

ANDERSON AFONSO NUNES

**RESTRINGINDO O ESPAÇO DE BUSCA NA
GERAÇÃO DE ESTRUTURAS DE
COALIZÃO UTILIZANDO GRAFOS**

Dissertação submetida ao Programa de Pós-Graduação em Computação Aplicada da Universidade Tecnológica Federal do Paraná como requisito parcial para a obtenção do título de Mestre em Computação Aplicada.

Curitiba PR
Agosto de 2015

ANDERSON AFONSO NUNES

RESTRINGINDO O ESPAÇO DE BUSCA NA GERAÇÃO DE ESTRUTURAS DE COALIZÃO UTILIZANDO GRAFOS

Dissertação submetida ao Programa de Pós-Graduação em Computação Aplicada da Universidade Tecnológica Federal do Paraná como requisito parcial para a obtenção do título de Mestre em Computação Aplicada.

Área de concentração: *Engenharia De Sistemas Computacionais*

Orientador: Drº Gustavo Alberto Giménez Lugo

Co-orientador: Drº Murilo V. G. da Silva

Curitiba PR
Agosto de 2015

Dados Internacionais de Catalogação na Publicação

N972r
2015 Nunes, Anderson Afonso
Restringindo o espaço de busca na geração de estruturas
de coalizão utilizando grafos / Anderson Afonso Nunes.-- 2015.
65 p. : il. ; 30 cm

Dissertação (Mestrado) - Universidade Tecnológica Federal
do Paraná. Programa de Pós-graduação em Computação Apli-
cada, Curitiba, 2015

Bibliografia: p. 64-65

1. Teoria dos grafos. 2. Inteligência artificial. 3. Sistemas mul-
tiagentes. 4. Informática - Dissertações. I. Giménez Lugo, Gus-
tavo Alberto, orient. II. Silva, Murilo Vicente Gonçalves da, coori-
ent. III. Universidade Tecnológica Federal do Paraná - Programa
de Pós-graduação em Computação Aplicada. IV. Título.

CDD: Ed. 22 -- 621.39

Biblioteca Central da UTFPR, Câmpus Curitiba

ATA DE DEFESA DE DISSERTAÇÃO DE MESTRADO Nº 30

Aos 20 dias do mês de agosto de 2015 realizou-se na sala C-301 a sessão pública de Defesa da Dissertação de Mestrado intitulada "Restrição no espaço de busca de Estruturas de Coalizão utilizando Grafos", apresentada pelo aluno **Anderson Afonso Nunes** como requisito parcial para a obtenção do título de Mestre em Computação Aplicada, na área de concentração "Engenharia de Sistemas Computacionais", linha de pesquisa "Sistemas Inteligentes e Lógica".

Constituição da Banca Examinadora:

Prof. Dr. Gustavo Alberto Gimenez Lugo, UTFPR - CT (Presidente) _____

Prof. Dr. Cesar Augusto Tacla, UTFPR – CT _____

Prof. Dr. André Luis Vignattti, UFPR _____

Em conformidade com os regulamentos do Programa de Pós-Graduação em Computação aplicada e da Universidade Tecnológica Federal do Paraná, o trabalho apresentado foi considerado _____ (aprovado/reprovado) pela banca examinadora. No caso de aprovação, a mesma está condicionada ao cumprimento integral das exigências da banca examinadora, registradas no verso desta ata, da entrega da versão final da dissertação em conformidade com as normas da UTFPR e da entrega da documentação necessária à elaboração do diploma, em até _____ dias desta data.

Ciente (assinatura do aluno): _____

(para uso da coordenação)

A Coordenação do PPGCA/UTFPR declara que foram cumpridos todos os requisitos exigidos pelo programa para a obtenção do título de Mestre.

Curitiba PR, ____/____/_____

"A Ata de Defesa original está arquivada na Secretaria do PPGCA".

Dedico este trabalho a Deus, por ter me dado força, serenidade, sabedoria e paciência para superar todos estes dias entre as salas de aulas, bibliotecas, laboratório e as estradas da vida em busca desta conquista profissional e pessoal. Aos meus pais Odete Nunes e Afonso Nunes e a minha noiva Ana Flavia Fontes, orgulho de minha alma, de um amor sem limites e por simplesmente tudo!

Agradecimentos

Primeiramente a Deus que permitiu que tudo isso acontecesse, ao longo de minha vida, e não somente nestes anos como universitário, mas que em todos os momentos é o maior mestre que alguém pode conhecer.

À Instituição pelo ambiente criativo e amigável que proporciona.

Ao meu orientador, pelo empenho dedicado à elaboração deste trabalho.

Aos meus pais, pelo amor, incentivo e apoio incondicional.

Em especial ao meu pai que para mim foi muito importante.

A minha noiva Ana Flavia Fontes, companheira de todas as horas que me impulsionou a seguir em frente, e pela compreensão pelas horas que estive ausente.

A todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

Resumo

O problema de geração de estruturas de coalizão (CSG) envolve o particionamento do conjunto de agentes em todos os subconjuntos(ou, coalizões) possíveis. O que torna esse problema desafiador é o número de coalizões possíveis crescer exponencialmente a medida que novos agentes são inseridos, o número de coalizões é $(2^n - 1)$ onde n é o número de agentes. Entretanto, em muitas aplicações do mundo real, existem limitações inerentes nas coalizões possíveis: por exemplo, determinados agentes podem ser proibidos de estar na mesma coalizão, ou a estrutura de coalizão pode ser obrigada a conter coalizões do mesmo tamanho. Quando consideramos CSG restrito por grafos, onde a viabilidade de uma coalizão é restrita por um grafo de sinergia dos agentes, a complexidade computacional pode ser a mesma ou menor, dependendo do que se considera uma coalizão válida. Os grafos de sinergia são representações dos agentes como sendo os vértices e as suas relações são as arestas. Este trabalho é um estudo sobre a utilização de restrições envolvendo grafos como uma heurística sobre as coalizões para o problema enumeração de coalizão, de forma a considerar uma coalizão factível ou não de acordo com a densidade do subgrafo induzido pelos agentes. Os trabalhos atuais, que utilizam os grafos de restrição como heurística para reduzir a complexidade computacional, consideram uma coalizão válida somente se o subgrafo formado pelos agentes da coalizão é conexo. Verificou-se experimentalmente para grafos com a propriedade *power law*, comum em uma variedade de grafos reais, que restringir uma coalizão válida como sendo um subgrafo conexo pode não ser uma redução significativa. Entretanto a utilização de um subgrafo com restrições mais fortes, em particular uma clique garante uma redução exponencial do número de coalizões consideradas. Não existem teoremas que possam calcular qual a quantidade de subgrafos conexos ou mesmo o número de cliques em um grafo do tipo *power law*. No presente trabalho foi possível calcular experimentalmente para grafos *power law* com ate 17 vértices, sendo que também são apresentados resultados analíticos para grafos estrela $(K_{n-1,1})$. Os grafos estrela são uma aproximação aceitável, pois formam um *hub*, estrutura característica de grafos *power law*. Como trabalhos futuros podem ser citados: o mapeamento de domínios para os quais a restrição de clique seria adequada, além do desenvolvimento de um algoritmo que incorpore a restrição diretamente na contagem de coalizões validas.

Palavras-chave: Formação de coalizões, Geração de estruturas de coalizão, Formação de Coalizões em grafos.

Abstract

The coalition structures generating problem (CSG) involves partitioning the set of agents in all possible subsets (or coalitions). What makes this problem challenging is the number of possible coalitions that grows exponentially as new agents are inserted. The number of coalitions is $(2^n - 1)$ where n is the number of agents. However, in many real-world applications, there are inherent limitations on possible coalitions: for example, some individuals may be prohibited from being in the same coalition or coalition structure may be required to contain coalitions of the same size. When we consider CSG restricted by graphs where the viability of a coalition is restricted by a synergy graph, the computational complexity can be maintained or eventually be smaller depending on what is considered a valid coalition. Synergy graphs are representations of the agents as being the vertices and their relationships are the edges. This work is a study on the use of restrictions involving graphs as a heuristic about coalitions for the problem coalition enumeration in order to consider a feasible coalition or not according to the density of the subgraph induced by the agents. Current works using the restriction graphs as heuristics to reduce the computational complexity, consider a coalition valid only if the subgraph formed by the agents of the coalition is connected. In this work it is experimentally verified for power law graphs, present in a variety of real graphs, that restricting availability coalition as a connected subgraph may in not prohibited a significant gain. However, they using a subgraphs with strong restrictions, in particular a clique, guarantees an exponential reduction in the number of considered coalition. There are no theorems calculate subgraphs or even the number of cliques on a type power law graph. In the present work it was possible to calculate values experimental for graphs of up to 17 vertices, being also presented analytics results for star graphs $(K_{n-1,1})$. Star graphs are an acceptable approximation, as they account for hubs, a characteristic structure of power law graphs. As future works can be cited the study of domains where the clique restriction is adequate as well as the development of an algorithm that incorporates the restriction for coalition counting.

Keywords: Coalition formation, Coalition formation over graphs, Coalition formation generation.

Sumário

Resumo	xi
Abstract	xiii
Lista de Figuras	xvii
Lista de Tabelas	xix
Lista de Abreviações	xx
1 Introdução	1
1.1 Objetivos	3
1.2 Organização do trabalho	3
2 Definições	5
2.1 Agente	5
2.2 Coalizões	5
2.3 Formalização do problema de geração de estruturas de coalizão	7
2.4 Complexidade de encontrar a estrutura de coalizão ótima	8
2.5 Grafos	9
2.6 Considerações	11
3 Partição em Grafos Para Cálculo de Coalizões	13
3.1 Algoritmos para formação de estruturas de coalizões	13
3.1.1 Algoritmo CSS1 - <i>Coalition-Structure-Search-1</i>	14
3.1.2 Algoritmo DP	15
3.1.3 Algoritmo <i>IP</i>	17
3.1.4 Algoritmo <i>IDP</i>	18
3.1.5 Algoritmo $IP^{+/-}$	20
3.1.6 Algoritmo CCF - Formação de coalizões com restrições	21
3.1.7 Algoritmo DyCE	21
3.1.8 Algoritmo CFSS	22
3.1.9 Algoritmo DyPE	22
3.1.10 Algoritmo TSP	23
3.2 Comparação de algoritmos	23
3.3 Uso de restrição sobre as coalizões possíveis	25
3.4 Considerações	25

4	Estruturação da Pesquisa	27
4.1	Criação de Programa Java Para geração dos grafos e contagem de subgrafos . . .	28
4.2	Geração dos grafos	29
4.2.1	Grafos <i>power law</i>	29
4.2.2	Grafos estrela	29
4.3	Análise do número de coalizões possíveis (sem considerar restrições)	30
4.4	Análise do número de subgrafos factíveis (coalizões que são subgrafos conexos)	30
4.5	Análise do número de subgrafos factíveis (coalizões que são cliques)	30
4.6	Experimentos	31
4.7	Considerações	31
5	Abordagem do Problema	33
5.1	O número de coalizões factíveis pode ser exponencial mesmo em grafos esparsos	34
5.2	Generalizando conceitos de coalizões factíveis	37
5.3	Considerações	37
6	Resultados e Análise	39
6.1	Resultados para grafos completos	39
6.2	Resultados para grafos estrela	40
6.3	Resultados para grafos <i>power law</i>	41
7	Conclusão	43
A	Anexos	47
A.1	Notações	47

Lista de Figuras

2.1	Grafo estrela $k_{5,1}$	10
3.1	Grafo de estruturas de coalizão para quatro agentes.	14
3.2	Representação do espaço de busca realizada pelo algoritmo IP. O conjunto da agentes $A = a_1, a_2, a_3, a_4$ e os elementos de N foram expandidos. Exemplo adaptado para melhor entendimento de Rahwan [2007].	18
3.3	Grafo de estruturas de coalizão para quatro agentes. As arestas em destaque mostram todos os caminhos que o DP poderia percorrer para encontrar a solução ótima. O IDP , elimina a maioria destas arestas, de forma que cada vértice contenha apenas uma aresta direcionada para ele(no sentido do nível inferior para anterior) Rahwan and Jennings [2008b].	19
4.1	Etapas do método.	28
5.1	Exemplo de um grafo da relação.	35
5.2	Apresenta 3 exemplos de subgrafos induzidos, as 3 coalizões destacadas nos grafos são válidas mas não factíveis. Na imagem (a) se observa uma os vértices e arestas destacados, esse subgrafo destacado pode ser considerada uma coalizão factível quando a restrição considerada é um subgrafo conexo, mas não pode ser uma coalizão factível se a restrição a ser considerada é clique. (b) como apresentado na descrição da imagem (a) os vértices e arestas destacados somente formam uma coalizão factível se a restrição considerada é a de subgrafo conexo. (c) apresenta alguns vértices isolados, todos eles poderiam estar presentes em coalizões onde o único vértice presente é ele mesmo.	36
6.1	Grafo <i>loglog</i> do número de possíveis coalizões quando são considerados grafos completos de agentes.	39
6.2	Grafo <i>loglog</i> do número de coalizões possíveis e factíveis em um grafo estrela. É possível observar que ambas as linhas estão sobrepostas, o número de subgrafos possíveis e o de factíveis. Considerando as coalizões factíveis como sendo subgrafos conexos	40
6.3	Número de coalizões possíveis, SICs e cliques em grafo estrela.	41
6.4	Gráfico <i>loglog</i> que apresenta o número de subgrafos possíveis, factíveis e cliques em grafos <i>power law</i> em relação ao número de vértices.	42

Lista de Tabelas

2.1	Comparativo entre o crescimento do número de coalizões e estruturas de coalizão para n agentes.	7
2.2	Exemplo de função característica para um conjunto $A = \{a_1, a_2, a_3, a_4\}$	8
2.3	Exemplo de função de partição para um conjunto $A = \{a_1, a_2, a_3, a_4\}$	9
3.1	Algoritmos de formação de coalizões	24

Lista de Abreviações

CSG	<i>Coalition Structure Generation</i>
CCF	<i>Constrained Coalition Formation</i>
PPGCA	Programa de Pós-Graduação em Computação Aplicada
UTFPR	Universidade Tecnológica Federal do Paraná
SIC	Subgrafo Induzido Conexo

Capítulo 1

Introdução

Formação de coalizão é uma abordagem fundamental para estabilizar a colaboração entre agentes, cada qual com capacidades e propriedades individuais [Shehory and Kraus, 1998]. Em particular, geração de estruturas de coalizão (*Coalition Structure Generation - CSG*) representa a principal tarefa computacional neste cenário.

Este trabalho considera uma determinada classe de problemas, conhecidos como jogos de restrição em grafos (*graph-restricted games*) [Myerson, 1977].

Em aplicações que envolvem múltiplas partes interessadas, a formação de grupos ou coletivos de agentes é fundamental para alcançar os objetivos individuais ou globais. Cada grupo de agentes pode colaborar de maneira distinta. A construção de grupos com agentes para colaboração tem sido o principal foco das técnicas de formação de coalizão.

Nas aplicações do mundo real, as redes sinérgicas formadas pelos agentes, podem restringir a formação de coalizões [Voice et al., 2012a,b]. As restrições ocorrem devido a fatores, como: infraestrutura de comunicação, relações sociais ou confiança, ou ainda restrições físicas entre os agentes.

Recentemente, algumas abordagens têm apresentado uma proposta de modelar as restrições em forma de um grafo, conhecido como grafo de restrições, os agentes são representados pelos vértices e as arestas representam as possíveis conexões entre os agentes para formar uma coalizão [Voice et al., 2012a,b]. Consequentemente, uma coalizão é considerada factível somente se seus membros representam os vértices de um subgrafo conexo do grafo de restrições.

Em sistemas multiagentes, os agentes tendem a melhorar suas performances por meio da formação de coalizões. A geração de estruturas de coalizão é equivalente ao problema de partição completa de conjuntos [Yun Yeh, 1986] e de acordo com Voice et al. [2012a] tem aplicação em muitos campos, como: ciência da computação (robótica, alocação de tarefas), ciências políticas e econômicas, pesquisa operacional, *e-commerce*, gerenciamento de desastres e aquisição de informação de dispositivos embarcados; isso tem gerado um grandes desafios para projeto de algoritmos.

O número de coalizões possíveis geradas em um conjunto A de n agentes é dado pela fórmula $2^n - 1$ e o número de estruturas de coalizão pode ser obtido pelo número de *Stirling* de segunda ordem, esses valores serão apresentados com mais detalhes no capítulo 2.

No problema de geração de estruturas de coalizão, um conjunto A de n elementos e uma função de valoração $v : P(A) \rightarrow \mathbb{R}$, onde $P(A)$ denota o conjunto potência¹ de A . O que

¹Conjunto potência é o conjunto formado por todos os subconjuntos de A incluindo o próprio conjunto A e também o conjunto vazio.

se pretende é dividir o conjunto dado em subconjuntos disjuntos A_1, \dots, A_m (ou, coalizões) para que a soma total $\sum_{i=1}^m v(A_i)$ seja maximizada.

Em termos computacionais a formação de coalizões envolve calcular o valor de cada possível coalizão e encontrar qual é a estrutura de coalizão ótima. O problema de geração de estruturas de coalizão pode ser dividido em três fases de acordo com Sandholm et al. [1999]: (i) Geração de estruturas de coalizão; (ii) Resolver o problema de otimização de cada coalizão; (iii) Dividir os ganhos obtidos pela solução entre os agentes.

O foco neste trabalho é interno a primeira fase proposta por Sandholm et al. [1999], a geração de coalizões, mais precisamente, sobre a quantidade de possíveis coalizões que serão geradas e avaliadas, considerando que nem todas as coalizões possíveis são válidas. A redução no número total de coalizões possíveis tem um impacto significativo no número de estruturas de coalizões a serem analisadas.

Trabalhos que estudam o problema de formação de coalizões em uma configuração abstrata [Michalak et al., 2010; Rahwan and Jennings, 2007, 2008b; Shehory and Kraus, 1998], ou seja, consideram que todo subconjunto de agentes é uma possível coalizão, são limitadas em resolver o problema devido ao desempenho computacional, como exemplo, resolvem problemas que envolvem 30 agentes para a enumeração de todas as coalizões e 28 agentes para o caso de geração de estruturas de coalizão. Entretanto, isso se mostra ineficiente quando se leva em conta que somente algumas coalizões podem ser consideradas factíveis.

É proposta uma nova abordagem sobre o problema de enumeração de coalizões, de forma que a existência de uma coalizão pode ser descrita usando uma aproximação, que restringe as relações à aquelas que possam existir em redes sociais ou de comunicação [Voice et al., 2012b].

Como exemplificação de uma situação real onde todos os agentes deveriam conhecer os demais agentes de uma coalizão, é possível citar: todos os membros que irão participar do próximo projeto de uma grande empresa devem obrigatoriamente residir na mesma cidade. Nesse caso a restrição entre os membros é a localização na mesma cidade. Também é possível pensar um grupo de robôs que está em uma mesma sala, somente podem formar coalizões os robôs que estão dentro de uma certa região delimitada dentro desta sala. As restrições onde todos os agentes de dentro de uma coalizão devem necessariamente conhecer (estar localizados no mesmo local) os demais pode ser representada através da clique de um grafo.

Além de considerar somente os subgrafos conexos, este trabalho tem uma abordagem que somente coalizões que formam uma clique do grafo são válidas, desta forma todos os agentes de uma coalizão obrigatoriamente devem estar conectados aos demais. Considerar clique como uma restrição não foi aplicado em nenhum dos trabalhos utilizados na revisão bibliográfica. Mesmo sendo uma restrição forte para a formação de coalizões, os cliques, o trabalho apresenta uma redução possível para tratar o problema em um grande número de agentes.

A representação de coalizões factíveis é equivalente ao problema de enumeração de subgrafos conexos, cada coalizão pode ser um subgrafo conexo do grafo de agentes. Esse enfoque têm se mostrado promissor para acelerar o processo de enumeração de coalizões em ordens de grandeza, em particular quando os agentes são ligados em estruturas conhecidas, como as árvores de baixo nível de ramificação [Voice et al., 2012b].

Se busca verificar os limites da utilização de grafos como restrições para a formação de coalizões, por exemplo, uso de subgrafos conexos Voice et al. [2012b] para reduzir o espaço de busca de coalizões. Em decorrência da análise é apresentada uma abordagem que não apenas

considera subgrafos conexos mas também cliques. Dessa maneira, somente são consideradas factíveis as coalizões que todos conhecem os demais agentes da coalizão.

Experimentos apresentados neste trabalho apontam para evidências que a utilização de cliques como coalizões factíveis (restrição mais forte) podem reduzir exponencialmente o número de coalizões a serem enumeradas quando um grafo tem propriedades comumente encontradas, como nos grafos *power law*. Isso pode representar a possibilidade de utilização em conjuntos que possuem um maior número de agentes envolvidos, o que hoje pode não ser possível pela complexidade do problema.

1.1 Objetivos

Objetivo Geral

- Verificar os limites da utilização de grafos como restrições na formação de coalizões e, em particular, o impacto ao utilizar clique como critério para a formação de coalizões.

Objetivos Específicos

- Analisar as condições para redução do número de coalizões utilizando grafos;
- Comparar o desempenho de algoritmos que utilizem esses enfoques;
- Verificar o impacto da escolha de cliques como critério de formação de coalizões;
- Apresentar evidências experimentais que ilustrem as possíveis vantagens da utilização de grafos como restrições de formação de coalizões, em particular nos grafos *power law*;

1.2 Organização do trabalho

A dissertação foi dividida em 7 capítulos. Os capítulos 2 e 3 se referem aos conceitos teóricos utilizados neste trabalho e uma revisão da literatura dos algoritmos para formação de coalizões. Os capítulos 4, 5 e 6 correspondem ao núcleo do trabalho desenvolvido, finalizando com o capítulo de conclusão:

- O capítulo 2 apresenta as definições fundamentais sobre formação de coalizões e teoria dos grafos.
- O capítulo 3 apresenta uma revisão bibliográfica sobre os principais algoritmos de formação de coalizão. Ao final do capítulo é apresentada uma tabela comparativa entre os algoritmos, nessa tabela é apresentado como os algoritmos estão ligados com a geração de coalizões levando em consideração as restrições apresentadas em grafos.
- O capítulo 4 apresenta a estruturação da pesquisa. As etapas que foram desempenhadas para a realização deste trabalho. São descritas as fases para a geração dos experimentos e como os resultados foram levantados e analisados para a geração dos resultados.
- O capítulo 5 apresenta qual a abordagem de geração de coalizões quando são levadas em consideração as restrições apresentadas em um grafo de restrições. Nesse grafo os agentes são representados pelos vértices e a ligação entre os agentes é representado pelas arestas.

- O capítulo 6 apresenta os resultados e análise dos experimentos realizados.
- O capítulo 7 apresenta a conclusão da dissertação.

Capítulo 2

Definições

Neste capítulo é apresentada a formalização necessária de agentes, coalizões e grafos. Essas definições são necessárias para a apresentação dos resultados nos próximos capítulos.

2.1 Agente

Um agente pode ser definido como um sistema de computador que está localizado em um ambiente e que é capaz de agir de forma autônoma neste ambiente com propósito de atingir seus objetivos [Wooldridge, 2009], [Wooldridge and Jennings, 1995]. Agir de forma autônoma se refere à capacidade de operar sem a intervenção de seres humanos ou de qualquer outro sistema. Os agentes possuem um conjunto de ações que podem influenciar em seu ambiente. O problema chave do agente é decidir qual das ações devem ser executadas, a fim de cumprir seus objetivos.

2.2 Coalizões

Dois conceitos são primordiais para o entendimento deste trabalho: coalizão possível e coalizão factível.

- **Coalizão possível:** Uma coalizão **possível** é toda coalizão que pode ser gerada. Formalmente, o número de coalizões possíveis é $2^n - 1$, onde n é o número de vértices.
- **Coalizão factível:** Uma coalizão é **factível** se e somente se atende a uma determinada restrição sobre o subconjunto de vértices do grafo. Neste trabalho, as restrições consideradas para uma coalizão ser factível são: i) um subgrafo conexo ou ii) uma clique.

Quando o problema de geração de estruturas de coalizão é tratado no modelo comum nenhuma coalizão pode ser descartada, pois se considera que qualquer coalizão não verificada pode acabar impactando nas demais coalizões. Ao tratar o problema considerando as restrições é possível, utilizando uma heurística, descartar algumas coalizões que não são de interesse serem validadas.

As coalizões geralmente são criadas para um objetivo direto e de curto período de vida, essa definição apresentada em Horling and Lesser [2004]. Coalizões são formadas com

um objetivo específico e se desfazem quando o mesmo é atingido, quando não se adequam ao propósito pretendido ou quando a rentabilidade da coalizão é perdida.

De acordo com Rahwan [2007], o paradigma baseado em coalizões é capaz de modelar cenários onde não existe uma autoridade central para resolver os possíveis conflitos existentes entre as decisões dos agentes. Outra característica que pode definir as coalizões é que dentro da coalizão os agentes podem cooperar para atingir determinado objetivo.

Essa característica não existe para agentes de coalizões diferentes (exceto se os objetivos das coalizões tenham como propósito a interação entre as mesmas). A estrutura organizativa das coalizões geralmente é plana, não existe uma liderança interna na coalizão¹.

A formação de coalizões tem conquistado grande atenção em pesquisas recentes e tem provado sua utilidade em grande escala de aplicações no mundo real e em sistemas multiagentes.

O processo de formação de coalizões pode ser resumido em três fases principais [Rahwan, 2007]:

- **Calcular os valores das coalizões:** Computar o valor de todas as coalizões possíveis.
- **Geração da estrutura de coalizão:** Computar o conjunto disjuncto de coalizões que tenham o valor total máximo.
- **Distribuição dos resultados:** Determinar qual a recompensa que cada agente pode obter com o resultado de suas ações tomadas na coalizão como um todo.

A geração de estruturas de coalizão (do inglês: *Coalition Structure Generation - CSG*) é um desafio existente no processo de formação de coalizões. Dado os valores de cada coalizão, deve-se determinar como particionar o conjunto de agentes em coalizões exaustivas e disjunctas. Cada uma das partições é chamada de estrutura de coalizão, por exemplo, dado um conjunto de agentes $A = \{a_1, a_2, a_3\}$, existem cinco possíveis estruturas de coalizão: $\{\{a_1\}, \{a_2\}\{a_3\}\}$, $\{\{a_1\}, \{a_2, a_3\}\}$, $\{\{a_2\}, \{a_1, a_3\}\}$, $\{\{a_3\}, \{a_1, a_2\}\}$, $\{\{a_1, a_2, a_3\}\}$.

Essa configuração de formação de estruturas de coalizão é conhecida como jogos de função característica, onde o valor de uma coalizão é dado por uma função característica, essa função não é necessariamente conhecida pelos agentes.

Mas nem sempre a formação de coalizões possui essa característica, alguns casos o valor de uma coalizão pode depender de ações dos membros que não pertencem a coalizão por conta de efeitos externos positivos e/ou negativos de acordo com a configuração utilizada, como exemplo: quando são consideradas as externalidades. Externalidades negativas entre coalizão e membros são muitas vezes causadas pelo compartilhamento de recursos Sandholm et al. [1999].

Denotado por n o número de agentes, e o conjunto de agentes sendo $A = \{a_1, a_2, \dots, a_n\}$. Todo subconjunto não vazio de A é uma **coalizão possível**, e denotamos como \mathcal{C} o conjunto de todas coalizões. De uma maneira mais formal, temos: $\mathcal{C} = \{C : C \subseteq A, C \neq \emptyset\}$. O que implica em:

$$|\mathcal{C}| = 2^n - 1 \quad (2.1)$$

Uma definição formal para **estrutura de coalizão** pode ser encontrada em Rahwan et al. [2012]: a mesma é denotada por $CS = \{C_1, C_2, \dots, C_{|CS|}\}$, sendo uma partição exaustiva do conjunto A em coalizões disjunctas, isso deve satisfazer as condições seguintes:

¹Embora possa existir um líder da coalizão representando o grupo como um todo.

- $\forall i \in \{1, \dots, |CS|\}, C_i \neq \emptyset$;
- $\bigcup_{i=1}^{|CS|} C_i = A$;
- $\forall i, j \in \{1, \dots, |CS|\} : i \neq j, C_i \cap C_j = \emptyset$.

De acordo com Sandholm et al. [1999], um problema enfrentado para a geração das estruturas de coalizão é quanto o número de estruturas de coalizão que cresce rapidamente, considerando o número de agentes. O número exato de estruturas de coalizão é denotado por \mathcal{P}^A , ou ainda por:

$$\sum_{i=1}^a Z(a, i) \quad (2.2)$$

onde, $Z(a, i)$ é o número de estruturas de coalizão com i coalizões. O número $Z(a, i)$ - também conhecido como o número de *Stirling* de segunda ordem, é calculado pela seguinte recorrência:

$$Z(a, i) = iZ(a-1, i) + Z(a-1, i-1), \quad (2.3)$$

onde $Z(a, a) = Z(a, 1) = 1$.

Esta recorrência pode ser compreendida ao considerar a adição de um novo agente para um jogo com $a-1$ agentes. O primeiro termo, $iZ(a-1, i)$ conta o número de estruturas de coalizão formadas pela adição de um novo agente para uma das coalizões existentes. Existem i possibilidades porque as estruturas existentes da coalizão tem i coalizões. O segundo termo, $Z(a-1, i-1)$, considera a adição do novo agente na mesma coalizão, e portanto somente estruturas de coalizão com apenas $i-1$ coalizões são contadas.

É possível observar a diferença entre os valores $|\mathcal{C}|$ e $|\mathcal{P}^A|$ o crescimento é exponencial ao número de agentes. Um exemplo é apresentado na tabela 2.1.

Tabela 2.1: Comparativo entre o crescimento do número de coalizões e estruturas de coalizão para n agentes.

n	1	3	4	5	7	9	11	13	15
$ \mathcal{C} $	1	7	15	31	127	511	2.047	8.195	32.783
$ \mathcal{P}^A $	1	5	15	52	877	21.147	678.570	27.644.437	1.382.958.545

Fonte: Adaptado de [Rahwan et al., 2012]

2.3 Formalização do problema de geração de estruturas de coalizão

O problema de geração de estruturas de coalizão é encontrar a estrutura de coalizão ótima $CS^* \in \mathcal{P}_A$ que pode ser definido para jogos de função característica como:

$$CS^* = \arg \max_{CS \in \mathcal{P}^A} \sum_{C \in CS} v(C) \quad (2.4)$$

enquanto para jogos de função de partição:

$$CS^* = \arg \max_{CS \in \mathcal{P}^A} \sum_{C \in CS} w(C, CS) \quad (2.5)$$

Nos exemplos apresentados para **jogos sem definição de externalidade**, na tabela 2.2, temos $CS^* = \{\{a_1, a_3\}, \{a_2, a_4\}\}$. Isso se deve ao fato que $v(\{a_1, a_3\}) + v(\{a_2, a_4\}) = 10$ é um valor maior que qualquer outra estrutura de coalizão existente neste jogo [Rahwan et al., 2012].

Em jogos com externalidade, o valor de uma coalizão depende da maneira que os outros agentes são particionados. O resultado da função de partição w reflete o desempenho de $C \in CS$ [Rahwan et al., 2012].

Para jogos que envolvem externalidades entre as coalizões, apresentado na tabela 2.3, $CS^* = \{\{a_1\}\{a_2, a_3, a_4\}\}$ pois $w(\{a_1\}, CS^*) + w(\{a_2, a_3, a_4\}, CS^*) = 8$ é o maior valor de estrutura de coalizão entre todas as existentes no jogo.

Tabela 2.2: Exemplo de função característica para um conjunto $A = \{a_1, a_2, a_3, a_4\}$.

$v(\{a_1\}) = 1$	$v(\{a_1, a_2\}) = 3$	$v(\{a_2, a_4\}) = 6$	$v(\{a_1, a_3, a_4\}) = 6$
$v(\{a_2\}) = 2$	$v(\{a_1, a_3\}) = 4$	$v(\{a_3, a_4\}) = 4$	$v(\{a_2, a_3, a_4\}) = 8$
$v(\{a_3\}) = 2$	$v(\{a_1, a_4\}) = 1$	$v(\{a_1, a_2, a_3\}) = 5$	$v(\{a_1, a_2, a_3, a_4\}) = 7$
$v(\{a_4\}) = 1$	$v(\{a_2, a_3\}) = 5$	$v(\{a_1, a_2, a_4\}) = 5$	height

Fonte: Adaptado de [Rahwan et al., 2012]

2.4 Complexidade de encontrar a estrutura de coalizão ótima

De acordo com Sandholm et al. [1999] encontrar a estrutura de coalizão ótima é computacionalmente custoso (*NP-hard*). O tamanho da entrada de dados e o número de estruturas de coalizões são os fatores que influenciam diretamente na complexidade do problema.

O **tamanho da entrada** é exponencial no número de agentes; A entrada contém um valor, v_S para cada coalizão. O valor é associado individualmente a cada coalizão, para uma entrada contendo um número n de agentes, o número total de $(2^n - 1)$ coalizões, o valor -1 representando a exclusão da coalizão vazia que não é levada em consideração, na ordem de $O(2^n)$. Isso vale para jogos de função característica, quando não se leva em conta as possíveis externalidades que possam existir entre as coalizões.

Ao levar em conta as externalidades o custo computacional é de $O(n^n)$, a razão para isso é que para toda coalizão a entrada contém todos os valores possíveis para a coalizão e seus relacionamentos com as demais coalizões possíveis.

Sandholm et al. [1999] comentam que conceitualmente pode-se excluir o valor de algumas coalizões na entrada ou decidir ignorar alguma parte para melhorar o tempo de execução. No entanto excluir uma simples coalizão ou valor da entrada pode causar a perda no valor das

Tabela 2.3: Exemplo de função de partição para um conjunto $A = \{a_1, a_2, a_3, a_4\}$.

$w(\{a_1\}, \{\{a_1\}\{a_2\}\{a_3\}\{a_4\}\}) = 1$	$w(\{a_4\}, \{\{a_1, a_2, a_3\}\{a_4\}\}) = 0$
$w(\{a_1\}, \{\{a_2, a_3\}\{a_1\}\{a_4\}\}) = 1$	$w(\{a_1, a_2\}, \{\{a_1, a_2\}\{a_3\}\{a_4\}\}) = 3$
$w(\{a_1\}, \{\{a_2, a_4\}\{a_1\}\{a_3\}\}) = 0$	$w(\{a_1, a_2\}, \{\{a_1, a_2\}\{a_3, a_4\}\}) = 2$
$w(\{a_1\}, \{\{a_3, a_4\}\{a_1\}\{a_2\}\}) = 0.5$	$w(\{a_1, a_3\}, \{\{a_1, a_3\}\{a_2\}\{a_4\}\}) = 4$
$w(\{a_1\}, \{\{a_1\}\{a_2, a_3, a_4\}\}) = 0$	$w(\{a_1, a_3\}, \{\{a_1, a_3\}\{a_2, a_4\}\}) = 3$
$w(\{a_2\}, \{\{a_1\}\{a_2\}\{a_3\}\{a_4\}\}) = 2$	$w(\{a_1, a_4\}, \{\{a_1, a_4\}\{a_2\}\{a_3\}\}) = 1$
$w(\{a_2\}, \{\{a_3, a_4\}\{a_1\}\{a_2\}\}) = 1$	$w(\{a_1, a_4\}, \{\{a_1, a_4\}\{a_2, a_3\}\}) = 1$
$w(\{a_2\}, \{\{a_1, a_3\}\{a_1\}\{a_4\}\}) = 2$	$w(\{a_2, a_3\}, \{\{a_2, a_3\}\{a_1\}\{a_4\}\}) = 5$
$w(\{a_2\}, \{\{a_1, a_4\}\{a_1\}\{a_2\}\}) = 1$	$w(\{a_2, a_3\}, \{\{a_2, a_3\}\{a_1, a_4\}\}) = 4$
$w(\{a_2\}, \{\{a_1, a_3, a_4\}\{a_2\}\}) = 0$	$w(\{a_2, a_4\}, \{\{a_2, a_4\}\{a_1\}\{a_3\}\}) = 6$
$w(\{a_3\}, \{\{a_1\}\{a_2\}\{a_3\}\{a_4\}\}) = 2$	$w(\{a_2, a_4\}, \{\{a_2, a_4\}\{a_1, a_3\}\}) = 4$
$w(\{a_3\}, \{\{a_1, a_2\}\{a_3\}\{a_4\}\}) = 1$	$w(\{a_3, a_4\}, \{\{a_3, a_4\}\{a_1\}\{a_2\}\}) = 4$
$w(\{a_3\}, \{\{a_1, a_4\}\{a_2\}\{a_3\}\}) = 2$	$w(\{a_3, a_4\}, \{\{a_3, a_4\}\{a_1, a_2\}\}) = 3$
$w(\{a_3\}, \{\{a_2, a_4\}\{a_1\}\{a_3\}\}) = 1$	$w(\{a_1, a_2, a_3\}, \{\{a_1, a_2, a_3\}\{a_4\}\}) = 5$
$w(\{a_3\}, \{\{a_1, a_2, a_4\}\{a_3\}\}) = 0$	$w(\{a_1, a_2, a_4\}, \{\{a_1, a_2, a_4\}\{a_3\}\}) = 5$
$w(\{a_4\}, \{\{a_1\}\{a_2\}\{a_3\}\{a_4\}\}) = 1$	$w(\{a_1, a_3, a_4\}, \{\{a_1, a_3, a_4\}\{a_2\}\}) = 6$
$w(\{a_4\}, \{\{a_2, a_3\}\{a_1\}\{a_4\}\}) = 0.5$	$w(\{a_2, a_3, a_4\}, \{\{a_1, a_3, a_4\}\{a_1\}\}) = 8$
$w(\{a_4\}, \{\{a_1, a_3\}\{a_2\}\{a_4\}\}) = 0.5$	$w(\{a_1, a_2, a_3, a_4\}, \{\{a_1, a_2, a_3, a_4\}\}) = 7$
$w(\{a_4\}, \{\{a_1, a_2\}\{a_3\}\{a_4\}\}) = 0.5$	height

Fonte: Adaptado de [Rahwan et al., 2012]

demais estruturas de coalizão e impactar no resultado. O tamanho da entrada está diretamente ligado ao número de agentes envolvidos.

2.5 Grafos

As definições de grafos apresentadas nesta seção são baseadas em Diestel [1997]. É fornecida uma introdução aos conceitos da teoria dos grafos, que fornece as ferramentas necessárias no decorrer deste trabalho para juntar os conceitos de formação de coalizões com a caracterização combinatória de rede(grafos).

Um grafo simples não direcionado G é um par $G = (V, E)$ onde V é um conjunto finito de elementos chamado de vértices de G , e E é uma coleção de dois elementos de V que representam as conexões entre os vértices, chamados de arestas de G . O número de vértices de um grafo é representado por v e o número de arestas por e .

- Um grafo completo é um grafo em que cada par de vértices é conectado por uma aresta. O grafo completo com v vértices é denotado por K_v . O número máximo de arestas em um grafo completo K_v é dado pela equação 2.6.

$$\frac{n(n-1)}{2} \quad (2.6)$$

- Um *subgrafo* H de um grafo G é induzido se para qualquer par de vértice v e u de H , $\{v, u\}$ é uma aresta de H se e somente se é uma aresta de G , ou seja, H é um subgrafo

induzido de G se ele tem exatamente as arestas que aparecem em G sobre o conjunto de vértices.

- Um grafo G é um grafo *bipartido* se os seus vértices podem ser divididos em dois conjuntos disjuntos V_1 e V_2 de modo que cada aresta conecta um vértice em V_1 para um em V_2 . Um grafo *bipartido* completo, $G = (V_1 \cup V_2, E)$, é um grafo *bipartido* tal que para qualquer dois vértices, $v_1 \in V_1$ e $v_2 \in V_2$, $\{v_1, v_2\}$ formam uma aresta em G . Um grafo *bipartido* completo com $|V_1| = m$ e $|V_2| = n$, é denotado por $K_{m,n}$
- Um *caminho* em um grafo é uma sequência de vértices ligados por arestas.
- O *grau* do vértice $v \in V$, onde v é um vértice, de um grafo $G = (V, E)$ é o número de vértices adjacentes a v . Um grafo é dito regular de grau r quando todos seus vértices possuem o mesmo grau r . Quando um vértice não possui nenhuma aresta adjacente a ele o mesmo é dito isolado.
- Um *ciclo* de um grafo é o caminho formado do seu vértice de partida sendo também o mesmo vértice de destino e comprimento maior que três, formalizando temos: dado que v é um vértice onde $v \in V$ então, v_1, \dots, v_k, v_{k+1} sendo que $v_1 = v_{k+1}$ e $k \geq 3$, por exemplo: um triângulo é um ciclo de comprimento 3(três).
- Uma *clique* de um grafo G é um subgrafo completo de G . O *número clique* $w(G)$ é a cardinalidade da clique máxima, uma clique máxima é aquela de maior cardinalidade possível do grafo;
- Um *conjunto independente* de um grafo G é um subconjunto de vértices no qual não existam dois vértices adjacentes.
- Uma *árvore* é um grafo conexo que não contém ciclos.
- Uma *árvore geradora* de um grafo G é qualquer árvore de G que contenha todos os vértices de G . É claro que todo grafo dotado de árvore geradora é conexo, reciprocamente, todo grafo conexo tem, pelo menos, uma árvore geradora.
- Um *grafo estrela* $K_{n,n-1}$ é um grafo bipartido de n vértices que possui um conjunto independente com um único vértice e o outro com $n - 1$ vértices, exemplo figura 2.1.

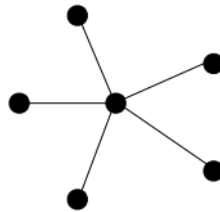


Figura 2.1: Grafo estrela $k_{5,1}$.

2.6 Considerações

Neste capítulo foram apresentados os conceitos básicos e necessários para a sequência dos experimentos e resultados que serão apresentados nos próximos capítulos. O principal ponto deste trabalho visa utilizar os conceitos existentes na teoria dos grafos como um auxílio na entrada de dados para os algoritmos existentes de formação de coalizões. Como discutido no próximo capítulo um dos pontos críticos para esses algoritmos é o tamanho exponencial da entrada. Os conceitos presentes na teoria dos grafos podem nos fornecer alguma aproximação para utilizar como restrição para esse problema.

Capítulo 3

Partição em Grafos Para Cálculo de Coalizões

Geração de estruturas de coalizão é equivalente ao problema de particionamento completo de conjunto Yun Yeh [1986], um dos problemas fundamentais em otimização combinatória, que tem aplicações em diversos campos: ciências da computação, ciências políticas sociais e econômicas e pesquisa operacional.

Em um problema de geração de estruturas de coalizão, temos um conjunto N de n elementos e uma função de valoração $v : P(N) \rightarrow \mathbb{R}$, onde $P(N)$ denota o conjunto potencia de N , e o problema é dividir o conjunto dado em subconjuntos disjuntos exaustivos (ou, coalizões) N_1, \dots, N_m para que o total da soma dos valores, $\sum_{i=1}^m v(N_i)$, seja maximizada. Dessa maneira buscamos encontrar a partição com maior valor (ou, uma estrutura de coalizão) sobre o conjunto N .

Segundo Voice et al. [2012a], o problema de geração de estruturas de coalizão se tornou um importante tema de pesquisa em inteligência artificial e sistemas multi agentes, como uma ferramenta para agentes autônomos formarem equipes eficientes.

O modelo clássico que trata o problema de geração de estruturas de coalizão não considera o conjunto de elementos como uma estrutura, mas somente como itens isolados. Apresentamos agora uma revisão do estado da arte dos principais algoritmos existentes para a formação de coalizões.

3.1 Algoritmos para formação de estruturas de coalizões

Segundo Rahwan and Jennings [2008b], o problema de geração de estruturas de coalizão pode ser resolvido através de três diferentes classes de algoritmos, conforme:

- **Algoritmos baseados em heurísticas:** algoritmos desta classe geralmente estabelecem restrições na criação de coalizões. A vantagem é a velocidade com que geram uma solução válida. Porém, os algoritmos desta classe não conseguem garantir a geração de soluções ótimas e, mesmo que uma solução ótima venha a ser encontrada, os mesmos não são capazes de verificar este fato. Alguns trabalhos que utilizam heurística para tratar a entrada dos dados para a geração de coalizão são: Ramos and Bazzan [2012] e Ramos et al. [2014].

- **Algoritmos baseados em programação dinâmica:** criados para o problema de particionamento de conjuntos, o qual é parecido com o de geração de estruturas de coalizão, a grande vantagem é na garantia de encontrar a solução ótima em $O(3^n)$, onde n é o número de agentes, mas o grande problema é a quantidade de memória utilizada para o processamento dos dados.
- **Algoritmos *anytime*:** Essa classe une características de ambas abordagens já tratadas anteriormente. Funcionam gerando soluções iniciais dentro de um limite do ótimo, que então são melhoradas enquanto houver tempo disponível. Embora não gerem a solução ótima de maneira rápida, geram soluções aceitáveis em um tempo curto.

3.1.1 Algoritmo CSS1 - *Coalition-Structure-Search-1*

O *CSS1* Sandholm et al. [1999] é o que pode ser considerado o primeiro algoritmo *anytime* criado para a geração de estruturas de coalizão, estabelecendo um limite de qualidade para a melhor solução até um determinado momento Rahwan [2007]. O algoritmo funciona da forma de representar o problema de geração de estruturas de coalizão como uma busca no grafo de estruturas de coalizão, figura 3.1.

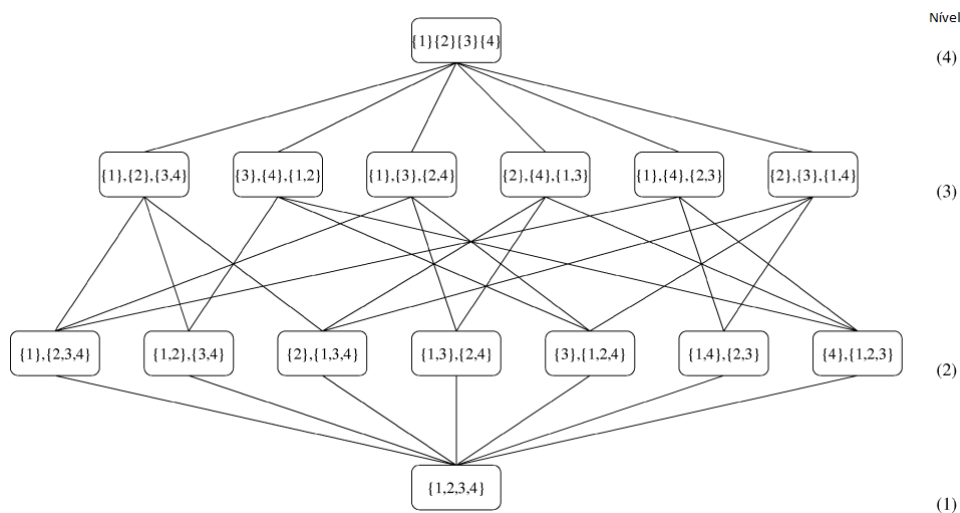


Figura 3.1: Grafo de estruturas de coalizão para quatro agentes.

Fonte: (Adaptado de Sandholm et al. [1999])

De acordo com Sandholm et al. [1999] é possível estabelecer um limite $\beta = \alpha$ da solução ótima através da busca nos dois níveis inferiores do grafo de estruturas de coalizão, conforme ele apresenta no teorema 3.1.1 Sandholm et al. [1999].

Para estabelecer tal limite é necessário que se percorra todas as coalizões possíveis. Esta busca pode ser realizada nos dois níveis inferiores do grafo, pois nesta porção do espaço de busca cada coalizão aparece exatamente uma única vez, ou seja, nenhum outro limite pode ser melhor para esta porção de busca. Portanto foi provado que nenhum algoritmo pode estabelecer um limite sem efetuar uma busca em pelo menos 2^{a-1} estruturas de coalizão, onde a é o número de agentes.

Teorema 3.1.1 (Busca mínima para estabelecer um limite) Para limitar β , é suficiente realizar uma busca nos dois níveis inferiores do grafo de estruturas de coalizão. Desta maneira, o limite é $\beta = \alpha$ e o número de vértices visitados é 2^{a-1} .

Ainda em Sandholm et al. [1999], é definido que a cada novo nível que o grafo é pesquisado o limite pode ser reduzido, a busca adicional deve iniciar pelo nível superior e ser executada de forma descendente. O motivo desta melhora é definida no teorema 3.1.2 Sandholm et al. [1999]. Sobre esse fato, ao realizar a busca no nível superior o limite é reduzido pela metade, ou seja, $\beta = \frac{a}{2}$. A cada dois novos níveis percorridos, o divisor no limite é acrescido de uma unidade.

Teorema 3.1.2 (Diminuindo o limite com busca adicional) Após realizar uma busca no nível l , o limite β é $\lceil \frac{a}{h} \rceil$ se $a \equiv (h-1)$ e $a \equiv l \pmod{2}$. Caso contrário o limite β é $\lfloor \frac{a}{h} \rfloor$, onde $h = \lfloor \frac{a-l}{2} \rfloor + 2$.

Estes dois teoremas, [Sandholm et al., 1999], deram origem ao algoritmo **CSS1**, *Coalition-Structure-Search-1*. A ideia do algoritmo é basicamente definir uma solução inicial dentro do limite $\beta = a$ e, enquanto houver tempo para continuar rodando, utilizar uma busca em largura para otimizar a solução, ou seja, o algoritmo pode ser interrompido a qualquer momento, que uma solução com um limite aceitável e confiável dentro do seu ótimo pode ser retornado. O algoritmo *CSS1* é apresentado no algoritmo 1.

Algoritmo 1 Algoritmo CSS1 para geração de estruturas de coalizão

Pesquisar os dois níveis inferiores do grafo de estruturas de coalizão

Efetuar busca em largura a partir do nível superior do grafo de estruturas de coalizão enquanto houver tempo ou até que o grafo inteiro tenha sido percorrido.

Retornar a estrutura de coalizão com maior valor encontrada

Sandholm et al. [1999] apresenta resultados teóricos sobre os estudos realizados sobre seu algoritmo, fazendo uma análise das garantias de desempenho no pior caso, seu trabalho foi pioneiro em provar que um limite inferior pode ser estabelecido após a busca em um número mínimo de estruturas de coalizão.

De acordo com Rahwan [2007] o algoritmo *CSS1* acaba necessitando uma busca muito grande no grafo de estruturas de coalizão, considerando um número de 24 agentes, após efetuar uma busca nos níveis 1, 2 e 24, o que corresponde a 8×10^6 estruturas de coalizão, temos $\beta = 12$, ou seja, mesmo que a busca tenha sido grande a solução que foi encontrada seria, no pior caso, 12 vezes pior que a solução ótima, o que significa que, a garantia é que a solução encontrada não seja pior que 8,3% da solução ótima. Isso quer dizer que tomar como base somente as garantias apresentadas, pode acabar não sendo a melhor alternativa para trabalharmos com cenários reais utilizando essa abordagem.

3.1.2 Algoritmo DP

O algoritmo *DP*, ou do inglês *Dynamic Programming*, foi proposto por Yun Yeh [1986]. Projetado para resolver o problema de particionamento de conjuntos. Sandholm et al. [1999] comenta que o problema de geração de estruturas de coalizão é semelhante ao problema de particionamento de conjuntos, pois ambos os problemas tem como objetivo particionar um

conjunto de elementos em subconjuntos de forma a maximizar os valores, bem estar social, dos agentes envolvidos.

O algoritmo possui duas etapas: Solucionar os subproblema para assim poder solucionar o problema principal, cada subproblema deve ser solucionado individualmente. Duas tabelas são mantidas em memória: f_1 , que contém a solução de cada coalizão; e f_2 , que contém o valor da função característica de cada coalizão. A execução acontece pelas coalizões de tamanho 1, como uma coalizão com apenas um elemento não pode ser dividida, a solução é ela mesma, ou seja, $f_1[C] = C$ e por consequência $f_2[C] = v(C)$. Após as coalizões de tamanho 1, as coalizões seguintes são as de tamanho 2, 3 e assim sucessivamente até a coalizão de tamanho máximo, ou seja, a quantidade total de agentes $|A|$.

Algoritmo 2 Algoritmo *DP* para geração de estruturas de coalizão

```

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

```

Nas coalizões de tamanho maior que dois sempre existe ao menos uma divisão possível a ser realizada, por exemplo, $C_1 = \{a_1, a_2, a_3\}$, pode ser dividida de três formas diferentes: $\{\{a_1\}\{a_2, a_3\}\}$, $\{\{a_2\}\{a_1, a_3\}\}$ ou $\{\{a_3\}\{a_1, a_2\}\}$. Supondo que a primeira coalizão tivesse o maior valor entre as três, então $f_1[\{a_1, a_2, a_3\}] = \{\{a_1\}\{a_2, a_3\}\}$ e consequentemente, $f_2[\{a_1, a_2, a_3\}] = v(\{a_1\}) + v(\{a_2, a_3\})$. Temos que verificar que os subproblemas dever ser resolvidos de forma ascendente, para que começando com coalizões de tamanho 1 até $|A|$.

Após todo o processo sobre os subproblemas o algoritmo inicia a solução do problema original. Assume-se, inicialmente, que a coalizão de tamanho $|A|$ é a CS^* , ou seja, a coalizão ótima. Então de forma recursiva se substitui cada coalizão $C \subseteq CS^*$ por sua solução f_1 , o encerramento do processo é quando $C = f_1[C]$ para todo $C \subseteq CS^*$.

Assim o algoritmo *DP* é capaz de encontrar a estrutura de coalizão com complexidade de $O(3^a)$ [Yun Yeh, 1986], onde a é a quantidade de agentes, vide algoritmo 2.

3.1.3 Algoritmo *IP*

O algoritmo *IP*, ou partição de inteiros, do inglês *Integer Partition*, proposto por Rahwan [2007], refere a classe de algoritmos *anytime* para a geração de estruturas de coalizão. O algoritmo é baseado em uma representação sobre as estruturas de coalizão. O algoritmo é capaz de realizar uma busca direcionada e eficiente, garantindo redução no tempos de execução e na complexidade na comparação com outras abordagens, como será apresentado na sequência.

A representação espaço de busca é baseado no agrupamento de estruturas de coalizão em configurações. A configuração de uma estrutura de coalizão explicita o tamanho das coalizões que a compõem. Como exemplo, ambas as estruturas de coalizão $\{\{1\}, \{2, 3\}\}$ e $\{\{3\}, \{1, 2\}\}$ são compostas por uma coalizão com um elemento e por uma coalizão com dois elementos; ou seja, ambas possuem a mesma configuração $[1, 2]$, as coalizões são agrupadas por tamanho.

As definições utilizadas pelo algoritmo *IP*. Seja $CL_s \in (CL)$ o conjunto de coalizões de tamanho $s \in 1, 2, \dots, a$. Seja $G_1, G_2, \dots, G_{|G_{CS}|} \in G_{CS}$ o conjunto de todas as configurações possíveis. Seja $F : G_{CS} \rightarrow 2^{CS}$ uma função que recebe uma configuração e retorna todas as estruturas de coalizão daquela, tal que $N = \{F(G_1), F(G_2), \dots, F(G_{|G_{CS}|})\}$ representa uma lista onde cada elemento é um conjunto de estruturas de coalizão de mesma configuração, ou seja, $N_1, N_2, \dots, N_{G_{CS}} \in N$ denota os elementos que pertencem ao conjunto N , é possível observar um exemplo prático na figura 3.2.

De forma simplificada o algoritmo aplica técnicas de divisão e conquista ao espaço de busca, definindo limites para as configurações e retirando as estruturas que possuem pouca possibilidade de serem a solução ótima, os limites são refinados sempre que a busca é aprofundada, desta forma a poda cada vez maior do espaço de busca, então direcionada para um espaço de busca relativamente pequeno.

O melhor desempenho do algoritmo é obtido quando a busca é direcionada a solução com um limite pré-definido em relação ao ótimo, o limite é ajustado pelo parâmetro β . Quando o $\beta = 1$ significa que se deseja encontrar uma solução ótima, ou seja, quanto mais perto de 1 mais próximo do ótimo se busca.

O algoritmo *IP* possui três etapas:

1. **Pré-processamento:** nesta fase são calculados os valores $v(C)$ máximo e médio de cada lista, para todos $CL_s \in CL$. Esta fase envolve a busca em todas as coalizões possíveis.
2. **Escolha da configuração ótima:** utilizando os valores obtidos na etapa anterior para gerar a lista $G \in G_{CS}$ e selecionar o elemento $F(G)$ que tem a maior probabilidade de conter a estrutura de coalizão ótima CS^* . Somente uma configuração é selecionada, da qual é possível analisar todas as CS que a compõem na fase seguinte.
3. **Busca da CS^* :** são determinados os valores $V(CS) \in F(G)$ para encontrar CS^* , esta fase é a que possui o maior custo computacional.

Rahwan [2007] demonstra empiricamente, a qualidade do algoritmo *IP* comparado com o algoritmo *DP*; Apresentando um cenário com 20 agentes o algoritmo *IP* é capaz de obter resultados que se encontram, no pior caso, no limite de 1% da solução ótima, utilizando 0,0043% do tempo e 77% da memória utilizada pelo algoritmo *DP*.

$$\begin{aligned}
G_1 &= [4], F(G_1) = N_1 = \{\{a_1, a_2, a_3, a_4\}\} \\
G_2 &= [2, 2], F(G_2) = N_2 = \left\{ \begin{array}{l} \{\{a_1, a_2\}, \{a_3, a_4\}\} \\ \{\{a_1, a_3\}, \{a_2, a_4\}\} \\ \{\{a_1, a_4\}, \{a_2, a_3\}\} \end{array} \right\} \\
G_3 &= [1, 3], F(G_3) = N_3 = \left\{ \begin{array}{l} \{\{a_1\}, \{a_2, a_3, a_4\}\} \\ \{\{a_2\}, \{a_1, a_3, a_4\}\} \\ \{\{a_3\}, \{a_1, a_2, a_4\}\} \\ \{\{a_4\}, \{a_1, a_2, a_3\}\} \end{array} \right\} \\
G_4 &= [1, 1, 2], F(G_4) = N_4 = \left\{ \begin{array}{l} \{\{a_1\}, \{a_2\}, \{a_3, a_4\}\} \\ \{\{a_1\}, \{a_3\}, \{a_2, a_4\}\} \\ \{\{a_1\}, \{a_4\}, \{a_2, a_3\}\} \\ \{\{a_2\}, \{a_3\}, \{a_1, a_4\}\} \\ \{\{a_2\}, \{a_4\}, \{a_1, a_3\}\} \\ \{\{a_3\}, \{a_4\}, \{a_1, a_2\}\} \end{array} \right\} \\
G_5 &= [1, 1, 1, 1], F(G_5) = N_5 = \{\{a_1\}, \{a_2\}, \{a_3\}, \{a_4\}\} \\
G_{CS} &= \{G_1, G_2, G_3, G_4, G_5\} \\
N &= \{N_1, N_2, N_3, N_4, N_5\}
\end{aligned}$$

Figura 3.2: Representação do espaço de busca realizada pelo algoritmo IP. O conjunto da agentes $A = a_1, a_2, a_3, a_4$ e os elementos de N foram expandidos. Exemplo adaptado para melhor entendimento de Rahwan [2007].

Mesmo sendo considerado como um dos algoritmos do estado da arte para a geração de estruturas de coalizão, quando consideramos n como sendo o número de agentes, o algoritmo *IP* apresenta uma complexidade de $O(n^n)$, essa complexidade é maior que apresentada pelo algoritmo *IDP* ($O(3^n)$), um dos algoritmos que representa o estado da arte em programação dinâmica.

3.1.4 Algoritmo *IDP*

Uma versão melhorada do algoritmo *DP* é apresentada em Rahwan and Jennings [2008b]. O objetivo do algoritmo é reduzir a quantidade de divisões avaliadas sem que isso comprometa a busca pela solução ótima.

Rahwan and Jennings [2008b] fizeram valer uma observação sobre o grafo de estruturas de coalizão, estabelecendo que cada vértice necessita de apenas uma aresta direcionada a ele para garantir que a solução ótima seja encontrada. Isso é válido pois o algoritmo *DP* pode percorrer qualquer dos caminhos para encontrar a solução, mesmo que exista um único caminho, sendo assim não há a necessidade de se manter os vários caminhos. Desta maneira, eliminando os caminhos, reduz de maneira considerável a quantidade de divisões de uma coalizão a se avaliar. A figura 3.3 demonstra esse conceito, as arestas em destaque apresentam diferentes caminhos que podem ser percorridos para se alcançar a solução ótima. Mas como

pode ser observado, se existi-se somente um caminho a solução seria encontrada da mesma forma.

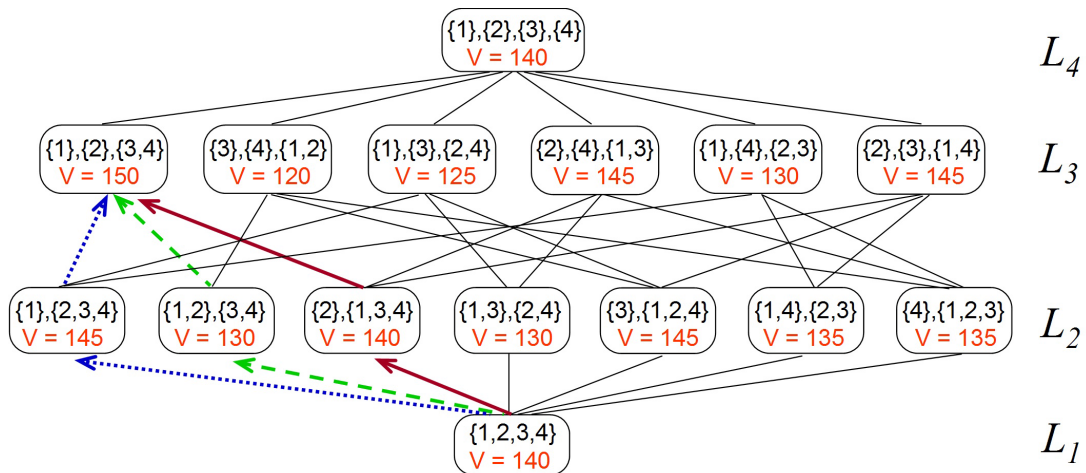


Figura 3.3: Grafo de estruturas de coalizão para quatro agentes. As arestas em destaque mostram todos os caminhos que o DP poderia percorrer para encontrar a solução ótima. O IDP, elimina a maioria destas arestas, de forma que cada vértice contenha apenas uma aresta direcionada para ele (no sentido do nível inferior para anterior) Rahwan and Jennings [2008b].

Observando este conceito, Rahwan and Jennings [2008b] define E^* como o conjunto de todas as arestas essenciais para a representação do problema, desconsiderando caminhos redundantes, com base em E^* , cada nodo tem exatamente uma aresta direcionada para ele. O conjunto E^* pode ser encontrado com base na equação 3.1.4:

$$E^* = \left(\bigcup_{s'' \leq n - (s' + s'')} E^{s's''} \right) \cup \left(\bigcup_{s' + s'' = n} E^{s's''} \right) \quad (3.1)$$

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

Com base nessa equação, Rahwan and Jennings [2008b] introduz o teorema que para cada estrutura de coalizão CS contendo um número s de coalizões ($s \in \{2, \dots, n\}$), onde n é o número total de agentes, existe uma aresta em E^* que conecta CS com outra estrutura de coalizão contendo $(s - 1)$ coalizões. Com base neste teorema, foi provado em Rahwan and Jennings [2008b] que a parte, $\left(\bigcup_{s' + s'' = n} E^{s's''} \right)$, da equação 3.1.4, engloba todas as arestas da coalizão de tamanho n até as coalizões de tamanho $(n - 1)$, já a parte $\left(\bigcup_{s'' \leq n - (s' + s'')} E^{s's''} \right)$, garante que apenas uma aresta chega a cada vértice. A prova com todos os detalhes deste teorema é encontrado no trabalho Rahwan and Jennings [2008b].

Os algoritmos de *IDP* e *DP* são basicamente os mesmos, com uma pequena diferença, esta diferença é para considerar somente as arestas em E^* . A mudança corresponde à validação ($|C''| \leq n - s$ ou $|C'| + |C''| = n$), que adapta a equação ao algoritmo *DP*. O algoritmo *IDP* é apresentado no algoritmo 3.

Com essa diferença aparentemente pequena entre os algoritmos *DP* e *IDP*, os experimentos apresentados em Rahwan and Jennings [2008b] demonstram que o desempenho do

algoritmo *IDP* é bem superior ao *DP*. Para um experimento com 25 agentes existe uma redução entre 33,3% e 66,6% na utilização da memória, complementar a isso o *IDP* executa apenas 38,7% das operações realizadas pelo *DP*, considerando o mesmo cenário. Com isso é possível afirmar que o algoritmo *IDP* representa o estado da arte para a geração de estruturas de coalizão através da programação dinâmica junto com a programação linear.

Algoritmo 3 Algoritmo IDP para geração de estruturas de coalizão

```

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

```

O algoritmo *IDP* foi melhorado em Rahwan and Jennings [2008a] incorporado a elementos do algoritmo *IP*, chamado de *IDP-IP*, esse algoritmo remove um número maior de arestas do grafo de estruturas de coalizão que o *IDP*, tornando o grafo desconexo, gerando assim vários subespaços. O algoritmo *IP* aplica técnicas de divisão e conquista, podando espaços e encontrando a solução ótima. Em alguns casos, o *IDP-IP* é capaz de encontrar a solução ótima em um tempo menor que o *IP* consumindo uma menor quantidade de memória que o *IDP*.

3.1.5 Algoritmo $IP^{+/-}$

Este algoritmo é baseado no algoritmo *IP*, proposto por Rahwan [2007] que leva em consideração as externalidades do ambiente para a geração das estruturas de coalizão. Externalidades podem ser entendidas como objetivos conflitantes ou sobrepostos entre os agentes que estão envolvidos no sistema. Isso não é levado em conta em outras abordagens dos algoritmos de geração de estruturas de coalizão, mesmo sendo uma situação muito comum no mundo real, pois a formação de uma coalizão pode muitas vezes influenciar no comportamento das demais coalizões existentes.

O algoritmo $IP^{+/-}$ apresentado em Rahwan et al. [2012] é o primeiro algoritmo de geração de estruturas de coalizão a levar em conta os fatores externos a coalizão para o seu

processamento. Isso por um simples fato, o problema de geração de estruturas de coalizão já é complexo, como mostrado anteriormente na demonstração dos outros algoritmos, devido ao tamanho exponencial do espaço de busca, mas em particular pois quando são levadas em conta as externalidades a complexidade dado n agentes passa de $O(2^n)$ para $O(n^n)$.

3.1.6 Algoritmo CCF - Formação de coalizões com restrições

O primeiro estudo sistemático sobre formação de coalizões envolvendo restrições foi realizado por Rahwan et al. [2011] - *constrained coalition formation (CCF)*.

Diferente do modelo convencional que considera todos os subconjuntos possíveis do conjunto de agentes como uma potencial coalizão, o trabalho de Rahwan et al. [2011], discorre a ideia de que em muitas aplicações do mundo real existem limitações que podem ser aplicadas sobre as coalizões possíveis, como exemplo: as estruturas de coalizão devem ter o mesmo tamanho, ou um agente pode estar impedido de estar com outro agente na mesma coalizão.

No trabalho apresentado por Rahwan et al. [2011] é descrito um processo que converte as restrições em uma entrada de dados estruturada de forma que permite aos algoritmos de formação de coalizões identificar, sem qualquer redundância computacional todas as coalizões factíveis. Isso é utilizado como entrada para o algoritmo encontrar qual a melhor estrutura de coalizão com restrição, além de demonstrar que essa abordagem supera as demais existentes.

O ponto chave do algoritmo desenvolvido por Rahwan et al. [2011] é a capacidade de se evitar a verificação de todas as coalizões possíveis (2^n dado n agentes) para verificar somente as factíveis, ou seja, se a coalizão satisfaz ou não as restrições.

Isso é realmente crucial, porque o número de coalizões factíveis pode ser significativamente menor do que $2^n - 1$ e, como cada uma dessas coalizões requer um número grande de operações, as que não são factíveis fazem que essas operações não sejam executadas desnecessariamente.

Outros desafios encontrados no desenvolvimento por Rahwan et al. [2011] foi evitar cálculos redundantes, por exemplo, geração de múltiplas coalizões que satisfazem várias restrições ao mesmo tempo, outro desafio importante em *CCF* é a forma como as coalizões factíveis podem ser combinadas para gerar estruturas de coalizão com estas coalizões.

3.1.7 Algoritmo DyCE

O algoritmo *DyCE* proposto por Voice et al. [2012b], usa programação dinâmica para encontrar a estrutura de coalizão ótima através de progressivas divisões sobre a solução corrente da melhor partição. Seu modo de operação é bastante similar ao algoritmo *IDP*, entretanto para melhorar a velocidade da busca pela estrutura de coalizão ótima usa o algoritmo *SlyCE* para enumerar todas as coalizões admissíveis, ignorando as coalizões impossíveis durante a busca Voice et al. [2012b].

Resolve o problema de formação de coalizões em grafos sinérgicos (*Synergistic graph*) através da exploração das arestas em um grafo de estruturas de coalizão. Entretanto, *DyCE* requer uma quantidade exponencial de memória em relação ao número de agentes, $O(2^n)$, e não possui uma abordagem *anytime*. Arestas entre os agentes representam algum link de sinergia vital: comunicação, confiança ou restrições físicas.

Uma coalizão é dita impossível se os membros não formam um subgrafo conexo, o que significa que partes da coalizão são isoladas das outras. No trabalho são propostos dois

algoritmos chamados *D-SlyCE* e *DyCE*, para enumerar todas coalizões admissíveis de forma distribuída e encontrar a estrutura de coalizão ótima respectivamente.

3.1.8 Algoritmo CFSS

Bistaffa et al. [2014a] propõem o algoritmo *CFSS*. Baseado na contração de arestas, uma operação básica em teoria dos grafos que envolve: (i) remover uma aresta em um grafo e (ii) juntar os dois nós que eram anteriormente unidos pela aresta. Sobre essa operação o algoritmo repete o processo de contração de diferentes arestas, a fim de eventualmente visitar todas as estruturas de coalizão. Para aumentar a velocidade de busca, uma técnica *branch-and-bound* é usada sempre que o algoritmo visita um vértice.

Os resultados apresentados pelo algoritmo *CFSS* mostram que quando os grafos são muito esparsos, o algoritmo é 4 ordens de grandeza mais rápido quando comparado ao algoritmo *DyCE*. Além disso, *CFSS* é a primeira abordagem que fornece uma solução aproximada *anytime* com garantia de qualidade para sistemas muito grandes, exemplo: sistemas com mais de 2.700 agentes Bistaffa et al. [2014a].

3.1.9 Algoritmo DyPE

Vinyals et al. [2013] propõem um algoritmo de programação dinâmica chamado *DyPE*. Antes de explicar como o algoritmo trabalha vamos primeiro brevemente descrever como a programação dinâmica funciona para jogos gerais, em vez de jogos de restrição em grafos. Aqui a ideia principal: para computar uma partição ótima do conjunto de agentes, A :

- Primeiro, computar uma partição ótima de cada subconjunto estrito de A .
- Depois disso, examinar todas as possíveis formas de divisão de A em duas metades, e substituir uma das metades com sua partição ótima.

O pseudo código do algoritmo pode ser observado no algoritmo 4.

Algoritmo 4 Algoritmo $DyPE(v(\cdot), A, G, O)$

```

1:  $C \leftarrow \emptyset$  // Subproblema atual
2:  $S \leftarrow \emptyset$  // Subespaço atual
3:  $i \leftarrow |A|$  // Começa a explorar o ultimo agente na ordenação  $O$ 
4: while  $(C, i) \leftarrow \text{nextSubproblem}(C, i)$  do
5:    $P[C] \leftarrow -\infty$ 
6:   while  $C' \leftarrow \text{nextConnectedSet}(C', i, C)$  do
7:      $V \leftarrow v(C') + \sum_{C'' \in \Phi(C \setminus C')} P[C'']$ 
8:     if  $P[C] < V$  then
9:        $P[C] \leftarrow V$  // Atualiza o valor do subproblema
10:       $P[C] \leftarrow C'$  // Atualiza o melhor subespaço
11:     end if
12:   end while
13: end while
14: return  $bestCS(A)$ 

```

Algoritmo 5 Algoritmo *nextSubproblem*(C, i)

```

1: if  $i = 1$  then
2:   if  $C = \emptyset$  then
3:     return  $(A, i)$  // Somente a grande coalizão
4:   end if
5:   return  $\emptyset$ 
6: else
7:   while  $C \leftarrow \text{nextConnectedSet}(C, i, \{i, \dots, |A|\})$  do
8:     if  $A \setminus C(\text{conexo})$  then
9:       return  $(C, i)$ 
10:    end if
11:   end while
12:   return nextSubproblem( $\emptyset, i - 1$ ) // Chama recursivamente com o agente anterior da ordenação
13: end if

```

3.1.10 Algoritmo TSP

O algoritmo proposto em Rahwan and Michalak [2014] é uma junção dos algoritmos *DyPE* Vinyals et al. [2013] com o *CFSS* Bistaffa et al. [2014a]. Ambos algoritmos são baseados em ideias muito diferentes; um é baseado na contração de arestas, enquanto o outro é baseado em uma pseudo árvore.

O nome do algoritmo TSP é um acrônimo onde TS para *Tree-Search* e P para *Pseudotree*.

3.2 Comparação de algoritmos

CFSS é o algoritmo *anytime* considerado o estado da arte. A solução melhora quanto mais tempo é dado ao algoritmo trabalhar sobre os resultados.

DyPE é superior em termos de complexidade computacional.

Por exemplo, dado um grafo completo de n agentes, *DyPE* roda em $O(3^n)$, enquanto *CFSS* roda em $O(n^n)$. Isto porque *CFSS* depende muito da técnica *branch-and-bound* que, no pior dos casos, pode falhar para podar até mesmo uma única solução, resultando numa pesquisa de força bruta.

Na tabela 3.1 observamos uma comparação superficial dos algoritmos citados neste capítulo.

Tabela 3.1: Algoritmos de formação de coalizões

Sigla	Tipo	Técnica	Nome	Grafo	Tempo	Ano	Intuição	Observações	Limites
CSS1	anytime	-	Coalition-Structure-Search-1	Completo	-	1999	Percorrer os dois níveis inferiores do grafo de estruturas de coalizão.	-	-
DP		Dynamic Programming	Dynamic Programming	Completo	$O(3^n)$	1988	Programação dinâmica	Representação na forma de matrix.	Requer uma significante quantidade de memória.
IP	anytime	Integer programming	Integer Programming	Completo	$O(3^n)$	2008 ~ 2009	(i) Uma nova representação do espaço de busca e eficiente estratégia de busca para explorar a representação; (ii) Utilização da técnica <i>branch-and-bound</i> que pode rapidamente identificar as melhores estruturas de coalizão durante a busca e podar uma grande porção das que não são ótimas.	Encontrar solução ótima em 0,082% do tempo utilizado pelo algoritmo DP (para 27 agentes). Encontrar solução para 17 agentes em tempo razoável e para 27 agentes em menos de 90 minutos, enquanto o algoritmo DP pode levar 2 meses para a melhor solução.	Testes com 27 agentes em 4.89×10^3 seconds
IDP		Dynamic Programming	Improved Dynamic Programming	Completo	-	2008	-	Utiliza 38.7% das operações utilizadas pelo DP para 25 agentes e usa somente 33.3% da memória para o melhor caso e 86.8% no pior caso, quando comparado ao algoritmo DP.	-
IDP-IP	Híbrido	Dynamic Programming	Improved Dynamic Programming + Integer Programming	Completo	$O(3^n)$	2008 ~2012	-	Junção dos algoritmos IP e IDP	-
IP+		Integer programming	Integer Programming with Externalities	Completo	-	2012	-	Leva em consideração as externalidades dos agentes.	-
ODP	Híbrido	Dynamic Programming	Optimal Dynamic Programming	Completo	-	2013	-	Algoritmo DP otimizado	-
DyCE		Dynamic Programming	Dynamic programming for optimal connected coalition structure Evaluation	Subgrafo conexo	$O(2^n)$	2012	-	O primeiro a enumerar valores para coalizões válidas para mais de 50 agentes. O primeiro a encontrar a estrutura de coalizão ótima para mais de 30 agentes em minutos, enquanto os demais algoritmos levavam meses.	30 agentes; Não tem abordagem anytime.
CFSS	Anytime	-	Coalition Formation with Sparse Synergies	Subgrafo conexo	$O(n^4)$	2014	Analisa a formação de coalizões sobre os grafos sinérgicos, ou seja, os grafos que mais se aproximam da topografia encontrada em redes sociais, de comunicação e de confiança.	Conceito baseado em contração de arestas. Abordagem de busca branch and bound.	2.700 agentes com garantia de qualidade. Anytime
DyPE		Dynamic Programming e Branch and bound	Dynamic programming Pseudotree-based optimal coalition structure Evaluation	Subgrafo conexo	$O(3^n)$	2013	Utilização da técnica branch-and-bound	-	-
TSP	Híbrido	-	Tree-Search Algorithm	Subgrafo conexo	-	2014	CFSS + DyPE	Junção de duas ideias: uma baseada em contração de arestas e outra baseada em pseudo tree.	-

Fonte: Autoria própria.

3.3 Uso de restrição sobre as coalizões possíveis

O principal problema com as abordagens é assumir que todos os subconjuntos de agentes podem existir como coalizões. Dessa maneira, é possível analisar o seguinte exemplo: dado 100 agentes, mas somente 100 coalizões das $2^{100} - 1$ possíveis são admissíveis, os algoritmos teriam que passar por todas as $2^{100} - 1$ coalizões, com objetivo de validar e verificar cada uma das coalizões. Isso, na prática é inviável.

Recentemente, uma nova abordagem de enumeração de coalizões foi sugerida, em que o admissível de uma coalizão pode ser descrito usando uma abordagem em grafos, imitando o tipo de relacionamento que existe em redes de comunicação ou sociais Voice et al. [2012b].

A representação de coalizões admissíveis da origem ao problema de enumeração de subgrafos conexos, pelo que cada coalizão é um subgrafo conexo de um grafo e a enumeração de toda as coalizões requer a enumeração de todos os subgrafos de um grafo de relações agentes.

Assim Voice et al. [2012b] propuseram um algoritmo para resolver este problema de maneira eficiente e sem redundância além de sugerir técnicas para distribuir a computação entre os agentes da maneira mais justa possível. Esta técnica se mostrou adequada no processo de enumeração das coalizões em ordem de grandeza em particular quando os agentes estão ligados em estruturas do tipo árvore com baixo nível de ramificação.

O presente trabalho aborda o problema de formação de coalizões quando levado em conta as restrições apresentadas no grafo. Um exemplo utilizado tipicamente pelos trabalhos já apresentados é considerar que uma coalizão é válida somente quando o subgrafo de um grafo é um subgrafo conexo. Não existe um vértice (agente) isolado dentro de uma coalizão.

Ao considerar que somente subgrafos conexos são coalizões admissíveis o número de coalizões a serem geradas é realmente menor.

3.4 Considerações

Neste capítulo se apresentou uma breve descrição dos principais algoritmos de formação de coalizões. Foram apresentados algoritmos que não consideram nenhuma restrição, como [Sandholm et al., 1999] e também algoritmos como abordagens novas sobre o problema de formação de coalizões que somente consideram as coalizões factíveis, como [Bistaffa et al., 2014a; Voice et al., 2012b].

No próximo capítulo será analisada de qual forma esses algoritmos conseguem aplicar as restrições com a utilização de grafos e de que forma isso pode reduzir o número de coalizões a serem avaliadas para a geração de estruturas de coalizão.

Capítulo 4

Estruturação da Pesquisa

Este capítulo apresenta as etapas do método de pesquisa para o desenvolvimento deste trabalho com o propósito que o objetivo geral seja atingido.

Para o levantamento dos artigos relacionados a proposta deste trabalho, foram utilizados termos como: *coalition formation*, *game theory*, *coalition structure generation*, *graphs*, *constrained coalition formation*, *synergistic graph*, *Coalition structure generation over graphs*, *coalition structure generation on synergy graphs* e *coalition formation with sparse synergies*. Foram consideradas pesquisas nas bases *ACM Digital Library*, *IEEEEXplere Digital Library* e *CiteSeerX*, sendo as pesquisas realizadas entre os anos de 2013 e primeiro semestre de 2015. O processo de busca resultou em pouco mais que 75 artigos, sendo que, após uma seleção, de acordo com avaliação do autor deste trabalho, dezenas foram considerados como referência para este trabalho.

Durante os estudos iniciais e com o levantamento dos primeiros artigos foi observado o grande destaque do pesquisador *Talal Rahwan*¹. Foi realizado contato por *e-mail* com Rahwan, no qual auxiliou com dicas de artigos e disponibilizou em sua conta no *github* um programa na linguagem Java com a implementação de algoritmos de formação de coalizão. O programa se mostrou de grande importância no início da pesquisa.

De acordo com a hipótese apresentada neste trabalho é esperado que a utilização de restrições para a formação de coalizão reduza de maneira significativa o número de coalizões a serem analisadas pelos algoritmos de formação de coalizão. Como demonstrado na seção 2.2 o processo de formação de coalizões é exponencial ao número de agentes. Dessa maneira, é de fundamental importância tratar pontualmente o número de coalizões a serem analisadas.

A utilização de uma nova forma de abordagem sobre a formação de coalizões, se busca encontrar a maneira de reduzir a quantidade de coalizões a serem enumeradas para se encontrar a coalizão ótima, já que se considerado todas como factíveis a tarefa permanece complexa.

A outra parte deste trabalho consiste na apresentação dos conceitos básicos da teoria de grafos e coalizões, apresentados no capítulo 2. Os conceitos, trabalhos correlatos e informações contidas nos capítulos 2 e 3 constituem o conteúdo fundamental para a idealização da hipótese proposta. Por meio de um estudo inicial, a hipótese foi estabelecida conceitualmente, e experimentada por meio do método descrito neste capítulo.

A figura 4.1 exhibe as etapas do método. O método é composto por seis etapas que serão descritas a seguir:

¹Página pessoal <http://users.ecs.soton.ac.uk/tr/>

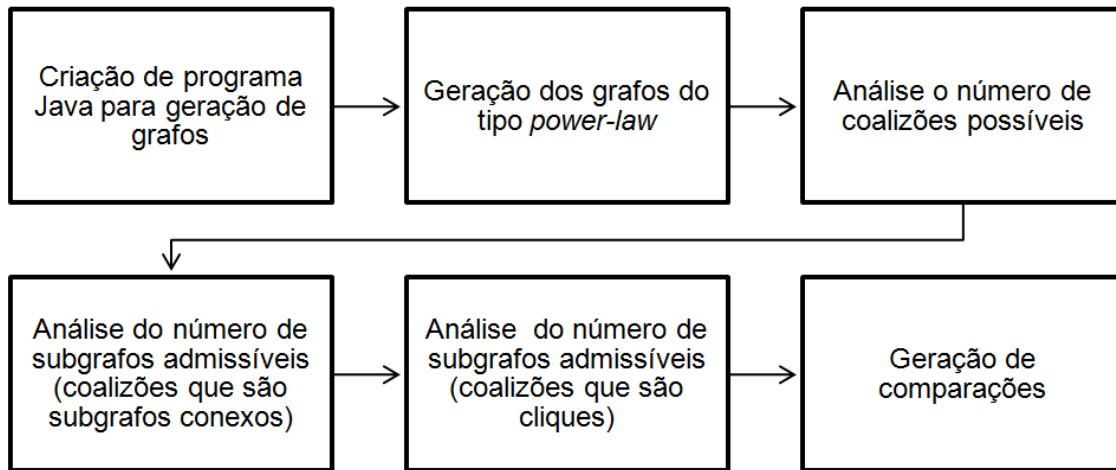


Figura 4.1: Etapas do método.

Fonte: Autoria própria.

4.1 Criação de Programa Java Para geração dos grafos e contagem de subgrafos

O objetivo desta etapa foi elaborar um programa para a geração dos grafos, codificado na linguagem Java² associado com a utilização da API disponibilizada pelo *software Gephi* Bastian et al. [2009], juntamente com esta API foi utilizado o *plugin Complex Generators*³, que facilitou a geração das redes complexas do tipo *power law*.

Os grafos do tipo *power law*, são redes geradas utilizando-se das leis de potência. Foi utilizado o padrão *Barabási-Albert* [Albert and Iászló Barabási, 2002], modelo de ligação preferencial, com parâmetros $K = 1, 2, 3$. Da maneira como o grafo G é construído, novos vértices são agregados aos k -vértices já existentes de tal forma que cada novo vértice é ligado a um vértice j que já existe com probabilidade $\frac{d_j}{\sum_j d_j}$ onde d_j é o grau do vértice j para todo $j \in G$.

Também foi realizada a implementação de um algoritmo para a completa enumeração de todos os subgrafos de um grafo, utilizada para enumerar os **subgrafos factíveis** para as hipóteses.

O programa é dividido em seis fases:

1. Geração de grafos do tipo *power law*;
2. Geração de coalizões: são geradas todas as coalizões possíveis para o conjunto de vértices do grafo.
3. Verificação se a coalizão atual é um subgrafo induzido conexo (não existe nenhuma restrição);
4. Verificar se a coalizão é uma clique;

²<http://www.oracle.com/technetwork/java/index.html>

³De acordo com as descrições disponíveis no sítio oficial de *plugins* do *Gephi* (<https://marketplace.gephi.org/plugin/complex-generators>)

5. Geração de arquivo *excel*(.xls) com os dados obtidos.

4.2 Geração dos grafos

A literatura indica que a utilização de grafos como restrição na geração de coalizões tende a reduzir o número total de coalizões possíveis, já que somente coalizões factíveis serão consideradas.

Também utilizamos os seguintes tipos de grafos:

- **Grafo power law;**
- **Grafo completo** uma aresta existe entre cada par de vértices e todas as coalizões são possíveis. O grafo completo é um grafo denso, mas será utilizado para realizar a comparação com o modelo de formação de coalizões comumente utilizado quando se considera que todas as coalizões são possíveis;
- **Grafos do tipo estrela** os grafos tipo estrela são uma primeira abordagem para a formação de coalizões pelo fato da sua característica particular $K_{n-1,1}$, onde n é o número de vértices do grafo.

4.2.1 Grafos *power law*

Para os grafos completos não existe a necessidade de geração pois o número de subgrafos completos podem ser calculados analiticamente: a) o número máximo de cliques [Wood, 2007]; b) o número de subgrafos completos partindo de um dado grafo com grau máximo [Cutler and Radcliffe, 2014]. Dessa maneira é possível evitar a geração de grafos completos e a enumeração de todas as clique e subgrafos.

Para os grafos completos o número total de arestas existentes em um grafo não direcionado é dado por $n(n * 2)/2$, onde n é o número de vértices(agentes) existentes no sistema. Isso representa que todos os subgrafos existentes são coalizões factíveis, ou seja, a enumeração de todas as coalizões é dada por $2^n - 1$.

Mas isso não acontece para grafos estrela, que não são grafos completos, ou seja, não possuem arestas ligando todos os vértices, ou ainda em outros grafos como os gerados no modelo *Barabási-Albert*. Esses grafos apresentam uma distribuição diferente de arestas entre seus vértices, o que acaba gerando uma redução no número total de subgrafos conexos, os quais são considerados como coalizões factíveis.

4.2.2 Grafos estrela

Para grafos estrela é possível obter os valores necessários, como o número de subgrafos conexos e cliques através de fórmulas exatas, outros detalhes serão apresentados nas próximas seções. A escolha do grafo tipo estrela é uma primeira abordagem para a formação de coalizões pelo fato da sua característica particular $K_{n-1,1}$, onde n é o número de vértices do grafo. A ocorrência deste tipo de grafo em redes *power law* é bastante comum, os vértices com essa característica recebem o nome de *hubs*.

4.3 Análise do número de coalizões possíveis (sem considerar restrições)

Nesta etapa foi analisado qual o número total de coalizões possíveis de serem geradas em um grafo de n vértices quando não são consideradas restrições entre os vértices, de modo que toda coalizão é válida. Esse modelo é equivalente ao problema de particionamento completo de conjuntos. A enumeração completa para qualquer grafo, considerando que cada vértice n é um agente, tem $2^n - 1$ coalizões possíveis.

A hipótese inicial é que com a utilização de restrições sobre a coalizão para que ela seja considerada válida pode reduzir o número de coalizões que serão analisadas pelo algoritmo que vai encontrar a melhor estrutura de coalizão. O número de coalizões possíveis de serem geradas em todos os tipos de grafos utilizados neste trabalho pode ser obtido pela fórmula matemática $2^n - 1$, onde n é o número de vértices do grafo.

4.4 Análise do número de subgrafos factíveis (coalizões que são subgrafos conexos)

O objetivo nesta etapa é detalhar quais os métodos que foram utilizados para a contagem de subgrafos induzidos conexos, quando somente é considerada uma coalizão factível se e somente se o subgrafo induzido gerado é conexo. A abordagem de considerar como coalizão factível somente o subgrafo induzido gerado pelos vértices é conexo, apresentada em Voice et al. [2012b].

Para a enumeração dos subgrafos induzidos conexos dos grafos *power law* foi implementado um contador no programa desenvolvido para a geração dos grafos. Durante o processamento das coalizões foi incluída uma verificação, por força bruta, sobre cada coalizão em questão sendo a mesma testada se forma um subgrafo induzido conexo. Para os grafos completos o total de subgrafos induzidos conexos pode ser obtido pela fórmula matemática $2^n - 1$. Que é o mesmo número total de coalizões possíveis. Para grafos estrela o número de subgrafos induzidos conexos é dado por $2^{n-1} + n - 1$, vide teorema 5.1.1.

Não foi levado em consideração a estrutura do grafo para a contagem de subgrafos conexos. Caso a consideração da estrutura do grafo seja levada em conta o número de vértices por grafo pode ser maior que os 17 vértices utilizados neste trabalho.

4.5 Análise do número de subgrafos factíveis (coalizões que são cliques)

O objetivo deste etapa é detalhar como as coalizões são considerados factíveis se e somente se são consideradas os subgrafos que formam uma clique do grafo de agentes. Em grafos *power law* a enumeração das cliques ocorreu pela força bruta, para cada subgrafo gerado o mesmo era testado para verificar se é uma clique. Quando considerados os grafos completos, o número de cliques do grafo é igual ao número de coalizões possíveis $2^n - 1$. Considerando que todos os vértices estão ligados, logo qualquer subconjunto é uma clique. Para os grafos estrela o número de cliques é calculado pela equação $(n - 1)$, onde n é o número de vértices.

4.6 Experimentos

Para os grafos *power law*, os experimentos deste trabalho foram ocorreram em computador de 64bits, PC Intel Core I5 com 6GB de RAM. Cada teste for executado 10 vezes para cada conjunto de grafos e os valores utilizados para a geração dos resultados são as médias das execuções realizadas.

Para os grafos completos e estrela os experimentos não foram necessários já que existem cálculos matemáticos que fornecem os valores necessários para as hipóteses apresentadas, a quantidade de subgrafos induzidos conexos e cliques. Foram realizados cálculos em função da variação do número de vértices dos grafos.

Para a plotagem dos gráficos de resultados foi utilizada a ferramenta online *SAGE*⁴, um software matemático *open-source*. Alguns resultados também foram tratados no *excel*

4.7 Considerações

Este capítulo descreveu as etapas do método utilizado no desenvolvimento e realização dos experimentos para a verificação das hipóteses levantadas, considerando o conteúdo avaliado na revisão da literatura. O capítulo descreveu também os métodos que foram utilizados para a geração dos grafos *power law*, incluindo a etapa de desenvolvimento de um programa na linguagem Java para análise e enumeração de subgrafos conexos e cliques. A primeira de todas as etapas do método consistiu na revisão e avaliação da literatura, estudo dos algoritmos de formação de coalizões, dos que não consideram restrições e os que as levam em conta, essa revisão é apresentada no capítulo 3. No próximo capítulo é apresenta uma abordagem sobre geração de coalizões envolvendo restrições com detalhes mais aprofundados.

⁴<http://www.sagemath.org/>

Capítulo 5

Abordagem do Problema

No contexto deste trabalho, cada vértice representa um agente e as arestas representam as relações que existem entre os agentes. Essa é a representação apresentada em outros trabalhos Voice et al. [2012b]; Bistaffa et al. [2014b,a]; Vinyals et al. [2013]; Rahwan and Michalak [2014].

Mesmo com essas aproximações, as restrições de coalizões factíveis, existe um limitador sobre os algoritmos de formação de coalizões, os mesmos não conseguem encontrar a melhor coalizão em tempo razoável, por exemplo, o algoritmo *DyPE* pode necessitar de um tempo muito maior do que o disponível para garantir que a coalizão encontrada é a melhor.

Com base nessa informação, este trabalho busca apresentar uma análise sobre a utilização da teoria dos grafos para buscar a redução do número de coalizões possíveis de serem geradas. O objetivo é verificar qual a aproximação que a teoria dos grafos pode fornecer de modo a facilitar a redução no número de coalizões factíveis.

O número total de coalizões possíveis é exponencial ao número de agentes (ou vértices do grafo). Entre os agentes existem relações que podem ser mapeadas em um grafo de restrições. Nesse grafo pode ser possível reduzir o número de coalizões possíveis de forma a ser tratável dentro de um limite. Esse grafo que representa os agentes e suas relações é denominado de **grafo de restrições** dos agentes ou simplesmente **grafo de agentes**.

No decorrer deste capítulo quando for realizada referência aos agentes será utilizado o termo vértice e o termo aresta para relações existentes entre os agentes. A utilização destes termos visa tornar o problema mais próximo aos conceitos da teoria dos grafos.

No modelo convencional de formação de coalizões, qualquer subconjunto de vértices é considerada uma coalizão possível, a situação é que nem todas as coalizões são factíveis quando observamos as situações reais. Um dos problemas fundamentais para a geração das coalizões é o tamanho da entrada dos dados para o algoritmo de formação de coalizões. A entrada de dados é exponencial em relação a quantidade de vértices. Relacionando esse conceito a abordagem baseada em grafos, o problema é exponencial a quantidade de vértices do grafo. Isso significa que mesmo para grafos com baixo número de vértices, por exemplo, 20 vértices, o número total de possíveis coalizões é maior que 1 milhão.

Neste trabalho, a busca pela redução na quantidade do número de possíveis coalizões a serem geradas, essas coalizões são denominadas de coalizão factível. A utilização de grafos como ferramenta para auxiliar no mapeamento de restrições não é uma abordagem criada por este trabalho. O objetivo é analisar qual o limite da utilização de grafos como restrições para a formação de coalizões. Se acredita que, a utilização das ligações existentes entre os vértices po-

dem ser utilizadas como restrições para a formação de coalizões, uma abordagem já apresentada em Voice et al. [2012b].

O presente trabalho não busca encontrar qual melhor estrutura de coalizão ou as melhores coalizões possíveis em um conjunto. Mas apresentar qual a influência que as relações apresentadas na forma de um grafo de restrições, como demonstrado em Rahwan et al. [2011], pode influenciar no número de coalizões factíveis e as condições quanto ao tamanho da entrada para os algoritmos de formação de coalizão já existentes como: [Rahwan et al., 2011; Bistaffa et al., 2014a; Voice et al., 2012b; Vinyals et al., 2013; Rahwan and Michalak, 2014] para classificar não as heurísticas, mas sim as aproximações quanto a enumeração das coalizões.

É esperado que a utilização das restrições sobre os vértices do grafo reduza o número de coalizões a serem avaliadas e torne a busca pela melhor coalizão menos custosa, pois o conjunto das coalizões factíveis no espaço de busca será menor que considerando todas as coalizões possíveis.

Nesse trabalho é analisado também qual o impacto de diferentes abordagens de restrição sobre as coalizões, são as coalizões definidas como **coalizão factível**. As restrições aplicadas sobre as coalizões se baseiam na densidade das coalizões e não sobre a densidade do grafo como um todo. Apesar que em um grafo muito denso, a quantidade de arestas será maior entre seus vértices, um grafo com baixa densidade terá um número menor de arestas, mas isso não impede que algumas de suas coalizões possa ter uma densidade alta. Isso pode acontecer no caso de um vértice centralizador, *hub*, que é um vértice altamente conectado.

Quando o vértice centralizador, *hub*, é observado isoladamente, pode ser considerado um subgrafo com topologia estrela ($K_{n-1,1}$), onde n é o número de vértices. A ocorrência de subgrafos com a topologia estrela em grafos *power law* é bastante comum, esse trabalho utiliza os grafos estrela como uma primeira abordagem aos grafos *power law*.

5.1 O número de coalizões factíveis pode ser exponencial mesmo em grafos esparsos

A definição para grafos $G = (V, E)$ ser categorizado como densos ou esparsos pode ser:

- **Grafos densos:** possuem muitas ligações, ou seja, um grafo é denso quando o número de arestas é proporcional ao quadrado do número de vértices.
- **Grafos esparsos:** possuem poucas ligações, ou seja, um grafo é esparsos quando o número de arestas é proporcional ao número de vértices.

Quando não são consideradas as restrições, o número de possíveis coalizões é $2^n - 1$, não é considerada a coalizão vazia.

Existe a possibilidade de aplicar restrições sobre o conjunto pelas relações existentes. Na figura 5.1 podemos observar um exemplo de um conjunto de vértices e suas relações representado através de um grafo, chamamos este de **grafo de restrições**, pois neste são observadas quais as ligações entre os vértices e quais são as coalizões factíveis.

Uma coalizão é válida se e somente se os vértices do grafo estão conectados em uma regra definida, por exemplo: formam um subgrafo induzido conexo ou uma clique de forma

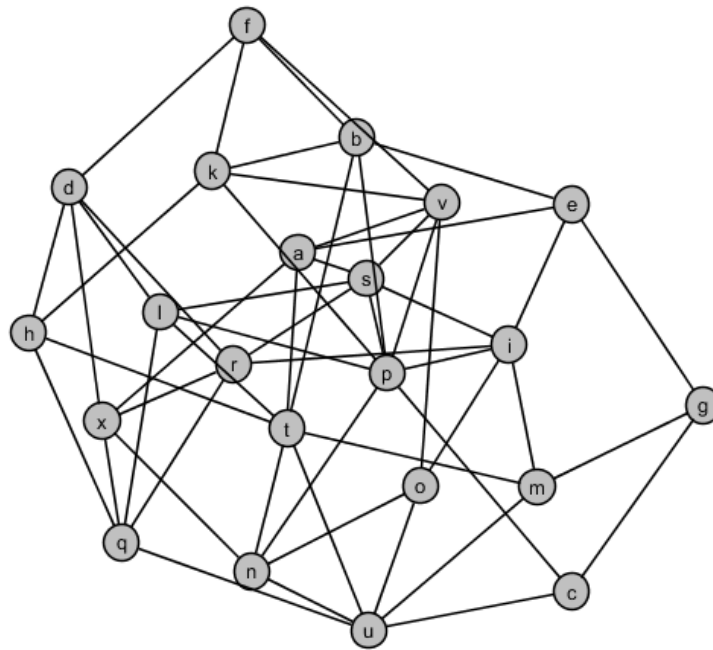


Figura 5.1: Exemplo de um grafo da relação.

Fonte: Autoria própria.

que existe uma aresta entre dois vértices quaisquer e nenhum dos vértices que estão presentes na coalizão está isolado.

Toda possível coalizão observada em um grafo é um **Subgrafo Induzido Conexo (SIC)**, um SIC é uma possível coalizão mas não necessariamente factível. A figura 5.2 apresenta alguns exemplos de coalizão válidas e factíveis.

Uma coalizão pode ser considerada factível se e somente se o subgrafo induzido for conexo.

Não existe uma fórmula para calcular o número de *SICs* de um grafo em função do número de vértices(v) e arestas(e), por exemplo: em árvores, onde temos que o número de vértices pode ser definido pela fórmula ($v = e - 1$), sendo v o número de vértices e e o de arestas. Neste caso o número de *SICs* varia de polinomial (Teorema 5.1.2) até exponencial (Teorema 5.1.1), dependendo da árvore.

Todo grafo, inclusive grafos esparsos, contém uma árvore geradora e portanto de acordo com o teorema 5.1.1 podem ter um número exponencial de subgrafos induzidos conexos (uma vez que o número de subgrafos da árvore geradora pode ser exponencial então o número de subgrafos conexos do próprio grafo é no mínimo o número de subgrafos da árvore geradora).

Embora nem todo grafo de n vértices contenha uma árvore geradora que seja uma estrela ($K_{n-1,1}$) isso é uma aproximação razoável para muitos grafos reais que seguem o modelo *power law*, ou seja, contém *hubs* que são vértices de grau alto.

O simples fato de o grafo ser esparsos não torna o problema menos complexo, mas se utilizar uma restrição sobre as coalizões que considera somente factível quando é uma clique, potencialmente isso pode colaborar em um ganho exponencial.

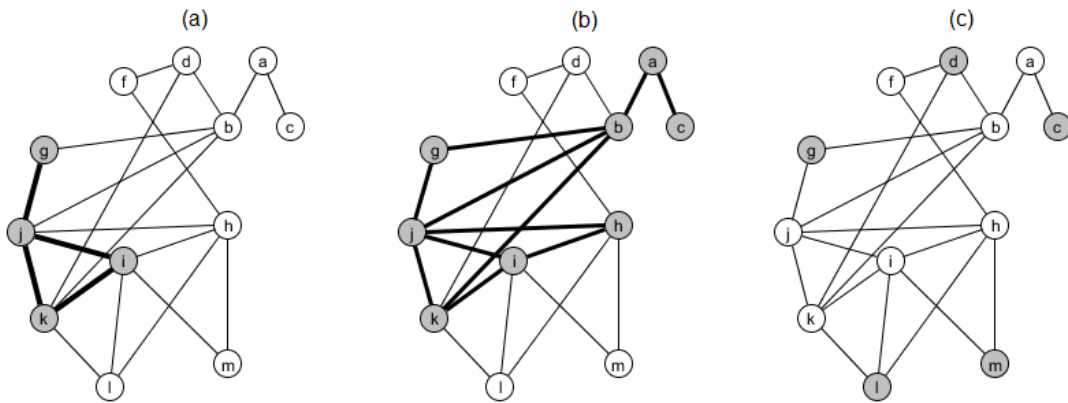


Figura 5.2: Apresenta 3 exemplos de subgrafos induzidos, as 3 coalizões destacadas nos grafos são válidas mas não factíveis. Na imagem (a) se observa uma os vértices e arestas destacados, esse subgrafo destacado pode ser considerada uma coalizão factível quando a restrição considerada é um subgrafo conexo, mas não pode ser uma coalizão factível se a restrição a ser considerada é clique. (b) como apresentado na descrição da imagem (a) os vértices e arestas destacados somente formam uma coalizão factível se a restrição considerada é a de subgrafo conexo. (c) apresenta alguns vértices isolados, todos eles poderiam estar presentes em coalizões onde o único vértice presente é ele mesmo.

Fonte: Autoria própria.

Teorema 5.1.1 O número de subgrafos induzidos conexos de um grafo estrela G com n vértices é $2^{n-1} + n - 1$.

Prova: Seja x o vértice de grau $n - 1$ de G e y_1, \dots, y_{n-1} os vértices de grau 1 de G . Seja $Y = \{y_1, \dots, y_{n-1}\}$. Vamos agora enumerar os SIC de G :

- SIC de G com 1 vértice: Existem n SIC (um para cada vértice isolado) de G .

- SIC de G com pelo menos 2 vértices: Note que neste caso para que um subgrafo de G seja conexo ele deve conter o vértice x . Portanto cada SIC de G com pelo menos 2 vértices é um subgrafo induzido por $x \cup Y'$, onde Y' é um subconjunto não vazio qualquer Y .

Como existem $2^{n-1} - 1$ tais conjuntos Y' , o número de SIC neste caso é $2^{n-1} - 1$.

Portanto o total de SIC é $2^{n-1} + n - 1$

Teorema 5.1.2 O número de subgrafos induzidos conexos de um P_n é $\sum_{k=1}^n k = \frac{n(n+1)}{2}$

Prova: Seja p_0, \dots, p_{n-1} os vértices do grafo P_n tal que p_i, p_{i+1} é uma aresta, onde $1 \leq i \leq n - 1$. Observe que neste caso um SIC de k vértices deve contar k vértices consecutivos p_i, \dots, p_{i+k} do caminho P_n . Em particular para $k = n$ existe 1 SIC, para $k = n - 1$ existem 2 subgrafos induzidos conexos, e assim por diante. Portanto o número total de SIC é $\sum_{k=1}^n k = \frac{n(n+1)}{2}$.

Observação 5.1.1 Qualquer árvore contém $2n - 1$ SIC que são cliques.

Prova: Como uma árvore não contém triângulos, os subgrafos que são cliques ou são vértices isolados ou são arestas da árvore. Portanto existem $2n - 1$ tais grafos.

5.2 Generalizando conceitos de coalizões factíveis

Restringir a densidades sobre o grafo como um todo pode não ter o ganho exponencial possível. Mas ao observar somente para o *SIC* é justificável aplicar restrições e considerar se a coalizão pode ser factível.

Neste trabalho é definido uma coalizão factível deve ser formada por subgrafo induzido do grafo principal, que devem atender a uma determinada restrição:

- **Um subgrafo conexo:** A coalizão é considerada factível se e somente se os vértices que pertencem a coalizão formam um *SIC*;
- **Uma clique:** A coalizão é considerada factível se e somente se os vértices que formam o subgrafo induzido que pertencem a coalizão formam uma clique, ou seja, todos os vértices da coalizão possuem uma aresta para os outros vértices;

A quantidade de cliques em um grafo de determinada família é uma questão típica da teoria extremal dos grafos [Wood, 2007].

Para grafos estrela o número de subgrafos induzidos conexos pode ser calculado pela equação 5.1, onde n é o número de vértices(v) do grafo:

$$2^{n-1} + n - 1 \quad (5.1)$$

5.3 Considerações

Este capítulo descreveu qual a abordagem sobre o problema de formação de coalizões que é realizada neste trabalho. A utilização de restrições para considerar a validade de uma coalizão não é novidade, como pode ser visto no trabalho de Sandholm et al. [1999], onde as coalizões são buscadas por ordem de tamanho em um grafo de coalizões. Mas uma nova abordagem proposta por [Voice et al., 2012b] apresenta os agentes como os vértices de um grafo e as relações entre os agentes são as arestas. Nessa abordagem as restrições são ditadas pelas arestas, se existe uma aresta entre os dois vértices logo eles podem estar presentes na mesma coalizão. Este capítulo apresentou qual a abordagem que este trabalho tem sobre essa restrição. Uma coalizão somente é factível se e somente se o subgrafo induzido forma: um subgrafo induzido conexo, uma clique. No próximo capítulo serão apresentados os resultados obtidos na realização dos experimentos.

Capítulo 6

Resultados e Análise

Nesta seção são apresentados os resultados obtidos durante o desenvolvimento dos experimentos realizados neste trabalho.

Em Voice et al. [2012b] apresenta uma abordagem baseada em grafos para a geração de coalizões. Outros trabalhos também restringem a formação de coalizões baseados em grafos [Vinyals et al., 2013; Bistaffa et al., 2014a,b; Ramos et al., 2014]. Estes trabalhos apresentam o modelo: um grafo onde os vértices representam os agentes e cada aresta pode ser interpretada como um canal de comunicação, uma relação de confiança, ou ainda uma facilidade de cooperação entre os dois vértices. Nos trabalhos citados uma coalizão somente é factível se e somente se a mesma é um subgrafo conexo do grafo original.

6.1 Resultados para grafos completos

No gráfico 6.1 é possível observar como o número de possíveis coalizões cresce em proporção ao número de vértices (agentes) envolvidos. No caso do gráfico 6.1, todas as coalizões são válidas, é como considerar um grafo de agentes completo, onde todo vértice tem relação com todos os vértices, nenhuma restrição é considerada.

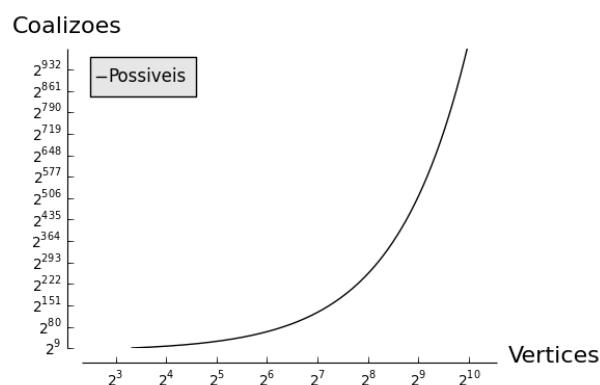


Figura 6.1: Grafo *loglog* do número de possíveis coalizões quando são considerados grafos completos de agentes.

Fonte: Autoria própria.

Para um grafo completo $G = (V, E)$, não é conhecida uma forma de reduzir o número de coalizões possíveis de serem geradas, pois todas as coalizões possíveis são também factíveis, sendo que qualquer restrição abordada neste trabalho não pode ser aplicada sobre esse tipo de grafo. Uma abordagem que pode ser utilizada sobre grafos completos poderia ser limitações sobre o tamanho de uma coalizão, limitando assim a quantidade mínima e/ou máxima de vértices que podem existir na coalizão [Voice et al., 2012b].

Com esse resultado, é possível então analisar qual o ganho, quanto ao número de coalizões factíveis nos tipos de grafo que não são completos e abordados neste trabalho: estrela e *power law*.

6.2 Resultados para grafos estrela

Quando se considera que uma coalizão factível é um subgrafo conexo, é possível verificar na figura 6.2 que para os grafos estrela o número de coalizões possíveis e factíveis é praticamente o mesmo. Isso se dá pela razão que um grafo estrela possui uma estrutura padrão de relação entre os vértices.

O número de coalizões possíveis e factíveis para grafos do tipo estrela $K_{n-1,1}$ pode ser calculado pelas formulas matemáticas, onde n é o número de vértices do grafo:

- Número de coalizões possíveis: $2^n - 1$
- Número de coalizões factíveis quando são considerados apenas subgrafos conexos: $2^{n-1} + n - 1$; (Teorema 5.1.1)

Com base nessas duas fórmulas para grafos do tipo estrela foi plotado o gráfico 6.2, é possível observar que ambas as linhas estão sobrepostas, o que significa que para um conjunto de vértices que acaba gerando grafo do tipo estrela a utilização de uma restrição baseada em subgrafos induzidos conexos pode não significar uma redução do número de coalizões atrativa.

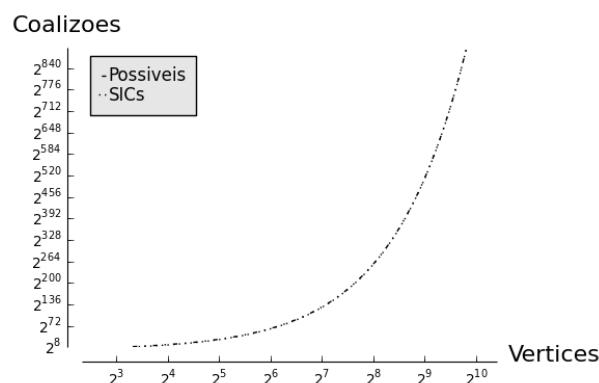


Figura 6.2: Grafo *loglog* do número de coalizões possíveis e factíveis em um grafo estrela. É possível observar que ambas as linhas estão sobrepostas, o número de subgrafos possíveis e o de factíveis. Considerando as coalizões factíveis como sendo subgrafos conexos

Fonte: Autoria própria.

Foi apresentado nos capítulos anteriores que a presença dos grafos estrela, ou *hubs* são muito comuns em redes do tipo *power law*, logo o interesse de trabalhar com os mesmos para a tarefa de formação de coalizões a restrição quanto a quantidade de coalizões necessariamente deve ter algum impacto significativo.

Quando observado o gráfico 6.2 se percebe que a restrição por subgrafos induzidos conexos como coalizões factíveis apesar de existir uma redução do número de coalizões proporcional ao número de vértices do grafo, não teve grande impacto, já que a curva de coalizões factíveis é exponencial assim como a de coalizões possíveis.

Realizando uma nova abordagem quanto a definição de coalizões factíveis, considerando somente os subgrafos que são uma clique do grafo, ou seja, somente é válida a coalizão se todos os vértices possuem relação com os demais vértices do subgrafo.

A contagem do número de cliques de um grafo estrela é dado pela fórmula matemática $(n - 1)$.

Sobre essa análise é possível observar o gráfico 6.3 que apresenta a comparação do número de coalizões possíveis e factíveis das duas maneiras: subgrafo conexos e cliques.

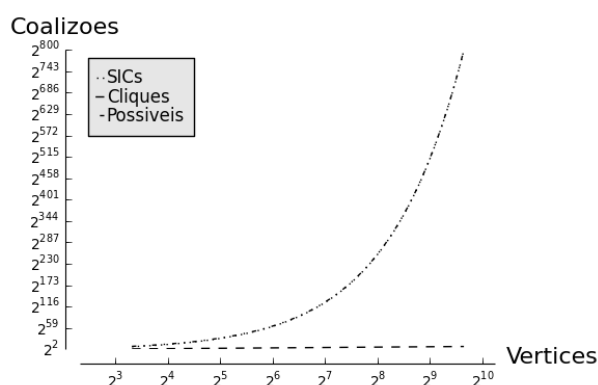


Figura 6.3: Número de coalizões possíveis, SICs e cliques em grafo estrela.

Fonte: Autoria própria.

É possível observar no gráfico 6.3 que quando somente o subgrafo de vértices forma um clique a quantidade de subgrafos possíveis é significativamente menor que a abordagem de subgrafos conexos. Desta forma a abordagem mais atrativa para considerar se uma coalizão é válida é a abordagem de verificar se o subgrafo é uma clique.

6.3 Resultados para grafos *power law*

Para os grafos *power law* que foram gerados de acordo como descrito no capítulo 4 os resultados são apresentados pela média dos valores obtidos nas execuções dos experimentos.

Quando consideramos os grafos *power law* sem nenhuma restrição para a formação das coalizões os valores são os mesmos que os de qualquer grafo completo de n vértices, $2^n - 1$, esse é o total de subgrafos induzidos ou possíveis coalizões.

Para calcular a quantidade de subgrafos induzidos conexos ou cliques, somente é possível com a enumeração de todos os subgrafos induzidos e o teste de cada. Essa tarefa foi

realizada pelo algoritmo criado para este fim como descrito no capítulo 4 e com base nesses resultados foi elaborado o gráfico 6.4.

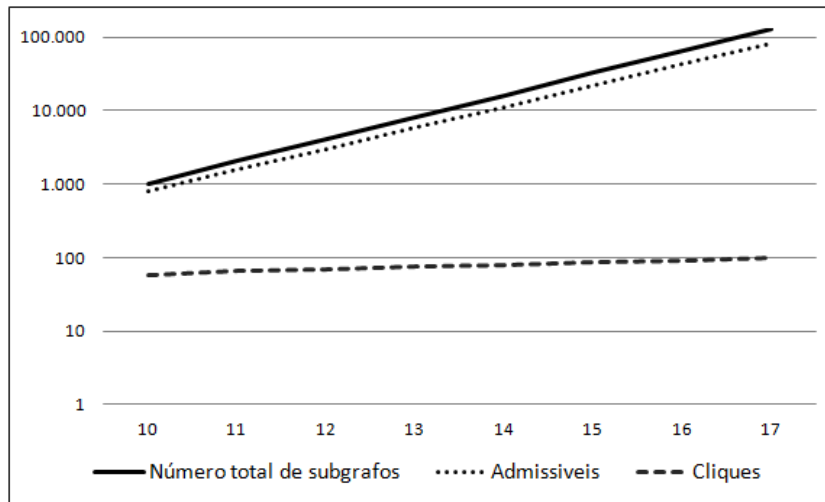


Figura 6.4: Gráfico *loglog* que apresenta o número de subgrafos possíveis, factíveis e cliques em grafos *power law* em relação ao número de vértices.

Fonte: Autoria própria.

No gráfico *loglog* observamos os resultados obtidos nos experimentos realizados sobre os grafos do tipo *power law* criados para verificar as hipóteses levantadas neste trabalho, os grafos foram limitados a 17 vértices, por conta de restrições técnicas de processamento.

Os resultados são apresentados por três linhas, cada uma delas representando uma restrição possível de ser aplicada. É observado que a utilização de subgrafos induzidos conexos como uma restrição para a formação de coalizões pode sim reduzir a quantidade de coalizões a serem avaliadas pelo algoritmo de geração de estruturas de coalizão. Mas se observado que o objetivo é reduzir a complexidade do problema que é exponencial, essa não é a melhor abordagem, pois a quantidade de coalizões factíveis continua exponencial ao número de vértices(agentes) do grafo.

Uma abordagem que pode ser avaliada para conseguir melhor ganho sobre a avaliação das coalizões é considerar como coalizões factíveis somente as que formam um subgrafo que seja uma clique. O que é um resultado semelhante ao observado na figura 6.3 que apresenta os resultados dos grafos estrela.

O ganho com a utilização de clique em grafos do tipo *power law* e estrela como uma restrição para considerar uma coalizão factível é exponencial.

Capítulo 7

Conclusão

Este trabalho abordou o problema de restrição de formação de coalizões utilizando grafos. Os agentes são tratados como um conjunto de vértices e as relações são as arestas. Uma possível coalizão é um subgrafo induzido que atende a uma determinada restrição. O objetivo geral deste trabalho foi verificar os limites da utilização de grafos como restrições na formação de coalizões e, em particular, o impacto ao utilizar clique como critério para a formação de coalizões.

Foram considerados tipos de grafos específicos para se observar o comportamento de restrições sobre a formação de coalizões. Os tipos de grafos utilizados neste trabalho foram: grafos do tipo *power law*, grafo estrela e grafos completos. Os grafos do tipo *power law* foram gerados de forma sintética por um programa em Java utilizando a API do *Gephi*. Já os valores para os grafos do tipo estrela são calculados diretamente através de fórmulas matemáticas.

Em todos os grafos utilizados para os experimentos o comportamento teve semelhança. A redução quando considerado somente que a coalizão seja um subgrafo induzido conexo teve pouco impacto sobre a quantidade de coalizões factíveis. Quando a restrição para se considerar uma coalizão factível foi considerar que o subgrafo seja uma clique o número reduziu, conforme pode ser observado no gráfico 6.3 para grafos do tipo estrela e no gráfico 6.4 para grafos do tipo *power law*.

Foi observado que a utilização de subgrafos induzidos conexos não reduz de forma significativa o número de coalizões que dever ser alisadas pelos algoritmos de geração de estruturas de coalizão. Entretanto, a utilização de outra forma de restrição para os subgrafos como a utilização de cliques pode tornar a restrição mais interessante para o número de coalizões factíveis.

A redução do número de coalizões pelos algoritmos que utilizam grafos como restrição depende diretamente de qual a restrição escolhida para se considerar uma coalizão factível. Como observado nos experimentos a utilização de uma restrição significa a qualidade da redução na quantidade de coalizões que deverão ser avaliadas pelos algoritmos de geração de estruturas de coalizão.

Na análise sobre os algoritmos considerados como estado da arte para a geração de estruturas de coalizão, se observou que eles fazem a utilização de restrições baseadas em grafos. Algoritmos que consideram uma coalizão factível somente quando a mesma forma um subgrafo conexo Vinyals et al. [2013], algoritmo este que consegue enumerar todas as coalizões factíveis para mais de 50 agentes, enquanto algoritmos que não consideram restrições, de forma que

todas as coalizões são consideradas válidas chegam a enumerar as coalizões para o limite de 17 agentes.

Foi verificado que a utilização de cliques como restrições para a formação de coalizões pode reduzir de maneira significativa a quantidade de coalizões que devem ser enumeradas. Se observa que nos gráficos gerados pelos experimentos realizados que realmente existe uma vantagem numérica na utilização de clique como uma restrição. A quantidade de subgrafos que são cliques é notoriamente menor que a quantidade de subgrafos induzidos conexos.

Realmente o número de coalizões possíveis de serem formadas é menor, apesar de não confirmar se essa redução torna o problema tratável para uma um número considerável de agentes. Se considerar uma rede social composta de 10.000 agentes, provavelmente nenhuma das restrições vai alcançar uma redução tratável para o problema.

A aplicação de algumas restrições existentes na teoria dos grafos como heurística para a redução no número de estruturas de coalizões factíveis. Neste, o foco não foi encontrar a melhor estrutura de coalizão existente, mas sim apresentar uma nova abordagem quanto ao agrupamento dos agentes para que sejam considerados uma coalizão válida.

Nos experimentos realizados foi possível observar que o número de coalizões é reduzido quando aplicamos a heurística que as coalizões devem ser conexas, ou seja, todos os vértices da coalizão devem estar ligadas uns aos outros. Essa regra teoricamente simples, todos os vértices devem estar ligados, o número de coalizões reduz, mas ainda não chega a ser um número computável.

Foi observado que quando é considerada uma coalizão válida somente se a mesma é uma clique de um grafo o número de coalizões factíveis é muito menor que a quantidade total. E notável que a restrição de coalizões factíveis somente quando consideramos os subgrafos conexos não é uma abordagem atrativa, pois a quantidade de subgrafos conexos ainda continua sendo exponencial ao número de vértices.

Mas quando consideramos como coalizões factíveis somente as que formam uma clique, a abordagem é notavelmente mais atrativa frente aos demais tratamentos para todos os tipos de grafos abordados neste trabalho. Mesmo com a variedade de algoritmos existentes para resolver o problema de geração de estruturas de coalizão nenhum deles é capaz de trabalhar com uma massa grande de agentes envolvidos. A utilização de uma restrição como cliques busca reduzir a quantidade de coalizões possíveis de serem geradas e assim tornar o problema mais simples, levando em consideração que o número de estruturas de coalizões esta relacionado com o número de coalizões.

Como trabalho futuro propõem o desenvolvimento de algoritmos que efetivamente utilizem a restrição de cliques para a formação de coalizões. Tendo como base este trabalho que demonstrou a redução significativa na quantidade de coalizões a serem consideradas para a formação de coalizões.

Como estudo futuro é possível verificar qual o comportamento para outros tipos de grafos e qual seriam outras restrições viáveis para a tratabilidade do problema de formação de coalizões. Além de considerar outras densidades para verificar qual o comportamento de acordo com a densidade aplicada sobre o subgrafo induzido pelos agentes da coalizão. Também verificar como essas abordagens podem ser utilizadas nos algoritmos atuais para encontrar a melhor estrutura de coalizão.

Referências Bibliográficas

- Albert, R. and Iászló Barabási, A. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, pages 74:47–97.
- Bastian, M., Heymann, S., and Jacomy, M. (2009). Gephi: An open source software for exploring and manipulating networks.
- Bistaffa, F., Farinelli, A., Cerquides, J., Rodríguez-Aguilar, J., and Ramchurn, S. D. (2014a). Anytime coalition structure generation on synergy graphs. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems, AAMAS '14*, pages 13–20, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.
- Bistaffa, F., Farinelli, A., Cerquides, J., Rodríguez-Aguilar, J. A., and Ramchurn, S. D. (2014b). Anytime coalition structure generation on scale-free and community networks. Paris, France.
- Cutler, J. and Radcliffe, A. (2014). The maximum number of complete subgraphs in a graph with given maximum degree. *Journal of Combinatorial Theory, Series B*, 104(0):60 – 71.
- Diestel, R. (1997). *Graph Theory*. Number 173 in Graduate Texts in Mathematics. Springer.
- Horling, B. and Lesser, V. (2004). A survey of multi-agent organizational paradigms. *Knowl. Eng. Rev.*, 19(4):281–316.
- Michalak, T., Sroka, J., Rahwan, T., Wooldridge, M., McBurney, P., and Jennings, N. R. (2010). A distributed algorithm for anytime coalition structure generation. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, pages 1007–1014, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.
- Myerson, R. B. (1977). Graphs and cooperation in games. *Mathematics of Operations Research*, 2(3):225–229.
- Rahwan, T. (2007). Algorithms for coalition formation in multi-agent systems.
- Rahwan, T. and Jennings, N. R. (2007). An algorithm for distributing coalitional value calculations among cooperating agents. *Artificial Intelligence*, 171(89):535 – 567.
- Rahwan, T. and Jennings, N. R. (2008a). Coalition structure generation: dynamic programming meets anytime optimisation. In *In Proceedings of the Twenty Third Conference on Artificial Intelligence*, pages 156–161.

- Rahwan, T. and Jennings, N. R. (2008b). An improved dynamic programming algorithm for coalition structure generation. In *In Proc. of the 7th Int. joint Conf. on Autonomous Agents and Multi-agent Systems (AAMAS)*, pages 1417–1420.
- Rahwan, T., Michalak, T., Elkind, E., Faliszewski, P., Sroka, J., Wooldridge, M., and Jennings, N. R. (2011). Constrained coalition formation. In *Proc. 25th Conference on AI (AAAI)*, pages 719–725.
- Rahwan, T., Michalak, T., Wooldridge, M., and Jennings, N. R. (2012). Anytime coalition structure generation in multi-agent systems with positive or negative externalities. *Artif. Intell.*
- Rahwan, T. and Michalak, T. P. (2014). Coalition structure generation on graphs. *CoRR*, abs/1410.6516.
- Ramos, G. d. O. and Bazzan, A. L. C. (2012). Reduction of coalition structure’s search space based on domain information: An application in smart grids. In *Proceedings of the 2012 Third Brazilian Workshop on Social Simulation*. IEEE Computer Society.
- Ramos, G. d. O., Burguillo, J., and Bazzan, A. L. (2014). Dynamic constrained coalition formation among electric vehicles. *Journal of the Brazilian Computer Society*, 20(1):1–15.
- Sandholm, T., Larson, K., Andersson, M., Shehory, O., and Tohmé, F. (1999). Coalition structure generation with worst case guarantees. *Artif. Intell.*, 111(1-2):209–238.
- Shehory, O. and Kraus, S. (1998). Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(12):165 – 200.
- Vinyals, M., Voice, T., Ramchurn, S. D., and Jennings, N. R. (2013). A hierarchical dynamic programming algorithm for optimal coalition structure generation.
- Voice, T., Polukarov, M., and Jennings, N. R. (2012a). Coalition structure generation over graphs. *J. Artif. Int. Res.*
- Voice, T., Ramchurn, S. D., and Jennings, N. R. (2012b). On coalition formation with sparse synergies. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1, AAMAS ’12*. International Foundation for Autonomous Agents and Multiagent Systems.
- Wood, D. R. (2007). On the maximum number of cliques in a graph. *Graphs and Combinatorics*, 23(3):337–352.
- Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. Wiley Publishing, 2nd edition.
- Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review*.
- Yun Yeh, D. (1986). A dynamic programming approach to the complete set partitioning problem. *BIT Numerical Mathematics*.

Apêndice A

Anexos

A.1 Notações

a	Agente.
A	Conjunto de Agentes.
n	Número de agentes do conjunto A .
G	Grafo.
V	Conjunto de vértices de um grafo.
v	Número de vértices de um grafo.
E	Conjunto de arestas de um grafo.
e	Número de arestas de um grafo.
C	Coalizão.
CS	Estrutura de Coalizão.