duas ou n coalizões. Ou seja, nos conjuntos E_1 , E_2 e E_n , respectivamente. Este primeiro passo do algoritmo efetua a busca nos dois níveis inferiores e no nível superior do grafo de estruturas de coalizão (Figura 3.1). Após, o algoritmo de (SANDHOLM et al., 1999) busca sequencialmente nos conjuntos restantes E_k , com k variando entre 3 e $n - 1$ ($3 \leq k \leq n - 1$). Porém, o algoritmo de Dang e Jennings busca apenas em alguns subconjuntos específicos de E_k . Mais especificamente, a busca ocorre no conjunto de todas as estruturas de coalizão que possuem k coalizões e no mínimo uma coalizão com cardinalidade não menor que $\lceil \frac{n(q-1)}{q} \rceil$, iterando em passos com q variando de $\lfloor \frac{n+1}{4} \rfloor$ até 2. A escolha destes valores como limitantes da iteração de q são explicados posteriormente. Para evidenciar a diferença entre o caminho da busca executada pelos algoritmos de (DANG; JENNINGS, 2004) e (SANDHOLM et al., 1999), o esquema da Figura 3.2 representa uma comparação entre a execução da busca pela estrutura da coalizão feita por cada um deles.

O Teorema 3 garante que a solução retornada está dentro de um limite do ótimo e que este limite é reduzido com a execução de cada passo de busca adicional, o que classifica o algoritmo de (DANG; JENNINGS, 2004) como *anytime*.

Teorema 3. (DANG; JENNINGS, 2004) Após terminar a busca em $SE(n, \lceil \frac{n(q-1)}{q} \rceil)$, a solução gerada pelo algoritmo está dentro do limite $\beta = 2q - 1$ do ótimo.

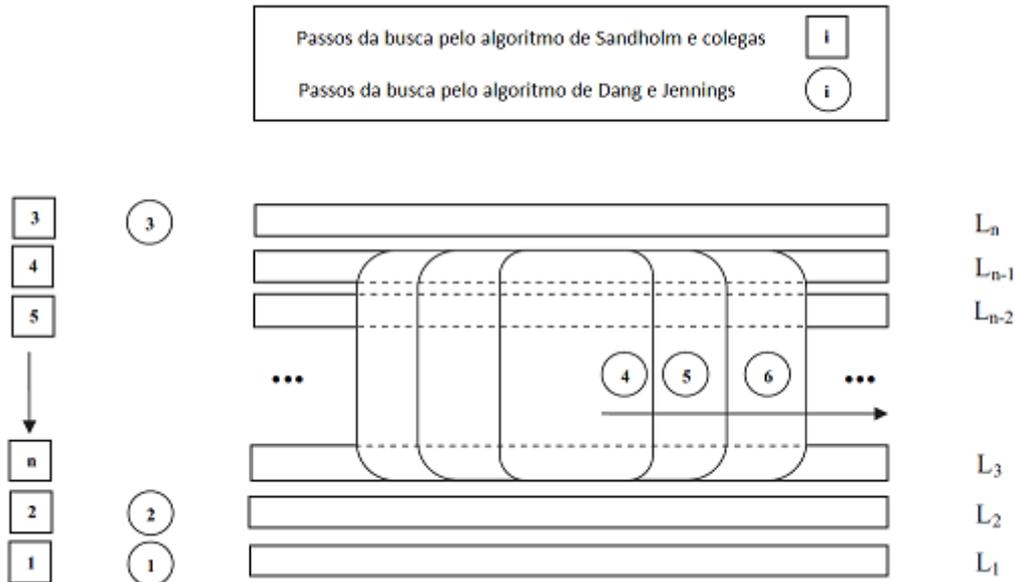


Figura 3.2: Comparação entre os caminhos percorridos pelos algoritmos de busca de (DANG; JENNINGS, 2004) e (SANDHOLM et al., 1999), com os valores dentro de círculos e quadrados representando, respectivamente, os passos dos dois algoritmos. Figura adaptada de (DANG; JENNINGS, 2004).

Com base no Teorema 3, é possível justificar a escolha de $\lfloor \frac{n+1}{4} \rfloor$ como o primeiro valor de q a ser iterado, pois considerando os Teoremas 1 e 2 de (SANDHOLM et al., 1999) que determinam que após buscar em E_1 , E_2 e E_n o limite estabelecido é $\beta = \lceil \frac{n}{2} \rceil$, deve-se escolher o maior valor possível de q tal que $2q - 1 < \lceil \frac{n}{2} \rceil$, ou seja $q = \lfloor \frac{n+1}{4} \rfloor$. A escolha do valor 2 como limitante inferior da iteração é justificada de maneira semelhante.

3.4 Uma nova representação

Rahwan et al. (RAHWAN et al., 2007b) apresentaram um novo algoritmo *anytime* para a geração da estrutura de coalizão, que utiliza uma nova representação, mais concisa, para o espaço de estados. O método dos autores é mais eficiente do que os algoritmos existentes até aquele momento em termos de tempo de computação. Pode gerar soluções que estão dentro de um limite baixo do ótimo ou que são ótimas, dependendo do objetivo desejado e explicitamente definido na entrada.

Os autores demonstram a eficácia de seu algoritmo empiricamente em comparação com um algoritmo de programação dinâmica, desenvolvido para o problema de particionamento completo de conjuntos (YEH, 1986). Os resultados mostram que o algoritmo de (RAHWAN et al., 2007b) foi 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 de programação dinâmica, em um sistema com 20 agentes.

Ao contrário dos algoritmos de (SANDHOLM et al., 1999) e (DANG; JENNINGS, 2004) vistos anteriormente, que observam os valores $v(CS)$ das estruturas de coalizão, o método de (RAHWAN et al., 2007b) realiza uma busca observando diretamente os valores $v(C)$ das coalizões.

De maneira simplificada, o método consiste em particionar o espaço de estados em um pequeno número de conjuntos de estruturas de coalizão que contém coalizões de de-

$$CL_S =$$

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

$$G = \{4\}, F(G) = \{ \{ \{a_1, a_2, a_3, a_4\} \}$$

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

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

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

$$G = \{1, 1, 1, 1\}, F(G) = \{ \{ \{a_1\}, \{a_2\}, \{a_3\}, \{a_4\} \}$$

Figura 3.3: Exemplo do espaço de estados para quatro agentes com os elementos $F(G)$ expandidos. Figura adaptada de (RAHWAN et al., 2007b). A notação apresentada é a mesma que a utilizada pelos autores.

terminados tamanhos, chamado de “*configuração*”. Por exemplo, as CS $\{\{a_1\}, \{a_2, a_3\}\}$ e $\{\{a_2\}, \{a_1, a_3\}\}$ fazem parte da mesma configuração $[1, 2]$. Sob esta representação do espaço, uma busca *online* (online significa que utiliza a informação obtida ao longo da execução da busca para guiar o processo de busca) é realizada utilizando uma heurística para guiar o processo. Esta heurística realiza cortes no espaço de busca, desconsiderando estruturas não válidas ou redundantes. Após buscar em apenas uma parte relativamente pequena do espaço, a heurística direciona a busca para as melhores configurações estimando a média e o limite superior dos valores das soluções em cada configuração. Ao concluir a busca em todas as estruturas de certa configuração, as estimativas sobre as outras configurações são refinadas. Com estas estimativas atualizadas, o algoritmo é capaz de efetuar ainda mais cortes no espaço de busca, ignorando as configurações que possuem um limite superior menor que o valor da melhor solução encontrada até o momento. A seguir, são apresentadas as definições essenciais utilizadas pelo algoritmo de Rahwan e colegas.

Seja $CL_s \in 2^A$ a lista de coalizões de tamanho $s \in \{1, 2, \dots, n\}$ tal que \mathcal{CL} é o

conjunto de todas as listas de coalizões ($\mathcal{CL} = \bigcup_{S=1}^n CL_S$). Seja $G_1, G_2, \dots, G_{|\mathcal{G}_{CS}|} \in \mathcal{G}_{CS}$ o conjunto de todas as possíveis configurações únicas. Ainda, seja $F : \mathcal{G}_{CS} \rightarrow \mathcal{CS}$ 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 da mesma configuração. A notação $N_1, N_2, \dots, N_{\mathcal{G}_{CS}} \in \mathcal{N}$ denota os elementos de \mathcal{N} . A Figura 3.3 mostra um exemplo com as listas CL_S e \mathcal{N} para um sistema com quatro agentes.

Como entrada, o algoritmo recebe o conjunto CL para o qual o valor da função característica $v(C)$ é conhecido para todas as coalizões C pertencentes a todas as listas $CL_S \in CL$. Um limite de aceitação $\beta \in [0, 1]$ do ótimo também é informado como entrada, significando que quando $\beta = 1$ o algoritmo deve parar apenas quando a solução ótima for encontrada e quando $\beta = 0$ qualquer solução encontrada é retornada. O algoritmo é constituído por três estágios:

1. Estágio de pré-processamento que percorre a entrada para obter os valores máximo e médio de cada $CL_S \in CL$.
2. Utiliza os valores do primeiro estágio para gerar a lista de $G \in \mathcal{G}_{CS}$ sem redundâncias e selecionar o elemento $F(G)$ no qual a busca pela estrutura de coalizão ótima deve ocorrer.
3. Determina os valores de cada estrutura de coalizão $CS \in F(G)$ para encontrar a solução CS^* .

O estágio de pré-processamento calcula os valores máximo e médio para cada lista $CL_S \in \mathcal{CL}$, denotados por max_S e avg_S . Inicialmente, os valores calculados são os das estruturas que contém uma coalizão (a grande coalizão) da lista $CL_{|A|}$ e que contém $|A|$ coalizões (ao somar os valores das coalizões na lista CL_1). A melhor dentre estas duas estruturas é guardada em CS' . Após, para cada lista CL_S , observa-se todas as suas coalizões C , guardando os valores máximo e médio ($max_S = \max_{C \in CL_S} v(C)$ e $avg_S = \frac{1}{|CL_S|} \sum_{C \in CL_S} v(C)$). A economia ocorre quando o algoritmo verifica as listas CL_S e $CL_{|A|-S}$ ao mesmo tempo. Para tanto, denota-se uma coalizão de índice c na lista CL como CL^c , onde $c \in 1, \dots, |CL|$. A atribuição de índices é feita de tal maneira que resulta que cada elemento $C \in CL_S$ no índice $c \in [1, \dots, S]$ é igual ao elemento $C' \in CL_{|A|-S}$ no índice $|CL_S| - c + 1$. Outra otimização feita é verificar ao mesmo tempo os valores de todas as estruturas que seguem a configuração $G = \{1, \dots, 1, i\}$. Com o término deste primeiro estágio uma solução é retornada com a garantia de estar a um limite $|A|/2$ do ótimo. Este limite é garantido pelo Teorema 1 de Sandholm et al. (Seção 3.2). A complexidade computacional do algoritmo de pré-processamento é $O(2^{|A|})$.

O objetivo do segundo estágio é encontrar o $F(G)$ que contém a estrutura CS ótima ou com valor dentro do limite estabelecido por β . Para tanto, inicialmente são enumeradas todas as configurações em \mathcal{G}_{CS} utilizando algum algoritmo que gere partições de coalizões de tamanho de um número inteiro. Por exemplo, para $|A| = 4$ as partições possíveis ($I_{|A|}$) são $\{4\}$, $\{3, 1\}$, $\{2, 2\}$, $\{2, 1, 1\}$ e $\{1, 1, 1, 1\}$. A seguir, utilizando os valores máximos e médios calculados no pré-processamento, calcula-se o limite superior (UB_G) e valor médio (AVG_G) para cada G . Estes valores são passados como entrada para essa etapa. Porém, antes compara-se a solução $V(CS')$ encontrada no estágio de pré-processamento com o limite superior considerando β . Caso o valor esteja dentro do limite estabelecido

como aceitável, o método para e retorna a solução CS' . Caso contrário, o processo continua para retornar o elemento $G \in \mathcal{G}_{CS}$ que possui o maior limite superior.

O terceiro estágio do algoritmo de (RAHWAN et al., 2007b) é responsável por realizar a busca nas estruturas de coalizão de G , caso não tenha ocorrido o caso de uma solução prematura ter sido obtida nos estágios anteriores. O algoritmo gera e verifica todas as estruturas de coalizão dentro do elemento particular G estabelecido pelo algoritmo do estágio anterior. Porém, ao invés de executar uma busca ingênua em um grande número de elementos, o algoritmo evita gerar coalizões sobrepostas (inválidas neste caso) e, mais importante ainda, evita estruturas de coalizão redundantes.

3.5 Algoritmo IP

Apesar do algoritmo proposto por (RAHWAN et al., 2007a) ser o melhor algoritmo do tipo *anytime* desenvolvido, ainda não era capaz de encontrar a solução ótima em um tempo menor que o algoritmo desenvolvido utilizando programação dinâmica, proposto em (YEH, 1986). Isso porque o algoritmo não procurava em todas as estruturas de coalizão pela ótima. Sendo assim, apenas poderia proporcionar um valor relativo ao ótimo.

Em (RAHWAN et al., 2009), uma nova proposta de algoritmo *anytime* permite localizar a estrutura de coalizão ótima em um tempo menor que o utilizado pelo algoritmo de programação dinâmica. Neste algoritmo (chamado de *IP*), foi proposto uma nova forma de busca e um algoritmo de *branch-and-bound*, tornando assim o algoritmo de busca *anytime* proposto mais rápido que qualquer outro. A representação do espaço se manteve em relação ao algoritmo de (RAHWAN et al., 2007b) e por isso não será descrito novamente.

O primeiro passo é calcular os limites máximos e mínimos de cada configuração. A computação é a mesma do utilizado em (RAHWAN et al., 2007b), porém um limite de pior caso é calculado para cada configuração G , sendo dada pela Equação 3.11, onde $AvG_{g_2}^* = \arg \max_{G \in g_2} (AvG_G)$ e $UB^* = \max_{G \in g} (UB_G)$. Assim, enquanto o limite de pior caso para o algoritmo de (SANDHOLM et al., 1999) é um inteiro (e, portanto, no mínimo 2, o que representa 50% do ótimo), o limite aqui não necessariamente é um inteiro e pode ser definido como 1 (o que equivale a buscar a CS^*).

$$LB^{min} = \frac{UB^{max}}{AvG_{g_2}^*}, \quad (3.11)$$

Depois do passo inicial de cálculo de limites, um subespaço de busca é selecionado, considerando os valores encontrados inicialmente. Esse subespaço é promissor no sentido de ter o maior UB que os demais e, portanto, pode ter o maior valor de estrutura de coalizão. As CS dentro do subespaço de G são procuradas e essa busca termina quando uma das seguintes condições for satisfeita:

- O valor de uma estrutura de coalizão é igual ao limite superior: essa é a melhor estrutura de coalizão do subespaço;
- Todas as possíveis estruturas de coalizão foram procuradas ou descartadas.

Essa etapa é diferente da proposta anteriormente por (RAHWAN et al., 2007b) e a velocidade com que a solução ótima é encontrada depende de quão próximo o limite superior está do valor ótimo. Uma vez que o subespaço tenha sido completamente examinado (e assumindo que a solução ótima não tenha sido encontrada), o algoritmo de busca passa a

procurar o CS^* em outro subespaço. Esse outro subespaço é definido selecionando-se o G tal que:

$$G = \arg \max_G \in g(UB_G). \quad (3.12)$$

É importante ressaltar que essa regra deve apenas ser utilizada na busca da CS^* . Como esse subespaço tem maior probabilidade de obter essa CS^* (uma vez que tem maior limite superior), ele deve ser primeiramente explorado em detrimento dos demais. Porém, se a busca for por uma estrutura dentro de um limite β do ótimo devemos selecionar os subespaços menores e que possam fornecer maior informação acerca de onde encontrar uma CS dentro do limite do ótimo. Esse algoritmo propõe que, caso a busca demore mais que o tempo disponível, seja possível alterar o limite β online (durante a execução do algoritmo) para que ele não seja equivalente a 1, mas sim um menor limite do ótimo.

Apesar dos avanços, procurar dentro de um subespaço continua sendo o gargalo do algoritmo, pois é possível que todo o espaço de busca tenha que ser expandido antes de se localizar a CS^* . A solução para minimizar os custos desta fase do algoritmo é identificar as CS inválidas ou redundantes, bem como eliminar os subespaços que contenham o limite máximo inferior ao melhor valor já encontrado. Além disso, um aprimoramento em relação a como "caminhar" pelas possíveis coalizões também é proposto. Cada CS válida de uma sub-região G é gerada somando os valores de cada uma das coalizões dentro de G . Ao mesmo tempo em que o valor das somas das coalizões é computado, também é possível computar o limite superior e inferior das demais coalizões.

O resultado do algoritmo proposto é capaz de achar a solução ótima em 0.082% do tempo necessário pelo algoritmo prévio do estado da arte. Além disso, é capaz de localizar a CS^* para mais de 17 agentes em um tempo considerado razoável (algumas horas).

3.6 O primeiro algoritmo distribuído com garantia de desempenho

(MICHALAK et al., 2010) apresentou o primeiro algoritmo distribuído para resolver o problema de geração de estruturas de coalizão com garantia de desempenho, chamado de $D-IP$. Até então, o único algoritmo existente para a resolução do problema tinha sido apresentado por (SHEHORY; KRAUS, 1998), mas esse algoritmo não possuía qualquer medida de desempenho. Apesar de ser simples de implementar, o algoritmo proposto por (SHEHORY; KRAUS, 1998) era baseado em uma abordagem gulosa e não se preocupava com questões de balanceamento de carga entre os agentes (MICHALAK et al., 2010). O algoritmo apresentado por (MICHALAK et al., 2010) é considerado o estado da arte para algoritmos distribuídos que buscam encontrar a CS^* .

Em (RAHWAN; JENNINGS, 2007) é descrito um algoritmo capaz de dividir entre os agentes o custo do cálculo dos valores das coalizões. Esse algoritmo serve como base para o proposto por (MICHALAK et al., 2010). A motivação central do algoritmo é que os custos de processamento e memória possam ser distribuídos entre os agentes sem que haja um contraponto no aumento expressivo de comunicação. Ou seja, que os agentes não sejam forçados a compartilhar toda a informação coletada com os demais agentes. Outra vantagem é a do sistema distribuído ser mais robusto a falhas. As particularidades dos recursos disponíveis para cada agente podem ser consideradas para otimizar o sistema, mas é comumente considerado (sem perda de generalidade) que todos os agentes possuem os mesmos recursos e poder computacional.

O algoritmo é dividido em três etapas, similares às aquelas encontradas no algoritmo IP

(Seção 3.5):

1. Cálculo distribuído dos valores das coalizões;
2. Cálculo dos valores máximos e médios de cada configuração;
3. Busca distribuída dos subespaços que podem conter a CS^* .

O algoritmo de (RAHWAN; JENNINGS, 2007) foi chamado de DCVC (*Distributing Coalition Value Calculations*) e é responsável pela primeira etapa do algoritmo D-IP. A ideia central por trás do DCVC é dividir de forma igualitária a função de cálculo de valor das coalizões entre os agentes. Para tal, primeiro as coalizões são divididas em listas de coalizões de um mesmo tamanho (conforme descrito em 3.4). Isso produz n listas, nomeadas de L_1, L_2, \dots, L_n tal que $L_s : s \in \{1, 2, \dots, n\}$ contém todas as coalizões de tamanho s ordenadas. Cada lista L_s é então dividida em $\lfloor \frac{|L_s|}{|A|} \rfloor$ segmentos. Cada agente possui um identificador único que irá indicar a ele por qual segmento ele é responsável. Rahwan e Jennings (2007) alegam que a existência desse identificador não afeta o aspecto distribuído do algoritmo pois é possível supor que os agentes tenham noção da sua posição na lista de agentes. Por fim, as coalizões que não foram calculadas (devido as listas possuírem um número de coalizões não divisível pelo número de agentes) são distribuídas entre os agentes (uma vez que as listas são ordenadas, os agentes podem calcular de forma direta quais valores de coalizões cada um deve calcular).

Um exemplo da divisão de coalizões entre os agentes pelo algoritmo DCVC pode ser visto na Figura 3.4. Apesar de o algoritmo DCVC produzir uma divisão simples das coalizões entre os agentes, ele nem sempre permite que os agentes possuam conhecimento próprio suficiente para formar estruturas de coalizão. No exemplo da Figura 3.4, nenhuma estrutura de coalizão pode ser formada utilizando-se apenas as coalizões calculadas por um agente. Dessa forma, é necessário modificar o algoritmo para que cada agente possa avaliar o valor de uma ou mais CS utilizando apenas as coalizões calculadas por ele, sem a necessidade de trocar informações com os demais agentes

A modificação começa na ordenação das listas. Enquanto as listas de $L_i : i \in \{1, 2, \dots, \lfloor \frac{|A|}{2} \rfloor - 1\}$ são divididas em segmentos igual ao DCVC, as demais listas L_{n-i} são divididas na ordem reversa àquela do DCVC. Essa abordagem permite que cada agente possua as coalizões necessárias para formar alguma das CS que são avaliadas na primeira etapa do algoritmo IP. É necessário a comunicação do valor da coalizão L_1 que cada agente calculou, a fim de que todos tenham o valor das coalizões de tamanho 1. A Figura 3.5 mostra como são divididas as coalizões no algoritmo D-IP.

Após o cálculo dos valores das coalizões, cada agente armazena o valor máximo e médio de cada lista, além da melhor CS encontrada até o momento. Para que se possa prosseguir ao segundo estágio do D-IP é necessário que cada agente transmita as informações armazenadas referentes ao maior valor da coalizão de cada lista, bem como a melhor CS encontrada. Em posse dessa informação, cada agente pode calcular o valor máximo e médio de cada lista, bem como a melhor CS encontrada entre todos os agentes.

Para o segundo passo do algoritmo, cada agente irá calcular os valores máximos e médios de cada subespaço, eliminando aqueles que não podem conter a CS^* . Este passo do algoritmo não requer comunicação entre os agentes, mas uma vez encontrados os subespaços mais promissores é necessário que os agentes comuniquem os valores das coalizões que podem fazer parte dele. Como cada lista tem um número exponencial de coalizões, se faz necessário determinar quais coalizões realmente podem fazer parte da CS^* antes de transmiti-las. Para isso, se utilizam dois filtros.

L_1	L_2	L_3	L_4	L_5	L_6
$a_1 \{ 6$	$a_1 \{ 5,6$	$a_1 \{ 4,5,6$	$a_1 \{ 3,4,5,6$	$a_1 \{ 2,3,4,5,6$	$a_3 \rightarrow 1,2,3,4,5,6$
$a_2 \{ 5$	$a_1 \{ 4,6$	$a_1 \{ 3,5,6$	$a_1 \{ 2,4,5,6$	$a_2 \{ 1,3,4,5,6$	
$a_3 \{ 4$	$a_2 \{ 4,5$	$a_2 \{ 3,4,5$	$a_2 \{ 2,3,5,6$	$a_3 \{ 1,2,4,5,6$	
$a_4 \{ 3$	$a_2 \{ 3,6$	$a_2 \{ 2,5,6$	$a_2 \{ 2,3,4,6$	$a_4 \{ 1,2,3,5,6$	
$a_5 \{ 2$	$a_3 \{ 3,5$	$a_3 \{ 2,4,5$	$a_3 \{ 2,3,4,5$	$a_5 \{ 1,2,3,4,6$	
$a_6 \{ 1$	$a_3 \{ 3,4$	$a_3 \{ 2,3,6$	$a_3 \{ 1,4,5,6$	$a_6 \{ 1,2,3,4,5$	
	$a_4 \{ 2,6$	$a_4 \{ 2,3,5$	$a_4 \{ 1,3,5,6$		
	$a_4 \{ 2,5$	$a_4 \{ 2,3,4$	$a_4 \{ 1,3,4,6$		
	$a_5 \{ 2,4$	$a_4 \{ 1,5,6$	$a_5 \{ 1,3,4,5$		
	$a_5 \{ 2,3$	$a_4 \{ 1,4,6$	$a_5 \{ 1,2,5,6$		
	$a_6 \{ 1,6$	$a_5 \{ 1,4,5$	$a_6 \{ 1,2,4,6$		
	$a_6 \{ 1,5$	$a_5 \{ 1,3,6$	$a_6 \{ 1,2,4,5$		
$a_1 \rightarrow 1,4$		$a_5 \{ 1,3,5$	$a_6 \rightarrow 1,2,3,6$		
$a_2 \rightarrow 1,3$		$a_6 \{ 1,3,4$	$a_1 \rightarrow 1,2,3,5$		
$a_3 \rightarrow 1,2$		$a_6 \{ 1,2,6$	$a_2 \rightarrow 1,2,3,4$		
		$a_6 \{ 1,2,5$			
		$a_4 \rightarrow 1,2,4$			
		$a_5 \rightarrow 1,2,3$			

Figura 3.4: Exemplo do resultado da divisão das coalizões entre os 6 agentes no algoritmo DCVC. Por questão de simplificação, as coalizões são apresentadas usando apenas os índices dos agentes. Retirado de (MICHALAK et al., 2010).

O primeiro estabelece que se o valor da coalizão é menor que o valor das soma das coalizões unitárias que o compõe, essa coalizão não pode fazer parte de CS^* e portanto não deve ser transmitida. Por exemplo, se $v\{a_1, a_2\} < v\{a_1\} + v\{a_2\}$, então a coalizão $\{a_1, a_2\}$ não é transmitida. Quando a distribuição dos valores das coalizões é feita utilizando-se uma distribuição uniforme, esse filtro irá eliminar aproximadamente 50% das coalizões de uma lista.

O segundo filtro estabelece que se a Equação 3.13 é satisfeita, então a coalizão c não será transmitida, pois não pode fazer parte da CS^* . Ou seja, se ao substituirmos em um subespaço o valor máximo de uma lista pelo valor de c e o resultado for menor que a melhor CS encontrada, então certamente c não faz parte da CS^* e podemos descartá-la.

$$v(c) + UB_{i_k} - Max_{|c|} < v(CS^*_{atual}) \quad (3.13)$$

Utilizando esses dois filtros, as demais coalizões são transmitidas entre os agentes e é iniciado o terceiro passo do algoritmo, onde os subespaços promissores são avaliados. Para distribuir a divisão do subespaço, as CS que o compõe são ordenadas (da mesma

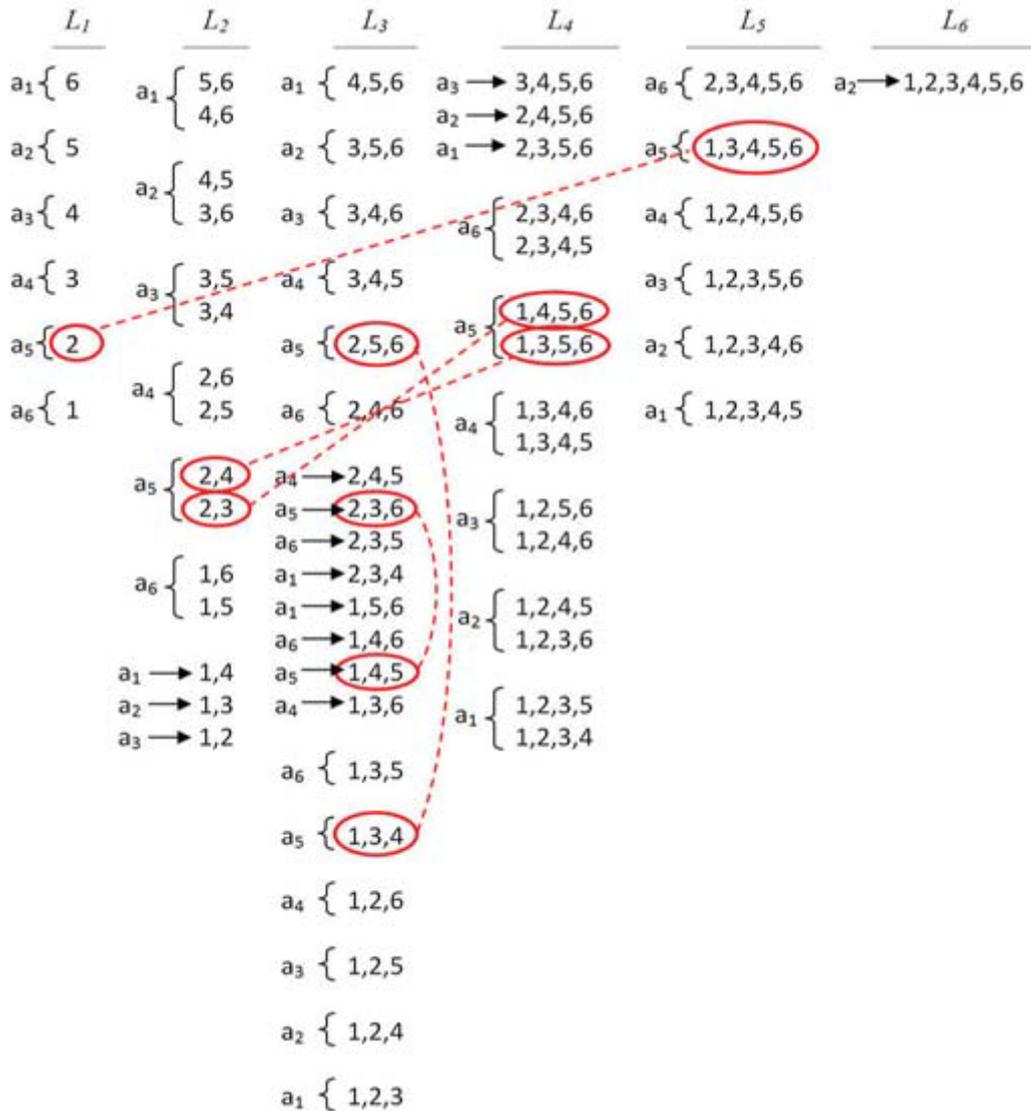


Figura 3.5: Exemplo do resultado da divisão das coalizões entre os 6 agentes no algoritmo D-IP. Exemplos de coalizões que formam estruturas de coalizão estão marcadas pela elipse e conectados pela linha tracejada. Retirado de (MICHALAK et al., 2010).

forma como é feito no algoritmo IP) e, assim como na primeira etapa do algoritmo DCVC, essas CS são divididas em segmentos. Cada agente é responsável pelo cálculo dos valores de coalizão de um determinado segmento.

3.7 Algoritmos para problemas com presença de externalidade (IP^+)

A pesquisa acerca da formação de estruturas de coalizão em ambientes que possuam externalidade é bastante recente, tendo no artigo de (MICHALAK et al., 2008) a primeira proposta para a resolução do problema de encontrar a CS^* . Isso porque até então não havia algoritmos ou heurísticas que pudessem estabelecer limites superiores e inferiores às CS levando em consideração externalidades. A partir de 2008, diversas abordagens para o problema foram propostas, desde novas representações para o espaço de busca até algoritmos que fossem capazes de percorrer esses espaços de forma mais eficiente. Porém, apenas em (RAHWAN et al., 2012) foi feito um estudo computacional acerca do

tema, onde é proposto um algoritmo capaz de resolver o problema para classes específicas e funções e métricas que podem ser utilizadas para realizar comparações entre algoritmos e heurísticas.

Em (RAHWAN et al., 2012) os autores desenvolveram um algoritmo para encontrar a CS^* em um sistema onde há externalidades tanto positivas quanto negativas e, portanto, deve ser modelado com um PF . As classes de externalidades apresentadas são PF_{sub}^+ e PF_{sup}^- , uma vez que as demais classes têm a CS^* sendo ou a grande coalizão ou a estrutura com coalizões individuais (Seção 2.3). Essas classes (com externalidade) são de grande importância principalmente para a área econômica, mas há também aplicações no âmbito de sistemas multiagente, como distribuição de conexões de rede ou divisão de recursos computacionais. No restante do texto, essas classes serão chamadas simplesmente de PF^+ e PF^- , respectivamente. Por considerar externalidades tanto positivas quanto negativas e ser fortemente baseado no algoritmo IP (Seção 3.5), os autores chamaram esse algoritmo de IP_-^+ .

O artigo propõe um algoritmo dividido em três etapas:

1. Calcular o valor máximo e mínimo de coalizões em PF^+ e PF^- ;
2. Identificar o número mínimo de estruturas de coalizões que necessitam ser exploradas antes de se estabelecer um limite inferior;
3. Melhorar o limite inferior da solução final em relação à CS^* quando é aumentado o espaço de busca;

3.7.1 Cálculo do valor máximo e mínimo de coalizões em PF^+ e PF^-

Para calcular o valor mínimo de cada coalizão é necessário tomar por base o seguinte teorema:

Teorema 4. Dado uma coalizão $C \subseteq A$, uma partição $P \in P_C$, uma estrutura de coalizão $CS \supseteq P$ e sendo $C = \{a_1, \dots, a_{|C|}\}$, temos que:

$$v(P, \{C \cup P\}) \leq v(P, CS) \leq v(P, a_1, \dots, a_{|C|} \cup P) \quad (3.14)$$

Partindo de uma estrutura de coalizão inicial CS , é possível formar outra estrutura de coalizão CS' ao fundir duas coalizões em uma. Após a formação de CS' , as demais coalizões em P não envolvidas na fusão permanecem com seu valor inalterado ou menor, devido às externalidades negativas. Isso equivale a dizer que:

$$v(P, CS') \leq v(P, CS) \quad (3.15)$$

Esse resultado não limita o valor das coalizões, mas sim das partições. Assim, o valor para cada partição pode ser calculado conforme a Equação 3.16.

$$LB_P(UB_P) = \sum_{p \in P} v(p, P \cup \vec{C}) UB_P(LB_P) = \sum_{p \in P} v(p, a_1, \dots, a_{|C|} \cup P). \quad (3.16)$$

Se assumirmos que $P = C$, podemos calcular o valor máximo que C pode assumir, bem como o valor mínimo.

3.7.2 Identificação do número mínimo de estruturas de coalizões que necessitam ser exploradas antes de se estabelecer um limite inferior

Uma vez calculado o máximo e mínimo para cada coalizão, é possível identificar o mínimo de espaço necessário para buscar um limite β do ótimo. A garantia de desempenho para algoritmos desenvolvidos para PFG é diferente daquele de CFG. Em CFG, para se obter um β mínimo é suficiente buscar nos primeiros dois níveis do grafo de estruturas de coalizão. Em PFG, a externalidade exige que sejam buscadas mais estruturas de coalizão, a fim de se obter estruturas de coalizão que levem em consideração os valores máximos das coalizões. Por isso, para estabelecer um limite β do ótimo em PF^+ é necessário buscar em CS diferentes daquelas para encontrar β em PF^- . A seguir, são descritos dois teoremas que indicam que é possível estabelecer um limite β do ótimo mesmo em PFG. As provas encontram-se em (RAHWAN et al., 2012).

Teorema 5. Para estabelecer um limite β no valor da estrutura de coalizão em PF^+ é necessário calcular o valor de todas estruturas de coalizão pertencentes às partições de inteiros tal que $|I_n| \leq 2$. O número de CS avaliadas é igual à 2^{A-1} e o limite é $\beta = A$.

Teorema 6. Para estabelecer um limite β no valor da estrutura de coalizão em PF^- é necessário calcular o valor das CS que pertencem ao subespaço: $\{[|A|]; [|A| - 1, 1]; [|A| - 2, 1, 1]; \dots; [1, 1, 1, \dots, 1]\}$. A avaliação dos valores dessas CS fornece um $\beta = \lceil \frac{|A|}{2} \rceil$ e o número de estruturas de coalizão avaliadas é $2^A - A + 1$.

3.7.3 Melhorando o limite inferior da solução final em relação à CS* quando é aumentado o espaço de busca

Após calcular os limites das coalizões e saber onde procurar pela estrutura de coalizão desejada, um novo algoritmo é proposto em (RAHWAN et al., 2012) para reduzir ainda mais a distância entre a solução encontrada e a ótima. O algoritmo tem como base o seguinte teorema:

Teorema 7. Seja X um conjunto de agentes e Y_s um conjunto contendo subconjuntos de X de tal forma que $\forall y \in Y_s, |y| \leq s$. Se considerarmos X um conjunto de todas as possíveis coalizões, Y_n um conjunto de todas as possíveis estruturas de coalizão, e se $Y'_n \subseteq Y_n$ é um conjunto de todas as possíveis CS onde o valor máximo de cada elemento de X aparece pelo menos uma vez, se procurarmos dentro de Y' temos uma solução que está dentro do limite $\beta = n$ em relação à CS*.

Baseado nesse teorema, a relação vinculada ao limite β pode ser reduzida utilizando duas etapas:

1. Adicionar coalizões ao X e agrupar em cada CS as coalizões que são iguais às coalizões adicionadas. Dessa forma, o número máximo de elementos de X que podem aparecer em uma CS é m , sendo que m é obrigatoriamente menor que o número de agentes. Por exemplo, se há três agentes, as possíveis estruturas de coalizão são: $\{\{a_1, a_2, a_3\}\}; \{\{a_1\}, \{a_2, a_3\}\}; \{\{a_2\}, \{a_1, a_3\}\}; \{\{a_3\}, \{a_1, a_2\}\}$ e $\{\{a_1\}, \{a_2\}, \{a_3\}\}$. Supondo que X contenha todas as possíveis coalizões, se adicionarmos $\{\{a_1\}, \{a_2\}\}$ à X e agruparmos a coalizão que corresponde à estrutura de coalizão (por exemplo, $\{\{a_1\}, \{a_2\}, \{a_3\}\}$ se torna $\{\{a_1, a_2\}, \{a_3\}\}$, o número máximo de elementos de X que podem aparecer em uma estrutura de coalizão cai de 3 para 2.

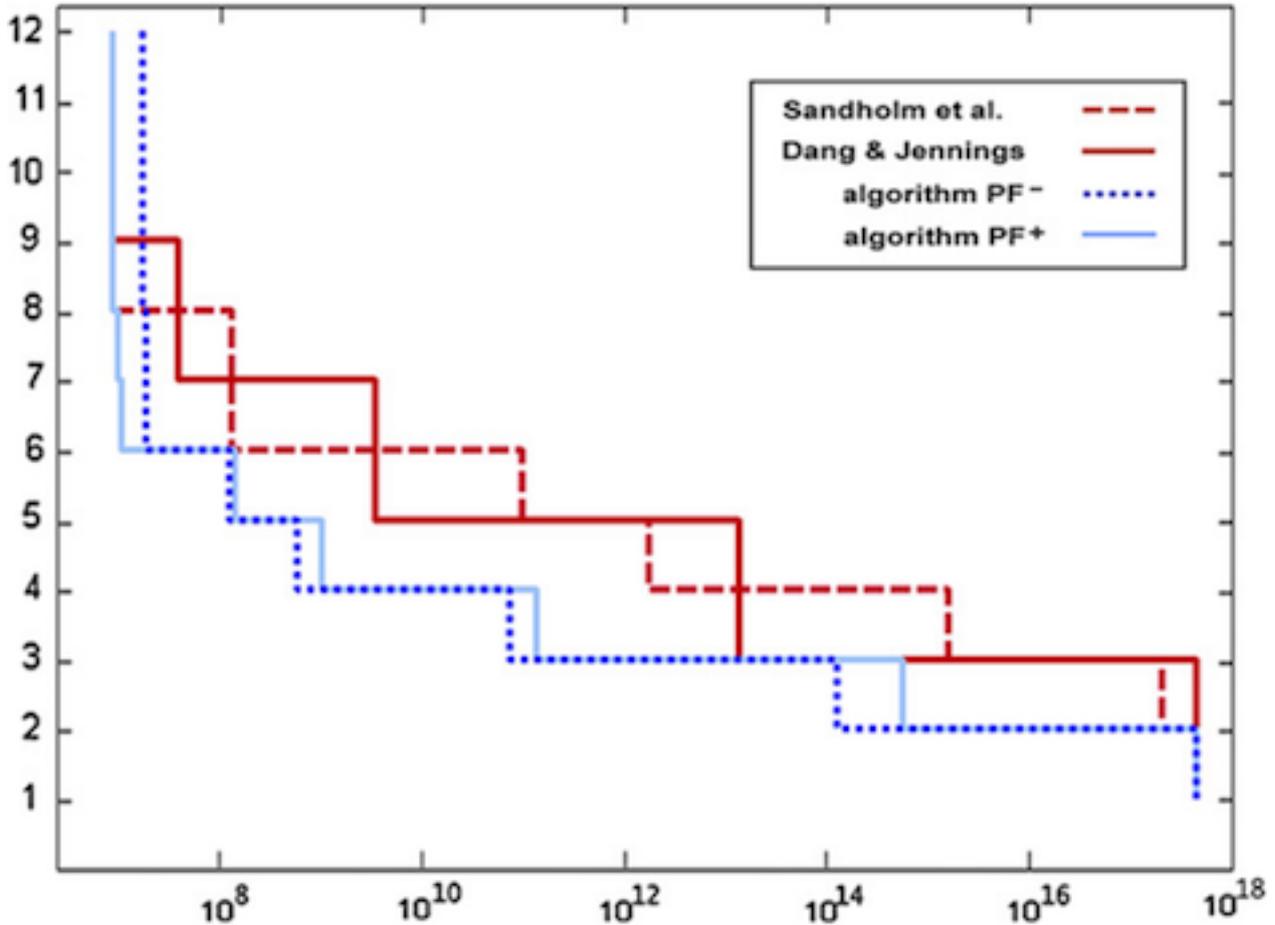


Figura 3.6: O limite inferior β em relação ao número de estruturas de coalizão procuradas, para 24 agentes. O eixo vertical representa o valor de β enquanto o eixo horizontal representa o número de estruturas de coalizão procuradas.

2. Para cada partição P que foi adicionada à X , examinar em que CS essa partição aparece com valor máximo. Dessa forma, procura-se em todas as CS onde o valor de X é máximo e podemos com isso reduzir o limite β .

Para validar o método utilizado, Rahwan e colegas realizaram experimentos com 24 agentes. Para experimentos com PF^+ , o algoritmo é expressivamente mais rápido que o seu similar em CFG. Para estabelecer um limite com $\beta = 3$, o número de estruturas de coalizão requeridas é de apenas 0.00003% daquelas requeridas por (SANDHOLM et al., 1999) e de 0.9% das requeridas por (DANG; JENNINGS, 2004).

Porém, quando os experimentos utilizaram externalidades negativas (PF^-), os resultados exigiram que mais estruturas de coalizões fossem buscadas em relação aos demais algoritmos para estabelecer um limite inicial. Após esse limite inicial ter sido encontrado, melhorar em relação ao ótimo se tornou muito mais rápido. A Figura 3.6, extraída do trabalho dos autores, demonstra essa comparação.

3.8 Discussão e revisão dos algoritmos existentes

Embora o algoritmo proposto por (SANDHOLM et al., 1999) (Seção 3.2) tenha sido capaz de prover garantias de desempenho e ser anytime, ele possui diversas limitações.

Dentre elas, pode-se destacar o fato de que a garantia era no máximo de 50% do valor do ótimo e a busca pela CS^* era realizada de forma exaustiva. Apesar de (DANG; JENNINGS, 2004) (Seção 3.3) terem melhorado tanto a garantia quanto a forma de busca, ainda utilizavam um algoritmo cujos valores de entrada eram os valores das estruturas de coalizão e a busca pela CS^* deveria ser realizada de forma exaustiva.

O algoritmo proposto por (RAHWAN et al., 2007a) (Seção 3.4) mudou a forma de representar o espaço de busca, que passou de um grafo de coalizões para um espaço de configurações. Esta nova abordagem permitiu que fosse possível gerar algoritmos que utilizam os valores das coalizões como entradas e também cujo valor de β não fosse apenas um número inteiro. Os algoritmos que seguiram e que hoje são considerados o estado da arte na área de CSG utilizam essa mesma abordagem, refinando apenas as formas como os valores das coalizões são calculados ou como a estrutura de coalizão é procurada dentro das configurações.

Em (RAHWAN et al., 2009) foi proposto o algoritmo IP (Seção 3.5), ainda utilizado nos mais recentes trabalhos da área. Este algoritmo é a base do IP^+ , considerado o melhor algoritmo para encontrar a estrutura de coalizão em PFG e descrito na Seção 3.7.

No campo de algoritmos distribuídos capazes de encontrar soluções para o problema de geração de estruturas de coalizão, destacam-se os algoritmos de (SHEHORY; KRAUS, 1998) (Seção 3.1) e (MICHALAK et al., 2010) (Seção 3.6). O algoritmo D-IP (MICHALAK et al., 2010) é o primeiro algoritmo distribuído a encontrar a solução com garantia de que é ótima para o problema. Ele distribui o cálculo dos valores das coalizões entre os agentes de forma ótima e considera que esses valores são gerados por uma função característica.

Apesar de todos os trabalhos já existentes na área, nenhum é capaz de abordar o problema quando este é distribuído e pertencente a PFG. Os algoritmos apresentados possuem limitações (por serem centralizados ou específicos para problemas de CFG) que os tornam extremamente eficientes para determinados cenários, mas impossíveis de aplicar em outros. Essa especificidade dos algoritmos estimulou a busca por um algoritmo que pudesse encontrar a CS^* tanto em CFG quanto em PFG, em ambientes descentralizados e centralizados. Com base nisso, é proposto o algoritmo desenvolvido nesse trabalho e descrito na Seção 4.

4 ABORDAGEM

4.1 Motivação

Jogos de partição têm recebido cada vez mais atenção do meio científico tanto nas áreas de economia e ciências sociais quanto na área de inteligência artificial. Além dos exemplos já providos na Seção 2.3, muitos outros exemplos de uso de PFG em sistemas multiagente podem ser fornecidos. Por exemplo, na área de telecomunicação, a união entre empresas diferentes e de distintas regiões vem possibilitando uma redução no custo de implementação de novas tecnologias e um avanço em novos mercados. Estas empresas têm um aumento no número de clientes, o que causa uma redução no número de clientes de companhias concorrentes. Assim, a formação de coalizões entre companhias exerce uma externalidade nas demais. Outro exemplo é o da empresa britânica Aerogistics, que forma coalizões entre pequenas empresas que manufaturam peças de aeronaves para que estas possam participar de grandes projetos nos quais sozinhas não teriam capacidade de participar (RAHWAN et al., 2012). O custo de padronização das peças manufaturadas depende das coalizões e da estrutura de coalizão que fazem parte do projeto.

Na proposta do uso de PFG em situações comuns no comércio virtual, descrita por (MICHALAK et al., 2009), tanto processos de formação de coalizão exógenos quanto endógenos possuem resultados favoráveis para o sistema. Em processos endógenos, um agente concentra uma grande quantidade de produtos para melhorar seu poder de barganha. De forma oposta, ao proceder nesse processo de forma exógena, quem concentra a distribuição de produtos comprados é o vendedor. No caso do trabalho de (MICHALAK et al., 2009), essa concentração é baseada em fatores geográficos. É interessante analisar estes cenários pois as características relativas à externalidade são desconhecidas dos agentes compradores. Por exemplo, um estoque limitado de produtos na loja poderia fazer com que apenas um grupo de compradores recebesse a compra na data prevista. Os agentes, ao formarem coalizões, podem não ter informações sobre esses dados e ser influenciados por outras coalizões desconhecidas.

Para um sistema centralizado que divida os agentes em coalizões, as influências da externalidade são observadas de forma direta. Porém, na vida real, um grupo de compradores não têm informações sobre outro grupo de compradores, impedindo que tais sistemas sejam abordados como centralizado. Cada comprador é um agente, que pode querer interagir com outros agentes para formar coalizões e aumentar seu poder de barganha. Obviamente, o maior poder de barganha de uma coalizão pode influenciar as demais, mas não é possível conhecer todas as estruturas de coalizões que podem ser formadas. Outro exemplo disso é quando uma agência de viagens compra muitas passagens promocionais de alguma companhia aérea. Esta agência busca ter maior poder de barganha junto aos consumidores que quiserem comprar as passagens promocionais através do controle da

maior parte dessas passagens. Porém, mesmo esta agência conhecendo todas as companhias aéreas e agências de viagem (os agentes desse sistema), não pode ter certeza que os demais agentes não irão se unir em uma grande coalizão para evitar esse que essa primeira agência tenha o maior número de passagens aéreas em promoção.

Nos cenários descritos acima, não há como um agente prever as estruturas de coalizão que serão formadas pelos demais agentes. Cada agente tem uma quantidade limitada de informação, podendo apenas inferir sobre o valor de sua coalizão e, às vezes, sobre algumas outras. Isso torna a representação centralizada desses cenários irreal e até mesmo de difícil paralelo com a realidade. É necessário que uma representação com agentes descentralizados seja produzida. Essa representação deve levar em consideração que os agentes não possuem (necessariamente) uma visão global do sistema.

Assim como em alguns cenários os agentes não conhecem todas as informações, em outros os agentes podem não querer distribuir informações com todos os agentes. De acordo com (KAZATZOPOULOS et al., 2006), sistemas centralizados possuem desvantagens ao lidar com informações de privacidade ou confidenciais. Considerando a questão de privacidade, alguns agentes em determinados cenários podem querer não fornecer toda a informação que conhecem, mesmo que isso afete a qualidade da solução do sistema. Por exemplo, quando uma reunião é marcada entre diversas pessoas, as agendas de compromissos pessoais dessas pessoas não são inteiramente compartilhadas com todos. Há um ou mais turnos de negociação, onde cada pessoa pode optar por aceitar a data prevista ou não. A criação de um grupo de pessoas que aceita uma determinada data (uma coalizão) pode trazer consequências negativas para as demais pessoas (externalidade), pois estas serão forçadas a aceitar a data mesmo não querendo. Caso todos compartilhassem todas as informações sobre suas agendas, uma solução melhor poderia ser encontrada, onde nenhuma pessoa sairia prejudicada.

Do ponto de vista econômico, há casos onde a formação de coalizões com o agente não pode ser alterada e coalizões externas à do agente independem dele. Por exemplo, quando pequenas empresas formam coalizões com grandes empresas, as pequenas empresas nem sempre tem autonomia para adicionar diferentes companhias na mesma coalizão. As pequenas empresas ficam "presas" aos acordos feitos com grandes empresas. Geralmente, a formação de coalizões distintas pode ser benéfica para a pequena empresa, mas ela não tem controle sobre essas novas coalizões (por exemplo, a união de uma empresa grande com outra menor pode trazer externalidade positiva para todas as demais empresas da estrutura de coalizão). Esse é um caso onde há diversos agentes, mas apenas alguns podem propor ou buscar novas coalizões.

Os exemplos anteriores têm em comum a ideia de que não pode haver um agente central que controle tudo. Em muitos casos, não é realístico assumir que alguém tem controle total sobre os agentes e pode definir de forma individual como estes irão se organizar. Nestes casos, é papel dos próprios agentes se regularem e encontrarem a melhor solução para os seus problemas, de forma a maximizar o desempenho do sistema quando necessário. Isso requer algoritmos que possam ser executados de forma distribuída, dividindo os custos e aceitando as particularidades de cada agente, como quantidade de recursos computacionais e preferências individuais.

Como já mencionado, algoritmos distribuídos possuem diversas vantagens em relação aos algoritmos centralizados. As divisões dos custos computacionais como memória e velocidade de processamento estão entre as principais. A adaptação às particularidades dos agentes, tais como preferência de ação, recursos existentes e objetivos próprios é outra vantagem de permitir que os agentes realizem as tarefas de forma distribuída. Mesmo

assim, ainda não há na literatura nenhum algoritmo capaz de encontrar a solução para o problema de geração de estrutura de coalizão em ambientes com externalidade que seja descentralizado. Todos os focos de pesquisa acerca de ambientes com externalidade são para desenvolvimento de algoritmos centralizados, onde um agente central possui todo o conhecimento e pode determinar a qual coalizão cada agente irá pertencer.

4.2 Calculando o valor de estruturas de coalizão em ambientes descentralizados com presença de externalidades

O algoritmo DCSGE proposto neste trabalho é dividido em duas etapas. A primeira é a avaliação dos valores das coalizões divididas em listas. Após essa avaliação, há uma troca de mensagens entre os agentes na qual informações relevantes quanto aos valores das coalizões e estruturas de coalizões analisadas são trocadas. Na segunda etapa, esses valores são utilizados para determinar quais configurações devem ser exploradas e quais outras coalizões devem ser transmitidas entre os agentes. Ao final da segunda etapa, os agentes percorrem as configurações buscando encontrar a CS^* , caso ela ainda não tenha sido encontrada. Cada etapa é detalhada a seguir.

4.2.1 Primeira etapa: cálculo distribuído dos valores das coalizões

Assim como no D-IP, cada agente inicia o processo recebendo um identificador próprio. Apesar de a primeira etapa ser similar àquela encontrada no D-IP, ela não pode prosseguir da mesma forma. Isso porque o valor das coalizões em ambientes com externalidade varia, dependendo da CS em que a coalizão está inserida. Portanto, faz-se necessário avaliar uma coalizão em relação à uma CS e deve-se fazer isso de tal forma a se obter informações relevantes ao problema e que possam ser utilizadas futuramente.

Conforme provado em (MICHALAK et al., 2008), é possível estabelecer os valores máximos e mínimos de coalizões dentro das classes PF^+ e PF^- . Uma vez que o valor é alterado apenas pela união ou divisão de outras coalizões, se a externalidade for positiva, o valor máximo de uma coalizão C será encontrado quando esta coalizão estiver em uma estrutura de coalizão de exatamente duas coalizões, ou seja, quando todos os demais agentes estiverem juntos em uma única coalizão. Por outro lado, o valor mínimo que essa mesma coalizão C pode obter é quando a CS em que está inserida contém apenas coalizões unitárias (de um agente). O Teorema 8 (apresentado por (MICHALAK et al., 2008)) apresenta estas conclusões:

Teorema 8. Considerando as configurações referentes à classe PF^- (PF^+) a coalizão C_1 tem seu valor máximo (mínimo) quando inserida na estrutura de coalizão $CS' = \{C_1, (a_1), \dots, (a_{(n-|C_1|)})\}$ e tem seu valor mínimo (máximo) quando em uma estrutura de coalizão $CS'' = \{C_1, C_{A \setminus C_1}\}$.

Tendo como base a necessidade de se conhecer os limites superiores de cada configuração, na primeira etapa os agentes devem avaliar os valores máximos e mínimos de cada lista de coalizões. Para isso, os agentes devem calcular os valores máximos e mínimos de cada coalizão, que podem ser obtidos de acordo com o Teorema 8.

A divisão das listas de coalizões em segmentos é semelhante àquela proposta em D-IP, ou seja: as listas $L_i : i \in \{1, 2, \dots, \lfloor \frac{|A|}{2} \rfloor - 1\}$ são divididas em segmentos e divididas entre os agentes, enquanto as demais listas L_{n-i} são divididas na ordem reversa. É importante utilizar a mesma divisão de segmentos que o D-IP porque ela garante que cada agente

terá informações sobre valores de estruturas de coalizão uma vez que a primeira etapa esteja terminada. Também, outra vantagem de se manter a mesma divisão é que o valor máximo (ou mínimo) de duas coalizões pode ser buscado ao mesmo tempo, pois as duas coalizões que fazem parte de CS'' podem ter seus valores avaliados simultaneamente, assim como o próprio valor de CS'' (linhas 9, 12 e 16 do Algoritmo 4.1). Por exemplo, para 5 agentes, as coalizões $C' = \{a_1, a_3\}$ e $C'' = \{a_2, a_4, a_5\}$ tem seu valor máximo (ou mínimo) definidos na $CS = \{C', C''\}$. Ao se calcular o valor de C' , é possível também calcular o valor de C'' e, somando os dois valores, o valor de CS' .

Algoritmo 4.1: Algoritmo para buscar valores máximos e mínimos para cada coalizão e o valor da melhor estrutura de coalizão.

Requer : A : o conjunto de agentes
: v : o conjunto de valores de cada coalizão
: ID : identificador global de cada agente
: L_s : lista contendo o conjunto de coalizões de tamanho s

Retorna: $MaxCS, Max_{L_s}, Valores(C, SC)$

```

1 Ordena_Agentes_Por_ID()
2 para cada  $i \in A$  faça
3   para cada  $s \in L_s$  faça
4      $Max_{L_s} = 0$ ;
5     Calcula_Tamanho_Segmento( $i, |L_s|$ );
6     para cada  $C \in L_s$  faça
7        $C' = \{A \setminus C\}$ ;
8        $CS' = \{\{C\}, \{C'\}\}$ ;
9        $CS'' = \{C, \{1\}, \{1\}, \dots, \{1\}\}$ ;
10      Calcula( $v(C, CS')$ );
11      se  $v(C, CS') > Max_{L_s}$  então
12        |  $Max_{L_s} \leftarrow V(C, CS')$ ;
13      fim se
14      Calcula( $v(C, CS'')$ );
15      se  $v(C, CS'') > Max_{L_s}$  então
16        |  $Max_{L_s} \leftarrow v(C, CS'')$ ;
17      fim se
18      Calcula( $v(CS')$ );
19      se ( $v(CS') > MaxCS$ ) então
20        |  $MaxCS \leftarrow v(CS')$ ;
21      fim se
22    fim para cada
23    Comunica( $Max_{L_s}$ )
24  fim para cada
25  Comunica( $MaxCS$ )
26 fim para cada

```

4.2.2 Segunda etapa: busca pela CS^* dentro das configurações

Tendo terminado a avaliação de todas as coalizões, torna-se necessário identificar o que deve ser transmitido aos demais agentes. Para a segunda etapa, é fundamental que

os agentes saibam os valores máximos e mínimos que cada lista pode ter (linhas 10 e 14 do Algoritmo 4.1), bem como a melhor estrutura de coalizão encontrada até então. Isso porque é possível descartar rapidamente configurações cujo valor máximo teórico é menor que o melhor valor encontrado até então. Portanto, a comunicação entre os agentes deve transmitir os valores máximos e mínimos de cada coalizão (os valores mínimos são utilizados posteriormente para inferir valores teóricos que a coalizão pode ter), bem como a melhor estrutura de coalizão encontrada por cada um (linhas 22 e 24 do Algoritmo 4.1). É importante ressaltar que, como cada agente é responsável pela avaliação de uma coalizão da lista L_1 , ao final da comunicação todos os agentes recebem todos os valores máximos e mínimos de todas as coalizões unitárias. Esses valores podem ser somados para se calcular o valor da CS formada por coalizões unitárias. Também, como há apenas uma coalizão na lista $L_{|A|}$, então essa coalizão é equivalente à CS formada por uma única coalizão, com $|C| = |A|$. Isso significa que todas as estruturas de coalizão, contendo exatamente duas coalizões, a grande coalizão (ou seja, a CS onde todos os agentes estão em exatamente uma única coalizão) e coalizões unitárias, tem seu valor calculado logo na primeira etapa do algoritmo.

Uma vez que todas as informações tenham sido trocadas entre os agentes, é verificado se a CS^* foi encontrada. Para saber se a CS^* foi encontrada, nenhuma configuração pode ter um valor de limite superior maior que a melhor CS encontrada até o momento. Caso a CS^* não tenha sido encontrada, os agentes utilizam suas informações para estabelecer o limite superior de cada configuração, já eliminando aquelas que não podem conter a CS^* . Como todos os agentes têm as mesmas informações a respeito do limite superior das listas, fazer com que todos os agentes calculem o limite superior de cada configuração criaria uma grande quantidade de computação redundante. Ao invés disso, as configurações são ordenadas e recebem um identificador, de 1 à $I(A)$, sendo $I(A)$ o número de partições inteiras para $|A|$ agentes (linha 1 do Algoritmo 4.2). Essas configurações são divididas em $\lfloor \frac{|A|}{|A|} \rfloor$ segmentos e cada segmento é designado a um agente. O agente deve calcular o limite superior da configuração e, caso esse limite seja menor que a melhor estrutura de coalizão encontrada até então, deve eliminar a configuração da lista (linhas 3 à 9 do Algoritmo 4.2).

Cada agente ordena as suas configurações remanescentes pelo valor do seu limite superior. Assim, cada agente tem uma lista de configurações onde possivelmente se encontra a CS^* . É necessário, agora, que essa lista seja transmitida aos demais agentes (linha 11 do Algoritmo 4.2). Para evitar a comunicação de todas as configurações, cada agente transmite apenas a configuração com maior limite superior. Isso não garante que as $|A|$ configurações com maior limite superior sejam de domínio de todos os agentes, pois algum agente pode ter ficado com o segmento onde as configurações mais promissoras se encontram. Portanto, após a primeira comunicação de configurações, todos os agentes que tiverem uma configuração com limite superior maior que o menor limite transmitido na comunicação anterior, transmite a sua configuração com maior valor de limite superior remanescente. Por exemplo, se há 10 agentes, na primeira comunicação são transmitidas 10 configurações e seus limites superiores. Se algum agente possui uma ou mais configurações que não foram transmitidas e cujo limite superior é maior que o limite de alguma das 10 configurações transmitidas, esse agente transmite a melhor dessas configurações aos demais. Esse processo segue até que nenhum agente possua configurações melhores que aquelas que foram transmitidas. Isso garante que as $|A|$ melhores configurações possíveis foram transmitidas a todos os agentes e, no pior caso, o número de configurações transmitidas é de $\frac{|A|*(1+|A|)}{2}$. A comunicação entre os agentes pode ocorrer mais vezes,

caso a CS^* não seja encontrada entre as configurações transmitidas. Após a comunicação, as configurações são ordenadas pelo seu valor de limite máximo e distribuídas entre os agentes (linha 1 do Algoritmo 4.3).

Algoritmo 4.2: Algoritmo para remover configurações que não podem conter a CS^* .

Requer : A : o conjunto de agentes
: β
: $Configurations$
: Max_{L_s} : o maior valor dentre todas coalizões da lista L
: $MaxCS$

Retorna: $Configurations$

```

1 Ordena_Configurations()
2 para cada config ∈ Configurations faça
3   UB_config ← 0
4   para cada CoalitionC ∈ config faça
5     UB_config ← UB_config + Max_{L_C}
6   fim para cada
7   se UB_config ≤ MaxCS * β então
8     Remover(config);
9   fim se
10 fim para cada
11 Comunica(Configurations);

```

Após cada agente transmitir as informações acima, é preciso trocar entre si os valores das coalizões que eles encontraram para cada lista que faz parte de alguma configuração. Uma vez que as listas de coalizões são exponencialmente longas, é necessário aplicar um filtro semelhante àquele utilizado no D-IP, mas levando em consideração que a externalidade produzida pela coalizão pode influenciar no resultado final da CS.

O primeiro filtro permite que sejam removidas da comunicação (e de qualquer CS) aquelas coalizões onde:

$$v_{maximo}(c) < \sum_{a_j \in c} v_{maximo}(\{a_j\}) \quad (4.1)$$

Ou seja, a soma das coalizões unitárias que compõe a coalizão c é maior que o valor da coalizão c . Porém, isso é válido apenas para a classe PF^- . É importante ressaltar que como a influência da externalidade nessas classes é ≤ 0 , essa regra é válida. Para o caso da classe PF^+ , o valor da coalizão c somado ao valor da externalidade produzida por ela pode superar o valor da soma das coalizões individuais que a compõe. Ou seja, essa regra não pode ser utilizada em simulações com externalidade positiva.

O segundo filtro permite verificar se a coalizão em questão é candidata a estar na CS^* . Esse filtro pode ser utilizado em qualquer uma das duas classes:

$$v_{maximo}(c) + UB_i - Max_{|c|} < v(CS^*_{atual}). \quad (4.2)$$

Ou seja, se for substituído no cálculo do limite superior de uma configuração o valor máximo da lista onde a coalizão c está inserida pelo valor de c e a nova soma for menor que o valor da $v(CS^*_{atual})$, a coalizão c é desconsiderada. Lembrando que os valores máximos

das coalizões estão sendo considerados e, portanto, não há nenhum cenário onde o valor da coalizão possa ultrapassar aquele considerado na Equação 4.2.

Uma vez que as coalizões que não podem fazer parte da CS^* foram removidas, as demais coalizões são transmitidas entre os agentes. O valor transmitido entre os agentes é o valor da coalizão livre de externalidade, que será o valor mínimo dela quando em um cenário com externalidade positiva ou o valor máximo dela, quando em um cenário com externalidade negativa. Com base nos dados transmitidos, os agentes podem iniciar a procura dentro do subespaço das configurações pela CS^* . Essa busca é realizada de forma idêntica àquela proposta por (RAHWAN et al., 2009), sendo apenas um agente responsável por todo um subespaço (linhas 7 a 15 do Algoritmo 4.3). Isso porque distribuir o espaço de busca entre os agentes impediria a poda dos subespaços que não podem conter a CS^* e criaria uma série de computações redundantes de valores e coalizões. Isto pode ser explicado pelo fato dos agentes não conhecerem os valores das coalizões quando inseridas em uma CS, ou seja, não possuírem informações necessárias para efetuar a poda dos subespaços a não ser que o valor da coalizão seja calculado por cada agente (o que resultaria em uma redundância desnecessária). Portanto, é importante que o agente que está efetuando a busca dentro do subespaço conheça os valores das coalizões que podem fazer parte do subespaço para assim efetuar a poda quando necessário.

Quando todas as configurações foram investigadas ou quando o valor da CS^*_{Atual} for maior que o limite superior da melhor configuração, o algoritmo para e retorna a CS^* (linhas 2, 3, 12 e 13 do Algoritmo 4.3).

Algoritmo 4.3: Algoritmo que busca dentro das configurações pela CS^* .

```

Requer :  A: o conjunto de agentes
           :   $\beta$ 
           :  Configurations
           :   $v(C, CS)$ 
           :   $Max_{CS}$ 
Retorna:   $CS^*$ 

1  Ordena_Config_UB()
2  se  $Max_{CS} \geq Max(Config_{UB}) \times \beta$  então
3  |   retorna  $Max_{CS}$ ;
4  fim se
5  para cada  $i \in A$  faça
6  |   para cada  $config \in Configurations(i)$  faça
7  |   |   para cada  $CS \in config$  faça
8  |   |   |    $v(CS) = 0$ ;
9  |   |   |   para cada  $C \in CS$  faça
10 |   |   |   |    $v(CS) \leftarrow v(CS) + v(C)$ ;
11 |   |   |   fim para cada
12 |   |   |   se  $v(CS) \geq Max(Config_{UB}) * \beta$  então
13 |   |   |   |   retorna  $v(CS)$ ;
14 |   |   |   fim se
15 |   |   fim para cada
16 |   fim para cada
17 fim para cada

```

4.3 Análise do Algoritmo

Nesta seção, é apresentada uma análise em relação a dois pontos importantes do DCSGE. O primeiro é a garantia de desempenho do algoritmo, mostrando que é possível observar a qualidade da solução encontrada até o momento sem a necessidade de se buscar todas as estruturas de coalizão possíveis. Essa é uma das características mais importante dos algoritmos recentemente projetados para resolver o problema de formação de estruturas de coalizão e permite interromper a execução do algoritmo caso uma solução suficientemente próxima do ótimo tenha sido encontrada. A outra análise que será apresentada é referente à comunicação entre os agentes e os efeitos que podem ocorrer caso ela não funcione de maneira ideal.

4.3.1 Garantia de desempenho

Depois que Sandholm e colegas provaram que é possível estabelecer um limite β do valor da estrutura de coalizão ótima ($\beta \geq \frac{V(CS_{atual}^*)}{V(CS^*)}$) sem a necessidade de pesquisar todo o espaço de busca (SANDHOLM et al., 1999), essa garantia de desempenho se tornou uma questão muito importante e presente em todas as propostas de algoritmo posteriores. Devido à natureza exponencial do problema, é importante que se possa estabelecer um limite a fim de não ser necessário buscar em todo o espaço de estados antes de se poder ter alguma garantia sobre o resultado encontrado. Além disso, nem sempre o valor ótimo é o desejado. Em muitos casos, um valor próximo do ótimo é suficiente para o usuário e a redução do tempo de execução do algoritmo é consideravelmente grande.

Com base nos teoremas 5 e 6 (Seção 3.7.2), é possível demonstrar que o algoritmo DCSGE proposto pode estabelecer um limite do ótimo de forma direta para PF^+ e pode ser facilmente alterado para fazer o mesmo em PF^- . No caso de PF^+ , ao final da primeira etapa, todos os valores de estruturas de coalizão contendo 1, 2 ou $|A|$ agentes já foram calculadas. Isso significa que todas as partições de inteiros tais que $|I_n| \leq 2$ têm seu valor conhecido e, portanto, pode-se garantir que a melhor estrutura de coalizão encontrada até então está em um limite $\beta = A$ do ótimo.

Para o caso de PF^- , é necessário modificar o DCSGE, acrescentando uma etapa adicional ao processo a fim de que se possa garantir que o resultado da primeira etapa está a um limite β do valor ótimo. Isso porque nenhuma estrutura de coalizão contendo mais de duas coalizões é investigada na primeira etapa (exceto aquela contendo exatamente $|A|$ coalizões). Para que seja possível estabelecer um limite sem que haja a necessidade de comunicação extra entre os agentes, para cada subespaço a ser investigado cada agente deverá calcular o valor da estrutura de coalizão onde a coalizão com maior cardinalidade já foi avaliada por ele. Por exemplo, na $CS = \{|A| - 3, 1, 1, 1\}$ cada agente que tiver pelo menos uma coalizão de tamanho $|A| - 3$ deve buscar o valor da CS para cada coalizão da lista $L_{|A|-3}$ que possuir. Lembrando que os subespaços que contém as CS $\{[A]; [A - 1, 1]; [1, 1, \dots, 1]\}$ não necessitam ser calculados pois os seus valores já são conhecidos.

Após a etapa inicial, é utilizado o valor de limite superior da melhor configuração para se estabelecer a qualidade da solução encontrada até o momento. A qualidade melhora à medida que $\frac{CS_N^*}{UB_{max}}$ se aproxima de 1. Assim, mesmo antes de completar o algoritmo pode-se saber qual a qualidade da solução encontrada até o momento.

É possível garantir que o algoritmo acha a CS^* , pois todas as configurações que podem conter um valor acima da CS_{atual}^* são inteiramente pesquisadas e o limite superior definido às configurações não pode ser ultrapassado em nenhum cenário, pois usa o melhor valor possível para cada coalizão. Sendo assim, não há como a configuração onde

a CS^* se encontra ser retirada do espaço de busca antes de ser avaliada. Também, não há como um agente não possuir o valor de uma coalizão necessária à formação da CS^* , pois nesse caso o valor será calculado pelo agente durante o processo de busca dentro do espaço da configuração.

4.3.2 Comunicação

As ocorrências de comunicação e o número de mensagens trocadas entre os agentes já foram discutidos na Seção 4.2. Entretanto, é necessário avaliar até que ponto a comunicação é essencial ao algoritmo e em que momentos ela se faz mais crucial.

A comunicação entre os agentes ocorre principalmente em 3 momentos. A primeira comunicação é a transmissão dos valores máximos e mínimos de cada coalizão, bem como a melhor CS encontrada até o momento. A segunda, é a comunicação das melhores configurações encontradas, onde cada agente transmite o valor máximo possível de cada configuração aos demais agentes e estas são ordenadas e divididas entre eles. Por fim, a terceira comunicação é a transmissão dos valores das coalizões necessárias para se buscar as estruturas de coalizão que fazem parte de cada configuração.

A primeira comunicação é a mais importante do algoritmo e essencial para o seu correto funcionamento. A transmissão das coalizões com melhores valores de cada lista é fundamental, pois os agentes usam esses valores para todas as demais etapas. Uma vez em posse desses valores, os agentes devem calcular os limites superiores de cada lista, de cada configuração e se uma coalizão pode ou não ser parte da CS^* . Caso haja alguma interferência, nessa primeira comunicação, as informações não transmitidas primeiramente afetam os cálculos dos valores superiores de cada lista. Um valor mais baixo que o correto pode equivocadamente ser assumido, o que fará com que o cálculo do valor máximo de uma configuração seja realizado com valores incorretos. Uma vez que esse erro pode apenas fazer os agentes assumirem valores mais baixos e nunca mais altos (já que o maior valor calculado pelo agente serve como limite superior das listas até que um valor mais alto seja comunicado para ele), os valores máximos das configurações que o agente deve calcular podem ficar abaixo da realidade e serem retiradas do espaço de busca erroneamente. Também, a utilização da Equação 4.3 (referente ao segundo filtro para transmissão dos valores das coalizões) com valores incorretos pode acarretar o envio de valores de coalizões que não podem fazer parte da CS^* . Por fim, o valor máximo possível para a CS^* pode ser calculado de forma incorreta e os agentes podem considerar alguma CS como sendo a CS^* de forma incorreta.

Durante a segunda comunicação, as melhores configurações são transmitidas entre os agentes e ordenadas de forma automática, sem necessidade de comunicação. Isso torna a comunicação nessa etapa necessária para a coordenação da busca pela CS^* dentro de cada configuração. Caso o valor de alguma configuração não seja corretamente transmitido por algum agente, a ordenação das configurações por ordem de valor é prejudicada e isso resultará em agentes com uma ordem diferente dos demais. Isso pode fazer com que mais de um agente percorra uma determinada configuração enquanto nenhum agente percorra outra. Por exemplo, a configuração listada como a quarta melhor da lista do agente com identificador 4 é percorrida por ele. Se, por algum motivo, o agente 3 não tiver recebido alguma das três melhores configurações, ele também irá buscar a CS^* na mesma configuração que o agente 4, sendo que a configuração 3 ficará sem ser investigada. Portanto, essa falha de comunicação terá como efeito uma possível redundância na busca de uma configuração e o resultado final será possivelmente menor que a estrutura de configuração ótima, tendo em vista que certas configurações não serão pesquisadas.

Por fim, a terceira comunicação afeta a busca dentro da configuração e pode impedir a localização da CS^* . Dentre as três comunicações, essa é a única onde o agente não tem como saber se houve problema na comunicação, tendo em vista que o número de transmissões recebidas não é explicitamente pré-determinado. Uma vez que apenas as coalizões que podem fazer parte da estrutura de coalizão ótima são transmitidas, uma falha na transmissão fará com que um agente desconsidere o valor daquela coalizão pelo restante do processo. Ou seja, uma coalizão que pode fazer parte de CS^* pode vir a ser ignorada e assim a estrutura de coalizão resultante do algoritmo não seria necessariamente a melhor possível.

4.4 Principais contribuições do algoritmo

Conforme descrito na Seção 3.8, não há atualmente um algoritmo capaz de encontrar a solução para o problema de geração de estruturas de coalizão em um ambiente que seja descentralizado e modelado como um PFG. Essa dificuldade advém da complexidade do problema, onde cada coalizão tem seu valor influenciado pela CS onde está inserida. Além disso, algoritmos produzidos para encontrar soluções em CFG não podem ser diretamente aplicados em PFG.

O DCSGE utiliza a segmentação das listas de coalizões proposta pelo algoritmo D-IP, modificando as atribuições de valores para as coalizões de forma que possa ser utilizado em cenários com externalidades. Essa alteração permite encontrar os valores máximos de cada coalizões e utilizá-los como base para determinar os valores máximos de cada configuração de forma distribuída. Com base na fundamentação teórica descrita em (RAHWAN et al., 2012) e no algoritmo IP_-^+ , o DCSGE traz alterações na busca pela CS^* dentro de cada configuração, levando em consideração o fator de externalidade e as limitações acerca das informações que cada agente possui.

O algoritmo DCSGE apresentado é o primeiro algoritmo na literatura capaz de encontrar a CS^* em ambientes distribuídos e cujos valores das coalizões sejam atribuídos por funções parciais. O trabalho de (RAHWAN et al., 2012) mostrou que jogos de função característica são uma instância dos PFG onde o valor da externalidade é exatamente zero ($e_c = 0$). Sendo assim, este trabalho se utiliza das principais pesquisas realizadas na área de formação de estruturas de coalizão para fornecer um algoritmo genérico que possa ser utilizado com qualquer dos dois tipos de atribuição de valores (CFG e PFG) e que é capaz de buscar estruturas de coalizão em ambientes distribuídos ou centralizados (para resolver problemas de forma centralizada, é necessário indicar para apenas um agente todos os identificadores globais, fazendo com que ele realize sozinho todos os cálculos).

5 EXPERIMENTOS E RESULTADOS

Este capítulo apresenta uma investigação a respeito do algoritmo DCSGE. Nele está descrita a metodologia utilizada para validação experimental do algoritmo, os experimentos, objetivos e resultados. São descritos e analisados os tipos de experimentos realizados, seus objetivos e seus resultados.

O capítulo encontra-se organizado em três seções. A Seção 5.1 descreve as métricas adotadas para validação; a Seção 5.2 descreve como são gerados os valores de externalidade para que sejam garantidas as propriedades das classes PF^+ e PF^- ; a Seção 5.3 descreve os parâmetros das simulações realizadas; a Seção 5.4 fornece uma análise dos resultados.

5.1 Metodologia

Sendo o DCSGE é dividido em duas etapas bem definidas (cálculos do valor máximo de cada configuração - Seção 4.3.1 - e busca pela CS^* dentro de cada configuração - Seção 4.3.2 -), é interessante analisar o impacto de cada etapa separadamente. Na primeira etapa do algoritmo, os agentes efetuam os cálculos relacionados aos valores máximos e mínimos das coalizões e buscam os valores de algumas estruturas de coalizão para as quais sabem os valores das coalizões que as compõem. Após, utilizam os valores máximos das coalizões para calcular os limites superiores das configurações. Nesta etapa do processo, são realizadas podas nas configurações, retirando aquelas cujo limite máximo teórico é menor que a melhor CS encontrada até o momento. Estas podas são um importante passo para a redução do espaço de busca do problema, e por isso é importante identificar quão grande é essa redução. Assim, para avaliarmos os efeitos da primeira etapa do algoritmo, é calculado o número de configurações que foram retiradas do espaço de busca em relação ao número total de configurações.

Na avaliação de algoritmos desenvolvidos para solucionar o problema de geração de estruturas de coalizão, uma das principais características apontadas é o número de estruturas de coalizão pesquisadas em relação ao número total de CS do problema. Como o gargalo do problema se encontra em enumerar e avaliar todas as CS, a redução do total de CS que necessitam ser avaliadas antes de se encontrar a CS^* é um critério importante nesse tipo de problema. Uma vez que o número de estruturas de coalizão cresce de forma exponencial com o número de agentes ($O(n^n)$), é importante que a quantidade de estruturas de coalizão calculadas em relação ao número total de CS diminua com o aumento do número de agentes. Isso aponta como uma das métricas de avaliação mais importantes: a relação entre o número de agentes e a porcentagem de estruturas de coalizão procuradas antes de se encontrar a CS^* . Sendo esta etapa o gargalo dos problemas de geração de estrutura de coalizão, o número de CS procuradas em relação ao número total de CS é a

métrica na qual serão focadas as análises.

5.2 Como a formação de uma coalizão influencia nas demais?

Um aspecto importante para o desenvolvimento de algoritmos a serem utilizados em PFG é como determinar o valor de uma coalizão. Esse valor é variável, determinado não apenas pela coalizão em si, mas também pelas demais coalizões que fazem parte do sistema. Também, é mandatório que o valor da coalizão seja afetado apenas pelo tipo de externalidade que se deseja, ou seja, apenas por externalidades ou positivas ou negativas. Isso significa que para quaisquer duas estruturas de coalizão CS' e CS'' , tal que CS'' foi formada por uma série de uniões de coalizões em CS' , o valor de todas as coalizões que não estiveram envolvidas nessas uniões não pode ser maior (no caso de PF^-) ou menor (no caso de PF^+) em CS'' que em CS' . (RAHWAN et al., 2012) propuseram uma equação capaz gerar valores aleatórios para as coalizões nas classes PF^- e PF^+ .

Para essa equação, assume-se que cada agente em cada coalizão é ordenado em ordem crescente e que cada coalizão em um CS também é ordenada de forma crescente, sendo essa ordem baseada no menor índice dentre os agentes que fazem parte dessa coalizão. Assume-se também que $l(a_i, c)$ seja a posição do agente i na coalizão c e $l(a_i, CS)$ seja a localização da coalizão que contém o agente i dentro da CS. Por exemplo, $l(a_3, \{a_1, a_3, a_5\}) = 2$ e $l(a_3, \{\{a_1, a_3\}, \{a_2, a_4\}\}) = 1$. Tendo essas definições por base, primeiramente são gerados valores aleatórios e positivos para as variáveis v_c , e_c e $e_{c,1}, e_{c,2}, \dots, e_{c,|c|}$ tal que $\sum_{j=1}^{|c|} e_{c,j} = e_c$. Com isso, o valor de c para qualquer CS a que pertença é dado por:

$$\omega(c, CS) = v_c - (+) \sum_{a_i \in \bar{c}} e_{c,l(a_i, \bar{c})} * \left(1 - \frac{l(a_i, CS\{c\}) - 1}{|\bar{c}|}\right). \quad (5.1)$$

A variável v_c representa o valor da coalizão sem o efeito da externalidade. O restante da equação representa a ação da externalidade sobre o valor da coalizão em uma determinada estrutura de coalizão. A variável e_c representa o máximo que a externalidade pode influenciar no valor de uma coalizão. Por exemplo, se queremos que a externalidade não influencie mais que 50% o valor da coalizão, basta previamente assumir que $e_c = 0.5 * v_c$. O valor de e_c não é fixo em 50%, mas sim limitado a ele. O valor deve ser calculado de forma individual para cada coalizão em cada CS. Ambas variáveis v_c e e_c não necessitam serem previamente definidas ou restringidas, podendo ser sorteadas de uma distribuição estatística qualquer.

5.3 Experimentos

Para validação e experimentação do algoritmo, foram realizadas simulações com diferentes números de agentes: de 10 a 16 agentes. A limitação em 16 agentes se deve ao tempo necessário para realizar cada simulação. Mesmo sendo possível utilizar mais agentes para as simulações, um dos principais problemas encontrados nas pesquisas de geração de estrutura de coalizão é o número exponencial de coalizões em relação ao número de agentes. Esse fator impede que simulações com um número elevado de agentes sejam realizadas.

As simulações foram realizadas tanto na classe de problemas PF^+ quanto PF^- . O objetivo foi avaliar o número de configurações removidas pela etapa um e o número de CS buscadas antes de se encontrar a CS^* em diferentes situações.

Os experimentos demonstram a influência que a externalidade negativa e positiva tem sobre resultados coletados. Além da externalidade, também são comparados os valores obtidos quando se busca a estrutura de coalizão ótima com valores de β iguais à 1 e 0.98.

Os valores das coalizões durante as simulações foram atribuídos utilizando às distribuições estatísticas uniforme e normal, de comum uso nessa área (SANDHOLM et al., 1999; RAHWAN et al., 2009). Na distribuição uniforme, os valores das coalizões podem variar entre 0 e $|C|$, enquanto na distribuição normal os valores variam entre 0 até $|C| \times N(\mu, \sigma^2)$, sendo $\mu = 1$ e $\sigma^2 = 0, 1$. Esses valores são influenciados pela externalidade presente na simulação, dependendo da CS onde se encontram. A externalidade pode assumir qualquer valor ou mesmo ser completamente aleatória. Para efeitos de simulação, utilizamos intervalos curtos de 5% para os valores de externalidade. O valor máximo da externalidade será 100% ($e_c = 1$), o que indica que uma coalizão pode, no máximo, dobrar seu valor (no caso de PF^+) ou ter seu valor igualado à zero (no caso de PF^-) como resultado da externalidade.

Conforme descrito no Seção 5.2, o valor de e_c determina a influência máxima que a externalidade pode ter sobre uma coalizão. Essa influência indica em quanto por cento o valor da coalizão irá ser alterado (Equação 5.1). Para cada coalizão dentro de cada estrutura de coalizão, é atribuído um valor de e_c com base em uma distribuição uniforme, variando de 0 a $v_C \times (x/100)$, sendo o x o valor máximo que a externalidade pode ter.

Todos os resultados apresentados são a média de 50 repetições.

5.4 Resultados

Nesta seção são apresentados os resultados das simulações realizadas com o algoritmo DCSGE. A Seção está dividida em duas subseções. A Seção 5.4.1 apresenta os resultados referentes aos experimentos em condições ideais, onde o objetivo é validar o algoritmo proposto frente às métricas mencionadas na Seção 5.3. A Seção 5.4.2 apresenta experimentos onde a comunicação entre os agentes não ocorre em condições ideais, ou seja, onde há perda de informação. O objetivo desta seção é avaliar os efeitos que a comunicação possui em relação ao resultado encontrado pelo algoritmo.

5.4.1 Experimentos em condições ideais

Inicialmente, são analisados os experimentos relativos à primeira etapa do algoritmo. Devido ao número elevado de combinações de parâmetros nas simulações, são apresentados apenas os gráficos dos resultados de algumas simulações. Por exemplo, 14 agentes, $\beta = 1$, PF^+ , $e_c = 0, 1$ e distribuição uniforme é apenas um exemplo de combinação de valores de parâmetros. Para cada aspecto analisado é apresentada uma figura contendo os resultados da simulação. A escolha de quais parâmetros apresentar graficamente neste trabalho foi feita considerando as simulações que melhor demonstram o comportamento do algoritmo. Todos os resultados para todas as simulações estão disponíveis no Anexo A deste trabalho.

A Figura 5.1 mostra a porcentagem de configurações que foram removidas do espaço de busca após a primeira etapa do DCSGE, considerando os dois tipos de distribuições. Pode-se perceber que, com a distribuição uniforme, o número de configurações removidas é maior. A explicação encontrada é que a externalidade acentua a diferença de valores entre as coalizões com maior valor e menor valor. Assim, a melhor estrutura de coalizão encontrada na primeira etapa possui um valor maior na distribuição uniforme do que na distribuição normal, já que a distribuição uniforme garante um maior número de coalizões

com valores altos ou baixos do que a distribuição normal. Nessa simulação, os valores de β e e_c são, respectivamente, 1 e $-0,1$.

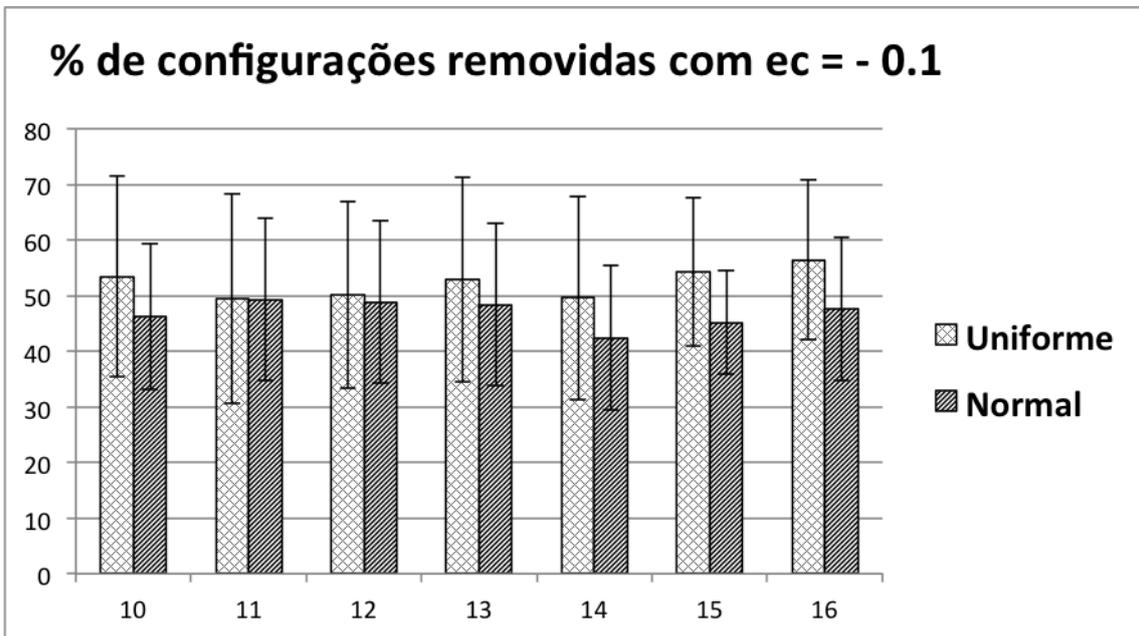


Figura 5.1: Porcentagem de configurações removidas considerando distribuição normal e uniforme em relação ao número de agentes e com $\beta = 1$.

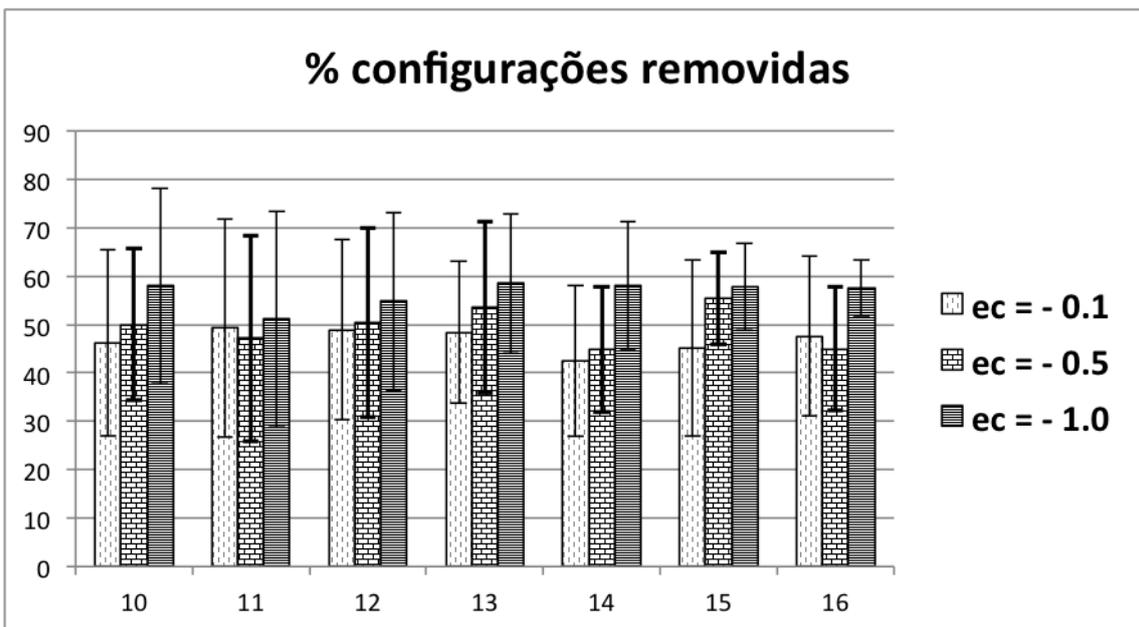


Figura 5.2: Porcentagem de configurações removidas considerando o valor da externalidade para distribuição normal com $\beta = 0.98$.

A Figura 5.2 mostra como a remoção de configurações varia quando variamos o valor da externalidade. Nessa simulação, foi utilizada uma externalidade negativa, com $e_c \in \{0, 1; 0, 5; 1\}$, $\beta = 0.98$ e distribuição normal. O efeito da externalidade torna-se ainda mais visível nesse experimento. Uma externalidade mais alta cria uma grande diferença

entre os valores das coalizões, tornando a remoção de configurações onde não é possível encontrar a CS^* mais fácil e abrangente.

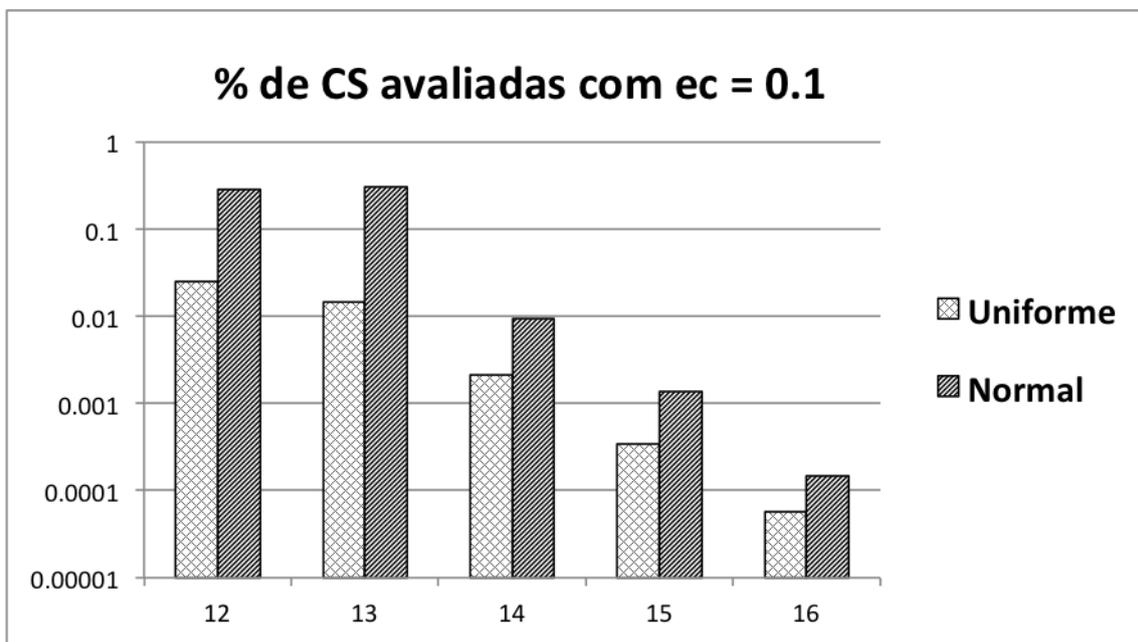


Figura 5.3: Porcentagem de estruturas de coalizão avaliadas considerando ambas funções de distribuição de valores e $\beta = 1$ (em escala logarítmica).

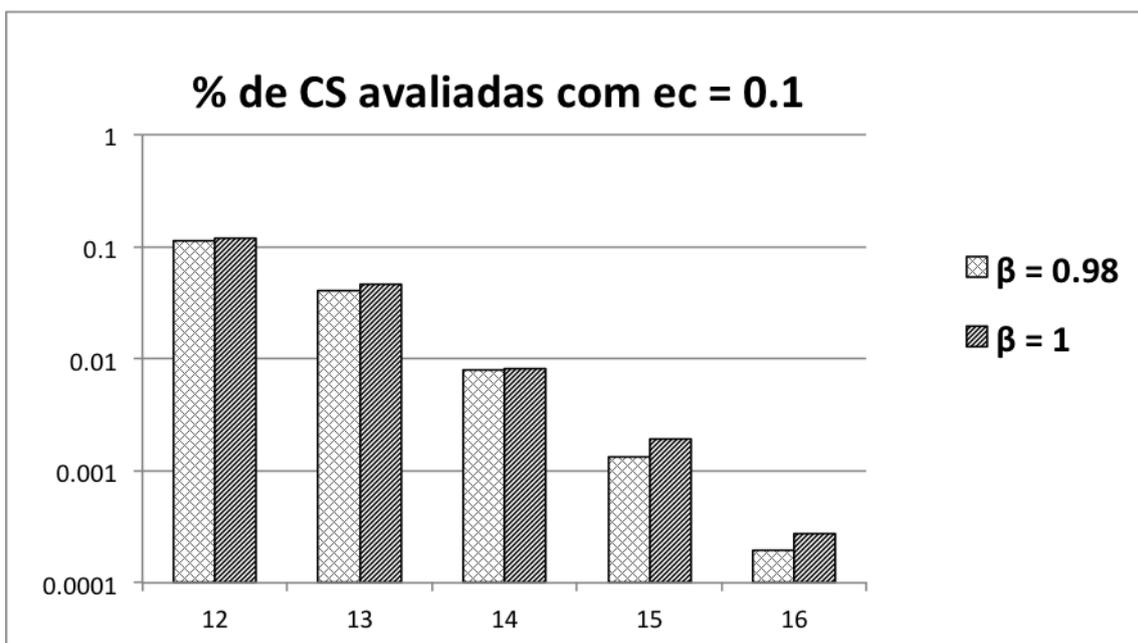


Figura 5.4: Comparação da porcentagem do número de estruturas de coalizão avaliadas com diferentes números de agentes utilizando dois valores de β e distribuição uniforme (em escala logarítmica).

Os experimentos referentes ao número de estruturas de coalizão calculadas demonstram a eficiência do algoritmo. A Figura 5.3 mostra a porcentagem de estruturas de coalizão procuradas antes de se encontrar a CS^* . Nessa simulação, foi utilizada uma ex-

ternalidade positiva com $e_c = 0.1$ e $\beta = 1$. Como demonstrado, nas simulações onde a distribuição uniforme foi utilizada, se tornou necessário procurar em um espaço menor para encontrar a solução. Também se pode verificar que, com o aumento do número de agentes, há uma diminuição na porcentagem de espaço procurado. Isso demonstra que o número de CSs procuradas não cresce na mesma proporção que o número total de estruturas de coalizão do problema.

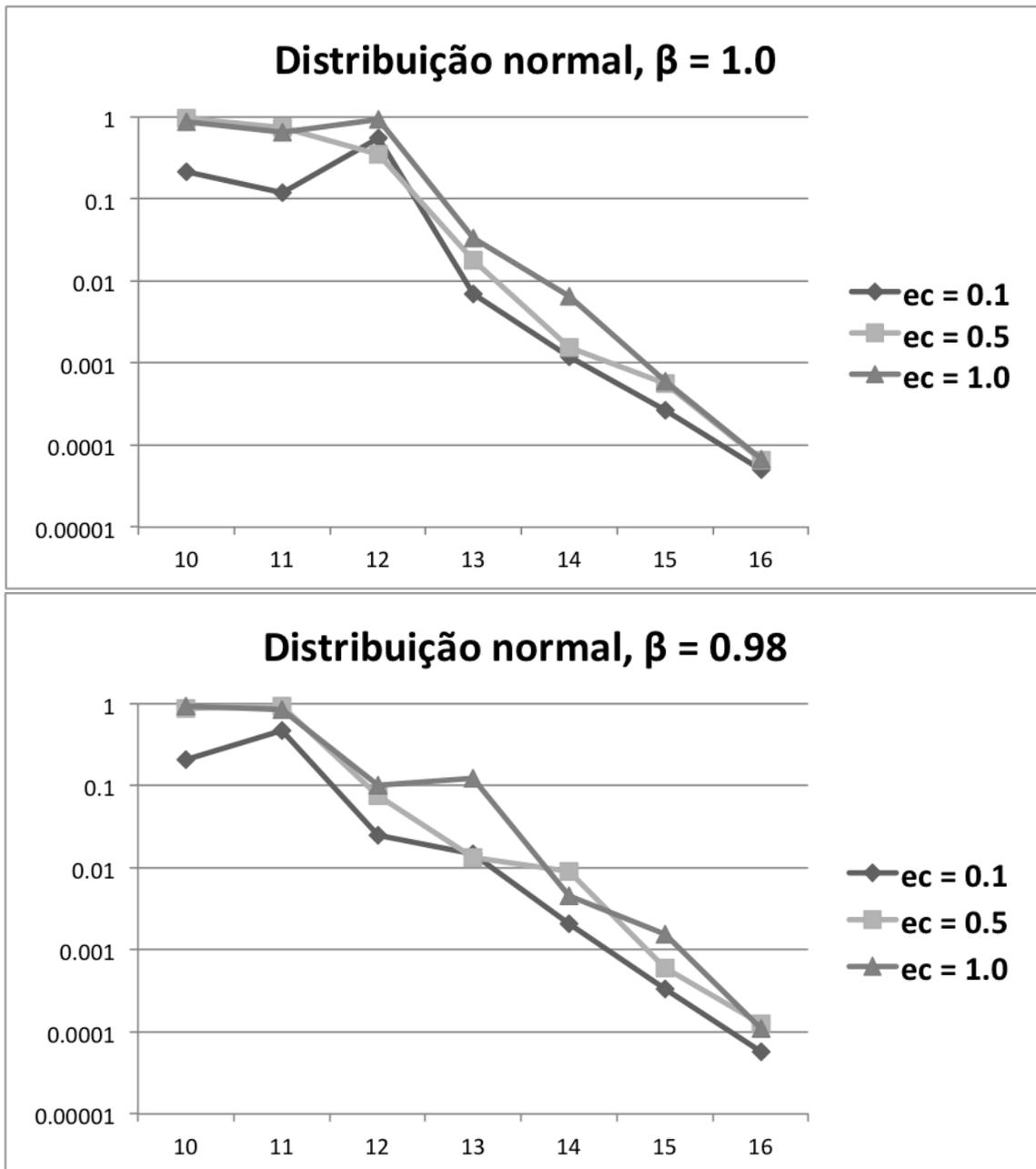


Figura 5.5: Variação da porcentagem do número de estruturas de coalizão avaliadas com diferente número de agentes e valores de externalidade.

Para analisar os efeitos de se buscar a CS^* com diferentes valores para β , foram realizadas simulações com distribuição uniforme, $e_c = 0, 1$ e $\beta = \{0.98; 1.0\}$. A Figura 5.4 mostra os resultados. Quando β possui um valor menor que 1, não se faz necessário encontrar a melhor estrutura de coalizão. É possível interromper a simulação quando o

valor da CS_N^* estiver perto do ótimo. Sendo assim, o número de CS procuradas para $\beta = 0.98$ foi menor que para $\beta = 1.0$. A Figura 5.5 mostra a comparação da porcentagem de CS avaliadas em relação ao β para cada número de agentes (de 10 a 16) e diferentes valores de externalidade ($e_c = 0, 1$, $e_c = 0, 5$ e $e_c = 1$). O comportamento anômalo dos resultados para 11 e 12 agentes é um comportamento recorrente do problema de formação de estruturas de coalizão. Durante as simulações realizadas não foi encontrada uma explicação satisfatória para esse comportamento.

Por fim, a última comparação realizada não é apenas relacionada ao algoritmo, mas também sobre os efeitos de cada externalidade. A Figura 5.6 mostra a porcentagem de configurações podadas para a mesma instância do problema (mesmos valores para β , e_c e distribuição) porém com externalidades opostas. É interessante observar que a externalidade negativa retira muito mais configurações do espaço de busca do que a externalidade positiva. Como resultado, a busca pela CS^* é mais rápida no segundo estágio do algoritmo. Isso ocorre porque na distribuição normal utilizada nos experimentos a maior parte das coalizões se encontra na região central da distribuição. Assim, essa região tem mais coalizões afetadas pela externalidade que as demais. Para externalidade negativa, a medida que os valores das coalizões ficam mais baixos, os poucos valores que permanecem altos possuem uma grande probabilidade de estar na CS^* , podendo-se eliminar uma grande parte das configurações. Quando a externalidade é positiva os valores das coalizões aumentam, tornando mais difícil podar configurações e as estruturas de coalizão. A Figura 5.7 mostra a porcentagem do espaço de estruturas de coalizão que foi buscado para encontrar a CS^* com 14 agentes, $\beta = 1$ e distribuição normal.

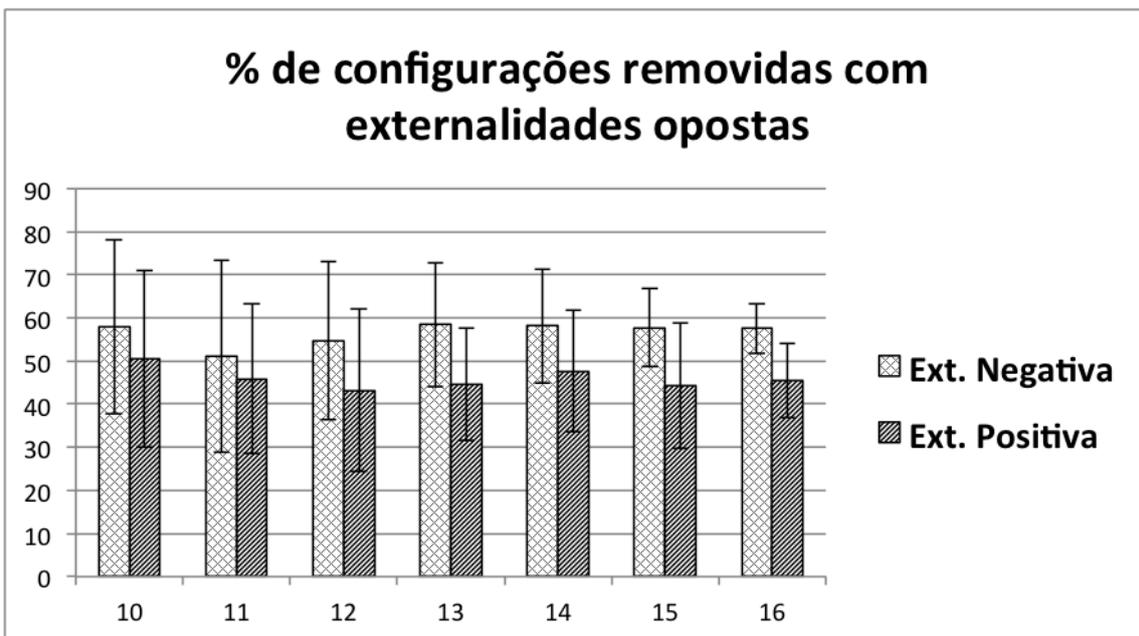


Figura 5.6: Comparação dos resultados da porcentagem de configurações removidas para valores opostos de externalidade e diferente número de agentes. Distribuição normal com $\beta = 0.98$ e $e_c = 1.0$

Não é possível comparar os resultados obtidos aqui com os resultados dos algoritmos D-IP e IP_-^+ por não se ter acesso à eles.

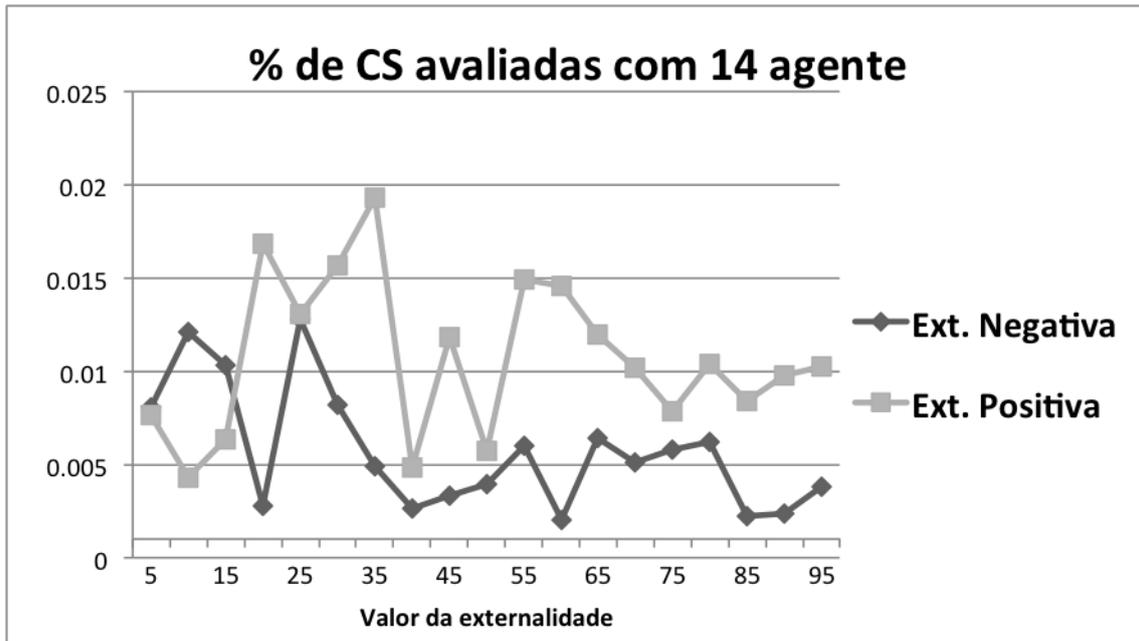


Figura 5.7: Comparação dos resultados da porcentagem do número de estruturas de coalizão avaliadas para diferentes valores de externalidade e 14 agentes, com $\beta = 1$ e distribuição normal.

5.4.2 Experimentos com falhas de comunicação

Os experimentos e resultados descritos nesta seção demonstram o efeito de falhas de comunicação na execução do algoritmo e no resultado final por ele encontrado. Assim como na Seção 5.4.1, apenas alguns experimentos serão demonstrados graficamente devido ao alto número de combinações de parâmetros, mas é possível consultar todos os resultados no Apêndice A. Para todos os experimentos demonstrados nessa seção, $\beta = 1$ e $e_c = 1.0$.

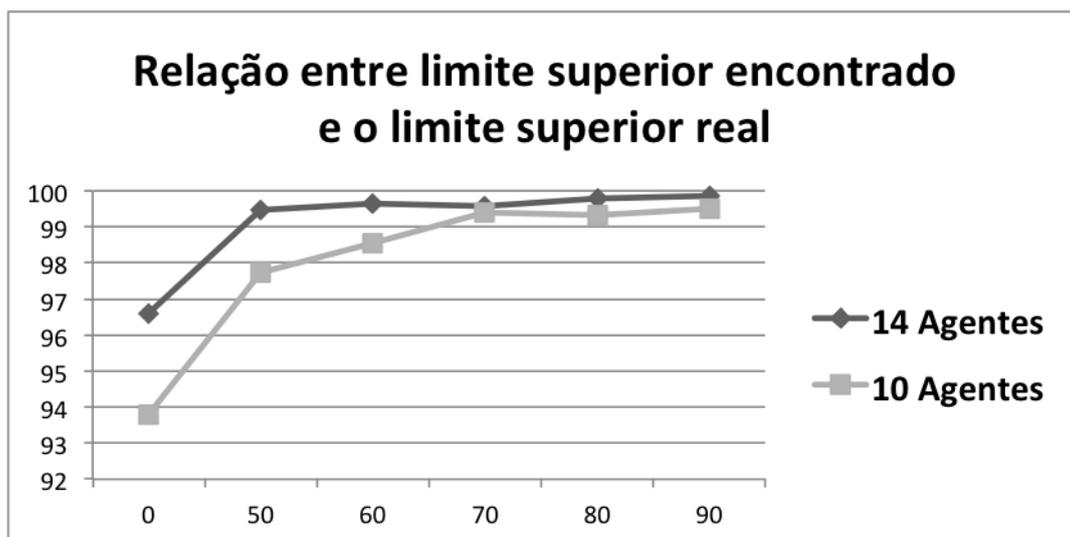


Figura 5.8: Relação entre o maior limite superior encontrado pelos agentes e o maior limite superior real em relação ao τ , utilizando-se a distribuição uniforme com $e_c = 1.0$ e $\beta = 1$.

Para explorar os efeitos da falha de comunicação são propostos cinco cenários, onde o que varia é a porcentagem do total de informação transmitida que é recebida pelos demais agentes. A quantidade de informação recebida é representada por τ e pode ter os seguintes valores: $\{0; 50; 60; 70; 80; 90\}$. No cenário onde $\tau = 0$, nenhuma comunicação enviada é recebida (0%), ou seja, não há qualquer tipo de comunicação entre os agentes e eles operam apenas com os valores de coalizões calculados por eles.

Primeiramente é demonstrado o efeito que a falha na comunicação pode provocar na primeira parte do algoritmo DCSGE, onde os limites superiores de cada configuração são estabelecidos. Estes limites definem quais configurações devem ser investigadas primeiro e também definem o limite máximo hipotético que a CS^* pode ter.

A Figura 5.8 demonstra o maior limite superior obtido por uma configuração para cada valor de τ em relação ao maior limite real que poderia ser obtido em condições perfeitas de comunicação, utilizando-se distribuição uniforme. Quando não há comunicação, o valor máximo obtido é muito abaixo do valor máximo real. À medida que a comunicação melhora, também melhora a relação entre o valor obtido e aquele que poderia ser obtido.

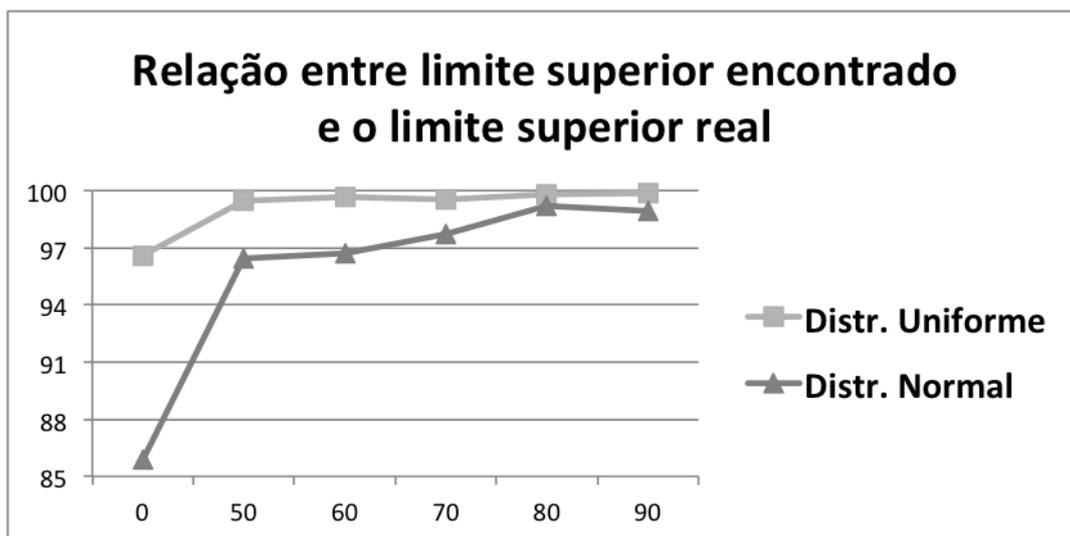


Figura 5.9: Comparação da relação entre o maior limite superior encontrado por diferentes tipos de distribuição e o maior limite superior real em relação ao τ , utilizando-se 14 agentes com $e_c = 1.0$ e $\beta = 1$.

Mesmo no caso de $\tau = 0$ o valor encontrado pelos agentes ultrapassa 90% do valor encontrado com comunicação sem falhas. Esse valor é ainda maior com 14 agentes que com 10 agentes. Isso porque as coalizões com alta cardinalidade possuem uma maior probabilidade de possuírem um maior valor que aquelas com menor cardinalidade (o valor de uma coalizão é fornecido pela distribuição e pelo número de agentes que ela possui). Com isto, é provável que coalizões com maior número de agentes tenham um maior impacto no valor da CS^* . Isso indica que a pequena diferença entre o maior limite da melhor configuração encontrado e seu valor real seja resultado de divisões ou fusões de coalizões pequenas, cujo impacto no valor final da CS^* não é tão representativo. No caso de 14 agentes, cada agente possui um número maior de coalizões a seu dispor para calcular o limite superior das configurações, tornando o valor encontrado ainda mais próximo do real.

A Figura 5.9 demonstra a diferença dos valores encontrados para o maior limite superior das configurações em relação aos limites reais para os dois tipos de distribuição.

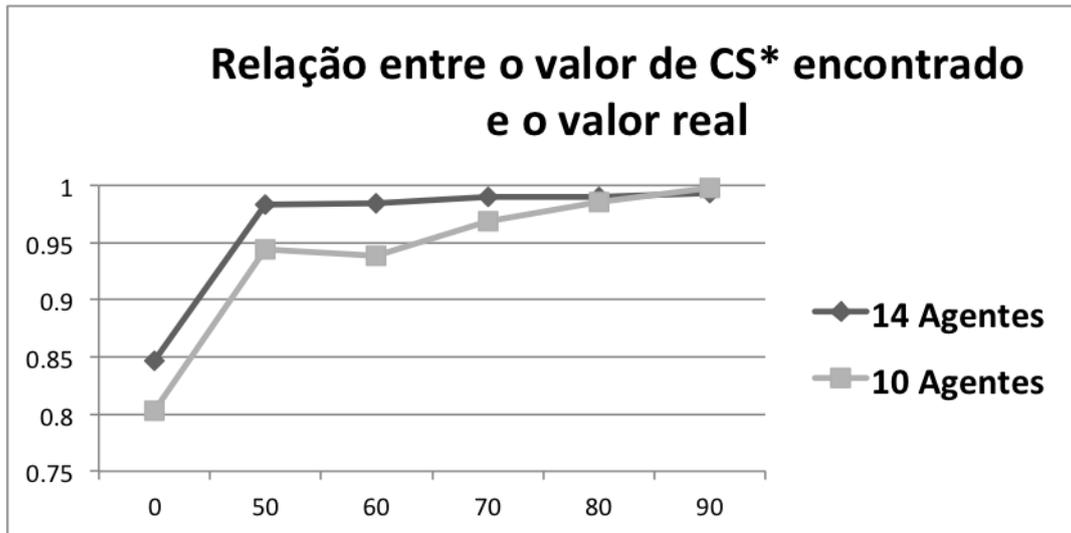


Figura 5.10: Comparação da relação entre a CS^* encontrada pelos agentes e o valor real da CS^* em relação ao τ , utilizando-se a distribuição uniforme com $e_c = 1$ e $\beta = 1$.

Como mencionado na Seção 5.4.1, a distribuição uniforme garante um maior número de coalizões com valores mais altos que a distribuição normal. Como os agentes não possuem acesso a uma grande quantidade de valores de coalizões, quando não há uma coalizões que exerça grande influência no valor da CS^* é necessário que o agente utilize os valores de diversas coalizões para encontrar a configuração com maior limite superior. Assim, a configuração com maior limite superior na distribuição normal é uma configuração com um maior número de coalizões que na distribuição uniforme (onde uma única coalizão pode representar grande parte do valor da CS^*).

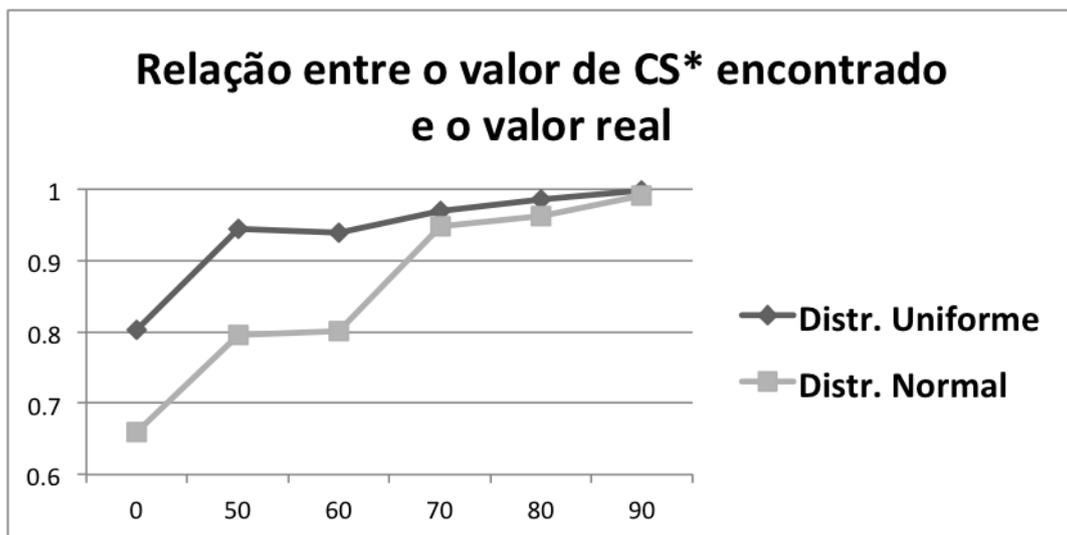


Figura 5.11: Comparação da relação entre a CS^* encontrada por diferentes distribuições e o valor real da CS^* em relação ao τ , utilizando-se 10 agentes com $e_c = 1$ e $\beta = 1$.

Conforme discutido na Seção 4.3.2, a falha na comunicação pode resultar em um valor incorreto para a CS^* . Isso porque as configurações podem ter seus limites superiores calculados de forma imprecisa (como demonstrado nas figuras 5.8 e 5.9), a segmentação das configurações entre os agentes pode ocorrer de forma incorreta ou mesmo os agentes

podem não receber os valores das coalizões transmitidos pelos demais agentes. A Figura 5.10 demonstra a relação entre o valor encontrado para a CS^* e seu valor verdadeiro, utilizando-se distribuição uniforme. Os agentes erroneamente atribuíram uma estrutura de coalizão com um valor abaixo do máximo à CS^* . À medida que a quantidade de informações dos agentes aumenta mais se aproxima o resultado encontrado do verdadeiro.

A última figura (Figura 5.11) mostra uma comparação entre os valores encontrados pelas distribuições uniforme e normal considerando 10 agentes. A falta de informações referentes aos valores das coalizões causa um prejuízo maior nos experimentos com distribuição normal que naqueles com distribuição uniforme. Conforme mencionado anteriormente, os agentes na distribuição normal necessitam de uma quantidade maior de valores de coalizões para produzirem boas aproximações do valor real da CS^* . Por outro lado, na distribuição uniforme os agentes podem conseguir boas aproximações utilizando apenas uma coalizão com um elevado número de agentes.

6 CONCLUSÕES

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

6.1 Conclusões

O problema de encontrar a estrutura de coalizão ótima (CS^*) recebe uma crescente atenção dos pesquisadores na área de inteligência artificial. Devido ao problema ser de complexidade exponencial, não é suficiente apenas encontrar uma solução. É necessário verificar a qualidade da solução encontrada em relação àquela requerida e poder fazer isso sem a necessidade de explorar todo espaço de busca. A complexidade do problema impõe diversas restrições, como por exemplo, o número limite de agentes com os quais se pode realizar simulações e as formas para estabelecer os valores das coalizões.

Diversos pesquisadores apresentam trabalhos que tiveram grande influência sobre a área. (SANDHOLM et al., 1999) apresentou um trabalho onde prova que o problema de geração de estruturas de coalizão é NP-Completo e que é possível estabelecer um valor limite do valor ótimo. Posteriormente, (RAHWAN et al., 2007a) desenvolveu uma nova forma de representar o espaço de busca do problema, dividindo-o em configurações. Ademais, ao contrário das pesquisas anteriores que buscavam diretamente os valores das CS, foi apresentado um algoritmo cuja entrada são os valores das coalizões. Essa representação e abordagem são ainda hoje utilizados nos trabalhos que tem por objetivo encontrar a CS^* . Em 2012 (RAHWAN et al., 2012) apresentou um estudo computacional acerca de formação de estruturas de coalizão quando considerado como PFG. Nesse trabalho, foi demonstrado que é possível estabelecer um limite do valor ótimo mesmo com a presença de externalidades e que é possível estabelecer os limites superiores e inferiores de cada coalizão (e, conseqüentemente, de cada configuração).

Este trabalho apresentou um estudo para tratar o problema de formação de estruturas de coalizão em PFG de forma distribuída. Até onde pode ser verificado, trata-se de um trabalho pioneiro e o algoritmo DCSGE foi apresentado juntamente com resultados de experimentos realizados. Conforme demonstrado, o DCSGE consegue garantidamente encontrar a solução ótima do problema, bem como soluções sub-ótimas com limite quando requerido. O algoritmo é capaz de excluir quase 50% das configurações do espaço de busca, deixando apenas aquelas que possuem um valor máximo teórico maior que a melhor estrutura de coalizão encontrada até então. Para todos os casos, menos do que 1% das CS foram pesquisadas. Para 16 agentes, só foi necessário avaliar 0.0001% das CS, o que representa uma redução considerável no espaço de busca. Também foi demonstrado o efeito de perda de comunicação entre os agentes e como isso influencia no valor da CS^* .

Quanto às aplicações desse trabalho, o DCSGE pode ser utilizado diretamente em

qualquer problema modelado como CFG ou PFG, porque CFG é um caso particular de PFG onde a externalidade é exatamente 0. Apesar de o algoritmo ser projetado para ambientes distribuídos, ele pode ser facilmente adaptado para cenários centralizados. Para tal, é necessário associar a um agente n valores de identificadores globais, fazendo com que ele simule a computação que todos os demais agente fariam. O restante do algoritmo permanece inalterado.

6.2 Trabalhos Futuros

Este trabalho mostrou um algoritmo capaz de encontrar a CS^* de forma distribuída e em PFG. Apesar de o algoritmo ter apresentado um resultado positivo, muitas melhorias para a abordagem do problema podem ser consideradas. Entre elas, aquela que parece ser mais promissora é limitar a influência da externalidade a um valor X ($e_c \leq X$) e utilizar essa limitação para aprimorar o cálculo do valor esperado de uma coalizão dentro de um CS. Para tanto, seria necessário conhecer quantas coalizões possui em uma configuração e utilizar esse valor para determinar o limite superior (e inferior) que cada coalizão pode ter. Isso resultaria em um aumento significativo na quantidade de informação que os agentes possuem e, como consequência, uma redução significativa no número de possíveis CS candidatas à CS^* .

Outra importante melhoria do algoritmo pode ser realizada utilizando um método mais eficiente para buscar CS dentro de configurações. O método utilizado nesse trabalho (método derivado do trabalho de (RAHWAN et al., 2009)) é muito ineficiente quando comparado com outros, como o de (RAHWAN et al., 2012). Porém, como os valores das coalizões não são fixos e os agentes não possuem todas as informações sobre esses valores, é difícil adaptar métodos mais eficientes para este tipo de situação.

REFERÊNCIAS

CATILINA, E.; FEINBERG, R. Market power and incentives to form research consortia. *Review of Industrial Organization*, Kluwer Academic Publishers, v. 28, n. 2, p. 129–144, 2006.

CHVÁTAL, V. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, INFORMS, v. 4, n. 3, p. 233–235, 1979.

DANG, V. D.; JENNINGS, N. R. Generating coalition structures with finite bound from the optimal guarantees. In: *INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS*. New York, USA: IEEE Computer Society: Washington, USA, 2004. p. 564–571.

DO, K. P.; NORDE, H. *The Shapley Value for Partition Function Form Games*. Tilburg: Center, 2007. v. 9, n. 2, 353-360 p.

FINUS, M.; RUNDSHAGEN, B. Endogenous coalition formation in global pollution control: a partition function approach. In: *The Endogenous Formation of Economic Coalitions*. Cheltenham: Edward Elgar, 2003, (Fondazione Eni Enrico Mattei Series on Economics and the environment). p. 199–243.

HORLING, B.; LESSER, V. A survey of multi-agent organizational paradigms. *Knowledge Engineering Review*, v. 19, n. 4, p. 281–316, 2005.

KAZATZOPOULOS, L. et al. An incentive-based architecture to enable privacy in dynamic environments. *Internet Research*, v. 16, n. 2, p. 171–189, 2006.

KOCZY, L. A. A recursive core for partition function form games. *Theory and Decision*, Springer Science Business Media, LLC, v. 63, n. 1, p. 41–51, 2006.

MICHALAK, T. et al. Optimal coalition structure generation in partition function games. In: *EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2008. p. 388–392.

MICHALAK, T. et al. A distributed algorithm for anytime coalition structure generation. In: *International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2010. p. 1007–1014.

MICHALAK, T. et al. Exogenous coalition formation in the e-marketplace based on geographical proximity. *Electron. Commer. Rec. Appl.*, Elsevier Science Publishers B. V.,

Amsterdam, The Netherlands, The Netherlands, v. 8, n. 4, p. 203–223, jul. 2009. ISSN 1567-4223.

RAHWAN, T.; JENNINGS, N. R. An algorithm for distributing coalitional value calculations among cooperating agents. *Journal of Artificial Intelligence*, v. 8, n. 171, p. 535–567, 2007.

RAHWAN, T. et al. Anytime coalition structure generation in multi-agent systems with positive or negative externalities. *Artificial Intelligence*, Elsevier Science Publishers Ltd., Essex, UK, v. 186, p. 95–122, jul. 2012.

RAHWAN, T. et al. An anytime algorithm for optimal coalition structure generation. *Journal of Artificial Intelligence Research (JAIR)*, v. 34, p. 521–567, April 2009.

RAHWAN, T. et al. Anytime optimal coalition structure generation. In: *National Conference on Artificial Intelligence*. Vancouver, Canada: AAAI: Menlo Park, CA, 2007. p. 1184–1190.

RAHWAN, T. et al. Near-optimal anytime coalition structure generation. In: *International Joint Conference on Artificial Intelligence*. Hyderabad, India: [s.n.], 2007. p. 2365–2371.

SANDHOLM, T. et al. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, Elsevier Science Publishers Ltd., Essex, UK, v. 1, n. 111, p. 209–238, 1999.

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

YEH, D. Y. A dynamic programming approach to the complete set partitioning problem. *BIT Numerical Mathematics*, BIT Computer Science and Numerical Mathematics, Lawrence, KS, USA, v. 26, n. 4, p. 467–474, 1986. ISSN 0006-3835.

ApêndiceA

A.1 Resultados completos considerando $\beta = 1$

A.1.1 Porcentagem de configurações removidas

A.1.1.1 Distribuição Uniforme

Externalidade Positiva

<i>Número de Agentes</i>	$e_c = 0.1$	$e_c = 0.5$	$e_c = 1.0$
10	40,20	39,07	40,73
11	35,57	40,57	39,90
12	41,07	42,25	35,00
13	44,07	42,07	40,55
14	40,67	43,15	40,14
15	43,82	41,59	48,19
16	46,45	46,10	51,56

Externalidade Negativa

<i>Número de Agentes</i>	$e_c = -0.1$	$e_c = -0.5$	$e_c = -1.0$
10	44,00	53,33	60,33
11	43,38	60,67	69,10
12	44,39	61,04	70,57
13	50,10	61,77	74,00
14	49,11	62,15	75,50
15	52,29	52,30	52,30
16	57,24	65,34	82,33

A.1.1.2 *Distribuição Normal****Externalidade Positiva***

<i>Número de Agentes</i>	$e_c = 0.1$	$e_c = 0.5$	$e_c = 1.0$
10	46,07	45,00	48,67
11	42,00	47,61	45,95
12	46,13	47,32	49,07
13	42,44	44,67	47,35
14	46,95	48,30	44,71
15	47,52	43,23	45,57
16	42,60	48,33	48,19

Externalidade Negativa

<i>Número de Agentes</i>	$e_c = -0.1$	$e_c = -0.5$	$e_c = -1.0$
10	45,10	51,67	59,17
11	46,33	53,90	60,10
12	46,61	55,70	62,68
13	49,25	57,67	66,25
14	45,75	53,49	66,83
15	48,73	53,83	55,02
16	52,43	55,17	69,95

A.1.2 Número de estruturas de coalizão avaliadas em relação ao número total de estruturas de coalizão do problema

A.1.2.1 Distribuição Uniforme

Externalidade Positiva

<i>Número de Agentes</i>	$e_c = 0.1$	$e_c = 0.5$	$e_c = 1.0$
10	0,212743436	0,975727872	0,875100000
11	0,119833503	0,735474218	0,652765581
12	0,556838896	0,345983847	0,940560000
13	0,007054789	0,017742820	0,033755398
14	0,001184236	0,001543045	0,006641137
15	0,000263620	0,000569330	0,000602146
16	5,01754E-05	6,61503E-05	6,84984E-05

Externalidade Negativa

<i>Número de Agentes</i>	$e_c = -0.1$	$e_c = -0.5$	$e_c = -1.0$
10	0,177869196	0,154062341	0,370519853
11	0,044554106	0,015867427	0,094958309
12	0,034429249	0,005690000	0,103400000
13	0,006516544	0,004319026	0,000731667
14	0,000836007	0,000165346	8,07432E-05
15	0,000128986	3,36892E-05	1,46456E-05
16	3,67833E-05	1,5311E-05	3,31692E-06

A.1.2.2 Distribuição Normal

Externalidade Positiva

<i>Número de Agentes</i>	$e_c = 0.1$	$e_c = 0.5$	$e_c = 1.0$
10	0,371823065	0,281050916	0,501238000
11	0,823470000	0,969994194	0,982992499
12	0,288873151	0,034045775	0,119873951
13	0,088184999	0,103927920	0,046505854
14	0,003180654	0,017122575	0,008140337
15	0,000297185	0,001803257	0,001927946
16	0,000107322	7,42509E-05	0,000273882

Externalidade Negativa

<i>Número de Agentes</i>	$e_c = -0.1$	$e_c = -0.5$	$e_c = -1.0$
10	0,496431472	0,254044579	0,267624143
11	0,250904122	0,268818987	0,457029385
12	0,412013675	0,148123791	0,205148984
13	0,106647010	0,025729671	0,037090613
14	0,003286323	0,001915388	0,003050450
15	0,000506726	0,000847886	0,000628572
16	5,71497E-05	0,000105872	8,24991E-05

A.2 Resultados completos considerando $\beta = 0.98$

A.2.1 Porcentagem de configurações removidas

A.2.1.1 Distribuição Uniforme

Externalidade Positiva

<i>Número de Agentes</i>	$e_c = 0.1$	$e_c = 0.5$	$e_c = 1.0$
10	41,40	36,53	37,80
11	44,62	45,38	39,33
12	44,55	43,69	42,82
13	40,13	45,09	45,32
14	41,70	41,82	45,27
15	45,29	51,06	40,24
16	44,37	44,66	49,53

Externalidade Negativa

<i>Número de Agentes</i>	$e_c = -0.1$	$e_c = -0.5$	$e_c = -1.0$
10	53,40	58,80	51,53
11	49,43	53,90	60,09
12	50,11	58,89	65,53
13	52,91	61,27	70,73
14	49,58	68,00	75,27
15	54,27	79,41	79,08
16	56,42	75,96	84,59

A.2.1.2 *Distribuição Normal****Externalidade Positiva***

<i>Número de Agentes</i>	$e_c = 0.1$	$e_c = 0.5$	$e_c = 1.0$
10	44,53	44,33	50,47
11	48,28	48,86	45,90
12	48,02	34,15	43,25
13	43,40	46,00	44,62
14	43,70	44,40	47,70
15	40,77	46,47	44,26
16	48,87	50,72	45,45

Externalidade Negativa

<i>Número de Agentes</i>	$e_c = -0.1$	$e_c = -0.5$	$e_c = -1.0$
10	46,20	50,00	58,00
11	49,28	47,14	51,09
12	48,82	50,36	54,79
13	48,39	53,58	58,49
14	42,40	44,83	58,15
15	45,15	55,37	57,74
16	47,62	44,99	57,57

A.2.2 Número de estruturas de coalizão avaliadas em relação ao número total de estruturas de coalizão do problema

A.2.2.1 Distribuição Uniforme

Externalidade Positiva

<i>Número de Agentes</i>	$e_c = 0.1$	$e_c = 0.5$	$e_c = 1.0$
10	0,208538047	0,876550000	0,947877000
11	0,469075497	0,950360000	0,852331000
12	0,024756087	0,076198626	0,101471435
13	0,014559685	0,013414500	0,124653041
14	0,002089395	0,009142253	0,004617719
15	0,000337169	0,000601830	0,001572208
16	5,7846E0-05	0,000126573	0,000111166

Externalidade Negativa

<i>Número de Agentes</i>	$e_c = -0.1$	$e_c = -0.5$	$e_c = -1.0$
10	0,078821298	0,078821298	0,067496788
11	0,038084825	0,043832206	0,362420974
12	0,010314954	0,010314954	0,048768093
13	0,002209591	0,002209591	0,000906156
14	0,000363868	0,000363868	7,16067E-05
15	0,000122820	0,000122820	2,44829E-05
16	2,99901E-05	2,99901E-05	2,93617E-06

A.2.2.2 *Distribuição Normal****Externalidade Positiva***

<i>Número de Agentes</i>	$e_c = 0.1$	$e_c = 0.5$	$e_c = 1.0$
10	0,575858935	0,789010000	0,874400000
11	0,769329354	0,907520000	0,974300000
12	0,284734282	0,045395365	0,113662771
13	0,304654293	0,060518175	0,040177723
14	0,009178488	0,017461894	0,007899561
15	0,001380336	0,000782097	0,001317907
16	0,000148111	0,000185754	0,000194378

Externalidade Negativa

<i>Número de Agentes</i>	$e_c = -0.1$	$e_c = -0.5$	$e_c = -1.0$
10	0,814993749	0,354026816	0,164728433
11	0,457254137	0,521770547	0,819100461
12	0,789598101	0,290557583	0,306897969
13	0,206777476	0,047140315	0,073449559
14	0,005736638	0,003665431	0,006020157
15	0,000884466	0,001662082	0,001242499
16	7,75161E-05	0,000196432	0,000161681