FRANK ARAUJO DE ABREU CARA

UMA HEURÍSTICA GANHA-GANHA PARA FORMAÇÃO DE COALIZÕES EM SISTEMAS MULTIAGENTES

São Paulo

2013

FRANK ARAUJO DE ABREU CARA

UMA HEURÍSTICA GANHA-GANHA PARA FORMAÇÃO DE COALIZÕES EM SISTEMAS MULTIAGENTES

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de mestre em Ciências

São Paulo

2013

FRANK ARAUJO DE ABREU CARA

UMA HEURÍSTICA GANHA-GANHA PARA FORMAÇÃO DE COALIZÕES EM SISTEMAS MULTIAGENTES

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de mestre em Ciências

Área de concentração:
Sistemas Eletrônicos

Orientador:
Márcio Lobo Netto

São Paulo

2013

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, ... de . . . . . de 2013

Assinatura do autor

Assinatura do orientador

FICHA CATALOGRÁFICA

Cara, Frank Araujo de Abreu
Uma Heurística Ganha-Ganha para Formação de Coalizões em Sistemas Multiagentes / F. A. d. A. Cara -- versão corr. -- São Paulo, 2013.
115 p.
Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.
1. Formação de Coalizões 2. Cooperação e Coordenação 3. Negociação 4. Sistemas Multiagentes I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II. t.

DEDICATÓRIA

Ao meu pai, Frank de Abreu Cara, in memoriam, pelo exemplo eterno de amor

AGRADECIMENTOS

Ao professor Márcio Lobo Netto pela profunda liberdade criativa que nos proporciona, apoio constante e, sobretudo, paciência.

RESUMO

Sistemas multiagentes muitas vezes apresentam características que os aproximam de sociedades de agentes e, como as humanas, possuem normas e organizações com o objetivo de coordenar as interações entre os seus membros. Coalizão é um tipo de estrutura organizacional temporária, montada com objetivos específicos. A teoria dos jogos estuda formalmente o fenômeno coalizional, se detendo em demonstrações de propriedades e características dessa estrutura. A área de sistemas multiagentes, por outro lado, tem mostrado significativo interesse nas estruturas coalizionais como forma de organizar a cooperação entre os agentes, dedicando-se ao desenvolvimento de algoritmos para formação de coalizões.

Esse trabalho apresenta um algoritmo de formação de coalizões para compartilhamento de recursos, denominado heurística ganha-ganha. Definimos um modelo que utiliza a abstração de recursos para representar tanto a posse de habilidades e objetos, quanto para representar os objetivos dos agentes. Um jogo de votação k-ponderado é utilizado para implementar o processo decisório de quais coalizões são válidas e o algoritmo testa iterativamente cada vizinhança de um agente na busca de associações vantajosas.

Demonstramos que o algoritmo incrementa monotonicamente o bem-estar da sociedade e converge para uma estrutura coalizional. Também mostramos empiricamente que a heurística é eficiente para compartilhamento de recursos em situações de abundância de recursos, montando coalizões em poucas iterações e com uma quantidade grande de agentes.

Palavras-chave: Formação de Coalizões. Cooperação e Coordenação. Negociação. Sistemas Multiagentes.

ABSTRACT

Multiagent systems frequently show characteristics that come closer to agent societies and, like the humans ones, have norms and organizations in order to coordinate the interactions of its members. Coalition is a type of temporary organizational structure, assembled with specific goals. Game theory formally studies the coalitional phenomenon focusing in the demonstrations of properties and characteristics of this structure. The area of multiagent systems, on the other hand, has devoted significant interest in coalition structures as a way to organize cooperation between its members, and has been dedicated to the development of algorithms for coalition formation.

This dissertation presents an algorithm to coalition formation named win-win heuristic. We define a model which uses the abstraction of resources to represent either, the possession of abilities and objects, or to represent the agent's target. A k-weight voting game is used to implement the decision process of what coalitions are worth and an iteratively algorithm tests each agent neighborhood in the pursue of better associations. We demonstrate that the algorithm monotonically increases the society welfare and converges to a coalitional structure. We also show empirically that the heuristic is efficient for resource sharing in situations of availability of resources, building coalitions with few iterations and a large amount of agents.

Keywords: Coalition Formation. Cooperation and Coordination. Negotiation. Multiagent Systems

LISTA DE ILUSTRAÇÕES

Figura 1 - Estrutura Coalizional 23
Figura 2 - Assuntos cobertos pela fundamentação teórica 31
Figura 3 - Estruturas Organizacionais 36
Figura 4 - Curva Normal do Valor Aproximado de Shapley 47
Figura 5 - Bem-estar Médio da Heurística 72
Figura 6 - Desvio Padrão Médio da Heurística 72
Figura 7 - Agente Coalizional 77
Figura 8 - Planejamento Cego 80
Figura 9 - Planejamento Cego com Avaliação Iterativa 81
Figura 10 - Replanejamento com Avaliação Iterativa 82
Figura 11 - Workbench do simulador 85
Figura 12 - Mundo em grid do simulador 86
Figura 13 - Diagrama de classes do simulador 87
Figura 14 - Quantidade de avaliações e bem-estar por carência de recursos 93
Figura 15 - Quantidade de avaliações e bem-estar por carência de recursos 95
Figura 16 - Quantidade de avaliações e bem-estar por carência de recursos 97
Figura 17 - Quantidade de avaliações e bem-estar por carência de recursos 98
Figura 18 - Quantidade de avaliações e bem-estar por carência de recursos 100
Figura 19 - Quantidade de coligações por carência 100

LISTA DE TABELAS

Tabela 1 - Exemplo de cálculo do valor de Shapley 44
Tabela 2 - Regras do agente coalizional 79
Tabela 3 - Ciclos do agente coalizional 83
Tabela 4 - Responsabilidades dos elementos do ambiente 84

LISTA DE CÓDIGOS FONTE

Listagem 1 - Trait Comportamento 111
Listagem 2 - Trait PlanSocDireto 112
Listagem 3 - Trait PlanDiretoSocIterativo 113
Listagem 4 - Trait PlanSocIterativo 114

LISTA DE ABREVIATURAS E SIGLAS

DPS	Distributed Problem Solving
IA	Inteligência Artificial
MARA	Multiagent Resource Allocation
SMAs	Sistemas Multiagentes

LISTA DE SÍMBOLOS

N	Conjunto de jogadores ou agentes de um jogo
S	Coalizão
$v(S)$	Valor da coalizão
φ_i	Valor de Shapley
i, j	Índices utilizados para representar agentes
$\mathcal{K}$	Conjunto de recursos do ambiente
k	Índice do k-ésimo recurso
Π_i	Plano de ação do agente i
rc_i	Vetor de recursos do agente i
rc_i^{Π}	Vetor de recursos do plano do agente i
rc_i^{+}	Vetor de recursos residuais do agente i
rc_S	Vetor de recursos da coalizão S
al_i	Vetor de alocação do agente i
$u_i(S)$	Benefício do agente i na coalizão S
$sw_u(S)$	Bem-estar utilitário dos membros da coalizão S

SUMÁRIO

1 INTRODUÇÃO . . . 15
1.1 VISÃO GERAL . . . 15
1.2 JUSTIFICATIVA . . . 17
1.2.1 Objetivos . . . 18
1.2.2 Metodologia . . . 18
1.2.3 Contribuições . . . 19
1.2.4 Aplicações . . . 19
1.3 ORGANIZAÇÃO DO TRABALHO . . . 20
1.4 SOBRE A NOTAÇÃO . . . 20
2 REVISÃO DA LITERATURA . . . 22
2.1 FORMAÇÃO DE COALIZÕES . . . 22
2.2 LEILÕES E NEGOCIAÇÕES . . . 26
2.3 SISTEMAS MULTIAGENTES E ALOCAÇÃO DE RECURSOS . . . 27
2.4 COMPARAÇÃO ENTRE OS MODELOS . . . 29
3 FUNDAMENTAÇÃO TEÓRICA . . . 31
3.1 SISTEMAS MULTIAGENTES . . . 32
3.1.1 Ambiente . . . 32
3.1.2 Agentes Sociais . . . 33
3.1.3 Organização Coalizional . . . 37
3.2 PLANEJAMENTO DISCRETO EM SISTEMAS MULTIAGENTES . . . 37
3.3 TEORIA DOS JOGOS COALIZIONAIS . . . 40
3.3.1 Valor de Shapley . . . 43
3.3.2 Aproximações ao Valor de Shapley . . . 46
3.4 COMPARTILHAMENTO DE RECURSOS . . . 49
3.4.1 MARA - Multiagent Resource Allocation . . . 49
3.4.2 CT-Game . . . 51
3.5 SUMARIZAÇÃO . . . 51
4 UMA HEURÍSTICA GANHA-GANHA PARA FORMAÇÃO DE COALIZÕES . . . 53
4.1 MODELO DE AGENTES BASEADO EM RECURSOS . . . 53
4.1.1 Recursos como *proxies* dos agentes . . . 53
4.1.2 Alocação através da Troca de Recursos . . . 56

4.2 DESENVOLVIMENTO DA HEURÍSTICA . . . 57
4.2.1 Utilidade do Agente e Bem-Estar da Coalizão . . . 58
4.2.2 Teste Ganha-Ganha . . . 59
4.2.3 Algoritmo de Formação de Coalizões . . . 61
4.3 CARACTERÍSTICAS DA HEURÍSTICA . . . 62
4.3.1 Monotonicidade . . . 62
4.3.2 Estratégia Gulosa . . . 65
4.3.3 Estabilidade do bem-estar . . . 66
4.3.4 Eficiência . . . 67
4.3.5 Distribuível . . . 67
4.3.6 Complexidade e Performance . . . 68
4.4 RESULTADOS . . . 68

5 AMBIENTE DE SIMULAÇÃO MULTIAGENTE PARA COMPARTILHAMENTO DE RECURSOS . . . 73
5.1 ARQUITETURA DO AMBIENTE . . . 74
5.1.1 Ambiente . . . 74
5.1.2 Elementos Conceituais de um Agente Coalizional . . . 77
5.1.3 Comportamentos dos Agentes Coalizionais . . . 79
5.1.4 Planejamento . . . 83
5.1.5 Coalizão . . . 83
5.1.6 Responsabilidades dos Elementos do Ambiente . . . 84
5.2 SOFTWARE DE SIMULAÇÃO . . . 85
5.2.1 Projeto do Simulador . . . 87
5.2.2 Implementação do Simulador . . . 89
5.3 SIMULAÇÕES E RESULTADOS . . . 90
5.3.1 Configuração do ambiente . . . 91
5.3.2 Variáveis Monitoradas . . . 91
5.3.3 Gráfico da Simulação . . . 92
5.3.4 Simulações . . . 93
5.3.5 Resultados . . . 98

6 CONCLUSÃO . . . 102
6.1 TRABALHOS FUTUROS . . . 103

APÊNDICE A – CÓDIGO FONTE . . . 110

1 INTRODUÇÃO

Esse trabalho apresenta um algoritmo para alocação de recursos através da formação de coalizões de agentes. O trabalho tem dimensão teórica e empírica. A dimensão teórica consistiu no desenvolvimento de um modelo de agentes baseado em recursos, provando-se a monotonicidade do algoritmo. Na parte empírica foi construído um ambiente de simulação, a partir do qual foi medido o desempenho do algoritmo com diversas distribuições de agentes. O principal resultado do trabalho foi desenvolver um algoritmo, provadamente monotônico, e eficiente em situações definidas.

Em sistemas multiagentes é recorrente a necessidade de que os agentes cooperem para atingir seus objetivos, especialmente quando não podem alcançá-los sozinhos, ou teriam melhores resultados trabalhando em conjunto. No entanto, o estabelecimento e manutenção da cooperação depende de inúmeros fatores e capacidades, tanto no âmbito do agente quanto no do ambiente, como, identificação de ações ou recursos sujeitos à cooperação, identificação dos parceiros candidatos, comunicação para o estabelecimento da cooperação, estabelecimento da cooperação em si com divisão de tarefas e recursos, monitoramento da performance e ganhos na cooperação.

Dessa forma, questões de cooperação surgem naturalmente quando falamos de *Sistemas Multiagentes* (SMAs). Nesse trabalho adotaremos *o compartilhamento de recursos como estratégia para alcançar a coordenação e desenvolvimento de cooperação entre agentes*. Para tanto utilizamos um modelo de agente baseado em recursos, em que estes representam os objetivos e capacidades dos agentes e desenvolvemos uma heurística para formação de coalizões.

1.1 VISÃO GERAL

Um sistema multiagente apresenta capacidades sociais quando implementa mecanismos que possibilitem a comunicação entre agentes, formação de organizações sociais, estabelecimento e garantia de normas ou execução de negociações entre agentes. Coalizões são estruturas organizacionais temporárias, sem hierarquia e formadas com o fim de alcançar um objetivo em comum. Segundo Sandholm et al. (1999), a formação de coalizões deve incluir três etapas:

1. Geração da estrutura coalizional: corresponde à atividade de identificar associações vantajosas e executar o protocolo de efetivação da associação. A identificação das coalizões corresponde à identificação de quais grupos de agentes tem sinergia, ou seja, quanto de valor é gerado pela associação. Além disso, essa atividade deve apresentar garantia de

convergência do processo e, preferencialmente, que o conjunto de coalizões seja o mais próximo possível de um ótimo.

2. Coordenação da ação interna da coalizão: agir segundo os objetivos da coalizão, maximizando os resultados da cooperação.

3. Divisão dos ganhos da coalizão: implementar a divisão dos recursos segundo o contrato prévio da coalizão. Essa atividade inclui decidir quem recebe quanto, ou o que, dos benefícios da coalizão. Um conceito chave nessa atividade é o de *divisão justa*, ou seja, como dividir os benefícios de forma que as maiores contribuições impliquem no maior retorno. O *valor de Shapley* (Osborne & Rubinstein, 1994) é uma medida apresentada para solucionar essa questão.

Nesse trabalho desenvolvemos o algoritmo chamado *heurística ganha-ganha*, **que consiste na geração da estrutura coalizional e divisão dos ganhos das coalizões (pontos 1 e 3) no domínio de alocação de recursos em SMAs.**

A alocação de recursos em sistemas multiagentes é a partição de um conjunto de itens denominados recursos (Rc) para um conjunto de agentes (N). Recursos representam itens do ambiente, tarefas ou mesmo habilidades que podem ser transferidos ou compartilhados entre os agentes. (Chevaleyre et al., 2006). Nesse modelo, os agentes possuem uma dotação inicial de recursos, e tem uma preferência determinada por esses recursos, dada por uma função utilidade (u_i). Assumimos que os agentes são individualmente racionais, ou seja, interessados na maximização da sua utilidade individual.

A *heurística ganha-ganha* é um algoritmo iterativo em que cada agente busca entre os outros agentes possibilidades de formação vantajosa de coalizões, onde associar-se em coalizão significa disponibilizar os seus recursos para serem distribuídos internamente, e receber recursos que atendam sua preferência, segundo a relevância de cada agente para a coalizão.

Demonstramos teoricamente que o algoritmo apresenta características importantes.

- O algoritmo incrementa monotonicamente o bem-estar social do ambiente.

- O algoritmo distribui com justiça os ganhos das coalizões.

- O algoritmo converge para uma alocação de recursos.

- Agentes podem entrar/sair *ad-hoc*, sem modificar os ganhos já obtidos.

Simulamos o algoritmo sobre um mundo em grade (Gal et al., 2005) onde cada agente está localizado em uma célula da grade, e tem uma célula objetivo a ser alcançada. Cada célula

precisa de um conjunto de recursos para ser ocupada, e cada agente possui uma dotação inicial de recursos. O agente monta o plano de ações que melhor atenda os seu objetivos, considerando a possibilidade de formar coalizões para compartilhar recursos.

1.2 JUSTIFICATIVA

Nesse trabalho foi desenvolvido um algoritmo para formação de coalizões em sistemas multiagentes no domínio de problemas de alocação de recursos. Ambas as áreas, formação de coalizões e alocação de recursos, tem recebido intensa atenção da comunidade de pesquisa em SMAs. A sua importância como tema de pesquisa se dá, por uma lado, pela necessidade de coordenação e cooperação em sistemas multiagentes, tema tratado inerentemente pela formação de coalizões, e por outro, pela abrangência do modelo de alocação de recursos que tem diversas aplicações potenciais.

A pesquisa em formação de coalizões utiliza-se do embasamento teórico fornecido pela teoria do jogos, que tem estudado o fenômeno coalizional sob uma perspectiva formal desde a fundação da área. Além disso, coalizões são uma das estruturas organizacionais mais genéricas, uma vez que não obrigam a definição de hierarquia ou tempo de duração da organização. Segundo Michalak et al. (2010),

> Formação de coalizões é um importante tópico em sistemas multiagentes, pois há muitas atividades que coalizões podem desempenhar melhor do que simples agentes: por exemplo, em comércio eletrônico, coalizões frequentemente conseguem compras (ou vendas) a menores (maiores) preços unitários do que indivíduos sozinhos. (Michalak et al., 2010).

O modelo de recursos, da mesma forma que a formação de coalizões, se beneficia da generalidade do conceitos, como pontua Wooldridge & Dunne (2006).

> Acreditamos que o foco em recursos é muito natural, dadas as preocupações de sistemas multiagentes e disciplinas relacionadas: limitação de recursos, e a necessidade de gerenciar e compartilhar eficientemente recursos em ambientes multiagentes, provêem uma das motivações fundamentais para Inteligência Artificial (IA) distribuída e sistemas multiagentes (Wooldridge & Dunne, 2006, pg. 836).

O modelo de recursos e a utilização de teoria dos jogos fazem parte da mesma abordagem, que busca na economia técnicas que auxiliem no controle de sistemas complexos.

> Sistemas distribuídos em que agentes de software autônomos interagem um com o outro, seja de forma cooperativa ou competitiva, podem frequentemente ser interpretados como sociedades de agentes; e podemos empregar

ferramentas formais da microeconomia para analisar tais sistemas. (Chevaleyre et al., 2004).

Dessa forma, essa abordagem mista, alocação de recursos e formação de coalizões apresenta excelentes perspectivas de avanço no controle de interações em sistemas multiagentes, tema relevante para a área de SMA e IA de um modo geral.

1.2.1 Objetivos

Desenvolver um algoritmo para alocação de recursos em sistemas multiagentes através da formação de coalizões. Avaliar as características desse algoritmo, tanto do ponto de vista teórico, quanto do empírico.

1.2.2 Metodologia

O trabalho teve uma abordagem mista, teórica e empírica. A heurística é analisada teoricamente, bem como são efetuadas execuções simuladas e medições de performance. A análise teórica busca estabelecer alguns resultados sobre a convergência e estabilidade do algoritmo. As medições, por sua vez, analisam a evolução do bem-estar sobre cenários de distribuição de recursos. Os recursos são distribuídos entre os agentes segundo uma distribuição normal, e os cenários correspondem a valores diferenciados dos parâmetros da distribuição (média e desvio padrão).

As medições foram efetuadas em dois ambientes diferentes, estático e dinâmico. No ambiente estático não há modificação nas configurações iniciais do ambiente, enquanto no ambiente dinâmico os objetivos dos agentes podem sofrer modificações de acordo com o sucesso das coalizões.

Dessa foma, o trabalho apresenta dois conjuntos de resultados empíricos:

1. Medições sobre a execução do algoritmo em um ambiente estático, onde é medido o comportamento do agente com relação aos parâmetros da distribuição, sem modificação no ambiente ou no conjunto de agentes.

2. Medições sobre um ambiente de simulação de mundo em grade, onde o algoritmo é avaliado sobre um ambiente mais dinâmico que permite a entrada/saída de agentes e alteração dos objetivos dos agentes.

1.2.3 Contribuições

Nessa dissertação desenvolvemos um novo algoritmo de alocação de recursos através da formação de coalizões. O algoritmo é eficiente para redistribuição de recursos em cenários de abundância dos mesmos, e pode ser executado para quantidade maiores de agentes do que os algoritmos usuais baseados em leilões combinatórios ou barganhas individuais. O algoritmo apresenta a importante propriedade de ser monotônico (incrementar monotonicamente o bem-estar no sistema), e pode ser classificado como uma heurística *greedy*.

Pontuando as vantagens do algoritmo, temos:

1. Utiliza o modelo de agentes baseados em recursos, tendo grande possibilidade de generalização para domínios específicos.
2. O algoritmo é eficiente em cenários com grande disponibilidade de recursos.
3. Executável para quantidades elevadas de agentes (> 20).
4. É uma *heurística gulosa* podendo ser interrompido a qualquer momento, alcançando uma configuração válida de coalizões.
5. Válido para coalizões que não são super-aditivas.

1.2.4 Aplicações

Formação de coalizões tem se provado útil em um número de cenários reais e sistemas multiagentes. Em Rahwan et al. (2009) temos exemplificado um conjunto grande de aplicações, em especial:

- *e-commerce*: Em Tsvetovat et al. (2001) compradores formam coalizões para comprar produtos em lote e conseguir vantagens de preços.
- *e-business*: Em Norman et al. (2004) grupos de agentes se formam para ocupar nichos específicos de mercado.
- Redes de sensores distribuídas: Em Dang et al. (2006) coalizões de sensores trabalham juntos para rastrear alvos de interesse.

1.3 ORGANIZAÇÃO DO TRABALHO

A dissertação foi dividida em 6 Capítulos. Os dois primeiros depois da introdução se referem aos conceitos teóricos utilizados no desenvolvimento do algoritmo. Os dois próximos Capítulos correspondem ao núcleo do trabalho desenvolvido, finalizando com um último Capítulo de conclusão.

Capítulo 2: Apresenta uma r**evisão bibliográfica** da área, focando-se nos temas de *formação de coalizões*, *leilões e negociações*, e por último da área denominada *alocação de recursos em sistemas multiagentes*.

Capítulo 3: Apresenta a **fundamentação teórica** do algoritmo. Cobre as técnicas e algoritmos utilizados na montagem da *heurística ganha-ganha*, entre elas: conceito de organização social, planejamento discreto em sistemas multiagentes e teoria dos jogos.

Capítulo 4: Apresenta o **algoritmo de formação de coalizões**. Para tanto, define o modelo de agentes baseados em recursos, o jogo de votação baseado em recursos, e os resultados teóricos do algoritmo. O Capítulo apresenta um Seção de **resultados** correspondentes aos testes do algoritmo em ambiente estático.

Capítulo 5: Apresenta a **simulação da formação de coalizões em um mundo em grade**. Descreveremos o projeto do ambiente de simulação, sua implementação e os resultados medidos na execução dos cenários de simulação.

Capítulo 6: Apresenta a **conclusão** da dissertação.

1.4 SOBRE A NOTAÇÃO

A notação para formação de coalizões tem a teoria dos jogos como referência, no entanto, esses algoritmos não são a principal preocupação dessa área do conhecimento. Dessa forma, há uma lacuna em padrões para representar os algoritmos de formação de coalizões, sendo usualmente utilizado um misto de notações da teoria dos jogos coalizionais, teoria dos conjuntos e análise combinatória. Além disso, por causa dessa falta de padronização, cada artigo ou técnica desenvolvida para formação de coalizões tem alguma peculiaridade na sua notação. O Capítulo 2 apresentará os principais artigos dessa área, e lá pode-se perceber que cada autor utiliza uma notação, apesar de haver alguma similaridade entre eles.

Outra dificuldade com relação à notação é que abordaremos outros temas que juntos formam a nossa solução para o problema de formação de coalizões. Esses temas cobrem organizações sociais e coalizões, planejamento, teoria dos jogos coalizionais e métodos aproximativos para o cálculo do valor de Shapley. São diferentes temas em áreas diversas, com algoritmos, definições e equacionamentos adequados ao seu fim, mas com abordagens diferentes entre si. O Capítulo 3 é dedicado a apresentação desse conteúdo.

Sumarizando, as dificuldades ocorrem em dois níveis.

1. Utilização de símbolos (letras, índices ou caracteres) diferentes para representar o mesmo conceito.

   Isso acontece entre os diversos algoritmos de formação de coalizões apresentados no Capítulo 2 e a heurística desenvolvida a partir do Capítulo 4. Adotamos como **padrão manter a notação original sempre que possível**. De outra forma seria muito trabalhoso e inseguro, no sentido de poder gerar inconsistências, tentar modificar os símbolos utilizados em cada algoritmo, buscando a homogeneidade entre eles. Consideramos que essa dificuldade é contornável pelo leitor, dado que os procedimentos iterativos são parecidos.

2. Utilização de símbolos iguais representando conceitos diferentes.

   Isso ocorre entre os temas do Capítulo 3 (fundamentação teórica), que por natureza são diferentes. Novamente adotamos o procedimento de manter a nomenclatura original dos textos de referência. Consideramos essa dificuldade contornável pois cada tema é abordado em Seção específica, portanto, os símbolos são utilizados dentro de um contexto. Além disso, as partes do trabalho que unificam os algoritmos, fazem apenas referência aos algoritmos gerais, não havendo repetição das formulações.

2 REVISÃO DA LITERATURA

Nesse Capítulo são analisadas e comentadas referências que cobrem o tema proposto. A área de sistemas multagentes é muito ampla, portanto, procuramos restringir a bibliografia, bem como a fundamentação teórica, às questões relevantes para formação de coalizões e compartilhamento de recursos em sistemas multiagentes. As referências apresentadas são utilizadas diretamente como base para o desenvolvimento do trabalho ou serverm como contraponto para discussão ou fundamentação.

A bibliografia que apresentaremos está relacionadas aos 2 assuntos principais do nosso trabalho: *formação de coalizões a alocação de recursos*. Adicionalmente apresentaremos bibliografia sobre o assunto *leilões e barganhas*, que são as estratégias mais utilizadas para alocação de recursos em sistemas multiagentes.

Uma dificuldade na apresentação da bibliografia é que cada artigo utiliza uma notação própria na representação e análise dos algoritmos ou modelos tratados. Não seria possível nesse trabalho apresentar detalhadamente cada abordagem. Por outro lado, acreditamos que as notações são inteligíveis, assumindo-se um público com alguma familiaridade com algoritmos iterativos ou combinatórios.

2.1 FORMAÇÃO DE COALIZÕES

Os benefícios da utilização de sistema multiagentes para tratar problemas difíceis é atualmente bem conhecido, no entanto, muitos dos esforços da pesquisa em Sistemas Multiagentes (SMAs) tem se voltado para questões relacionadas ao trabalho em equipe (Van De Vijsel & Anderson, 2004). Uma das principais questões envolvidas é como organizar os agentes de forma a potencializar os ganhos dessas associações e, por outro lado, uma das respostas apresentadas para esta questão é através da utilização de *coalizões*. O procedimento de *formação de coalizões* é uma das etapas envolvidas na utilização de *coalizões* como estrutura organizacional de um SMA. Nessa Seção apresentaremos artigos de formação de coalizões relevantes para o nosso trabalho.

Em Shehory & Kraus (1998) é desenvolvido um algoritmo de formação de coalizões para execução compartilhada de tarefas. Esse é o trabalho mais citado para divisão de tarefas utilizando um algorítmo combinatório. O procedimento de formação de coalizão utiliza um modelo de tarefas e capacidades em que cada agente do conjunto $\{A_1, A_2, \ldots, A_n\}$ possui um vetor de capacidades representado como um inteiro positvo $B_i = < b_1^i, \ldots, b_r^i >$, e cada

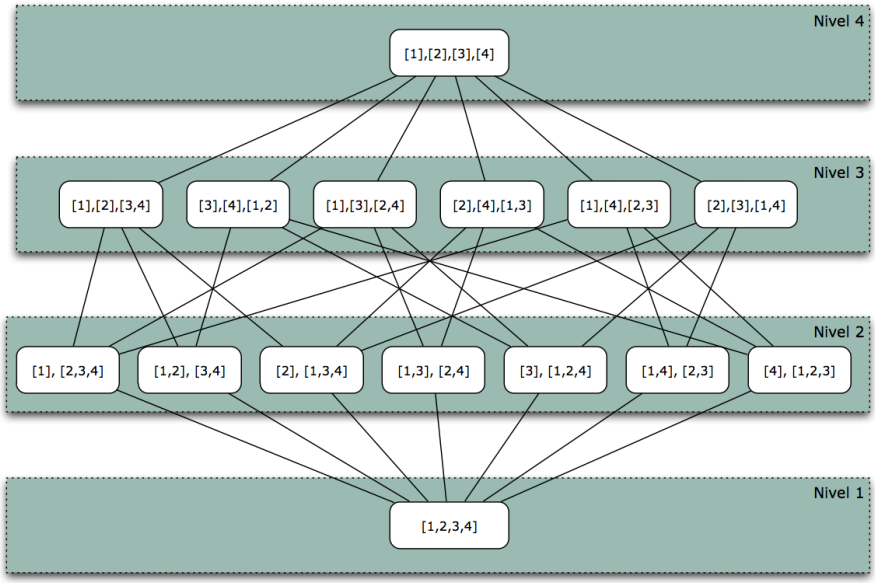


Figura 1 - Estrutura Coalizional

tarefa do conjunto $T = \{t_1, \ldots, t_m\}$ possui um conjunto de capacidades necessárias para sua execução, $B_j = < b_1^j, \ldots, b_r^j >$. O algoritmo é dividido em duas etapas. A primeira, executa o calculo distribuído do valor de todas as coalizões, e a segunda executa a escolha de qual coalizão formar. O cálculo da coalizão é efetuado sobre o vetor de capacidades da coalizão $B_C = \sum_{\forall A_i \in A} B_i$. O benefício esperado da execução da tarefa t_j pela coalizão C é $e_j := f(B_C, |C|)$ se $\forall i, b_j(i) \leq b_C(i)$ e $e_j = 0$ caso contrário. e_j é adicionado a uma lista E_C da coalizão C. O valor da coalizão, denotado por V_C será o maior valor de e_j e a relacionada tarefa t_j. Cada agente seleciona as coalizões de que fará parte pelo valor ponderado da coalizão $\frac{V_C}{|C|}$. Cada agente participa de apenas uma coalizão para realização de uma tarefa. Em Shehory et al. (1998) um exemplo prático de utilização do algoritmo é apresentado para o contexto de agentes para coleta e filtros de páginas web.

Sandholm et al. (1999) apresenta um estudo de complexidade na formação de coalizões e demonstra que é possível ter uma medida do esforço mínimo para que um algoritmo de formação de coalizões alcançe uma razão da estrutura coalizional ótima. Esse é um trabalho referencial na formação de coalizões por ter conseguido estabelecer um parâmetro de qualidade para as coalizões formadas. Especificamente, discute a geração de estruturas coalizionais em configurações onde existam demasiadas coalizões para serem enumeradas ou avaliadas. Estuda em que subconjunto das coalizões focar a busca de forma a garantir que seja montada uma estrutura coalizional com uma qualidade dentro de limites definidos da estrutura coalizional ótima. Suscintamente, CS é a estrutura coaizional, ou seja, uma partição do conjunto N de agentes, sendo $v(CS)$ o valor da estrutra coalizional, dado por $v(CS) = \sum_{S \in CS} v(S)$. Usu-

almente o objetivo é identificar uma estrutura coalizional que maximize o bem-estar coletivo dos agentes, representado por $CS^* = \arg\max_{CS \in 2^{2^N}} v(CS)$. Os autores buscam estabelecer quais estruturaras coalizionais avaliar para conseguir determinar o menor k que permita estabelecer $k \geq \frac{v(CS^*)}{v(CS^*_{visitadas})}$, onde $CS_{visitadas}$ representa as estruturas visitadas. Os autores demonstram que é necessário visitar os dois primeiros níveis do grafo de estruturas coalizionais (fig. 1), e o valor de k = |N|. Além disso, o número de nós do grafo que devem ser visitados é $2^N - 1$.

Em Bachrach & Rosenschein (2008) é apresentado um jogo denominado *jogo coalizional de habilidades* (em inglês, coalitional skill game). Essa é uma formulação do jogo coalizional $< N, v >$ (jogos coalizionais são apresentados na Seção 3.3 da fundamentação teórica), em que cada agente tem um conjunto de habilidades requeridas para completar diversas tarefas. A idéia é similar à de Shehory & Kraus (1998), mas a abordagem é aquela da teoria dos jogos "clássica", que busca a demonstração das propriedades do jogo, em especial dos **conceitos de solução**. O jogo é formulado da seguinte forma: $N = \{a_1, \ldots, a_n\}$ é o conjunto de agentes, $T = \{t_1, \ldots, t_m\}$ o conjunto de tarefas e $S = \{s_1, \ldots, s_k\}$ representa o conjunto de habilidades (*skill* em inglês). As habilidades que um agente possui é denotada por $S(a_i) \subset S$, e as habilidades requeridas para execução de uma tarefa $S(t_j) \subset S$. O total de habilidades de uma coalizão é igual a soma do conjunto de habilidades dos seus membros, e denotada por $S(C) = \cup_{a_i \in C} S(a_i)$. Se todas as habilidades para execução de uma tarefa são possuídas pela coalizão então temos $perform(C, t_j) \Leftrightarrow S(t_j) \subset S(C)$. Simetricamente temos $T(C)$ como todas as tarefas que a coalizão C pode realizar, ou seja, $T(C) = \{t_j \mid perform(C, t_j)\}$. A função utilidade $u : 2^T \rightarrow \mathbb{R}$ associa um valor a cada conjunto de tarefas. Os autores assumem que se $T_1 \subset T_2 \Rightarrow u(T_1) \leq u(T_2)$. Finalmente, para finalizar a formulação do jogo, a função característica é dada por $v(C) = u(T(C))$. Os autores demonstram a complexidade de identificação do core, valor de Shapley e índice de poder de Banzhaf.

Wooldridge & Dunne (2004) apresenta uma formulação para jogos coalizionais em que os agentes apresentam um conjunto de objetivos a serem realizados, chamado de *jogos coalizionais qualitativos*. Nesses jogos a função característica não é numérica, mas representa os conjuntos de objetivos que satisfazem o agente. Dessa forma, $v : 2^N \rightarrow 2^{2^G}$, onde G representa o conjunto de objetivos (G é utilizado do inglês *goals*). Cada agente possui um conjunto de objetivos G_i sobre o qual não tem preferência (qualquer objetivo realizado satisfará o agente). O problema nessa formulação é a dificuldade na representação da função característica. Uma representação direta que enumere todas as permutações de agentes e objetivos não é factível. Os autores propõem como solução a representação da função característica como uma fórmula da lógica proposicional $\Psi[N, G]$ sobre os agentes e objetivos, onde cada elemento desses conjunto representa um clausula. Dessa forma, um *jogo coalizional qualitativo* é re-

presentado como a estrutura $Q = < G, N, G_1, \ldots, G_n, \Psi >$. Uma extensão desse modelo para utilização da noção de recurso é proposto em Wooldridge & Dunne (2006). Nesse jogo, existe um conjunto R de recursos e duas funções de mapeamento: uma que associa a cada agente um conjunto de dotações de recursos (do inglês *endowment*), $\mathbf{en} : N \times R \rightarrow \mathbb{N}$ e uma função que define o conjunto de recursos requeridos para realização de um objetivo, $\mathbf{req} : G \times R \rightarrow \mathbb{N}$. Dessa forma, um *jogo coalizional de recursos* é representado como a estrutura $\Gamma = < N, G, R, G_1, \ldots, G_n, \mathbf{en}, \mathbf{req} >$.

Apt & Witzel (2009); Apt & Radzik (2006) apresentam uma estratégia de formação de coalizões baseada na transformações de partições da grande coalizão através de preferências parciais. Os autores demonstram que apenas com funções de junção (do inglês *merge*) e separação (do inglês *split*) é possível chegar a uma estrutura coalizional estável, desde que o critério de montagem de coalizão seja dado por uma ordem de preferência global (bem-estar social). O operador $\rhd$ define a ordem de preferência entre duas estruturas coalizionais, sendo definido como $CS_A \rhd CS_B \Leftrightarrow v(CS_A) > v(CS_B)$. O artigo define as operações de separação e junção seguindo a ordem de preferência entre estruturas coalizionais, assim, $\mathbf{merge} : \{S_1, \ldots, S_k\} \cup P \rightarrow \{\bigcup_{j=1}^{k} S_k\} \cup P$ se $\{\bigcup_{j=1}^{k} S_k\} \rhd \{S_1, \ldots, S_k\}$, e vice-versa para a função de separação. O artigo demonstra que se as funções de *split-merge* forem seguindas, eventualmente chega-se a uma estrutura coalizional estável sobre essas operações.

Em Saad et al. (2009b) é apresentado um tutorial de teoria dos jogos coalizionais com aplicações em engenharia de comunicações e redes sem fio. Os autores categorizam os jogos coalizionais em 3 tipos: **jogos coalizionais canônicos**, **jogos de formacão de coalizões** e **jogos coalizionais em grafo**. A aplicação de coalizões na análise e controle de redes de comunicações vem da necessidade do desenvolvimento de dispositivos móveis que participem de redes dinamicamente e que sejam capazes de tomar decisões em cenários de colaboração e competição (Saad et al., 2009b, pg. 77). Em Saad et al. (2009a) é desenvolvido o algoritmo para cooperação de usuários através de coalizões e transmissão via TDMA. O trabalho mostrou ganho de até 40% no retorno do usuário além de tratar eficientemente a mobilidade dos usários. O algoritmo é baseado em (Apt & Witzel, 2009) que utiliza operações de *merge-split* a partir de uma partição do conjunto de agentes (N), incrementando monotonicamente os ganhos do usuário ao executar-se as operações de *merge-split*.

Em Service & Adams (2010, 2011) é apresentado um algoritmo de formação de coalizões aproximado baseado em programação dinâmica.

Rahwan et al. (2009) desenvolve o algoritmo *IP*, em referência à *partição inteira*, que corresponde na identificação de limites superiores e inferiores de uma estratégia de dividir o espaço de busca da estrutura coalizional em partições inteiras. Uma partição inteira é um conjunto de números inteiros que somam um valor N, assim, por exemplo,

uma partição inteira de índice 4 (4 agentes) é igual ao conjunto de 5 partições distintas, $\{[4], [1,3], [2,2], [2,1,1], [1,1,1,1]\}$. O algoritmo divide o espaço de busca em subespaços P_G, onde G indica a quantidade de elementos da partição inteira associada. Por exemplo, $P_1 = \{P_{[4]}\}, P_2 = \{P_{[2,2]}, P_{[1,3],...}\}$, e cada subpartição de P_G contém as partições com a dada conformação, ex. $P_{[2,2]} = \{\{a_1, a_2\}, \{a_1, a_3\}, \{1_1, a_4\}, \dots\}$. Os autores demonstram que a partir do máximo e mínimo de valores de coalizões de tamanho $|S| \leq |N|$ é possível identificar limites de valores para os subespaços de P_G, diminuindo significativamente o esforço de busca da coalizão ótima. No entanto, a forma de recuperar o máximo e mínimo da coalizões não foi explorado. Dessa forma, a quantidade de avaliações do valor da coalizão na pior das hipóteses, continua sendo completo, ou seja, todas as coalizões são visitadas pelo menos uma vez, o que gera uma complexidade de $O(2^N)$. Em Rahwan & Jennings (2005) é apresentado o algoritmo original de programação dinâmica distribuída, e em Rahwan & Jennings (2008) a partição de espaços inteira. Em Michalak et al. (2010) é desenvolvida uma versão distribuída do algoritmo IP.

Finalmente, em Cara & Netto (2013) é apresentada uma proposta de representação da estrutura coalizional como um grafo adaptativo.

2.2 LEILÕES E NEGOCIAÇÕES

Desenvolvido por Smith (1980); Smith & Davis (1981) CNET é o protocolo mais utilizado para alocação de tarefas. Segundo Smith (1980) "*CNET é um protocolo de alto nível para comunicação entre nós de um solucionador de problemas distribuídos (distributed problem solver). Ele facilita o controle distribuído de tarefas executadas cooperativamente (denominada task-sharing)*". Além disso, "*a principal questão a ser resolvida no compartilhamento de tarefas é como estas serão distribuídas entre os nós de processamento. Deve haver um meio de nós com tarefas a serem executadas acharem os nós dispoíveis mais apropriados para a sua execução*". Para endereçar essas questões o protocolo CNET utiliza um mecanismo de leilão, em que o nó com necessidade de execução de um conjunto de tarefas publica um anúncio com as tarefas a serem executada e detalhes da tarefa. Cada nó disponível avalia sua capacidade de execução da tarefa, e estabelece uma medida por ela, e os habilitados fazem ofertas para a sua execução. O nó anunciante age como gerente das tarefas e seleciona a melhor oferta após um período determinado de tempo.

Leilões combinatórios são aqueles que alocam um conjunto de recursos para cada agente, buscando o perfil de alocações que maximize uma função de bem-estar social. Ou seja, dado um conjunto $\mathcal{Z}$ de itens para serem leiloados, um perfil de alocações Z é a enupla $(Z_1, Z_2, ..., Z_N)$, de $Z \subset \mathcal{Z}$, o leilão combinatório busca maximizar a função $sw(Z_1, ..., Z_N, v_1, ..., v_N)$,

tal que $sw = \sum v_i(Z_i)$. Segundo *Lehman, Muller & Sandholm (2006)* Lehmann et al. (2006), o problema de identificação da melhor alocação é um *problema de otimização combinatória*, e tem complexidade *NP-Hard*.

Osborne & Rubinstein (1990) desenvolveram um modelo formal de barganhas conhecido como *modelo de ofertas alternadas* que é dos mais influentes e geral na área. Nesse modelo as ofertas acontecem alternadamente entre dois agentes, segundo a estratégia de cada agente a partir de um espaço de propostas válidas. É definido um conjunto de regras de finalização, e o ciclo de propostas acontece até alguma regra de finalização ser atingida. Osborne & Rubinstein (1990) demostram que se os ciclos forem infinitos o conceito de *equilíbrio de Nash* é insuficiente para resolver o jogo, e que é necessário impor um limite aos ciclos de negociações e uma penalidade a ser aplicada por recusa. Com essas suposições é estrategicamente possível prever o melhor acordo entre as partes.

Uma técnica específica para alocação de tarefas foi desenvolvida em Rosenschein & Zlotkin (1994), que consiste na definição de uma função de custo individual (de cada agente) para um conjunto de tarefas, e a atribuição das tarefas para cada agente, que podem trocar tarefas através de acordos. Através desse *framework* simples são desenvolvidos os conceitos de *otimilidade de Pareto* e *racionalidade individual*. Formalmente, um *ambiente orientada à tarefas* é formado pela tupla $<T, Ag, c>$, em que a função custo c é caracterizada por $c{:}2^T{\rightarrow}\mathbb{R}$. Um *acordo* δ ocorre entre dois agentes através de enupla $\delta{:}<D_1, D_2>$, tal que $D_1{\cup}D_2 = T_1{\cup}T_2$. Temos que o custo do acordo para o agente i é $c_i(\delta) = c_i(D_i)$, e a função utilidade é definida por $u_i(\delta) = c(T_i) - c_i(\delta)$. A divisão de tarefas original $\theta :< T_1, T_2 >$ é denominada acordo conflituoso. No mesmo sentido de *jogos coalizionais* é definido o conceito de dominância $\delta_1 \succ \delta_2$, acordo 1 domina acordo 2, se $\forall i, u_i(\delta_1) \geq u_i(\delta_2)$, e $\exists i, u_i(\delta_1) > u_i(\delta_2)$. Um acordo δ é *Pareto ótimo* se $\nexists \delta' \succ \delta$. Ou seja, *uma acordo é Pareto ótimo se não há outro acordo que vá melhorar algum agente sem piorar outro*. Finalmente, um acordo é *individualmente racional* se é pelo menos preferível à tarefa original, ou seja, $u_i(\delta) > u_i(\Theta)$.

Kraus (2001) apresenta um *survey* de técnicas para construção de acordos em sistemas multiagentes. A autora avalia e categoriza as técnicas baseando-se em *teoria dos jogos* e economia.

2.3 SISTEMAS MULTIAGENTES E ALOCAÇÃO DE RECURSOS

Chevaleyre et al. (2005) apresenta uma visão geral da área de Multiagent Resource Allocation (MARA), enfatizando as preocupações que devem direcionar o projetista de um SMA para alocação de recursos. O artigo é uma excelente sumarização, apresesentando uma tipologia de funções de bem-estar social, aplicações e dinamica de execução das aplicações.

(Sandholm, 1998) define o modelo de contrato entre agentes para realização de tarefas. Os autores definem 4 tipos de contratos, sendo o principal deles o O-Contrato. O-Contratos representam uma negociação entre os agentes i, j para execução de uma tarefa $T_{i,j}$ por um preço $p_{i,j}$, assim um O-contrato é dado pelo par $< T_{i,j}, p_{i,j} >$. O autor demonstra que existe um caminho de O-contratos entre qualquer alocação inicial de tarefas e a alocação ótima, mas esse caminho pode passar por contratos **não** individualmente racionais. No entanto, o artigo não fornece um processo para montagem dos contratos.

Em Endriss et al. (2003) é efetuado um estudo sobre troca de recursos através de negociações. Nesse artigo estuda-se a evolução do bem-estar social (sw) a partir da execução de acordos, representados por $\delta = (A, A')$ [1], onde A representa uma alocação (uma partição dos recursos do sistema entre os agentes). Utiliza-se os modelos de contratos apresentados em Sandholm (1998) aplicados à alocações de recursos. Os autores trabalham com dois cenários de negociação para alocação de recursos: cenário monetário, em que agenttes podem efetuar pagamentos para compensarem outros agentes por acordos desvantajosos, e cenário não monetário. Os acordos são individuais, ou seja, são *O-Contratos* na nomenclatura de Sandholm (1998). Para os cenários monetários é demonstrado o teorema sobre *acordos racionais* que estabelece que em um acordo $\delta = (A, A')$, $sw(A) < sw(A') \Leftrightarrow u_i(A') - u_i(A) > p(i), \forall i \in A$, onde $p(i)$ é o pagamento efetuado entre os agentes participantes do acordo. Para o cenário monetário vale a regra $\sum p(i) = 0 \forall i \in N$, além de que $p(i) = 0$ se $A(i) = A'(i)$. O teorema da racionalidade do acordo é forte no sentido de que vincula a racionalidade individual ao aumento do bem-estar, vale dizer, que se o agente agir racionalmente o bem-estar social terá aumento em cada acordo. Com esse resultado os autores provam que uma sequência de acordos leva, eventualmente, ao acordo maximal. Para o cenário não monetário, por sua vez, não chega-se ao mesmo resultado sobre acordos racionais, ou seja, a individualidade racional não é suficiente para garantir-se o ótimo do bem-estar social. Nesse caso utiliza-se outro conceito mais flexível que é o de *individualidade cooperativa*, definido como $\delta(A, A')$ é cooperativamente racional $\iff u_i(A) \leq u_i(A') \forall i \in A$ e para pelo menos um agente j, $u_j(A) < u_j A')$, ou seja, pelo menos um agente tem ganho no acordo. Através dessa noção não é possível chegar-se ao acordo ótimo, no entanto os autores demonstrar que com acordos individuais que sejam cooperativamente racionais, chega-se a um acordo *Pareto ótimo*. Um acordo é Pareto ótimo se, e somente se, não há um acordo que seja melhor para um agente e pior para algum outro, ou seja, se não é possível melhorar sua situação sem piorar a de outro agente.

Em Chevaleyre et al. (2004, 2008) os autores apresentam um *framework* para alocação de pacotes de recursos, considerando cenários em que os agentes são modelados através

[1] δ em referência a *deal* em inglês.

de funções utilidade *k-aditivas*. O algoritmo é uma extensão do esquema de negociação denominado *O-Contratcts*.

A função de bem-estar social de Nash é estudada em Nongaillard et al. (2010) sob o ponto de vista de negociações bilaterais. A função de bem-estar social de Nash é dada por $sw_N = \Pi_{i \in N}\ u_i(A_i)$. Essa função apresenta características interessantes, a principal é que ao se maximizar a função de bem-estar social de Nash, diminui-se as diferenças entre as utilidades individuais dos agentes, ou seja, é uma função que estimula a divisão igualitária dos recursos. Tome o exemplo da seguinte configuração de 2 agentes com 6 unidades de recursos. A alocação que maximiza o bem-estar é 3 recursos para cada agente $sw_N = 9$ (qualquer outral alocação será menor que 9, como {4,2} ou {5,1}). Os autores apontam as dificuldades de cálculo, centralizada ou distribuído, em se utilizar essa função, e propôem um mecanismo de vizinhança que tornaria mais barata a sua utilização.

2.4 COMPARAÇÃO ENTRE OS MODELOS

Algoritmos de formação de coalizões podem ser divididos em 3 categorias (Rahwan & Jennings, 2008, pg. 1417):

1. Algoritmos de programação dinâmica: Essa classe de algoritmos foram desenvolvidas para solucionar o *problema de partição completa de conjuntos* [8] e o *problema de identificação de vencedor em leilões combinatórios* [3], sendo que ambos podem ser vistos como análogos ao *problema de geração da estrutura coalizional*. A vantagem desses algoritmos é que eles podem ser executados em $O(3^n)$, o que é exponencialmente melhor do que a busca exaustiva que requer $O(n^n)$. No entanto, o espaço armazenado é $2^n - 1$, visto que cada coalizão deve ser calculada.

2. Heurísticas: são algoritmos que buscam retornar soluções rapidamente, no entanto não provêem qualquer garantia de localizar uma solução ótima. Na realidade as soluções podem ser arbitrariamente piores que o ótimo pois nem sempre é possível verificar o fato de que uma solução é ótima.

3. Algoritmos *Anytime*: esses algoritmos agem gerando soluções iniciais garantidamente dentro de limites do ótimo, e então incrementam a qualidade das soluções. A característica desses algoritmos é que podem ser interrompidos a qualquer momento após a geração da solução inicial, e sempre conhece-se a distância máxima que a execução se encontra do ótimo. A questão que aqui se apresenta é quão custosa é a parte de inicialização, e qual o grau de incremento de soluções, uma vez que ambos os pontos podem ser extremamente custosos, o que de fato ocorre no caso de geração de estruturas coalizionais.

O primeiro algoritmo apresentado, Shehory & Kraus (1998) apresenta um modelo de tarefas e capacidades, e pontua as associações entre os agentes a partir de uma função das tarefas totais a serem realizadas por uma coalizão *versus* as capacidades dadas pelos seus agentes. O modelo é próximo do modelo que utilizaremos, no entanto, o domínio de problemas é de Distributed Problem Solving (DPS), ou seja,os agentes não são autônomos, no sentido de que não tem agenda própria. A função dos agentes é contribuir com capacidades para um objetivo comum.

O trabalho de Sandholm et al. (1999) é referência na análise do esforço necessário para alcançar uma estrutura coalizional ótima, apresentando parâmetros da distância mínima do ótimo atingida quando se pesquisam novos níveis da estrutura coalizional. No entanto, não é apresentada nenhuma alternativa além da pesquisa exaustiva de coalizões, o que é inviável para qualquer número significativo de agentes.

Bachrach & Rosenschein (2008) apresentam uma formalização chamada *jogos coalizionais de habilidades (coalitional skill games)*, no entanto não estabelecem nehum algoritmo prático para formação de coalizões, focando-se na demonstração das propriedades dos jogos. Além disso, os agentes assumem que o jogo é super-aditivo, uma vez que adicionando-se novas habilidades a uma coalizão não diminui-se o valor da coalizão.

A formulação de Wooldridge & Dunne (2004) reduz o problema de alocação de recursos ao problema de satisfação de uma fórmula proposicional. O problema é demonstrado ser *NP-Completo*, portanto seu tempo de execução é exponencial.

Em Apt & Witzel (2009) e Apt & Radzik (2006) é feito o estudo da formação de coalizões quando podemos estabelecer uma ordem de preferëncia entre estruturas coalizionais. Demonstra-se que a estrutura coalizional ótima é alcançada quando utilizamos apenas duas operações, *merge* e *split* de coalizões. O resultado é semelhante ao apresentado por Sandholm (1998) para negociações, no entanto, nenhum algoritmo prático de *quais operações de merge/split executar* é apresentado.

Rahwan et al. (2009) desenvolve um algoritmo baseado em *programação dinâmica* e numa nova estratégia de partição do espaço de busca, no entanto, todas as coalizões devem ser visitadas pelo menos uma vez.

3 FUNDAMENTAÇÃO TEÓRICA

Afim de darmos os subsídios para o desenvolvimento da heurística responsável pela formação das coalizões, apresentaremos primeiramente nossa visão de *agência* e *sociabilidade* em sistemas multiagentes. Apresentaremos um sumário de estruturas organizacionais em sistemas multiagentes e nos deteremos em *coalizões*, que é a estrutura organizacional adotada para compartilhamento de recursos. Dedicaremos também uma Seção ao tópico de planejamento discreto em sistemas multiagentes, tópico necessário para implementar os processos de decisórios do agente.

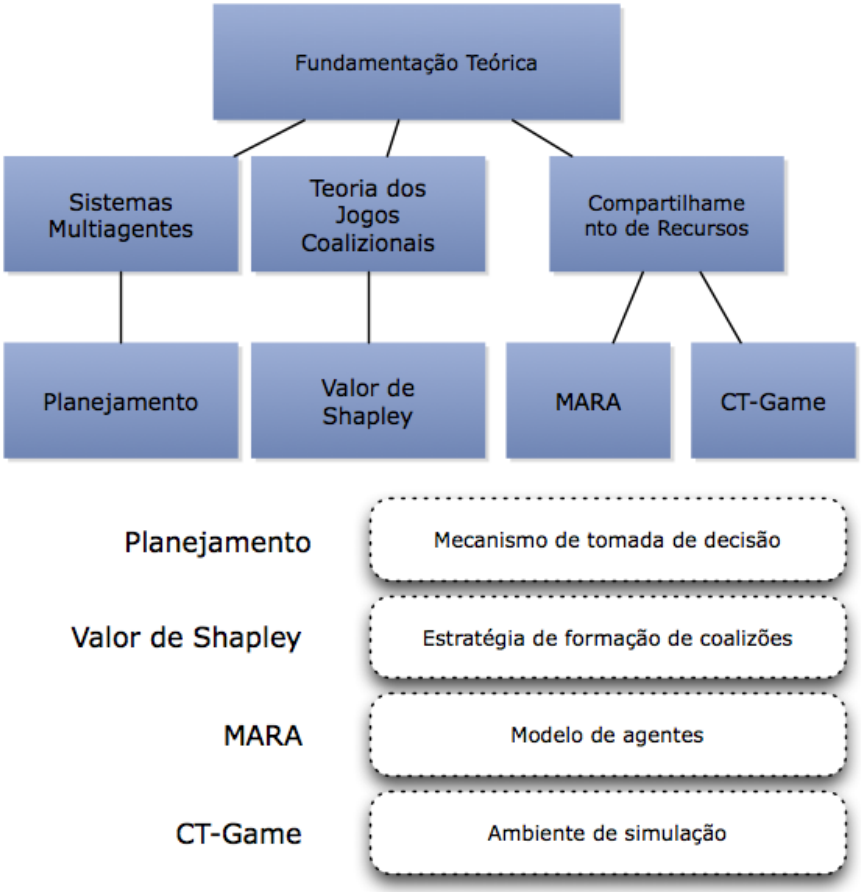


Figura 2 - Assuntos cobertos pela fundamentação teórica

A figura acima apresenta um mapa das seções e subseções desse capítulo, com a aplicabilidade de cada parte na dissertação. A árvore de tópicos está dividida em 3 grandes áreas: *Sistemas Multiagentes*, *Teoria dos Jogos Coalizionais* e *Compartilhamento de Recursos*. Os tópicos específicos dentro de cada grande área da dissertação, são planejamento, valor de Sha-

pley, *Multiagent Resource Allocation (MARA)* e *CT-Game*. A aplicabilidade de cada assunto está apresentada abaixo da árvore de seções. Em especial, a questão de planejamento foi abordada pois é utilizada na construção dos mecanismos de tomada de decisão dos agentes.

3.1 SISTEMAS MULTIAGENTES

Um sistema multiagente é aquele em que diversas entidades, denominadas agentes, estão presentes. A noção de *agência* varia dependendo do domínio, no entanto, há dois atributos que estão presentes na maior parte das definições:*autonomia* e *estar situado em um ambiente* (Woolridge, 2001). Autonomia relaciona-se a independência, liberdade, autodeterminação, ou seja, capacidade de tomar suas próprias decisões e perseguir os meios para realizá-las. No contexto de agentes computacionais adotaremos o sentido de autonomia, como a capacidade de seguir sua própria agenda. Por sua vez, situar-se em um ambiente significa ser capaz de perceber os elementos externos através de seus sentidos e agir sobre esse meio, obedecendo as regras do ambiente.

Do ponto de vista de arquitetura, cada agente consiste minimamente de um estado interno, um conjunto de ações possíveis, percepções e alguma forma de processo decisório para seleção de ações. No caso de SMAs os agentes compartilham um ambiente e podem ter uma representação interna do mesmo.

3.1.1 Ambiente

O ambiente em que o agente se encontra pode ter uma importante função reguladora. No nosso trabalho vamos adotar a perspectiva sumarizada por Weyns et al. (2007) do ambiente como cidadão de primeira ordem.

A prática corrente associa o ambiente aos recursos externos ao agente e também à infraestrutura de comunicação. Essas seriam as principais responsabilidades do ambiente, pensado apenas como o meio em que o agente se insere e com pouca influência na concepção sistema multiagente. No entanto, também são responsabilidades do ambiente:

1. O ambiente estrutura o SMA: onde é definida a estrutura física (topologia e localização), a estrutura de comunicação inter-agentes e extra-sistema e a estrutura social (grupos e sociedades).

2. O ambiente encapsula serviços e recursos: meio de acesso aos serviços da plataforma computacional.

3. O ambiente mantém a dinâmica: gerência os processos que impliquem em mudança de

estado do ambiente.

4. O ambiente é localmente observável e acessível: disponibiliza informações e permite acesso aos recursos, de acordo com o contexto em que o agente se encontra.

5. O ambiente define papeis: normatizações, regras e papeis podem estar encapsulados no ambiente. Tem alçada para impedir ações dos agentes, garantir contratos, ou mesmo atuar como árbitro em situações de conflito.

3.1.2 Agentes Sociais

Quando mais de um agente se encontra no mesmo ambiente, dependendo das suas capacidades e da finalidade do sistema, podem ocorrer interações. Essas interações podem ser simples, como troca de informações, ou complexas, como coordenação para realização de objetivos em comum. Segundo Davidsson (2001),

> O [sentido] uso do termo sociedade em SMA é análogo ao quando aplicado à humanos ou sistemas ecológicos. (...) e nesse contexto, o papel da sociedade é permitir que os seus membros possam coexistir em um ambiente compartilhado, e perseguir seus objetivos na presença de outros.

Dadas as características de autonomia e adaptatividade, agentes podem possuir atributos sociais similares aos dos humanos. Dessa forma, é válido dizer que um agente pode ser ou não confiável e que determinada configuração é justa ou não. Os agentes se relacionam com outros baseados em sua própria personalidade, e essas relações podem ser altruísticas, egoístas ou, ainda, baseadas em reciprocidade.

Dependendo do propósito da sociedade, ela deve apresentar as seguintes propriedades, em diferentes graus:

1. Abertura: possibilidade de agentes participarem do sistema sem restrições.

2. Flexibilidade: nível de restrição imposta pela sociedade às ações dos agentes.

3. Estabilidade: nível de previsibilidade das ações dos agentes.

4. Confiabilidade: nível pelo qual os agentes podem confiar na sociedade.

Uma característica da interação social é a capacidade de montar organizações como, por exemplo, times, grupos, coalizões e instituições. Segundo Horling & Lesser (2004), foi repetidamente mostrado que a organização de um sistema pode ter impacto na sua performance. Organizações podem ser utilizadas para limitar o escopo de interações, reforçar através da

quantidade de integrantes, reduzir ou gerenciar incerteza, explicitamente incrementar a redundância ou, formalizar objetivos de alto-nível que nenhum agente poderia sozinho.

É geralmente aceito que há um tipo de organização adequado à cada situação (Horling & Lesser, 2004, pg. 281), sendo que cada tipo inevitavelmente trará custos e benefícios na sua utilização. A Figura 3 apresenta as principais estruturas organizacionais, definidas abaixo.

- **Hierarquias:** são estruturas em forma de árvore, onde a informação flui apenas entre os nós conectados.

- **Holarquias:** grupos de hierarquias multi-nível. Baseados em sistemas biológicos, astrológicos e sociais. Nessa organização, cada grupo tem características derivadas dos seus elementos, e ao mesmo tempo esse grupo contribui com características para os grupos mais elevados.

- **Coalizões:** Coalizões são organizações orientadas a um fim e com tempo de vida curto. São formadas com um propósito em mente e se dissolvem assim que a necessidade não mais existe, as coalizões deixam de ter condições de obter o seu fim ou mesmo quando uma massa crítica de agentes se desassocia. Coalizões não tem uma estrutura organizacional fixa; não há uma relação hierárquica entre os seus membros. Coalizões podem ser mutuamente exclusivas com relação aos seus membros, mas podem também existir sistemas em que os agentes pertençam simultaneamente a uma ou mais coalizões.

- **Times:** Times consistem em um grupo de agentes cooperativos que concordaram em trabalhar juntos para realização de um objetivo em comum. A diferença entre times e coalizões é que os membros de um time trabalham com a finalidade de maximizar a utilidade do time e não a sua utilidade individual. Além disso, times tem uma existência mais longa. Ainda que possam ser alterados de tempos em tempos pela substituição de seus agentes, isso não é feito a qualquer momento. Vale tambem ressaltar que normalmente o seu tamanho é fixo.

- **Congregações:** Similarmente a coalizões e times, congregações de agentes são grupos de indivíduos que se uniram em uma organização tipicamente plana, a fim de alcançarem benefícios em comum. Diferentemente dos outros paradigmas, congregações são de longa duração e não formadas com um objetivo específico em mente. No lugar disso, congregações são formadas entre agentes com características similares ou complementares. Na Figura 3e as diferentes cores representam propósitos potencialmente heterogêneos subjacente a cada agrupamento, em comparação com coalizões, que tem um propósito único para formação. Indivíduos não tem necessariamente um objetivo simples ou fixo,

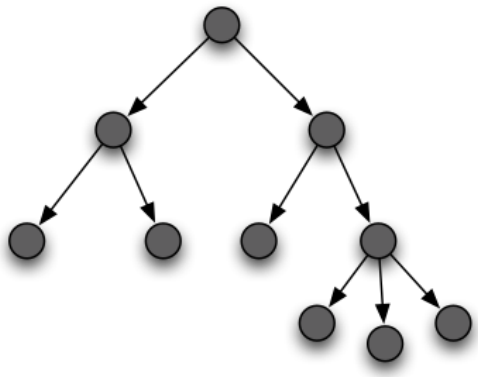
mas tem um conjunto constante de capacidades ou requerimentos que os motivam a se congregaram.

- **Sociedades:** Sociedades são construções sociais de longo termo e, diferentemente de outros paradigmas, são inerentemente sistemas abertos. Agentes de diferentes origens podem vir e ir, mas a sociedade persiste, agindo como um ambiente através do qual os participantes se encontram e interagem. Sociedades impõem estrutura e ordem, mas o arranjo específico de interações pode ser bastante flexível. Dentro de sociedades os agentes podem se agrupar em sub-organizações, ou serem completamente desrelacionados. Uma segunda característica que distingue sociedades é o conjunto de restrições impostas ao comportamento dos agentes, comumente conhecidos como normas sociais, leis ou convenções.

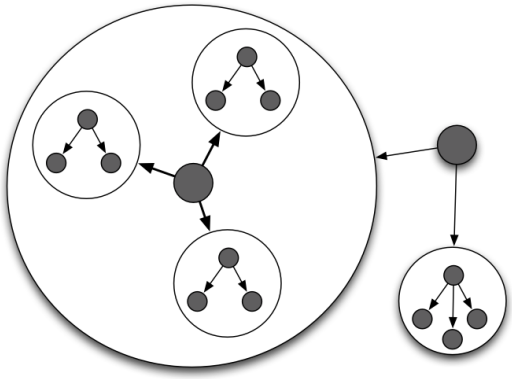
- **Mercados:** Em organizações baseadas em mercados, agentes compradores (mostrados em branco na Figura 3g) podem requisitar ou postar pedidos de compra de recursos, tarefas, serviços ou bens. Agentes também podem produzir itens no mercado para serem vendidos. Vendedores (mostrados em cinza), ou algumas vezes denominados leiloeiros, são responsáveis por processar pedidos e determinar o vencedor. Este arranjo cria um sistema produtor-consumidor que pode modelar economias reais de mercado.

Dentre as organizações apresentadas, no nosso trabalho nos deteremos no estudo de coalizões para compartilhamento de recursos entre agentes. Na próxima Seção discutiremos mais as características das coalizões.

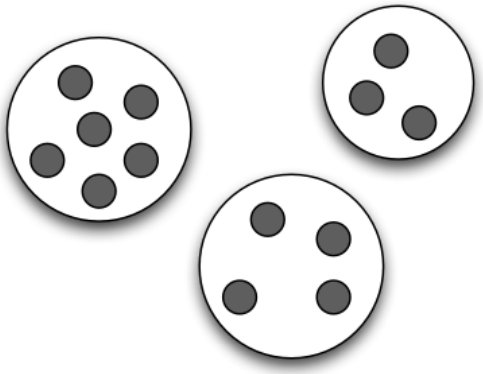
Figura 3 - Estruturas Organizacionais



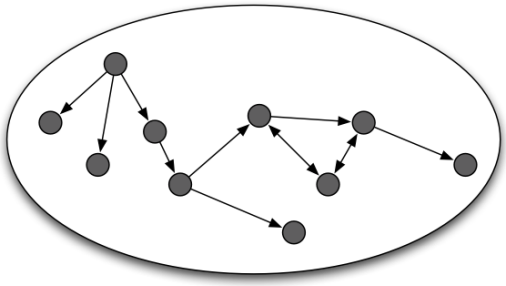
(a) *Hierarquia*



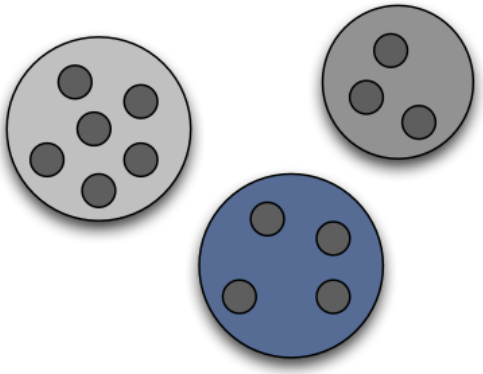
(b) *Holarquia*



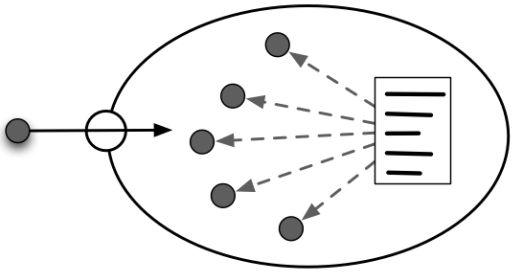
(c) *Coalizões*



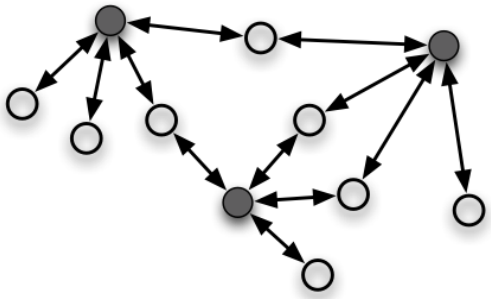
(d) *Time*



(e) *Congregações*



(f) *Sociedade*



(g) *Mercado*

3.1.3 Organização Coalizional

Coalizões são organizações formadas com a intenção explícita da cooperação (Davidsson, 2001), afim de compartilhar recursos ou capacidades segundo um método de divisão, e mantendo uma normatização das interações intra-coalizão. Geralmente coalizões são criadas com um propósito e tem vida curta, e a partir do momento que não são mais necessárias, ou que uma massa crítica de indivíduos a abandonam, são dissolvidas. Coalizões podem ser formadas tanto em populações de agentes altruístas como egoístas. e uma vez formadas podem ser tratadas como uma entidade única (Horling & Lesser, 2004).

A motivação da criação de coalizões é que haja sinergia entre as partes do grupo de forma a gerar valor. Formalmente, espera-se que em alguma dimensão a utilidade da coalizão seja super-aditiva [2], ou seja, se $v(S)$ identifica a utilidade ou valor da coalizão, $v(S) \geq v(S') + v(S'')$, onde $S = S' \cup S''$.

A formação de coalizões é resultado de um mecanismo, que pode ser centralizado ou distribuído. Quando centralizado um *oráculo* decide a melhor configuração coalizional que os agentes devem seguir, e basicamente nos deparamos com um problema de otimização combinatória. Quando o mecanismo de montagem é distribuído, depende da dinâmica de interação, e do aparato decisório individual dos agentes. Fatores como dinamicidade do ambiente, incerteza no retorno das coalizões, participação de um agente em mais de uma coalizão, ou falta de confiabilidade entre os agentes, são obstáculos na formação de coalizões, e devem ser endereçados pelo mecanismo proposto.

A noção de coalizão de indivíduos é extensivamente estudada na *teoria dos jogos*, que é apresentada na Seção 3.3.

É atributo de cada agente a capacidade de seguir seus próprios objetivos, segundo seus próprios meios. Com isso o agente auto-determina também as ações que deve tomar com o fim de alcaçar os seus objetivos. Esse é o processo denominado *planejamento*.

3.2 PLANEJAMENTO DISCRETO EM SISTEMAS MULTIAGENTES

Segundo Russell & Norvig (2009) planejamento é *a tarefa de elaborar uma sequência de ações que alcançarão um objetivo*. Nesse trabalho utilizaremos utilizaremos técnicas de planejamento discreto, uma vez que utilizaremos um mundo com espaço de estados discreto e contável, denotado por s.

[2] Veremos que nem toda função característica de coalizão é super-aditiva. Aqui estamos nos referindo ao fato de que uma determinada instância de coalizão S somente tem sentido em existir se a sua utilidade for maior que a utilidade das partes.

Utilizaremos a formalização definida em LaValle (2006). A idéia é que cada configuração do mundo recebe a denominação de um *estado* e representado por s. O mundo pode ser transformado através da aplicação de ações, denominadas ac, resultando em um novo estado ac'. Assim, $s' = f(s, ac)$. Seja $Ac(s)$ o conjunto de ações a possíveis a partir do estado s, dessa forma temos que $Ac = \bigcup Ac(s)$. O conjunto $S_G \subset S$ denota os estados alvos do planejamento. O finalidade do planejamento é encontrar uma sequência de ações que levem do estado inicial s_I a um estado final pertencente a S_G. O modelo é sumarizado abaixo:

1. Um conjunto de estados $s \in S$ finitos e contáveis.
2. Para cada estado um conjunto de ações $Ac(s)$.
3. Uma função de transição de estado que produza um novo estado a partir do atual, dada uma ação: $s' = f(s, ac)$.
4. Um estado inicial s_I.
5. Um conjunto de estados finais S_G.

Para um mundo em *grid 2D*, definimos os estados com sendo dados pela coordenada (i, j) com $i, j \in \mathbb{Z}$, e as ações representando movimentos nas direções norte, sul, leste, oeste, nordeste, noroeste, sudoeste, sudeste, respectivamente,
$Ac = \{(1,0), (-1,0), (0,1), (0,-1), (1,1), (1,-1), (-1,-1), (-1,1)\}$. Além disso, temos que $Ac(S) = Ac$, ou seja, toda ação pode ser executada a partir de qualquer estado. Temos que a função de transição é dada por $f(x, ac) = x + ac$, em que $s \in S$ e $ac \in Ac$ são tratados como vetores bidimensionais.

O modelo apresentado pode ser representado como um **grafo direto de transição de estados**, em que o conjunto de vértices são os estados do espaço S. Existe uma aresta $s \rightarrow s' \iff \exists\, ac \in Ac(s)\ tq.\ s' = f(s, ac)$. Adotando a representação do mundo como um grafo direto reduz-se a atividade de planejamento a um problema de busca em grafo, com a característica de que o grafo de transição de estados é revelado incrementalmente através da aplicação de ações.

Utilizaremos o algoritmo *A-Star (A^*)*, segundo LaValle (2006), que é uma extensão do algoritmo de busca de Dijkstra (1959). No algoritmo 1 está definido o método genérico de busca em grafos utilizado como base do algoritmo de Dijkstra e A^*. Nesse algoritmo, em qualquer ponto da busca haverá 3 tipos de estados:

1. **Desconhecidos:** Estados que ainda não foram visitados.
2. **Fechados:** Estados que foram visitados e para os quais todo possível próximo estado também foi visitado.

Algorithm 1 Busca a frente

1: *Q.add(s_I)* e marca s_I como visitado
2: **while** Q não vazio **do**
3: $s \leftarrow$ Q.getFirst()
4: **if** $s \in X_G$ **then**
5: **return** SUCESSO
6: **end if**
7: **for all** $ac \in Ac(x)$ **do**
8: $s' \leftarrow f(x, ac)$
9: **if** s' não visitado **then**
10: Marca s' como visitado
11: *Q.add(s')*
12: **else**
13: Resolve s' duplicado
14: **end if**
15: **end for**
16: **end while**
17: **return** FALHA

3. **Abertos:** Estados que foram visitados, mas que tenham próximos estados ainda não visitados.

O conjunto de estados abertos, inicialmente é formado por s_I, e é armazenado na fila de prioridade Q, para a qual uma **função de prioridade** deve ser definida. A diferença entre os algoritmos está na função de prioridade utilizada. Cada estado presente em Q tem seus próximos estados visitados e acrescentados na fila. Quando algum estado final é removido da fila, o algoritmo encontrou um plano de ação que levaria a um estado final. Os próximos estados s' podem ou não ter sido visitados. Caso já tenham sido, não precisam mais ser considerados pois, ou estão fechados, ou se encontram em Q.

O algoritmo de Dijkistra considera uma função custo $C(s)$, $C : S \rightarrow \mathbb{R}$, que associa um valor a cada estado, denominado **custo de chegada ao nó**. Para cada estado s, $C^*(s)$ é chamado o **custo de chegada ótimo** do estado origem s_I ao estado s. Este custo é calculado a partir de um valor associado a cada aresta, $l(s, ac)$, que representa o custo da tomada de uma determinada ação ac no estado s. O custo ótimo $C^*(s)$ é obtido somando-se o custo das arestas sobre todos os caminhos que levam de s_I até s. Se o custo não é ótimo, é escrito como $C(s)$. Em Dijkstra (1959) é provado que $C(s)$ converge para $C^*(s)$ se atualizarmos incrementalmente o valor de cada estado identificado como já visitado na linha 13 do algoritmo 1. Além disso, o que caracteriza o algoritmo de busca de Dijkstra é a modificação da função de prioridade [*Q.getFirst()*] no algoritmo 1 para selecionar primeiramente o estado em Q com menor custo $C(s)$. Dessa forma, duas modificações do *algoritmo de busca a frente* definem o algoritmo de busca de Dijkstra: primeiramente, a atualização dos custos $C(s)$ nos estados

abertos e, em segundo lugar, o ordenamento de Q pela ordem de custo $C(s)$.

O **algoritmo** A^*, por sua vez, é uma extensão do algoritmo de Dijkstra que tenta reduzir o número total de estados explorados, através da incorporação de uma heurística estimada do **custo de chegar ao destino**, escrito como $G(s)$ LaValle (2006). $G^*(s)$, similarmente, indica o **custo ótimo de chegar ao destino** mas, ao contrário de $C^*(s)$, não pode ser calculado incrementalmente. No entanto, para algumas aplicações uma boa estimativa de $G^*(s)$ do custo pode ser construída. No caso específico do planejamento de movimento em 2D, essa estimativa pode ser a distância euclidiana entre a posição (i, j) de s e a posição (i_G, j_G) de s_G. Escreveremos esse valor como $H(S)$. O algoritmo A^* consiste na utilização da função $C^*(s) + H(s)$ para ordenar a fila Q de estados abertos. Se $H(s)$ é uma estimativa inferior do custo ótimo de chega ao destino $G^*(s)$, então é garantido que A^* localize o caminho ótimo até s_G, além de incrementar a eficiência da busca, evitando que o algoritmo de busca visite nós muito distantes do objetivo.

3.3 TEORIA DOS JOGOS COALIZIONAIS

Teoria dos jogos é o instrumento matemático que utilizaremos para a análise das ações que agentes autônomos e auto-interessados tomam quando deparados com outros agentes, em situações em que um pode influenciar o ganho do outro. Segundo Osborne & Rubinstein (1994),

> A teoria dos jogos é um arcabouço de ferramentas desenhadas para nos ajudar a entender o fenômeno observado quando tomadores de decisão interagem. O pressuposto básico que esta subjacente à teoria, é que os tomadores de decisão perseguem objetivos exógenos bem-definidos (são racionais), e levam em consideração o seu conhecimento e expectativas sobre o comportamento dos outros tomadores de decisão. (Osborne & Rubinstein, 1994, pg. 1)

O trabalho de Von Neumann & Morgenstern (1944), *Teoria dos Jogos e Comportamento Econômico*, é considerado um dos fundadores da teoria dos jogos, e desde o seu surgimento a teoria tem sido aplicada extensivamente em áreas como economia, biologia, sociologia e inteligência artificial, para entender as relações entre indivíduos, ou coletividades, autônomas.

Os modelos de teoria dos jogos classicamente são divididos em duas categorias. Jogos *Não-Cooperativos*, quando as ações dos indivíduos são primitivas, ou os Jogos *Cooperativos*, quando as ações do coletivo de indivíduos é primitiva Osborne & Rubinstein (1994). Ainda, é possível pensar que jogos não-cooperativos tratam das ações antes de instituída qualquer forma de cooperação, enquanto os jogos cooperativos tratam da racionalidade após ser desenhada a cooperação.

Jogos cooperativos ou coalizionais se preocupam com as ações coletivas que cada grupo de jogadores pode tomar. O resultado do jogo coalizional é o conjunto de coalizões formado e o ganho advindo da formação das coalizões.

Diferentemente dos jogos não-cooperativos, os cooperativos se preocupam sobre o que cada grupo de jogadores pode obter. O ganho coletivo é o objeto de estudo, e os detalhes do funcionamento interno das coalizões não é considerado. Os jogos não-cooperativos, por sua vez, desenvolvem a racionalidade do que os jogadores individuais podem ou devem fazer.

Definição 1: Jogo coalizional com utilidade transferível. Um jogo com utilidade transferível associa um número $v(S)$ à coalizão S, interpretada como *benefício* da coalizão, e divide esse número, sem restrições, entre os membros da coalizão.

Formalmente, um jogo coalizional com utilidade transferível é uma tupla $< N, v >$, onde N é o conjunto finito de jogadores, e v é uma função que associa a cada $S \subset N$ um valor $v(S) \in \mathbb{R}$.

A função $v(S)$ é chamada de *função característica da coalizão* e, usualmente, representa o valor máximo que a coalizão ganha e que pode ser distribuído entre os seus membros. A divisão do valor entre os membros da coalizão não é preocupação da teoria dos jogos cooperativos, nem o é a origem da função característica, assumida como dada previamente à formação da coalizão. No entanto, famílias de funções características são analisadas e propriedades específicas demonstradas. Uma família importante das funções características define os *jogos super-aditivos*.

Definição 2: Jogos super-aditivos. Um jogo $< N, v >$ é super-aditivo se para qualquer S,$T \subset N$, se $S \cap T = \emptyset$ então $v(S \cup T) \geq v(S) + v(T)$.

Resultados importantes são demonstrados quando a função característica é aditiva, como a condição para o núcleo ser não vazio (veremos adiante o conceito de núcleo). Além disso, quando o jogo é aditivo é racional esperar que a melhor coalizão será a *grande coalizão* [3]. Parte significativa da comunidade de teoria dos jogos se preocupa com a solução do jogo coalizional considerando que a *grande coalizão* será formada, o que não será válido, no entanto, quando a função não for aditiva.

Da mesma forma que nos jogos não-cooperativos havia o conceito de *equilíbrio de Nash*, que dava uma noção de solução do jogo, no jogo coalizional temos o conceito de *estabilidade*. Uma coalizão é estável quando os seus membros não conseguiriam obter um ganho melhor em nenhuma outra coalizão. A noção de *núcleo de um jogo coalizional* é a que captura o conceito de estabilidade.

[3] Coalizão formada por todos os agentes presentes no jogo.

Definição 3: Núcleo de um jogo coalizional. Seja o jogo de utilidades transferíveis $< N, v >$. Chamaremos de perfil de *payoffs* factível na coalizão S qualquer vetor $(x_i)_{i \in S}$ de números reais, onde $x(S) = \sum_{i \in S} x_i$, e $x(S) = v(S)$. Chamaremos simplesmente de perfil de *payoffs* factível $(x_i)_{i \in N}$ quando S é a grande coalizão N.

Núcleo é o perfil de *payoffs* factível $(x_i)_{i \in N}$, tal que não há uma coalizão S com perfil de *payoffs* factíveis $(y_i)_{i \in S}$ para o qual $y_i > x_i, \ \forall i \in S$.

Um perfil de *payoffs* é uma distribuição da dotação total da coalizão $(v(S))$ entre os seus membros. Em um jogo coalizional de utilidade transferível, a distribuição dos recursos entre os membros da coalizão não é definida pelo jogo, ou seja, não existe um esquema de distribuição de recursos *à priori* que a coalizão deva executar. Qualquer distribuição de recursos $(x_1, \ldots, x_N)$ é factível se distribui apenas os recursos da coalizão. O núcleo, por sua vez, é um perfil de *payoffs* de forma que nenhum subgrupo de agentes da coalizão tem estímulo para formar outra coalizão e todos os membros receberem mais, ou seja, $\nexists S \subset N$ tal que $y_i > x_i \, \forall i \in S$, sendo y um perfil de *payoffs* em S. O núcleo captura a noção de estabilidade uma vez que nenhum agente pode ficar "mais satisfeito" do que na coalizão estável.

No entanto, o conceito de núcleo e estabilidade diz muito pouco sobre como deve acontecer uma divisão justa dos recursos dentro da coalizão. Para o nosso trabalho tal indagação é fundamental, uma vez que buscamos uma lógica que satisfaça os objetivos individuais dos agentes. A teoria dos jogos coalizionais apresenta uma resposta para essa indagação. É o *valor de Shapley*, que mensura o quanto um jogador agregou a uma determinada coalizão, e estabelece o valor justo que cada jogador i deveria receber do *payoff* $v(S)$ da coalizão S.

Definição 4: Jogos de Votação. Um jogo de votação por pesos é aquele formado por um conjunto de pesos w_i associado a cada agente, e uma cota q, característica do jogo. O jogo é considerado vencedor quando $v(S) = 1$ para $\sum_{i \in S} w_i > q$.

Um jogo de votação, com jogadores $1, \ldots, n$, pesos $w_1, \ldots, w_n$, e quota q é escrito $(q; w_1, \ldots, w_n)$. Jogos de votação acontecem frequentemente na nossa vida cotidiana. O exemplo mais obvio é a votação majoritária utilizada para escolha dos representantes legislativos [4]. O jogo de votação nos diz que uma coalizão vencedora tem membros cujos pesos ultrapassam um limiar ou quota. Veremos mais adiante que os jogos de votação são convenientes para calcular o valor de Shapley, um dos principais parâmetros para formação de coalizões no nosso modelo. Assim, modelaremos o compartilhamento de recursos como um jogo de votação.

[4] A votação majoritária é o jogo $(\frac{N+1}{2}; w_1 = 1, \ldots, w_n = 1)$

3.3.1 Valor de Shapley

Uma questão importante relativa aos jogos coalizionais é como dividir o valor resultante da coalizão entre os seus membros. Uma resposta à essa questão é o *Valor de Shapley* (Shapley, 1953). Esse valor é uma medida da contribuição que cada agente fornece à coalizão e, portanto, um parâmetro *justo* para executar-se a divisão dos ganhos advindos da coalizão. Mais que isso, o *valor de Shapley* apresenta 3 propriedades coerentes com o conceito de divisão *justa* (*simetria, jogador irrelevante e aditividade*), e *Shapley* demonstrou que qualquer número que apresente essas 3 propriedades será uma medida do *valor de Shapley*. No desenvolvimento dessa Seção utilizaremos a formulação apresentado em Shoham & Leyton-Brown (2009).

O valor de Shapley de um agente é denotado por φ_i.

- **Simetria:** Agentes que contribuam com o mesmo valor à coalizão devem ter o mesmo *valor de Shapley*. Ou seja, se $v(S \cup \{i\}) = v(S \cup \{j\})$ então $\varphi_i = \varphi_j$.

- **Jogador:** indiferente Se um jogador contribui com todas as coalizões apenas o que receberia sozinho, o seu *valor de Shapley* deve ser igual ao que receberia sozinho. Assim: $\forall S'$, $S' \subset S$ & $i \notin S$ tal que $v(S' \cup \{i\}) = v(\{i\})$ então $\varphi_i = v(\{i\})$.

- **Aditividade:** Dados dois jogos coalizionais definidos sobre o mesmo conjunto de agentes, com funções características v_1 e v_2, caso definamos um terceiro jogo com função característica $v = v_1 + v_2$, teremos que o valor de Shapley correspondente será $\varphi_1 + \varphi_2$.

Definição 5: Valor de *Shapley*. O valor de Shapley calcula a contribuição marginal de um agente considerando todas as ordens em que ele pode ser incluído em uma coalizão.

$$\varphi_i = \frac{1}{N!} \sum_{S \subseteq N/\{i\}} |S|!(|N| - |S| - 1)![v(S \cup \{i\}) - v(S)] \tag{1}$$

Estamos considerando N como a grande coalizão, e S como todas as subcoalizões sem considerar o agente i. Ou seja, S identifica todas as coalizões, inclusive a vazia em que i pode ser inserido.

O valor de Shapley pode ser entendido como a média da contribuição marginal de um agente sobre todas as possíveis formas de formar uma coalizão. A contribuição marginal é o valor que um agente agrega à uma coalizão, e é formalizado por $\mu_i = v(S \cup \{i\}) - v(S)$. Na equação (1) identificamos N como a grande coalizão, ou a coalizão sendo correntemente alterada [5]. $S \in 2^N$ identifica uma das subcoalizões formadas a partir de N excluindo-se o

[5] No nosso modelo N pode identificar uma coalizão qualquer, sendo avaliada por um agente.

$N : \{1,2\}$
$Ord(N) : \{(1,2),(2,1)\}$
$v(\{1\}) = 5, v(\{2\}) = 5, v(\{1,2\}) = 20$
$\mu_1(\emptyset) = 5,\ \mu_1(\{2\}) = v(\{1,2\}) - v(\{2\}) = 15 \Rightarrow \boldsymbol{\varphi_i} = (5+15)/2 = 10$
$N : \{1,2,3\}$
$Ord(N) : \{(1,2,3),(1,3,2),(2,1,3),(2,3,1),(3,1,2),(3,2,1)\}$
$v(\{1\}) = 5,\ v(\{2\}) = 5,\ v(\{3\}) = 5,\ v(\{1,2\}) = 10,\ v(\{1,3\}) = 10,\ v(\{2,3\}) = 20,\ v(\{1,2,3\}) = 25$
$\mu_1(\emptyset) = 5,\ \mu_1(\{2\}) = v(\{1,2\}) - v(\{2\}) = 5,\ \mu_1(\{3\}) = 5,\ \mu_1(\{2,3\}) = 5$
$\boldsymbol{\varphi_1 = 5}$
$\mu_2(\emptyset) = 5,\ \mu_2(\{1\}) = 5,\ \mu_2(\{3\}) = 15,\ \mu_2(\{1,3\}) = 15$
$\boldsymbol{\varphi_2 = 10},\ \boldsymbol{\varphi_3 = 10}$

Tabela 1 - Exemplo de cálculo do valor de Shapley

agente i. O valor é calculado considerando-se a contribuição marginal de i sobre todas as ordens em que ele poderia ser inserido. Dessa forma, existem $|S|!$ permutações antes de i ser inserido, e $|N - S - 1|!$ permutações após i ser inserido. Finalmente, a somatória de contribuições marginais é ponderada pelo total de ordenamentos possíveis $|N|!$. Na tabela 1 temos o exemplo do cálculo do valor de Shapley para dois jogos.

Exemplo do Cálculo do Valor de Shapley

Nessa tabela temos representados dois exemplos. Em cada um está definido o conjunto de agentes N e a função característica das coalizões $v(S)$ para $S \subseteq N$, ou seja, temos $v(S)$ definido para todo o espaço de coalizões do conjunto N. Os ordenamentos de N, $Ord[N]$, representam todas as permutações dos agentes, com a ordem em que cada agente foi inserido na coalizão. Para cada ordenamento um agente i está em uma posição, e o valor de Shapley é calculado a partir da contribuição marginal μ_i do agente, ou seja, a diferença do valor da coalizão antes e depois do agente ser inserido ($\mu_i(S) = v(S \cup \{i\}) - v(S)$). No primeiro exemplo, formamos as coalizões de 2 agentes, e calculamos a contribuição marginal de cada agente dada uma coalizão pré-existente. Para ilustrar, tomemos o agente 1: o valor do ag. 1 sozinho é $v(\{1\}) = 5$, por outro lado, o valor da coalizão {1,2} é $v(\{1,2\}) = 20$. Dessa forma, a contribuição marginal do ag. 1 para a coalizão vazia é $\mu_1(\emptyset) = 5$, pois por definição $v(\emptyset) = 0$. Para a coalizão {1,2}, a contribuição do ag. 1 é dado por $\mu_1(\{2\}) = v(\{1,2\}) - v(\{2\}) = 20 - 5 = 15$. O valor de Shapley do ag. 1 é calculado pela ponderação das contribuições

marginais sobre o total de coalizões. Dessa forma, $\varphi_1 = \frac{1 \times \mu_1(\emptyset) + 1 \times \mu_1(\{2\})}{2} = 10$. Agora tomemos o segundo exemplo, formado por 3 agentes {1,2,3}. A quantidade de permutações é $3! = 6$. Para calcular o valor de Shapley, precisamos avaliar a contribuição marginal do agente em coalizões com zero, um e dois membros. Como um agente apareçe como sendo o primeiro membro em dois ordenamentos, sendo o segundo ou terceiro em mais dois ordenamentos, temos o peso dois para cada contribuição marginal de um agente. Para ilustrar, tomemos o agente 2: $\mu_2(\emptyset) = v(\{2\}) = 5$. $\mu_2(\{1\}) = v(\{1,2\}) - v(\{1\}) = 10 - 5 = 5$. A colaboração do ag. 2 na coalizão {2,3}, por outro lado, é diferente da anterior, e é dada por $\mu_2(\{3\}) = v(\{2,3\}) - v(\{3\}) = 20 - 5 = 15$. E finalmente $\mu_2(\{1,3\}) = 25 - 10 = 15$. Dadas as contribuições marginais, podemos calcular o valor de Shapley do agente 2, $\varphi_2 = \Big[2 \times \mu_2(\emptyset) + \mu_2(\{1\} + \mu_2(\{3\} + 2 \times \mu_2(\{1,3\})\Big]/6$.

A complexidade do cálculo de φ_i está em calcular todos os ordenamentos da coalizão S para ponderar o valor de $v(S)$. A quantidade de ordenamentos é da ordem de $O(2^{|S|})$, portanto, *exponencial*.

Índice de Shapley-Shubik

O valor de Shapley tem um sentido interessante quando aplicado aos jogos de votação por pesos. Nesses jogos o valor de Shapley, chamado de índice de Shapley-Shubik, indica o poder de um agente de alterar o resultado da votação. Além disso, o índice de Shapley-Shubik tem uma fórmula simplificada para o seu cálculo.

Definição 6: Índice de Shapley-Shubik. O índice de Shapley-Shubik indica o valor de Shapley para um jogo de votação por pesos. Para o seu cálculo iniciaremos com a definição de um *pivot*, que é o agente que dado uma coalizão (ordenada) transforma a coalizão de perdedora para vencedora, ou seja, é o agente que quando inserido à coalizão a torna vencedora. Utilizaremos o predicado $pivot_i(S) : \{0,1\}$ como o indicador se um jogador i é pivot na coalizão S.

$$\varphi_i \leftarrow \frac{\sum_{S \in Ord[N]} pivot_i(S)}{N!}$$

Assim, o cálculo do índice de Shapley-Shubik é feito somando-se a quantidade de vezes que um agente é pivot sobre a quantidade de ordenamentos da coalizão.

3.3.2 Aproximações ao Valor de Shapley

Dada a complexidade de calcular a contribuição marginal sobre cada coalizão diversos métodos de aproximações foram desenvolvidos para o cálculo do valor de Shapley(Fatima et al. (2011), Fatima et al. (2008), Jain & Vazirani (2001), Bachrach et al. (2010)).

No nosso trabalho utilizaremos o método de aproximação descrito em Fatima et al. (2011) e Fatima et al. (2008), que consiste em aproximar φ_i, denotado por $\bar{\varphi}_i$, pela contribuição marginal do agente i em uma coalizão de tamanho X, dadas as médias (μ) e variâncias (ν) dos agentes sobre os seus pesos, através da distribuição $\mathcal{N}(\mu, \nu/X)$. A aproximação da contribuição marginal do agente i na coalizão X é denotada por $E\Delta_i^X$.

Os autores utilizam um método de aproximação baseado em amostragens construídas segundo a regra definida a seguir.

Definição 7: Regra de Amostragem ($\mathcal{R}_1$). Se $w_1, \ldots, w_X$ é uma amostra randômica de tamanho X retirada de uma distribuição com média μ e variância ν, então a média da amostra tem uma distribuição normal aproximada $\mathcal{N}$ com média μ e variância ν/X, e quanto maior X melhor a aproximação.

Uma curva normal da distribuição aproximada dos pesos é mostrada na Figura 4. Nela aparecem dois pontos, **a** e **b** que representam o intervalo de pesos para que um agente i seja pivot em uma amostra de tamanho X com peso w_i. Na curva esse intervalo é igual a probabilidade de que i contribua com $w_i - q$, e é igual ao valor de Shapley para o jogador. O intervalo é igual aos pontos $(\frac{w_i - q}{X}, \frac{q-\epsilon}{X})$ na curva de distribuição normal aproximada dos pesos.

Dessa forma,

$$E\Delta_i^X = \frac{1}{\sqrt{2\pi\nu/X}} \int_a^b e^{-X\frac{(x-\mu)^2}{2\nu}} \, \mathrm{d}x \qquad (2)$$

$$\text{onde } a = (q - w_i)/X,\ b = (q - \epsilon)/X$$

A aproximação do valor de Shapley será a média sobre a *contribuição média aproximada* de coalizões de todos os tamanhos ($1 \leq X \leq n$).

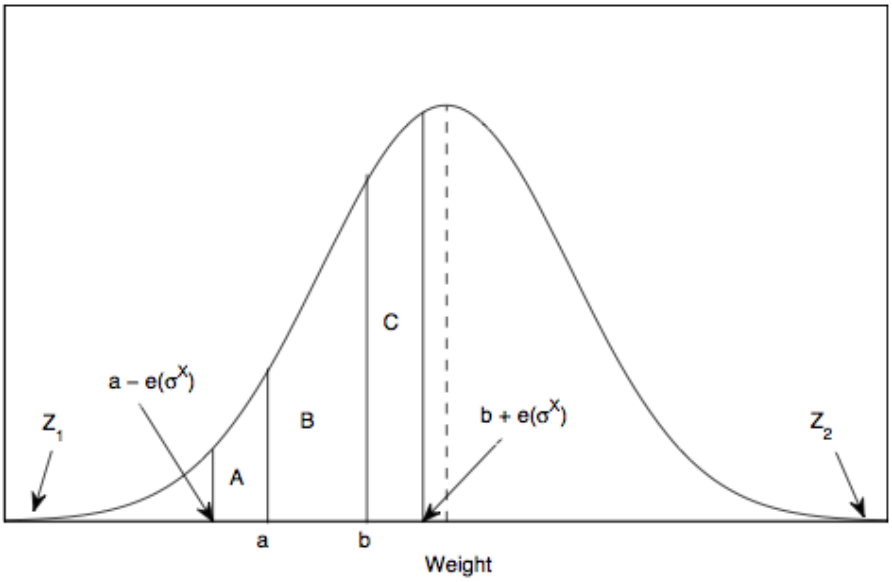


Figura 4 - Curva Normal do Valor Aproximado de Shapley. Extraído de (Fatima et al., 2008, pg. 15).

$$\bar{\varphi}_i = \frac{1}{n}\sum_{X=1}^{n} E\Delta_i^{X-1} \tag{3}$$

Para o **jogo de votação k-ponderado**, a aproximação é feita segundo os mesmos princípios daquela do jogo de votação simples, apresentado através das equações (2) e (3).

O *jogo de votação k-ponderado* $(v_1 \wedge \cdots \wedge v_g), 1 \leq g \leq k$, denotado como $v_g = [q^g; w_1^g, \ldots, w_S^g]$, tem o valor de Shapley calculado pela aproximação da contribuição marginal, denotada por $kE\Delta_i^X$. O valor da contribuição marginal do agente i para a coalizão S_X, é calculado pela equação:

$$\bar{\varphi}_i{}^k = \frac{1}{n}\sum_{X=1}^{n} kE\Delta_i^{X-1} \tag{4}$$

O valor da contribuição marginal ($kE\Delta_i^X$) é calculado considerado-se que o jogo é vencedor para a coalizão S_X se:

1. Existe pelo menos um jogo v_g para o qual i é pivot. Essa condição é medida pela probabilidade $PL_I^g(S_X)$. Assim,

   $PL_i^g(S_X)$: Probabilidade da coalizão S_X estar perdendo e a coalizão $S \cup \{i\}$ estar ganhando, ou seja, probabilidade de i ser pivot.

2. Para todo v_g, $v_g(S_X \cup \{i\}) = 1$. Essa condição é medida pela probabilidade $PW_i^g(S_X)$.

   $PW_i^g(S_X)$: Probabilidade de $S_X \cup \{i\}$ estar ganhando, seja com i pivot ou não.

Dados os conceitos de $PL_i^g(S_X)$ e $PW_i^g(S_X)$, calculamos a contribuição marginal considerando todos os possíveis modos de i ser pivot para o jogo de votação k-ponderado. Esse valor é dados para cada $0 \leq j \leq k-1$ como a probabilidade de i não ser pivot para os jogos 1 até j **e** i ser pivot para o jogo $j+1$ **e** os jogos $j+2$ até k estarem vencendo com i pivot ou não, que é igual ao termo abaixo:

$$\Pi_{g=1}^{j}(1 - PL_i^g(S_X)) \times PW_i^{j+1}(S_X) \times \Pi_{f=j+2}^{k} PW_i^f(S_X) \tag{5}$$

Considerando-se o termo (5) sobre todos os k jogos, temos:

$$kE\Delta_i^X = \sum_{j=0}^{k-1} \left(\Pi_{g=1}^{j}(1 - PL_i^g(S_X)) \times PW_i^{j+1}(S_X) \times \Pi_{f=j+2}^{k} PW_i^f(S_X) \right) \tag{6}$$

Onde $PL_i^g(S_X)$ e $PW_i^g(S_X)$ são calculados segundo a *regra de amostragem* pelas integrais definidas abaixo.

$$PL_i^g(S_X) = \frac{1}{\sqrt{2\pi\nu^g/X}} \int_{q^g - w_i^g}^{q^g - \epsilon} e^{-X \frac{(x-\mu^g)^2}{2\nu^g}} \, \mathrm{d}x \tag{7}$$

$$PW_i^g(S_X) = \frac{1}{\sqrt{2\pi\nu^g/X}} \int_{q^g - w_i^g}^{\infty} e^{-X \frac{(x-\mu^g)^2}{2\nu^g}} \, \mathrm{d}x \tag{8}$$

3.4 COMPARTILHAMENTO DE RECURSOS

3.4.1 MARA - Multiagent Resource Allocation

> Muitos problemas estudados pela comunidade de sistemas multiagentes podem ser considerados como instâncias abstratas do problema de alocação de recursos em sistemas multiagentes (Chevaleyre et al., 2005).

Alocação de recursos em sistemas multiagentes [6] é o processo de distribuir uma quantidade de recursos entre um grupo de agentes. Uma abordagem analítica-conceitual do processo de alocação de recursos é efetuada na área de teoria dos jogos, que foi discutida na Seção 3.3. Segundo Chevaleyre et al. (2006), a pesquisa da alocação de recursos abordada em conjunto com SMAs privilegia novas questões, graças ao poder de abstração que o paradigma de multiagentes proporciona. Em particular, *que tipos de itens estão sendo distribuídos? Como e por que eles estão sendo distribuídos? Como as preferências dos agentes são representadas, e como a eficiência de uma alocação é medida?* Para responder a essas questões são aplicados técnicas de teoria dos jogos juntamente com algoritmos computacionais.

Vale citar que existem aplicações da abordagem de alocação de recursos com SMAs em diversas áreas, como gestão de produção em fábricas, gestão de crises, transporte, *patrolling*, gestão de tráfego aéreo, comércio eletrônico e computação em grid. Detalharemos mais alguns conceitos relacionados que serão utilizados no restante do trabalho.

- **Recursos** são os itens sendo distribuídos. Representam um elemento do ambiente que pode ser transferido entre os agentes, e que de alguma forma influencia na realização dos objetivos do agente.

Podemos identificar diversos tipos de recursos: divisíveis ou indivisíveis, compartilháveis ou não, contínuos ou enumeráveis. Abordaremos o problema de alocação de tarefas como sendo similar ao de alocação de recursos. Uma diferença entre tarefas e recursos é que tarefas podem estar associadas à restrições sobre sua combinação coerente, no sentido de que tarefas tem como pré-requisitos outras tarefas, e tarefas podem ser equivalentes ou incompatíveis umas as outras.

---

[6] *MARA - Multiagent resource Allocation* é a área que busca unir sistemas multiagentes com economia, mais especificamente teoria dos jogos, com o objetivo de modelar sistemas em que os agentes participam ativamente na busca de alocações de acordo com suas preferências.

- **Alocação**: é uma distribuição particular dos recursos entre os agentes. No caso de recursos não-compartilháveis e indivisíveis é uma partição do conjunto de recursos.

- **Preferências**: agentes podem ter preferências sobre um grupo de recursos, ou mesmo ter preferências sobre recursos recebidos por outros agentes (chamados de externalidades). Os agentes podem ou não reportar verdadeiramente suas preferências. [7]

  Segundo Chevaleyre et al. (2006), "*preferências expressam a satisfação relativa ou absoluta de um indivíduo quando se depara com uma escolha entre diferentes alternativas*". Preferências podem ser estruturadas de diferentes formas:

  1. Cardinal: forneçe um valor para cada conjunto de preferências, ou seja, constroi uma função $v : 2^{Rc} \rightarrow \mathbb{R}$. Pode ser difícil valorizar conjuntos de valores, além de que a quantidade de conjuntos é exponencial.
  2. Ordinal: define uma ordem entre os recursos segundo a preferëncia do agente.
  3. Binário: indica se o agente aceita o recursos ou não.

- **Procedimento de alocação**: os procedimentos de alocação podem ser distribuídos ou centralizados. Um exemplo de procedimento centralizado é o de leilão e, por sua vez, de distribuído é o procedimento de negociação, ambos discutidos nas próximas seções.

- **Bem-estar social**: Quando o SMA é visto como uma sociedade de agentes, o objetivo pode ser encontrar um configuração que seja benéfica para a média dos agentes, ou que incremente uma característica global.

  Caso os agentes utilizem uma representação cardinal de preferência (função v), o conceito de bem-estar social também pode ser representado por uma função, denominada função utilidade coletiva, $sw : (u_1(\rho), \ldots, u_n(\rho)) \rightarrow \mathbb{R}$. A seguir alguns tipos de funções utilidade coletivas.

  - Função de bem-estar social utilitária: $sw_u(\rho) = \sum u_i(\rho)$, onde ρ é uma alocação de recursos.
  - Função de bem-estar social egualitária: $sw_e(\rho) = \min u_i(\rho)$.
  - Função de bem-estar social elitista: $sw_{el}(\rho) = \max u_i(\rho)$.

Na próxima Seção descreveremos o ambiente de teste que motivou o desenvolvimento do simulador.

[7] A pesquisa de incentivos e restrições para que os agentes manifestem verdadeiramente suas preferências é denominada *mechanism design*.

3.4.2 CT-Game

O CT-Game (Gal et al., 2010, 2005), é um ambiente para investigação de situações de tomada de decisão que ocorrem em domínios orientados à tarefas, onde as interações chaves são entre os objetivos (individuais ou de grupos), tarefas e recursos necessários à realização destes objetivos. O jogo tem analogia direta com situações de execução de tarefas, permitindo a transposição dos resultados obtidos para cenários reais. As regras do CT-Game são simples, o que o torna adequado à abstração de domínios particulares, mantendo o foco nas questões relativas à tomada de decisão, sem a especificação detalhada das características do domínio.

No jogo o mundo é representado por um *grid* onde cada agente se localiza inicialmente em uma célula. Todo agente tem uma ou mais células objetivos, as quais pode alcançar se movendo, desde que tenha os recursos para isso. As células do *grid* tem cores representando o recurso necessário para ocupá-la. Os agentes são inicializados com uma dotação de recursos, os quais serão consumidos para efetuar-se a movimentação no *grid* [8].

Diferentes cenários podem ser modelados através da configuração das características do jogo. O *payoff* de um agente é proporcional à quão próximo se encontra da realização dos seus objetivos ou pode levar em conta os objetivos coletivos. O grau de dependência do *payoff* de um agente com o dos outros distingue o trabalho em times. Por sua vez, variando-se a quantidade de células percebíveis do *grid* pode-se representar situações de informações incompletas.

O CT-Game original não prevê a realização de coalizões[9]. Faremos a adaptação do jogo à estratégia de compartilhamento de recursos, que será descrita no próximo Capítulo.

3.5 SUMARIZAÇÃO

Nesse Capítulo apresentamos a visão de agente como entidade autônoma capaz de seguir seus próprios objetivos e auto-determinar as ações necessárias para atingí-los. Agentes também

[8]A correspondência com domínios orientados à tarefas se dá através da analogia de movimento no *grid*, com a execução de tarefas complexas. Cada tarefa necessita de uma habilidade específica, no jogo representado por uma peça de uma cor. Os jogadores possuem habilidades restritas, e portanto dependem da colaboração de outros.

[9]No jogo original os agentes fazem proposições de trocas diretamente a outros agentes. Um agente pode aceitar uma troca, ou mesmo retroceder dela após a sua aceitação. Questões como comprometimento com as propostas e confiabilidade do agente podem ser configuradas e medidas. No nosso modelo, por outro lado, as interações se dão intra-coalizão, estando normatizadas previamente e o sistema garante o cumprimento destas normas.

apresentam características de sociabilidade, que aqui se resumem à capacidade de montar organizações coalizionais com o fim de compartilhamento de recursos.

O processo decisório do agente está dividio em ações físicas, definidas através da montagem de planos de ações, e ações sociais, restritas as decisões necessárias para a montagem de coalizões.

Recursos são representações genéricas de capacidades ou itens do ambiente que podem ser transferidos entre os agentes. Apresentamos a área de alocação de recursos em sistemas multiagentes (MARA) que desenvolve as noções de utilidade, bem-estar e preferência para modelar o processo de alocação de recursos distribuído.

4 UMA HEURÍSTICA GANHA-GANHA PARA FORMAÇÃO DE COALIZÕES

Nessa Seção descreveremos um procedimento de formação de coalizões baseado nos conceitos da *teoria dos jogos coalizionais* apresentados na Seção 3.3. Iniciaremos com a apresentação de um modelo de agente social que utiliza a noção de recursos para a montagem de coalizões. Definido o modelo de agente, passaremos a apresentação do algoritmo para formação de coalizões e da heurística utilizada, com a demonstração teórica da monotonicidade do algoritmo. Finalizaremos com a medição empírica dos resultados do algoritmo em um conjunto de cenários de prova.

4.1 MODELO DE AGENTES BASEADO EM RECURSOS

Consideramos agentes como entidades tomadoras de decisões que são capazes de seguir uma agenda própria. A nossa noção de tomada de decisão está restrita a capacidade de montagem de planos para atingir os objetivos internos do agente. No modelo assumimos que os agentes são capazes de montar os próprios planos, no entanto, os detalhes dessa faculdade são externos ao modelo. Além da faculdade de planejamento, os agentes devem ser capazes de traduzir os seus planos em um conjunto de recursos necessários para a sua realização. O conceito de recurso foi tema da Seção 3.4.1. A seguir definimos formalmente como esse conceito é utilizado no nosso modelo.

4.1.1 Recursos como *proxies* dos agentes

Como citado Chevaleyre et al. (2006), recursos são itens sendo distribuídos e que representam um elemento do ambiente ou unidade de trabalho, e que, de alguma forma, influenciam na obtenção dos objetivos dos agentes. Consideraremos recursos como unidades indivisíveis e compartilháveis, sendo representados por um vetor, denominado *vetor de recursos*. Os recursos são indivisíveis pois uma quantidade unitária do recurso não pode ser subdividida. Por sua vez, os recursos são compartilháveis pois uma determinada quantidade de recursos pode ser livremente transferida entre os agentes.

Formalmente, os tipos de recursos disponíveis no ambiente são representados por um con-

junto $\mathcal{K} = \{1, 2, \dots\}$, onde cada elemento $k \in \mathcal{K}$ representa um tipo de recurso, cuja quantidade é dada por um vetor rc.

Exemplificando, se o ambiente contém 3 tipos de recursos, vamos dizer recursos $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}$, teremos $\mathcal{K} = \{\mathfrak{a}, \mathfrak{b}, \mathfrak{c}\}$, sendo que os agentes conterão uma quantidade positiva de cada recurso. Dessa forma, digamos que o agente 1 contenha quantidade $10, 9, 8$ de $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}$, respectivamente. Representaremos essa distribuição por um vetor rc_1, o índice identifica o agente, igual a $rc_1 = (10, 9, 8)$. A quantidade de um determinado recurso é acessada por $rc_1(k), k \in \mathcal{K}$, assim, para acessar a quantidade do recurso $\mathfrak{a}$ utilizaremos $rc_1(\mathfrak{a})$ [10].

Definição 8: Vetor de Recursos dos Agentes. Um vetor de recursos é uma n-upla $(rc_i(k_1), rc_i(k_2), \dots, rc_i(k_{|\mathcal{K}|})) \in \mathbb{N}^{|\mathcal{K}|}$, com $k_j \in \mathcal{K}$ e $rc_i(k_j) \in \mathbb{N}$ é a quantidade de recursos k_j disponiveis para o agente i.

Como colocado anteriormente, recursos são a principal representação das necessidades e disponibilidades dos agentes. Eles tem o papel de *representar os agentes*, no sentido de que são, primeiramente, traduções dos estados internos e, além disso, medida da relevância do agente no processo de formação de coalizões. Adiantando, um *jogo de votação K-ponderado* é executado para calcular o valor de Shapley de cada agente na coalizão. O peso de cada agente é proporcional à razão da quantidade de recursos disponíveis sobre requeridos, e a alocação é executada baseada no valor de Shapley. O vetor de recursos da coalizão é definida de maneira similar.

Definição 9: Vetor de Recursos da Coalizão. O vetor de recursos da coalizão é a soma de recursos de cada membro da coalizão. Dessa forma, dada a coalizão S $\mathbf{rc_S} = \sum_{i \in S} rc_i$.

O agente apresenta uma função de tradução de planos em recursos

Cada agente é capaz de montar um plano de ação para atingir seus objetivos. Outra habilidade inata dos agentes é sua capacidade de transformar os planos em recursos necessários para sua execução. O conceito de *vetor de recursos do plano* captura essa noção. **Dessa forma, entendemos que cada agente dispõe de uma função de mapeamento** $plan_i : \Pi \rightarrow \mathbb{N}^{\mathcal{K}}$**, tradutora de plano em necessidades de recursos**. Entendida de outra forma,

[10] A maneira mais usual de acessar os valores do vetor de recursos rc_1 seria utilizar as coordenadas do vetor, fazendo uma referência ao conjunto $\mathcal{K}$ ordenado. Assim $rc_1(\mathfrak{a})$ seria $rc_1(1)$, $rc_1(\mathfrak{b})$ seria $rc_1(2)$, e assim por diante. Adotamos a referência direta ao membro de $\mathcal{K}$ para economizar um índice, tornando a notação mais enxuta, sem prejuízo do modelo.

essa função é a forma do agente representar suas preferências sobre o conjunto de recursos. Nessa Seção não nos deteremos na função $plan$ de planejamento.

Assumiremos como dado o plano de ações, bem como a tradução em recursos. No Capítulo 5 implementaremos um algoritmo para planejamento e tradução sobre o mundo em grade.

Definição 10: Vetor de recursos do plano. O vetor de recursos do plano, rc^{Π} é a quantidade de recursos necessários para realização do plano de ação Π.

Definição 11: Vetor Residual de Recursos. O vetor residual de recursos é a quantidade de recursos não utilizada pelo agente na realização dos seus planos. Denominamos o vetor de recursos residuais como rc_i^+.

A seguir apresentaremos o *jogo de votação k-ponderado* a partir do qual o valor de Shapley é calculado.

Definição 12: Jogo de Votação k-ponderado para Alocação de Recursos. É o jogo $(q_S, w1, \ldots, w_S)$ que define o valor de Shapley para cada agente $i \in S$.

$$\text{Função característica } v(S) = \begin{cases} 1 & \text{se } \sum_{i \in S} w_i >= q_S \\ 0 & \text{caso contrario} \end{cases}$$

$$\text{Onde } q_S = \min\Big\{ \sum_{\substack{k \in \mathcal{K} \\ i \in S}} rc_i^{\Pi}(k), \sum_{\substack{k \in \mathcal{K} \\ i \in S}} rc_i(k) \Big\},$$

$$\text{and } w_i = \sum_{k \in \mathcal{K}} rc_i(k)$$

Definição 13:. O valor de Shapley é definido como o índice de poder de Shubik-Shapley para o jogo definido anteriormente (Definição 12).

Até agora nos detivemos na definição de *vetores de recursos*, elemento que tem a capacidade de representar tanto demandas dos agentes (rc_i^{Π}), como suas dotações próprias (rc_i). Por representarem as capacidades ou possibilidades dos agentes, bem como seus objetivos ou necessidades, dizemos que os recursos são *representantes* dos agentes. Mais que isso, através dos do jogo de votação *k-ponderado* (def. 12), que é baseado em na quantidade de recursos residuais dos agentes, podemos atribuir um valor para a capacidade de contribuição do agente em uma coalizão para compartilhamento de recursos. Esse é o *valor de Shapley*.

O *valor de Shapley* é calculado segundo a equação (3) de aproximação do índice de *Shapley-Shubik*. Utilizamos o valor aproximado para evitar o cálculo da contribuição marginal sobre todas as permutações de agentes da coalizão. A aproximação, no entanto, tem erro pequeno para coalizões com quantidade grande de agentes, dessa forma, utilizamos a aproximação para coalizões com mais de 6 agentes. No caso de coalizões menores calculamos o valor de Shapley pela definição.

Baseado no *valor de Shapley* podemos fazer a redistribuição dos recursos na coalizão, tentando com isso atender melhor aos planos de ações dos agentes. A melhora ou piora em uma distribuição de recursos será capturada pelo conceito de *utilidade* no escopo individual e *bem-estar social* no escopo coletivo. Discutiremos na próxima Seção a redistribuição dos recursos na coalizão, que é feita através de um procedimento de alocação de recursos.

4.1.2 Alocação através da Troca de Recursos

Alocação é uma distribuição particular de recursos entre os agentes e, para recursos compartilháveis e indivisíveis, uma partição dos recursos originais. No procedimento descrito a seguir, os recursos são distribuídos internamente na coalizão segundo a contribuição de cada agente, medida pelos *valor de Shapley* (φ_i). Além disso, a alocação é executada como consequência da participação em coalizões, ou seja, é uma norma do modelo a qual todo agente deve obedecer. Em outros termos, toda vez que um agente participa de uma coalizão ele disponibiliza os seus recursos para serem compartilhados dentro da coalizão, cedendo recursos e, por outro lado, recebendo recursos de outros membros [11]. Portanto, dentro de uma coalizão há a troca de recursos entre os seus membros. A alocação é construída através do procedimento apresentado na definição 14 e do algoritmo 2.

Definição 14: Alocação dos recursos na coalizão. A alocação dos recursos dentro da coalizão segue a ordem definida pelo *valor de Shapley* dos agentes. O procedimento de alocação é definido pelo algoritmo 2, e constrói o *vetor de alocação* $al_i \in \mathbb{N}^{|\mathcal{K}|} \ \ \forall i \in S$.

No algoritmo 2 vemos um laço onde cada agente é iterado através da sua ordem de relevância (valor de Shapley) e recebe uma quantidade dos recursos da coalizão (linha 5). Essa quantidade

[11] Dependendo do comportamento do agente, ele pode disponibilizar parte ou todos os recursos. O algoritmo exige apenas que o agente siga a norma de troca de recursos intra-coalizão.

Algorithm 2 Alocação de Recursos da Coalizão

Require: Estarem calculados
$\varphi_i \leftarrow$ Shapley value $\forall i \in S$
$rc_S \leftarrow \sum rc_i, \quad rc_S^{\Pi} \leftarrow \sum rc_i^{\Pi}; \quad \forall i \in S$
Ensure: Vetor al_i montado.

1: $rc_S^{-al} \leftarrow rc_S$: Vetor residual.
2: $S^{Ord} \leftarrow Ord[ag_i, \varphi_i], \forall i$: Ordenação dos agentes pelo valor de Shapley
3: **for all** $i \in S^{Ord}$ **do**
4: **for all** $k \in \mathcal{K}$ **do**
5: $al_i(k) \leftarrow \min\{rc_i^{\Pi}(k), rc_S^{-al}(k)\}$
6: $rc_S^{-al}(k) \leftarrow rc_S^{-al}(k) - al_i(k)$
7: **end for**
8: **end for**
9: **if** $rc_S^{-al} \neq 0$ **then**
10: Divide o vetor residual proporcionalmente ao valor de Shapley
11: **end if**

é o valor mínimo entre as necessidades do agente e os recursos disponíveis. Na linha 6 os recursos da coalizão são consumidos e o próximo agente pode ser processado. Pode haver um remanescente de recursos após todos os agentes serem atendidos (linha 9), dessa forma, esses recursos são distribuídos aos agentes numa base proporcional (linha 10).

Uma alocação ocorre toda vez que um agente avalie os ganhos possíveis da participação em uma coalizão, ou entre numa coalizão, dessa forma, é uma parte fundamental no processo de formação de coalizões. O processo de formação de coalizões que chamamos *heurística ganha-ganha* assume o procedimento de alocação de recursos intra-coalizão como descrito. Na próxima Seção definiremos a *heurística ganha-ganha*, e demonstraremos as condições de convergência do algoritmo.

4.2 DESENVOLVIMENTO DA HEURÍSTICA

O objetivo da heurística é possibilitar a troca de recursos entre os agentes, segundo algum critério de racionalidade. O processo completo deve cobrir as seguintes etapas:

1. Montagem de uma coalizão.

2. Troca dos recursos entre os seus membros.

3. Uso dos recursos pelos agentes para cumprir os seus planos.

A etapa 3 não faz parte da heurística de formação de coalizões. É responsabilidade individual do agente prover os meios para execução das ações, cujos recursos são atendidos **pela** coalizão. As etapas 1 e 2 estão conjugadas e compõem o processo de formação de coalizões.

Diferentemente das etapas de formação de coalizões propostas por Sandholm et al. (1999), não temos bem separadas as etapas de *geração da estrutura coalizional*, *ação interna para realizar o objetivo da coalizão* e *divisão dos ganhos*. No modelo de alocação de recursos, a própria divisão destes recursos é o objetivo da coalizão, e isso é eficiente a partir do momento que é vantajoso para o agente. Dessa forma, há uma circularidade entre as ações formar coalizão e alocar recursos, no sentido de que: **a coalizão é formada para alocar recursos, mas os membros da coalizão dependem da alocação para decidir se há ganho na sua efetivação**. Resolvemos essa retro-alimentação entre formação e alocação, criando coalizões que podem ser desfeitas se o resultado não for satisfatório para ambas as partes, o agente sendo associado e os demais da coalizão.

Nas próximas seções detalharemos a etapa de formação da coalizão. No Capítulo 5 apresentaremos um exemplo do uso de recursos para o agente cumprir os seus planos.

Iniciaremos com a definição de utilidade, critério de satisfação do agente, e do bem-estar social. Definiremos o procedimento de alocação que corresponde divisão dos recursos internos a uma coalizão, e efetivação da coalizão, que corresponde ao assentimento individual e coletivo de que essa coalizão é vantajosa.

4.2.1 Utilidade do Agente e Bem-Estar da Coalizão

A dimensão dos benefícios de uma coalizão para o agente é medida através de uma *função utilidade*, enquanto o benefício para a coalizão da participação do agente, por uma *função de bem-estar social*.

Por benefício para o agente nos referimos à proporção do plano de ação que o agente consegue realizar, dada uma alocação. Assim, a função utilidade é definida como $u : (\mathbb{N}^{\mathcal{K}}, \mathbb{N}^{\mathcal{K}}) \rightarrow [0, 1]$, ou seja, é a função que leva um par de vetores de recursos para um intervalo real entre zero e um. Essa função pode ser representada como uma razão entre a soma da quantidade de recursos alocada sobre a quantidade de recursos necessárias para realização do plano.

Definição 15: Utilidade do agente i na coalizão S. É a utilidade que o agente i alcança,

dada a alocação al_i^S e o plano Π_i.

$$u_i(S) = \frac{\sum \min(al_i^S(k), rc_i^{\Pi}(k))}{\sum rc_i^{\Pi}}$$

O bem-estar social corresponde a uma medida da satisfação de uma população. Dada que a função utilidade é uma medida do benefício da participação de um agente em uma coalizão, adotaremos como o bem-estar dessa coalizão, a média das utilidades dos seus membros.

Definição 16: Bem-estar de S. O bem-estar social da coalizão S é a média da utilidade dos agentes.

$$sw(S) = \frac{\sum_{i \in S} u_i(S)}{|S|}$$

Nesse ponto temos os elementos para definir o *teste ganha-ganha*, que é o critério de racionalidade para formação de coalizões.

Propriedade 1: A ordem de entrada de um agente na coalizão não altera a sua utilidade. Essa propriedade afirma que $u_i(S \cup \{i\}) = u_i(\{i\} \cup S)$. Essa afirmação é verdadeira porque a alocação de recursos é feita baseada no valor de Shapley, que considera a importância de um agente dada todas as ordens de participação na coalizão.

4.2.2 Teste Ganha-Ganha

O *teste ganha-ganha* corresponde aos critérios utilizados na montagem de coalizões. Estes englobam um critério de racionalidade individual e dois de racionalidade coletiva. O critério de racionalidade individual dirige o agente a buscar prioritariamente o seu bem-estar próprio. Esse critério é formalizado através da noção do aumento de utilidade, lembrando que no nosso modelo a utilidade é uma medida da proporção do plano que tem demandas de recursos atendidas.

Definição 17: Racionalidade individual. A associação do agente i com a coalizão S é individualmente racional se, e somente se, $u_i(S \cup \{i\}) > u_i(T)$, onde $T, S \subset N$, ou seja, são coalizões, e $i \in T$.

A definição acima ressalta que um agente racional busca melhorar sua satisfação individual. Qualquer coalizão S, é melhor que T, da qual o agente faz parte, se aumentar a utilidade, sendo uma potencial candidata para o agente se unir.

Entendendo de outra forma, um agente apresenta um comportamento racional se conseguir mais recursos para satisfazer seu plano de ação. No entanto, esse critério nada diz sobre os outros membros da coalizão, ou seja, a associação de i pode ser tanto benéfica quanto prejudicial aos membros de S. Por isso um critério de racionalidade coletiva é incorporado no processo de julgamento dos benefícios de uma coalizão.

Definição 18: Racionalidade Coletiva. Uma coalizão S é coletivamente racional se, e somente se, aceitar novos membros quando o bem-estar dos seus agentes, antes e depois da entrada do novo membro, não piora. Essa regra é formalizada pela comparação entre o bem-estar de S antes da associão do agente i, e a média da utilidade dos membros de S após a associação com i, excluindo-o, no entanto, da média.

$$\frac{\sum_{j \in S \cup \{i\}, j \neq i} u_j(S \cup \{i\})}{|S|} \geq sw(S)$$

Vale ressaltar que a média da utilidade é o valor do bem-estar, sendo assim, estaremos comparando o bem-estar entre os mesmos conjuntos de agentes, ou seja, todos os membros da coalizão S original. Essa regra nos garante que não há piora média para a coalizão S pela participação do novo membro i.

A partir dos dois conceitos de racionalidade podemos definir o *teste ganha-ganha*.

Definição 19: Teste Ganha-Ganha. Designa-se por *teste ganha-ganha* a verdade das seguintes condições:

$$u_i(S \cup \{i\}) > u_i(T) \tag{9}$$

$$\frac{\sum_{s \in S \cup \{i\}, s \neq i} u_s(S \cup \{i\})}{|S|} \geq sw(S) \tag{10}$$

onde $S, T \subset N$ e $i \in T$. Avalia-se a entrada de i em S segundo as regras acima.

Um ponto a enfatizar é que o teste ocorre no momento em que i avalia o benefício de *abandonar* uma coalizão T e unir-se a uma nova coalizão S. O teste pode ser positivo ou não dependendo do atendimento das condições impostas pelo teste. Além disso, o teste pressupõe o armazenamento dos valores de utilidade $u_i(T)$ e $sw(T)$ antes de qualquer alteração nessa coalizão.

Demonstraremos na Seção 4.3.1 que o *teste ganha-ganha* é suficiente para garantir o incre-

mento no bem-estar sempre que um novo agente se associa a uma coalizão. Esse resultado é intuitivo a partir das noções de racionalidade apresentadas, pois há melhora para o agente individual e para a média dos outros agentes afetados.

4.2.3 Algoritmo de Formação de Coalizões

Algorithm 3 Algoritmo de formação de coalizões com a utilização do valor de Shapley

Require: Agent's plans $\Pi_i \forall i$
Require: Mapping $\Pi_i \rightarrow rc_i^{\Pi}$
Require: Max stationary iterations: λ

1: $sw_N^{(0)} \leftarrow sw_u(N))$; $t \leftarrow 0$
2: **while** $(sw_N^{(t)} = sw_N^{(t-1)} = ... = sw_N^{(t-\lambda)})$ **do**
3: **for** $i \leftarrow Ord^N$ **do**
4: $S_i \leftarrow \{i\} \vee$ *i*'s coaliton if there is one.
5: Initialize the utility array (u_i^j).
6: **for** $\forall j, j \notin S_i$ **do**
7: $S_i^j \leftarrow \{i\} \cup S^j$
8: $al_i \leftarrow alocate(S_i^j)$ (alg. 2)
9: $u_i^j \leftarrow \frac{\sum al_i}{rc_i^{\Pi}}$ (def.15)
10: **end for**
11: $j \leftarrow \arg\max_j u_i^j$
12: **if** (*double-win test*) holds (def. 19) **then**
13: **join**(S_i, i, S^j)
14: **end if**
15: **end for**
16: $sw_N^{(t)} \leftarrow sw_u(N)$ (def. 16)
17: **if** $sw_N^{(t)} = sw_N^{(t-1)} = sw_N^{(t-2)}$ **then**
18: $Ord^N \leftarrow resort(Ord^N)$
19: **end if**
20: $t \leftarrow t + 1$
21: **end while**

A partir das definições anteriores, podemos formalizar o procedimento de formação de coalizões como expressado no Algoritmo 3.

O Algoritmo 3 apresenta o processo de formação de coalizões. O primeiro requerimento para a sua execução é que os agentes tenham montado os seus planos. Como pontuado na Seção 4.1.1, o procedimento é independente do mecanismo de planejamento utilizado pelos agentes, por isso, a única restrição é que os agentes tenham uma função de tradução que transforme planos em recursos requeridos. O algoritmo recebe a **razão de busca** (λ) como

parâmetro, que é a quantidade de iterações passadas que são consideradas para assumir que tenha havido convergência. Discutiremos sobre esse tema na Seção 4.3.3.

A execução do algoritmo é iniciada armazenando-se o bem-estar corrente para futuras comparações, o qual também é armazenado após cada iteração (linha 16). Utilizamos a notação $sw_N^{(t)}$ para fazer referência ao array circular de λ posições no qual o bem-estar é armazenado. A linha 2 apresenta a condição de parada e na linha 3 temos o laço principal que itera por todos os agentes em uma ordem específica. Todos os agentes são visitados através do laço na linha 6, no qual o algoritmo de alocação é executado para cada candidato, permitindo a avaliação da utilidade após a alocação realizada (line 9). Cada avaliação é armazenada no array na posição u_i^j para futura comparação. Na linha 11 capturamos o melhor candidato, e na próxima linha executamos o *teste ganha-ganha*, i.e., a condição para que o agente se una à coalizão é que haja ganhos para ambas as partes, representado pelo incremento na utilidade do agente e no bem-estar da coalizão.

Finalmente, o último elemento a ressaltar do algoritmo é o **reordenamento dos agentes** que é executado na linha 18. Isto é feito para evitar a mesma ordem de testes entre os agentes. A avaliação empírica demonstrou que variando a ordem das visitas atinge-se um melhor bem-estar social.

4.3 CARACTERÍSTICAS DA HEURÍSTICA

A *heurística ganha-ganha* apresenta algumas características próprias, entre elas a *monotonicidade do bem-estar*, é uma *heurística gulosa* e é um procedimento *distribuível* de montagem de coalizões. Nessa Seção exploraremos essas características.

4.3.1 Monotonicidade

Vale a pena ressaltar que os agentes trocam recursos como parte do processo de formação de coalizões. Esse procedimento, associado à racionalidade individual dos agentes, por um lado, e a manutenção do bem-estar das coalizões, por outro lado, agem como estímulo para a o incremento de ambas as funções. De fato, todas as ações dos agentes objetivam o incremento da sua própria utilidade, seja quando se unem a uma coalizão, ou quando a abandonam. Além disso, o bem-estar da coalizão é garantidamente incrementado quando novos agentes são

aceitos e a troca de recursos mantém-no inalterado quando os agentes abandonam a coalizão.

O fato da utilidade ficar inalterada após um agente abandonar uma coalizão está relacionado ao próprio conceito de utilidade do agente. Um agente melhora a sua situação quando dispõe de recursos mais adequados à sua utilização. Da mesma forma, ao melhorar a sua situação individual, melhora o bem-estar geral da sociedade. Esse tipo de abordagem é similar aos contratos de negociação, e pode ser entendido como uma generalização da negociação individual para negociação em grupo.

Sumarizando, se o algoritmo 3 for seguido, como uma consequência do **procedimento de alocação** e do *teste ganha-ganha*, teremos as seguintes regras garantidas:

Propriedade 2: Abandono da coalizão T não decrementa a utilidade dos membros de T. Essa propriedade é efeito direto da troca de recursos executada na efetivação de um novo membro.

Propriedade 3: O procedimento $join(T, i, S)$ incrementa a utilidade do agente i.

Propriedade 4: O procedimento $join(T, i, S)$ não altera a utilidade dos agentes de T.

Finalmente, após qualquer execução do procedimento $join$ o bem-estar do sistema é aumentado. Essa afirmação é demonstrada pelo teorema 1, que apresenta a monotonicidade da função de bem-estar se seguirmos os procedimentos colocados anteriormente.

Teorema 1: A função de bem-estar é monotônica seguindo-se o procedimento de alocação e o teste ganha-ganha.

Demonstração. Dadas as coalizões $(T \cup \{i\})$ e S queremos demonstrar que se seguirmos o *procedimento de alocação* e o *teste ganha-ganha*, como definidos em def. 14 e def. 19, o bem-estar não é decrementado pela execução de $join(T, i, S)$. Para entendimento, ressaltamos que:

1. $(T \cup \{i\})$ e $(S \cup \{i\})$ identificam coalizões.

2. $join(T, i, S)$ é o procedimento de abandono da coalizão $(T \cup \{i\})$ pelo agente i, para associação numa nova coalizão $(S \cup \{i\})$.

Observação: Nessa demonstração adotaremos a notação de $T \cup \{i\}$ para a coalizão inicial,

ou seja, $i \notin T$. Essa notação é ligeiramente diferente da adotada nas definições 17, 18 e 19, mas os resultados são equivalentes. Optamos por essa abordagem diferente na demonstração pois precisávamos de elementos para representar explicitamente a transição do agente entre coalizões. Por outro lado, para as definições gerais esse elemento de detalhamento seria desnecessário, tornando a expressão dos conceitos ainda mais extensa.

Temos que N é o conjunto de todos os agentes. Para a demonstração é suficiente assumir que $N = T \cup S \cup \{i\}$, uma vez que qualquer agente $l \notin T, S$ não é afetado pela execução de $join(T, i, S)$. Além disso, vamos definir N^0 como sendo a partição dos agentes em $\{(T \cup \{i\}), S\}$ e N^1 o inverso, ou seja, $\{T, (S \cup \{i\})\}$. Além disso qualquer agente não pertencente a T, S é inafetado por hipótese pela transição de N^0 em N^1. Dessa forma, N^0 representa a configuração dos agentes antes de $join(T, i, S)$ e N^1 após a execução do procedimento.

$$N^0 = \{(T \cup \{i\}), S\}$$
$$N^1 = \{T, (S \cup \{i\})\}$$

Assim, queremos demonstrar que se vale $join(T, i, S)$, temos $sw(N^0) \leq sw(N^1)$.

$$join(T, i, S) \Rightarrow sw(N^0) \leq sw(N^1) \tag{11}$$

Etapa 1: inequação de utilidades

Primeiramente vamos transformar a parte direita de (11) em uma relação de funções utilidade. Destacando-se essa relação, temos:

$$sw(N^0) \leq sw(N^1) \tag{12}$$

O que é equivalente a:

$$sw((T \cup \{i\}) \cup S) \leq sw(T \cup (S \cup \{i\})) \tag{13}$$

Pela def. 16 de bem-estar, temos que $sw(S) = \sum_{l \in S} u_l(S)/S$.

$$sw((T \cup \{i\}) \cup S) = \frac{\sum_{j \in T, j \neq i} u_j(T \cup \{i\}) + u_i(T \cup \{i\}) + \sum_{l \in S} u_l(S)}{N} \tag{14}$$

$$sw(T \cup (\{i\} \cup S)) = \frac{\sum_{j \in T} u_j(T) + u_i(S \cup \{i\}) + \sum_{l \in S, l \neq i} u_l(S \cup \{i\})}{N} \tag{15}$$

Aplicando (14) e (15) em (13) e simplificando N, temos:

$$\sum_{j \neq i} u_j(T \cup \{i\}) + u_i(T \cup \{i\}) + \sum u_l(S) \leq \sum u_j(T) + u_i(S \cup \{i\}) + \sum_{l \neq i} u_l(S \cup \{i\}) \tag{16}$$

Dado que pela propriedade 2 o bem-estar não é afetado pelo abandono de um agente, temos que $\sum_{j \neq i} u_j(T \cup \{i\}) = \sum u_j(T)$. Aplicando em (16), temos que:

$$u_i(T \cup \{i\}) + \sum u_s(S) \leq u_i(S \cup \{i\}) + \sum_{s \neq i} u_s(S \cup \{i\}) \tag{17}$$

O que equivale a:

$$\sum_{s \neq i} u_s(S \cup \{i\}) - \sum u_s(S) \geq -(u_i(S \cup \{i\}) - u_i(T \cup \{i\}))$$

Podemos representar suscintamente essa inequação como:

$$\Delta sw(S, i) \geq -\Delta u_i(T, S) \tag{18}$$

Etapa 2: equivalência da inequação

A volta é verdade, aplicando as operações inversas e assumindo o procedimento de alocação. Assim, se (18) vale, significa que houve incremento no bem-estar.

Etapa 3: teste ganha-ganha executa a inequação

Mas assumimos também o *teste ganha-ganha* o que garante eq. 10 pela def. 19. De fato, eq. 10 exige $\frac{\sum_{s \in S \cup \{i\}, s \neq i} u_s(S \cup \{i\})}{|S|} \geq sw(S)$, o que representa $\Delta sw(S, i) > 0$, portanto eq. 18 é garantido pelo *teste ganha-ganha*. □

4.3.2 Estratégia Gulosa

O algoritmo é executado sequencialmente para qualquer agente no ambiente. Durante cada iteração, o agente ativo inspeciona os outros, buscando por oportunidades de incremento de utilidade. Dessa forma, cada agente adota uma **estratégia gulosa** uma vez que apenas considera o próprio bem-estar. O mesmo para a coalizão, que aceita somente agentes que não

interfiram negativamente no seu bem-estar.

Algoritmos que adotem uma estratégia gulosa compõem uma familia de algoritmos de busca que executam escolhas considerando apenas os aspectos locais das variáveis envolvidas no processo de tomada de decisão. Como pontuado anteriormente, o *procedimento de formação de coalizões* apresenta essa característica e, consequentemente, é uma forma de algoritmo guloso para identificação de estruturas coalizionais e, por se tratar de um algoritmo guloso, pode não encontrar a estrutura coalizional ótima. Apesar disso, mostraremos na próxima Seção que o algoritmo alcança resultados satisfatórios para o problema de compartilhamento de recursos.

Considerando a tendência de convergência para uma configuração local, incrementamos a variabilidade da busca através de uma **função de reordenamento**. Essa função modifica a sequência na qual os agentes são ativados e é executada sempre que há uma manutenção no bem-estar por um determinado intervalo de iterações. A função de reordenamento utiliza uma estratégia de ativação similar a da **razão de busca**, que determina a convergência do bem-estar.

4.3.3 Estabilidade do bem-estar

Enquanto buscamos uma nova estrutura coalizional, necessitamos de um critério de parada para determinar se o algoritmo encontrou uma configuração local ótima. Isso é feito pelo parâmetro λ, que representa a quantidade de iterações com bem-estar estagnado a partir da qual se considera que houve a convergência para um valor final.

Como o valor do bem-estar é incrementado monotonicamente, e somente é modificado por alteração na estrutura coalizional (um agente se une ou abandona uma coalizão), eventualmente uma configuração na qual não há incremento possível de utilidade para qualquer agente, é localizado. Dessa forma, através do teste dos λ valores de bem-estar, o procedimento de formação de coalizões pode determinar se o valor do bem-estar se estabilizou e interromper novas iterações.

4.3.4 Eficiência

O algoritmo é eficiente para alocação de recursos em cenários de disponibilidade de recursos. Veremos na Seção de resultados (Seção 4.4) as medições para cenários com maior e menor disponibilidade de recursos. O algoritmo alocou eficientemente os recursos, no sentido de que fez uma redistribuição destes que levou a altos níveis de bem-estar na sociedade, e o fez em poucas iterações.

4.3.5 Distribuível

O algoritmo de formação de coalizões apresentado é potencialmente *distribuído* [12]. Essa característica é resultante do modelo de tomada de decisão, que é baseado na configuração de recursos de cada agente e homogêneo para todos eles.

O fundamento de todo o processo decisório do agente é a ausência ou presença de recursos. Por isso, como apresentado anteriormente, recursos funcionam como *procuradores* dos agentes. O mesmo vale para as coalizões, pois tem como objetivo a alocação de recursos que, por outro lado, indicam se a coalizão é vantajosa ou não. Consequentemente, as funções que a heurística aplica são, direta ou indiretamente, baseadas em recursos. Dessa forma, a única necessidade para um agente executar o processo decisório de formação de coalizão é conhecimento da disponibilidade e necessidade de recursos de outros agentes, o que diminui grandemente a quantidade de informações a serem mantidas entre nós de processamento.

Além disso, o mesmo processo decisório de formação de coalizões é adotado para todos os agentes. Com isso, cada agente avalia as coalizões candidatas da mesma forma, seguindo os critérios de racionalidade individual e coletiva. Como o processo é rigorosamente igual, independentemente do agente, ele pode se replicado em unidades de processamento diferentes. Ainda, como o processo de tomada de decisão necessita de poucas informações, apenas dois vetores de reais por agente, e informação de como o sistema está particionado em coalizões, a quantidade de comunicação é leve.

Por outro lado, é necessário o desenvolvimento de um mecanismo de sequenciamento da formação de coalizão. Por exemplo, quando dois agentes decidem se unir a mesma coalizão. Se a avaliação ocorrer ao mesmo tempo, uma delas deverá ser invalidada, pois um não considerou

[12] A implementação feita, bem como as medições, não foram feitas na versão distribuída

a presença do outro na coalizão. No entanto, esse pode ser um mecanismo de *controle de alterações otimista*, ou seja, assume-se que não haverá modificação concorrente das coalizões, mas caso ocorram, essa situação poderá ser identificada e uma das alterações descartada. O agente que teve sua solicitação de associação descartada poderá iniciar outra avaliação, sob a nova configuração das coalizões.

4.3.6 Complexidade e Performance

O algoritmo avalia para cada agente todos os vizinhos, buscando possibilidades de melhores associações, o que resulta em $O(n^2)$ avaliações por iteração. O algoritmo de alocação de recursos na coalizão implica no cálculo do valor de Shapley para cada agente, o que tem complexidade $O(k^2n)$, e da distribuição dos recursos pelo valor de Shapley, que em complexidade $O(kn)$, dessa forma, o algoritmo tem complexidade $O(k^3n^4)$, por iteração.

Para uma quantidade grande de agentes, vamos supor 100 agentes, e 5 recursos, o algoritmo executaria 125E8 iterações, o que é relativamente rápido considerando-se o poder de processamento disponível nos atuais computadores. Em uma máquina Intel core Duo com 2,54 Ghz de CPU, um loop que insere inteiros em uma lista circular levou aproximadamente 4 minutos para execução dessa quantidade de iterações. Para tomada de decisão em tempo real outros estudos deveriam ser realizados, levando-se em conta a possibilidade de executar o processo distribuído. Potencialmente o algoritmo tem uma boa perspectiva de escalabilidade, ou seja, poderá ter a sua performance aumentada com a adição de novos nós de processamento.

4.4 RESULTADOS

Nesta Seção apresentaremos o resultado das simulações da *heurística ganha-ganha*. Essas simulações foram executadas sobre um ambiente estático, com a finalidade de verificar o comportamento do algoritmo com relação aos ganhos de bem-estar. O ambiente é estático pois não há entrada nem saída de agentes, nem variação no plano de ação dos agentes. Todos os agentes recebem uma dotação inicial de recursos e uma demanda de recursos, representando o plano de ação a ser executado. Os agentes executam a sessão de formação de coalizão segundo a heurística, medindo-se o bem-estar do conjunto total de agentes ao final da execução.

As simulações são compostas de 3 parâmetros:

1. **Quantidade de agentes (P_{ag})**: variamos a quantidade de agentes, de 20 a 80 [13].

2. **Disponibilidade de recursos (P_{aval})**: Identifica a proporção de recursos que o agente terá disponível para compartilhamento.

3. **Carência do agente (P_{lack})**: Identifica o percentual de recursos que o agente não dispõe.

   A diferença entre a *disponibilidade* e a *carência* é que a disponibilidade informa quantos recursos o agente possui a mais do que o necessário para realização dos seus planos, enquanto a *carência* informa quais recursos o agente não tem. Evidentemente os dois parâmetros influenciam a quantidade de recursos do agente. Veremos adiante que cada um deles altera o vetor de recursos de uma forma, e são configurados para estimular que os agentes participem de coalizões.

Chamamos de *configuração da simulação* a uma atribuição de valor para cada um dos parâmetros da simulação ($P_{ag}, P_{aval}, P_{lack}$). Foram montadas 24 configurações de simulação, e medido o valor de *bem-estar* do ambiente, e o desvio padrão dessa medida. O valor medido é referente a média de 10 execuções. Detalharemos, a seguir, o significado de cada parâmetro e os valores que cada um deles assumiu nas simulações.

Cada agente recebeu K tipos diferentes de recursos, distribuídos entre os agentes através de uma distribuição normal $\mathcal{N}(\mu, \sigma^2)$. Fixamos o valor de μ como 10, σ como 3, e K igual a 5. Ou seja, cada agente recebeu uma dotação inicial do recurso $k \in \mathcal{K}, \mathcal{K} = \{1, 2, 3, 4, 5\}$ segundo a distribuição normal $\mathcal{N}(10,\ (3)^2)$. Com isso monta-se $rc_i(k)$ para cada agente, o que define a dotação inicial de recursos.

Por outro lado, a demanda do agente é dada pelo vetor de recursos $rc_i^{\Pi}(k)$, definido como um percentual da distribuição normal da dotação inicial de recursos do agente, ou seja, a demanda de recursos do agente é dada por $\mathcal{N}(P_{aval} \times 10,\ (P_{aval} \times 3)^2)$. Percebe-se que quanto maior P_{aval}, mais recursos dos agente serão utilizados, resultando em menor disponibilidade de recursos para compartilhamento.

[13] O maior valor já é bastante significativo, uma vez que os algoritmos disponíveis na literatura são adequados para pequenas quantidades de agentes (<15 agentes).

Além disso, como $0 \leq P_{aval} \leq 1$, a dotação inicial tenderá a ser sempre maior que a demanda de recursos, ou seja, estaríamos trabalhando com cenários de absoluta abundância de recursos. Em tal cenário não haveria estímulo para montagem de acordos, uma vez que a maior parte dos agentes estaria *satisfeito*. Para que conseguíssemos variar a *disponibilidade de recursos do agente* e manter o estímulo para a formação de coalizões, usamos o parâmetro P_{lack}.

O parâmetro P_{lack} representa a *carência de recursos do agente*, pois força que um recurso k esteja indisponível (quantidade zero do recurso k), e que, por outro lado, o agente tenha alguma demanda desse recurso. Operacionalmente, utilizamos o parâmetro P_{lack} como o percentual médio de recursos que o agente não possui. Identificamos se o agente possui algum recurso, tirando-se randomicamente um número entre 0 e 1. Se esse número for maior que o percentual de carência, então o agente possui uma quantidade desse recurso dada pela distribuição normal previamente apresentada.

Sumário

A dotação inicial de recursos e a demanda de recursos serão dadas pelas distribuições gaussianas, levando-se em conta a carência de recursos do agente (P_{lack}). A demanda de recursos será proporcional à dotação inicial segundo o parâmetro (P_{aval}). Assim:

$$rc(k) \approx \begin{cases} \mathcal{N}(\mu,\ \sigma^2), & \text{se } rand(1) \geq P_{lack} \\ 0 & otherwise \end{cases}$$

$$rc^{\Pi}(k) \approx \begin{cases} \mathcal{N}(P_{aval} \times \mu,\ (P_{aval} \times \sigma)^2), & \text{se } rand(1) < P_{lack} \\ 0 & otherwise \end{cases}$$

- Temos 3 parâmetros: P_{ags}, P_{aval} e P_{lack}
- Serão montados 24 conjuntos de configurações segundo 4 valores de P_{ags}, 3 valores de P_{aval} e 2 valores de P_{lack}.
  - $P_{ags} \in \{10, 20, 40, 80\}$.
  - $P_{aval} \in \{0.5, 0.75, 1.0\}$.

- $P_{lack} \in \{0.3, 0.5\}$.

- Cada execução será executada 10 vezes.

- Será medido o bem-estar dos agentes. O bem-estar é dado pela utilidade média dos agentes.

Na página 72 apresentamos os gráficos representando as simulações.

Análise dos Resultados

Os resultados podem ser vistos na Figura 5, que mostra o bem-estar médio dos agentes, e na Figura 6 que mostra, por sua vez, o desvio padrão médio das medidas do bem-estar [14]. As figuras mostram 24 conjuntos de simulações, com 10 execução de 10 iterações cada uma.

Através da Figura 5 podemos perceber que o bem-estar se situa entre 0.70 e 0.99, o que implica que pelo menos 70% dos recursos necessários para execução dos planos puderam ser levantados através do compartilhamento de recursos nas coalizões. O bem-estar se mostrou melhor na presença de recursos abundantes, o que pode ser percebido nos 3 melhores conjuntos, todos com carência de 30%. A mesma tendência de ter comportamento mais eficiente na presença de pouca carência de recursos se mostra no desvio padrão, que fica entre 2% e 20% nos 3 cenários citados.

Por outro lado, os cenários com *carência* de 0.5 mostram as dificuldades de montar coalizões que possam atender a todas as demandas dos agentes. A tendência demonstrada é que com o aumento da *carência de recursos*, o *bem-estar* passa a diminuir.

[14] O bem-estar apresentado é a média de 10 execuções. Por isso apresenta-se o desvio padrão dessas medidas.

	Agents			
Aval. // Lack	10	20	40	80
0.5 // 0.3	1.00	1.00	1.00	0.99
0.75 // 0.3	0.95	0.96	0.99	0.97
1.0 // 0.3	0.93	0.96	0.92	0.90
0.5 // 0.5	0.94	0.95	0.93	0.86
0.75 // 0.5	0.88	0.83	0.79	0.73
1.0 // 0.5	0.84	0.76	0.74	0.68

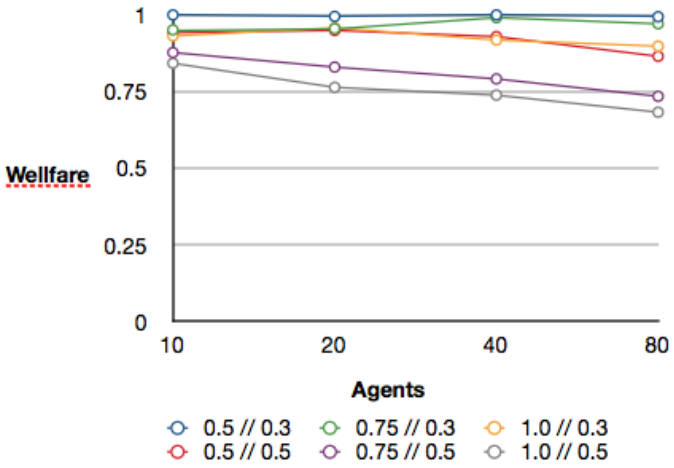


Figura 5 - Bem-estar Médio da Heurística

	Agents			
Aval. // Lack	10	20	40	80
0.5 // 0.3	0.00	0.01	0.00	0.02
0.75 // 0.3	0.16	0.13	0.04	0.11
1.0 // 0.3	0.22	0.18	0.17	0.25
0.5 // 0.5	0.20	0.17	0.19	0.28
0.75 // 0.5	0.28	0.32	0.37	0.39
1.0 // 0.5	0.35	0.37	0.40	0.44

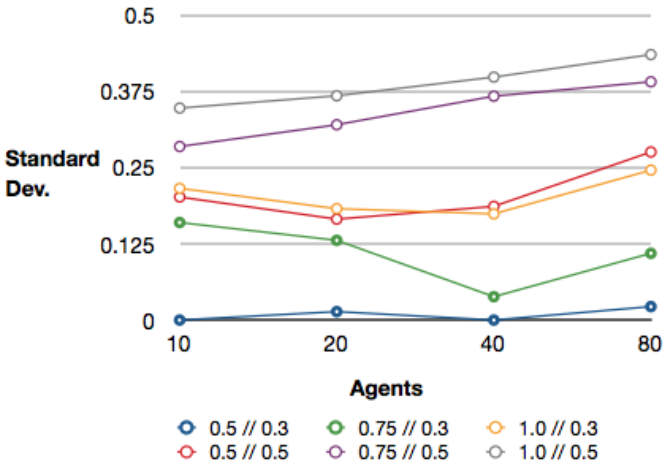


Figura 6 - Desvio Padrão Médio da Heurística

5 AMBIENTE DE SIMULAÇÃO MULTIAGENTE PARA COMPARTILHAMENTO DE RECURSOS

Nesse Capítulo apresentaremos o ambiente de teste utilizado para avaliar agentes que tenham capacidade de montar coalizões para compartilhamento de recursos. No Capítulo anterior detalhamos o algoritmo de formação de coalizões denominado *heurística ganha-ganha*. Executamos medições de performance do algoritmo, considerando-se um ambiente estático, onde os agentes não variam seus objetivos nem há variação no conjunto participantes do ambiente. Além disso, não nos detivemos na montagem de uma arquitetura de agente que permitisse incorporar as capacidades coalizionais ao processo de tomada de decisão.

Faz parte do modelo que adotamos a separação entre planejamento e interação social. Assumimos que os agentes são capazes de representar o seu plano de ação através de um conjunto de recursos demandados para a sua execução. Apresentaremos a implementação de uma função de mapeamento de planos em recursos utilizando o ambiente de teste desenvolvido por Gal et al. (2010). Demonstraremos as funcionalidades do simulador desenvolvido e apresentaremos as medições executadas sobre esse ambiente de execução.

Assim,desenvolvemos um simulador para avaliar os aspectos que surgem na interação entre agentes planejadores, quando integramos a formação de coalizões ao processo decisório dos agentes. Nesse sentido nos preocupamos com os seguintes tópicos que complementam os resultados teóricos e medições efetuadas no Capítulo anterior.

- Implementação da capacidade de mapeamento de planos em recursos.
- Estudo da interação entre o processo de planejamento de o processo de formação de coalizões.
- Teste da execução do algoritmo de formação de coalizões em um ambiente dinâmico.

Primeiramente apresentaremos a Seção *Arquitetura do Sistema Multiagente*, que descreve os elementos conceituais dos agentes, enfatizando a relação entre planejamento e formação de coalizões. A próxima Seção, *Software de Simulação* apresenta o projeto do simulador, e finalmente, a Seção *Simulações e Resultados* apresenta as medições.

5.1 ARQUITETURA DO AMBIENTE

A arquitetura do ambiente apresenta os grandes componentes do simulador. Seu objetivo é separar responsabilidades de forma a organizar o projeto do sistema em áreas distintas. Dividimos a arquitetura em 4 componentes principais:

1. Ambiente: representa todos os elementos externos ao agente, incluindo o espaço em que o agente se encontra, a estrutura de vizinhança entre os agentes, e a infra-estrutura de comunicação entre os agentes.

2. Elementos conceituais do agente: representam a estrutura interna do agente, especialmente a divisão entre planejamento e processo de formação de coalizões. O ciclo de interação entre o processo de planejamento de ações e de formação de coalizões é executado por uma sequência de ações que denominamos *comportamento do agente*.

3. Comportamento dos agentes: representa o fluxo de ações entre o processo de planejamento e formação de coalizões, em especial, define se há retroalimentação entre planejamento e montagem de coalizões, e organiza o projeto do simulador de forma que esses fluxos sejam intercambiáveis.

4. Coalizões: representa o conjunto de agentes associados. Tem a responsabilidade de impor as normas da sociedade, em específico garante que o *teste ganha-ganha* será respeitado.

Passaremos agora a detalhar os componentes da arquitetura.

5.1.1 Ambiente

O simulador contém um **mundo de grade** como ambiente no qual os agentes estão situados. O ambiente foi baseado no trabalho de Gal et al. (2010) que detalhamos na Seção 3.4.2. Segundo essa abordagem, o mundo é dividido em células discretas, com dimensão de $R \times R$ células, em que R representa a quantidade de células de uma linha/coluna da grade. Cada agente pode ocupar uma posição do ambiente e se mover entre elas. Para ocupar uma célula o agente deve possuir os recursos demandado por ela. Para simplificação do controle dos agentes, não há restrição quanto a quantidade de agentes que ocupam uma célula.

As **ações físicas** dos agentes se restringem a movimentos nas 8 direções (norte, sul, leste, oeste, noroeste, nordeste, sudoeste, sudeste). Para que executem as movimentações, os agentes devem conter os recursos requeridos para ocupação da célula alvo, e esse recurso, ao executar-se a ação, é consumido do agente. As **ações sociais** consistem na junção ou abandono de coalizões. Essas, por sua vez, são acordadas e executadas diretamente entre os agentes, sem mediação do ambiente. O *teste ganha-ganha*, que é parte da heurística, é executado conjuntamente pelo agente e pela coalizão. Da parte do agente pela racionalidade individual que é seu principal critério de decisão sobre a viabilidade de uma coalizão, enquanto pela coalizão, como norma de formação.

Os recursos são inicialmente configurados no ambiente como recursos necessários para ocupação de cada posição. Cada agente, por sua vez, recebe uma dotação própria de recursos, e ocupa uma posição inicial no mundo. O objetivo de um agente é determinado por uma posição a ser alcançada, o que permite a montagem de um plano de ação, composto dos movimentos necessários para atingir-se a célula alvo. Os recursos são conservativos, ou seja, a medida que os agentes ocupam posições do ambiente, não esgotam os seus recursos [15].

A simulação é iterativa. Os agentes são ativados individualmente, recebendo o estado do ambiente e executando o processo de escolha da ação a ser executada.

Esse esquema permite o teste de cenários em que há carência de recursos. Dessa forma os agentes são obrigados a formarem coalizões para suprirem suas necessidades. Esse ambiente virtual é um campo rico de testes para situações de formação do coalizão, possibilitando a criação de cenários variados de interação ou demanda de recursos, sem a necessidade de aprofundamento em domínios específicos de problemas.

Distribuição de recursos no ambiente

Os recursos do ambiente e dos agentes, bem como a posição inicial e alvo dos mesmos, são determinadas *randomicamente*. Para a distribuição dos recursos próprios do agente foi adotada uma estratégia simular a das simulações da *heurística*, que consiste na dotação inicial de recursos dos agentes segundo uma distribuição normal. No entanto, diferentemente do

[15] Apesar de termos implementado uma versão de ambiente em que os recursos são conservativos, não haveria impedimento para que eles fossem **não-conservativos**. A heurística não sofreria alteração por essa mudança, e seria responsabilidade do ambiente comandar o consumo de recursos dos agentes.

simulador da *heurística*, os recursos do plano não são dados diretamente como demanda, mas são montados através de um plano de movimentação do ambiente.

Cada posição do ambiente contém um conjunto de recursos demandados para a sua ocupação. Esses recursos são definidos através de uma distribuição normal que é parâmetro da simulação. Os recursos demandados pelo plano de ação dos agentes, por sua vez, dependem da posição inicial, da posição alvo, do caminho escolhido e dos recursos demandados pelas posições desse caminho.

Dessa forma, os parâmetros da simulação são:

1. Média e desvio padrão da distribuição de recursos próprios dos agentes: os recursos são distribuídos segundo uma distribuição normal $\mathcal{N}(\mu, \sigma)$ para cada dimensão k do vetor de recursos. As distribuições de cada dimensão são independentes.

2. Média e desvio padrão da distribuição de recursos das células do ambiente: os recursos necessários para ocupação de cada célula também seguem uma distribuição normal dada por $\mathcal{N}(\mu_{cell}, \sigma_{cell})$.

   Um outro parâmetro é a relação entre o plano montado e o processo de formação de coalizões,

3. Comportamento coalizional dos agentes: o comportamento coalizional define o fluxo entre o procedimento de planejamento e de formação de coalizão. Esse fluxo pode ser unidirecional ou com retro-alimentação (o resultado da formação de coalizão pode alterar os planos). Na simulação testamos algumas configurações possíveis para esse fluxo. Os comportamentos serão detalhados na Seção 5.1.3.

Na próxima Seção apresentaremos a arquitetura interna do agente coalizional. Representaremos cada componente interno, detalhando sua função e inter-relação.

5.1.2 Elementos Conceituais de um Agente Coalizional

Chamamos de *agente coalizional* o agente que implementa a *heurística ganha-ganha* em um ambiente multiagente, participando de interações com outros agentes coalizionais. Uma vez que o agente se insere em um SMA, segundo a concepção adotada, apresenta capacidade de receber informações do ambiente através de sensações, e agir sobre o ambiente, segundo as categorias de ações descritas anteriormente.

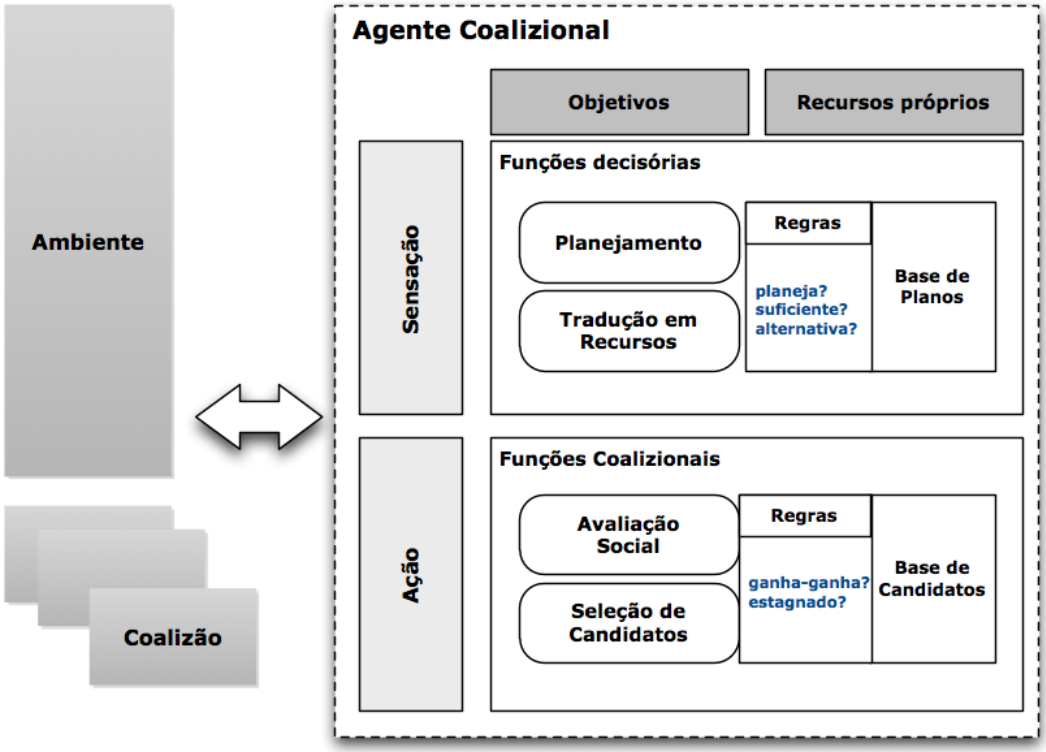


Figura 7 - Agente Coalizional

Na Figura 7 temos representadas as funções principais de um agente com capacidade de participar de coalizões. Dividimos essas funções em decisórias e coalizionais. O grupo de funções decisórias agem sobre os planos de ação, das quais a **função de planejamento** executa a montagem e alteração dos planos de ação e a **função de tradução** calcula os recursos necessários para execução do plano corrente. O grupo de **funções coalizionais**, por sua vez, avalia e seleciona os agentes passíveis de associação. As funções coalizionais são divididas em **função de avaliação social**, responsável por montar alocações de recursos em coalizões candidatas, e **função de seleção de candidatos**, que seleciona a coalizão

candidata que gere o melhor retorno e atenda ao *teste ganha-ganha*. As funções coalizionais implementam a **heurística ganha-ganha** segundo apresentada no Capítulo 4.

Há a interdependência entre as funções decisórias e coalizionais, que é expressa na necessidade de um plano de ação ser mapeado em um conjunto de recursos demandados, para que a formação de coalizões possa ser executada. Dados isso, entende-se que o plano direciona o objetivo do processo de formação de coalizões. No entanto, a dependência existe, também, na direção contrária, uma vez que um plano é inviável sem os recursos para a sua execução. Dessa forma, as associações do agente podem influenciar o plano selecionado.

Chamamos de **comportamento do agente** o controle do fluxo entre as duas categorias de funções (decisórias ou coalizionais) perante a ausência de recursos.

Identificamos 3 comportamentos diferentes para o agente:

- **Comportamento A**: Planejamento e avaliação em uma passagem: O agente planeja uma vez e avalia os candidatos que possam auxiliar com recursos para o atendimento ao plano em apenas uma passagem.
- **Comportamento B**: Planejamento em uma passagem e avaliação iterativa; O agente planeja uma vez e a avaliação é executada até que o ambiente esteja **estagnado** ou os recursos sejam **suficientes**.
- **Comportamento C**: Planejamento e avaliações iterativos: O agente planeja as ações mas pode replanejar caso o ambiente esteja **estagnado** a ainda exista **alternativa** de outro plano. A avaliação coalizional e montagem de coalizão é feita iterativamente utilizando a *heurística ganha-ganha*.

5.1.3 Comportamentos dos Agentes Coalizionais

Nesta Seção descreveremos os comportamentos do agentes coalizionais. Os comportamento estão representados nas Figuras 8, 9 e 10. Essas figuras apresentam diagramas com as ações e os testes que compõem cada comportamento. Os testes são representados por losangos, e nos comportamentos representam regras para avaliar a situação do ambiente e tomar decisões sobre o fluxo de cada comportamento. Na tabela 2 temos listadas essas regras.

planeja	Teste utilizado para verificar a necessidade de planejamento. É positivo quando o agente não contém um plano válido ou o ambiente sofreu modificação relevante.
suficiente	Teste utilizado para verificar se o agente tem todas as necessidades de recursos atendidas. É positivo quando o agente detém os recursos necessários para execução completa do plano.
estagnado	Teste utilizado para verificar se houve a estabilização da montagem de coalizões. É positivo quando não há melhora no bem-estar por λ iterações.
ganha-ganha	Teste utilizado para verificar se uma coalizão entre um agente e outro pode ser realizado. É positivo quando o agente e o bem-estar da coalizão terão melhora com a associação.
alternativas	Teste utilizado para verificar se executa replanejamento após tentativa de montagem de coalizão. É positivo quando ainda existem outros planos a serem explorados que tenham custo menor que o corrente,

Tabela 2 - Regras do agente coalizional

Alguns desses testes são específicos de determinados comportamentos. Os testes estão listados em por ordem de generalidade (os mais gerais são utilizados em maior quantidade de comportamentos).

A. Planejamento e Avaliação em uma Passagem

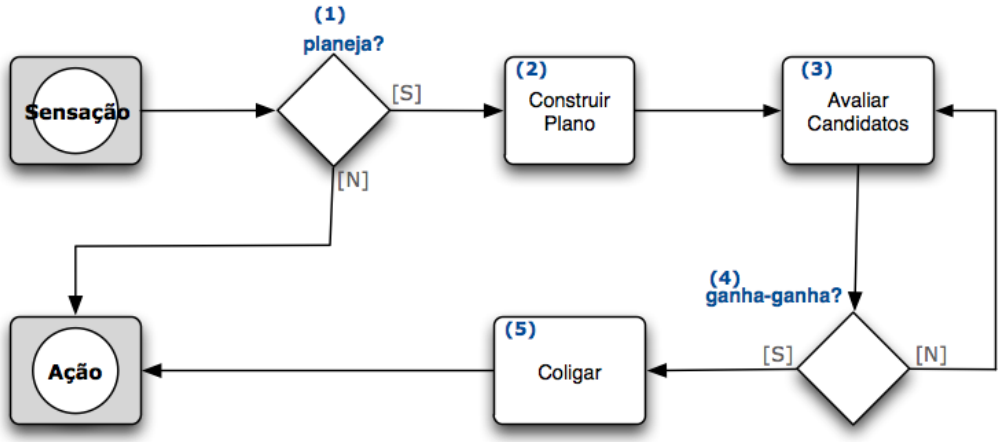


Figura 8 - Planejamento Cego

Na Figura 8 temos o comportamento mais simples do agente, que consiste na construção de um plano de ação, seguida da tentativa de montagem de uma coalizão que atenda a esse plano, executando-se ambas ações em apenas uma passagem.

O fluxo representa o processo de tomada de decisão do agente. O evento inicial é a recepção de um estímulo sensorial do ambiente, que pode gerar um novo plano de ação ou a execução imediata de uma ação em resposta ao estímulo (teste **1**). Um novo plano é construído na fase **2**, *construir plano*, que fornece os subsídios para a *avaliação de candidatos* à associações em coalizões. Associações estas que pretendem complementar os recursos próprios dos agentes para possibilitar a realização dos planos. As funções coalizionais estão representadas pelas fases **3,4** e **5**, sendo ciclo $(3) \rightarrow (4) \rightarrow (3)$ a representação uma iteração da algoritmo 3. O ciclo busca os agentes que forneçam o maior incremento de utilidade e ao mesmo tempo passem no *teste ganha-ganha*. A coalizão é montada com o agente que atenda aos requisitos, finalizando-se o fluxo com a ação sobre o ambiente.

Percebe-se que caso não haja um plano válido, este é construído sem a interferência das funções coalizionais do agente, uma vez que não há retroalimentação entre as fases **3,4** e **2**.

A simulação 1 detalhada na Seção 5.3 avalia os resultados desse comportamento no ambiente de teste.

B. Planejamento em uma Passagem e Avaliação Iterativa

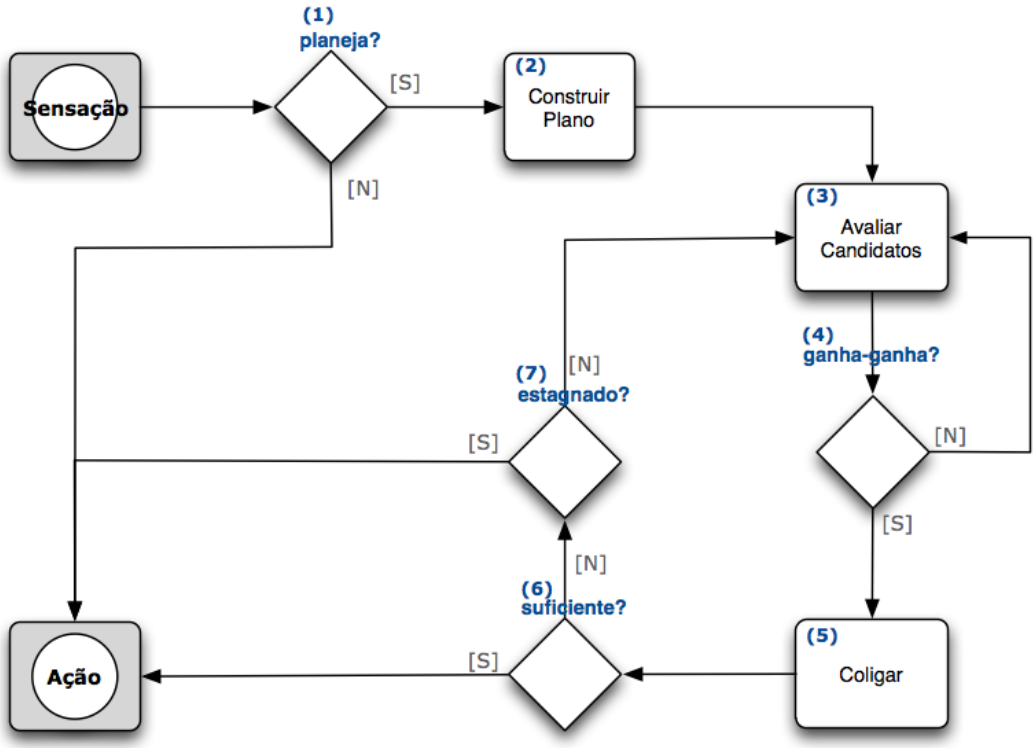


Figura 9 - Planejamento Cego com Avaliação Iterativa

Na Figura 9 temos o comportamento de avaliação iterativa de candidatos, que consiste na implementação completa da heurística ganha-ganha. Neste caso o agente executa os ciclos de avaliação iterativamente até que não identifique mais aumento no bem-estar social, ou sua demanda de recursos seja completamente atendida.

O fluxo inicia-se da mesma forma que no caso do *planejamento e avaliação em uma passagem*, ou seja, construindo-se o plano de ação sem interferência das funções coalizionais. A diferença daquele comportamento se dá no ciclo de incremento do bem-estar social, representado pelas fases $(3) \rightarrow (4) \rightarrow (5) \rightarrow (6) \rightarrow (7) \rightarrow (3)$ do fluxo. Esse ciclo avalia iterativamente novas coalizões até que não haja mais melhora no bem-estar social do ambiente ou todas as necessidades estejam satisfeitas.

C. Planejamento e Avaliação Iterativa

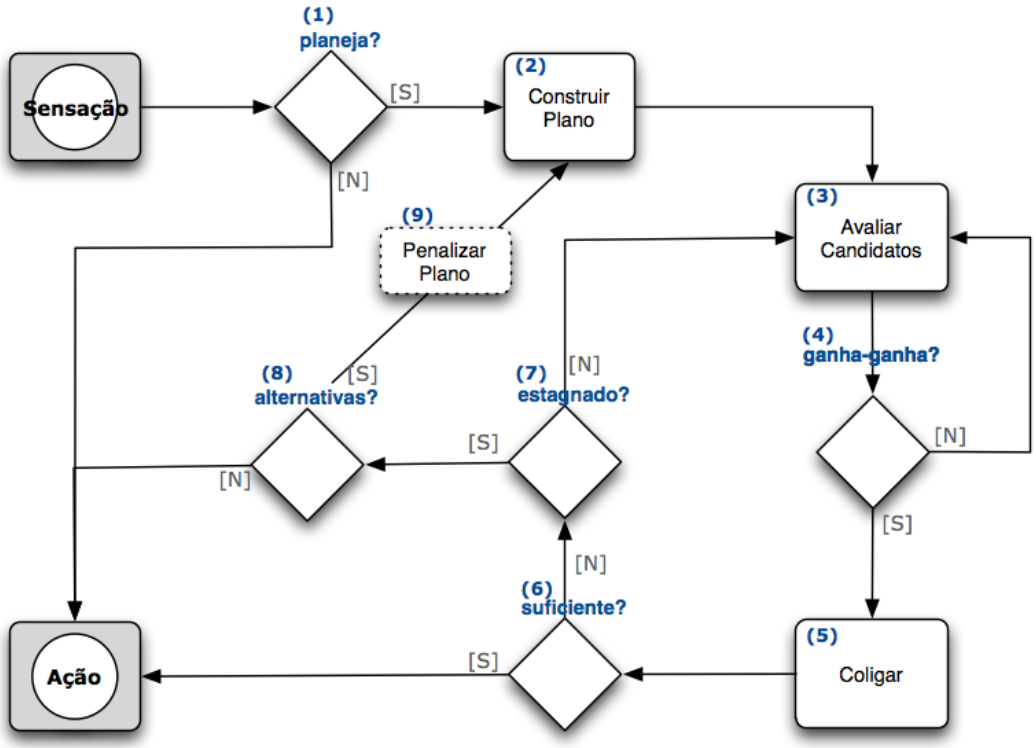


Figura 10 - Replanejamento com Avaliação Iterativa

Na Figura 10 temos o comportamento com planejamento e avaliação iterativos. Esse comportamento inclui, além da implementação da *heurística ganha-ganha*, a retroalimentação entre as funções coalizionais e decisórias.

O ciclo $(2) \rightarrow (3) \rightarrow (4) \rightarrow (5) \rightarrow (6) \rightarrow (7) \rightarrow (8) \rightarrow (9) \rightarrow (2)$ do fluxo representa o planejamento, avaliação coalizional e replanejamento. Esse ciclo maior será utilizado na simulação 3 e é o comportamento completo do agente, incluindo formação de coalizões e replanejamento. O replanejamento ocorre quando o teste **8** é positivo, o que indica que existem alternativas ainda válidas para alcançar os objetivos do agente.

A retroalimentação possibilita o replanejamento das ações quando o agente identifica que não há possibilidade de realização completa do plano de ação corrente. Essas situações são definidas pela regra de *estagnação*, que avalia que não há mais melhora no bem-estar social do ambiente por λ iterações, sendo λ um parâmetro da simulação. Esse teste pode ser evoluído para identificar outras situações, como variações no ambiente ou mudanças de prioridade.

Ciclo	Caminho	Descrição
Avaliação	$(3) \rightarrow (4) \rightarrow (3)$	Quando o agente não contém um plano válido ou o ambiente sofreu modificação relevante
Coalizional	$(3) \rightarrow (4) \rightarrow (5) \rightarrow (6) \rightarrow (7) \rightarrow (3)$	Quando o agente detém os recursos necessários para execução do plano
Replanejamento	$(2) \rightarrow (3) \rightarrow (4) \rightarrow (5) \rightarrow (6) \rightarrow (7) \rightarrow (8) \rightarrow (9) \rightarrow (2)$	Quando não há melhora no bem-estar

Tabela 3 - Ciclos do agente coalizional

A Tabela 3 sumariza todos os ciclos do agente coalizional.

5.1.4 Planejamento

O planejamento é executado através do algoritmo A^*, definido na Seção 3.2, que constrói o caminho mais curto entre a célula origem e a célula objetivo do agente, considerando uma função $c(x, y) \in \mathbb{R}$ de custo. Cada agente contém uma função custo própria, com valor inicial 1, que varia dependendo do *comportamento do agente* e dos recursos disponíveis para o agente.

A função $c(x, y)$ é utilizada para direcionar o algoritmo A^* que considera esse valor na composição da função $f(x, y) = c(x, y) + g(x, y) + h(x, y)$.

Apenas o **comportamento 3** altera o valor de $c(x, y)$ para indicar que o plano corrente não pode ser realizado, e assim possibilitar a seleção de um novo caminho quando for executado o replanejamento. Esse procedimento está representado pela fase 9 da Figura 10.

5.1.5 Coalizão

A coalizão, no contexto do simulador [16], contém os dados da coalizão social (que agentes participam da coalizão) e os procedimentos de cálculo da valor de Shapley e alocação de recursos. As *Coalizões* podem ser **temporárias**, quando montadas apenas para cálculo da utilidade na fase de *avaliação de candidatos*, ou **efetivas**, quando passam pela fase de *seleção de candidatos*. As *Coalizões* são independentes dos agentes e fazem parte do ambiente.

[16] Aqui estamos descrevendo um elemento da arquitetura. Esse elemento está relacionado à *coalizão* no sentido de estrutura organizacional, no entanto, contém outras responsabilidades associadas ao projeto do sistema.

5.1.6 Responsabilidades dos Elementos do Ambiente

Nessa Seção discriminaremos as responsabilidades de cada elemento do ambiente em relação à implementação da *heurística ganha-ganha*.

Elemento	**Responsabilidade**	**Colaboração**
Ambiente	Armazenar o posicionamento dos agentes. Executar as ações dos agentes garantindo as normas.	Agentes para gerar as ações
Agente		
Planejamento	Monta sequência de ações para alcançar a célula objetivo.	Tradução
Tradução	Transforma um plano em recursos necessários (rc_{Π}).	
Avaliação social	Mensura a utilidade de coalizão com outros agentes, montando coalizões candidatas. A utilidade de uma coalizão é calculada a partir da alocação dos recursos da coalizão.	Coalizão para executa a alocação dos recursos.
Seleção de candidatos	Selecionar a melhor coalizão montada na avaliação social. A melhor coalizão é aquela que tem a maior *utilidade* para o agente.	
Coalizão	Alocar os recursos das coalizões para os membros, seguindo a ordem definida pelo *valor de Shapley*.	

Tabela 4 - Responsabilidades dos elementos do ambiente

5.2 SOFTWARE DE SIMULAÇÃO

Para a avaliação da *heurística* em um ambiente multiagente, implementamos um simulador de mundo em grid, segundo definido na Seção de planejamento (sec. 3.2). Nessa Seção serão expostas as funcionalidades do simulador, bem como o seu projeto.

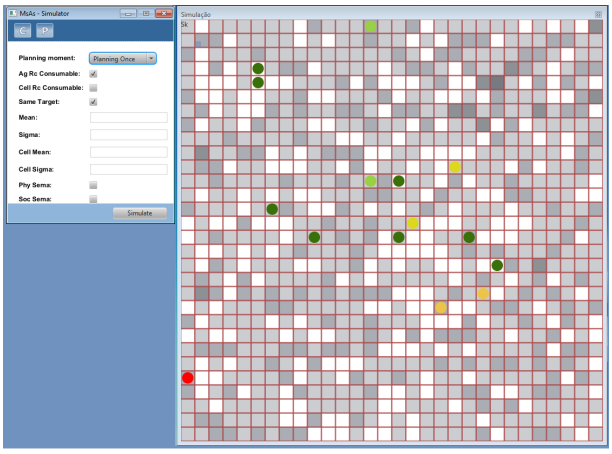


Figura 11 - Workbench do simulador

A Figura 11 apresenta o *workbench* do simulador. Na janela da esquerda temos os parâmetros de configuração da simulação, e na direita a representação do mundo em grid. Os parâmetros são aqueles que configuram o comportamento do agente e a distribuição de recursos dos agentes e do ambiente.

- Comportamento do agente: um dos 3 comportamentos apresentados.
- Recursos dos agentes consumíveis: define se os recursos dos agentes são consumidos após utilizados para ocupar uma célula.
- Objetivo comum: define se cada agente terá uma célula objetivo selecionada randomicamente, ou se todos terão a célula central como objetivo.
- Média da distribuição de recursos: a média (μ) da curva normal de distribuição de recursos entre os agentes.
- Sigma da distribuição de recursos: o *sigma* (σ) da curva normal de distribuição de recursos entre os agentes.

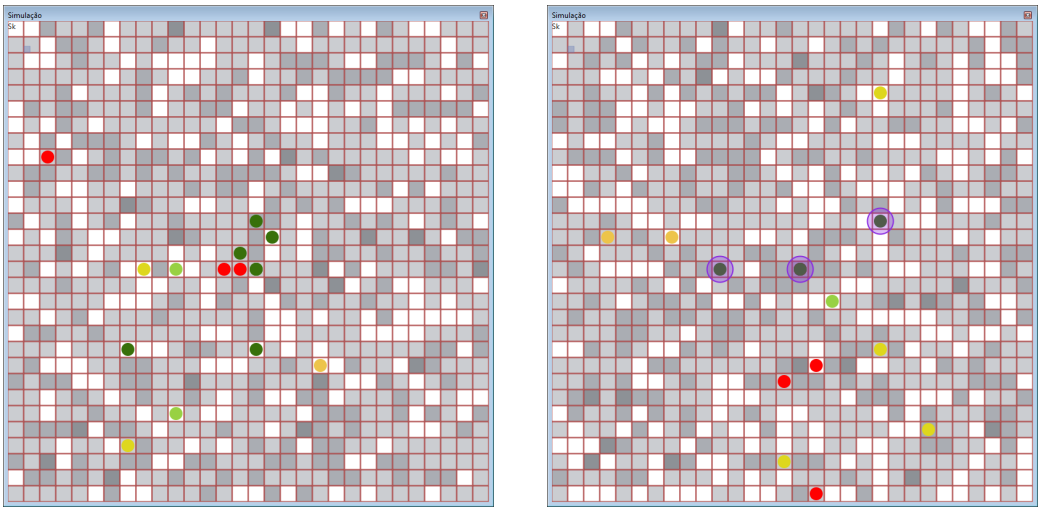
- Escala da distribuição de recursos do ambiente: percentual da média e sigma da curva normal de distribuição de recursos utilizado para distribuir os recursos demandados por cada célula.

- Semáforo físico: recurso de acompanhamento da simulação que informa que a simulação deve parar toda vez que um agente estiver impedido de executar movimento.

- Semáforo social: recurso de acompanhamento da simulação que informa que a simulação deve parar toda vez que um agente tiver executado avaliações de candidatos.

A visualização do mundo representa os agentes com círculos coloridos. A cor do agente representa o seu índice de satisfação, variando do vermelho (menos satisfeito) ao verde escuro (completamente satisfeito).

No *workbench* do simulador temos, ainda, dois botões que ativam os recursos de visualização de planos e coalizões. A Figura 12 -b representa o workbench do simulador com a função de visualização das coalizões ativadas (circulos roxos).



(a) *Execução* **(b)** *Coalizão*

Figura 12 - Mundo em grid do simulador

5.2.1 Projeto do Simulador

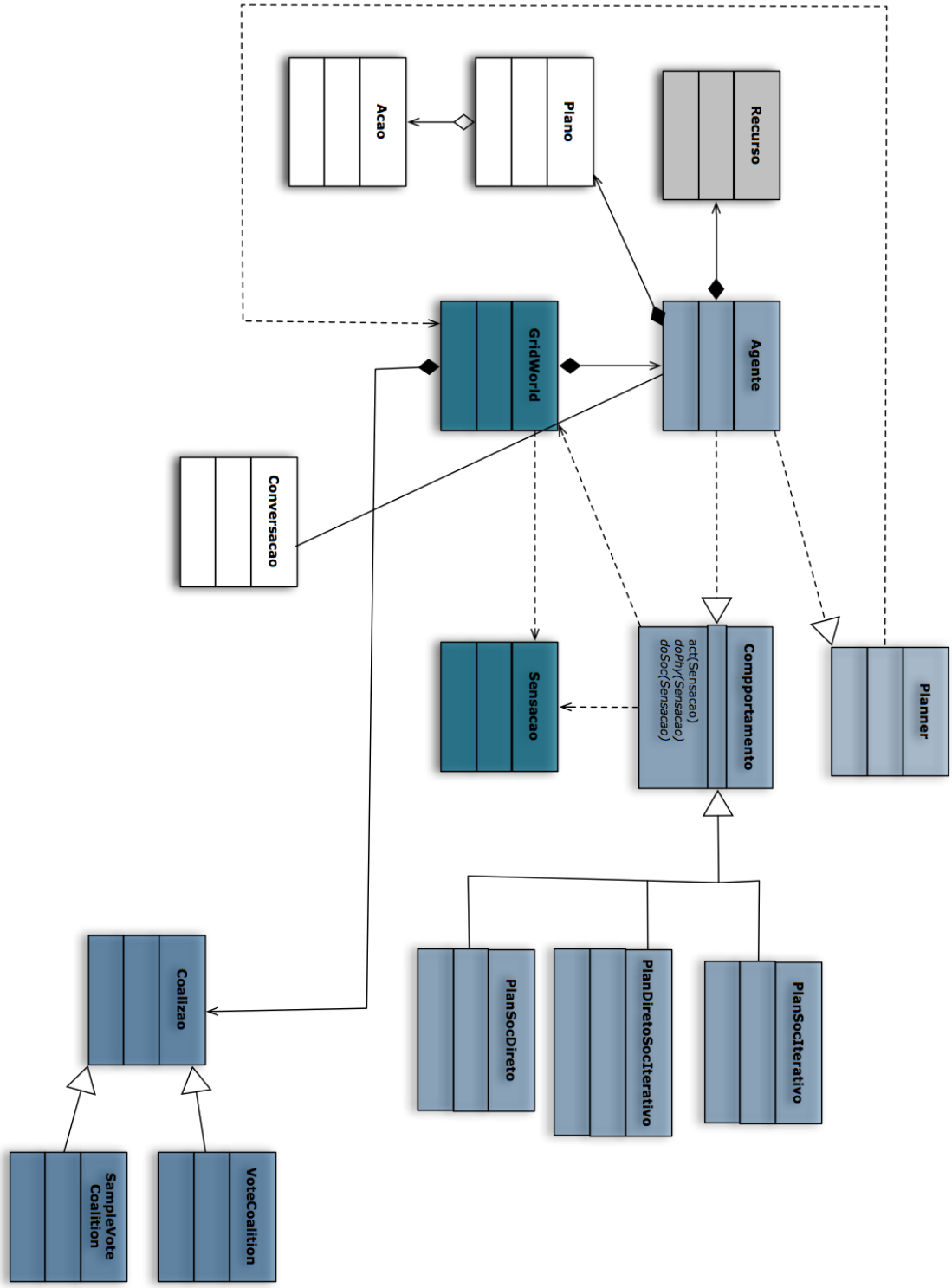


Figura 13 - Diagrama de classes do simulador

Os principais elementos do projeto são agrupados em 3 categorias:

1. **Elementos de projeto do ambiente:** Classes e traits referentes à implementação do ambiente, entre elas, o mundo grid (GridWorld), células do grid (GridCell) e sensação do ambiente (Sensacao). As responsabilidades dos elementos:

   - GridWorld: matriz de células e índices para acesso aos dados contidos nas células. Esses índices são: localização do agente ($Ag \Rightarrow Cell$), recursos da célula ($Rc \Rightarrow Cell$).
   - GridCell: cada célula contém uma lista com os agentes ali localizados, o recurso necessário para sua ocupação e a sua coordenada (x, y).
   - Sensacao: dado bruto que representa a visão do agente referente aos recursos de cada célula, coalizões montadas e seus membros, e agentes vizinhos.

     Tipicamente o agente recebe a sensação bruta, que é o estímulo externo do ambiente, e a transforma em uma percepção interna, mas no nosso modelo simplificado a própria sensação representa a percepção do agente.

2. **Elementos de projeto do agente:** Classes e traits utilizados na implementação do agente, suas funções de decisão e sociais. São esses elementos a classe Agente, o planejador (Planner) e o controlador de comportamento (Comportamento e suas subclasses).

   - Agente: representa o agente propriamente dito. O agente armazena os recursos próprios e demandados pelo plano. O agente implementa o cálculo da função utilidade.
   - Planner: implementa o algoritmo A^*, recebendo o ponto de origem, o ponto de objetivo e uma função de custo ($(x, y) \rightarrow \mathbb{R}$).
   - Comportamento: controle de fluxo entre o planejamento, e montagem de coalizão.

3. **Elementos de projeto da coalizão:** Classes e traits referentes à implementação das coalizões, contendo o cálculo do valor de Shapley, e algoritmo de alocação de recursos na coalizão. As subclasses de coalizão implementam o algoritmo aproximado ou por permutação.

5.2.2 Implementação do Simulador

A implementação do simulador foi efetuada na linguagem *Scala* (Odersky et al., 2007). *Scala* é uma linguagem de programação de propósito geral, fortemente tipada, que integra elementos de linguagens funcionais e orientadas à objetos (Odersky & al., 2006). O compilador *Scala* gera *bytecode* para máquinas virtuais java (versão 1.5 ou superior), o que possibilita interoperabilidade completa da linguagem com bibliotecas e *frameworks* escritos em java. Entre as principais características da linguagem, vale ressaltar:

- **Scala como linguagem funcional:** *Scala* é funcional no sentido de que toda função é um *valor*. A linguagem implementa funções como elementos de primeira ordem, podendo ser instanciadas, atribuídas, copiadas e criadas dinamicamente. Outra característica de linguagens funcionais é a facilidade de manipulação de listas, com métodos de transformação, pesquisa e mapeamento de listas nativos da linguagem. Funções podem ser aplicadas em listas ou elementos de listas com agilidade.

- **Scala como linguagem orientada à objetos:** *Scala* é uma linguagem orientada a objetos pura, no sentido de que todo valor é um objeto. Tipos e comportamentos de objetos são descritos por **classes** e **traits**. A linguagem permite extensão de classes e implementa um mecanismo de composição através de *mixins*. Mixins são uma alternativa à extensão pois possibilitam a introdução de comportamento dinamicamente e a implementação de herança múltipla.

- **Scala como linguagem extensível:** *Scala* foi concebida para facilitar a criação de linguagens de domínio específicas através de três elementos:

  1. Qualquer método pode ser utilizado de forma infixa ou sufixa.
  2. Parâmetros podem ser declarados opcionalmente.
  3. Implementação de *curring*: métodos podem ser chamados com uma lista de parâmetros. Quando um elemento da lista não está definido, é criada uma nova função que recebe os parâmetros ausentes como atributo.
  4. Parênteses são opcionais.

Controle do Comportamento

O controle do comportamento é responsabilidade da classe **Comportamento** e suas especializações **PlanSocDireto**, **PlanDiretoSocIterativo**, e **PlanSocIterativo**, representando, respectivamente, os comportamentos **A**, **B** e **C**. As classes estão apresentadas nas listagens 1, 2 e 3, a seguir.

A estratégia utilizada para a implementação é a denominada *template method*, que consiste em um método base (método *act*) que delega a execução para outros métodos (*doActPhy* e *doActSoc*), representando as ações físicas e sociais. Cada comportamento específico implementa variações sobre os métodos *templates*, e, eventualmente, pode mudar o comportamento de controle sobrescrevendo o método base.

Utilizamos um recursos específico da linguagem *Scala* que é chamado *traits*, que consiste em um mecanismo para implementação de herança múltipla e extensão dinâmica (em *runtime*).

5.3 SIMULAÇÕES E RESULTADOS

Nessa Seção apresentaremos os resultados e análises sobre simulações realizadas no ambiente em grade descrito. A eficiência do algoritmo de formação de coalizões foi analisada na Seção resultados do Capítulo 4. Nesta Seção, por outro lado, buscamos as informações para entender a relação entre planejamento e formação de coalizões, e também as características próprias da interação de agentes coalizionais em um ambiente dinâmico.

Montamos 3 conjuntos de simulações, referentes aos comportamentos **A, B** e **C**, definidos na Seção 5.1.3, respectivamente, *planejamento e avaliação em uma passagem*, *planejamento em uma passagem e avaliação iterativa* e, por último, *planejamento e avaliações iterativas*. A simulação ocorreu no mundo em grade de tamanho fixo (raio $R = 30$) e quantidade fixa de agentes (15 agentes). Para cada elemento do conjunto de simulações, além do comportamento dos agentes, variaram-se os parâmetros da distribuição de recursos. A demanda de recursos de um agente depende do seu plano de ação, que, por sua vez, depende do seu posicionamento inicial e da sua célula objetivo. Os posicionamentos iniciais foram dados randomicamente, e adotamos uma célula objetivo central (*target(x,y)* $= (15, 15)$) comum para todos os agentes.

Detalharemos seguir os parâmetros de configuração.

5.3.1 Configuração do ambiente

Foram executados três (3) conjuntos de simulações, referentes aos comportamentos dos agentes. Cada conjunto é formado por cinco (5) configurações da **carência de recursos no ambiente (**$lack$**)**, e cada valor de $lack$ é utilizado em dez (10) simulações, totalizando um total de 150 simulações executadas para comparação dos valores medidos.

Dessa forma, cada simulação é definida por uma instância dos parâmetros **comportamento**, **lack**, **identificador**, onde,

$$\begin{aligned} comportamento &\in \{A, B, C\} \\ lack &\in \{0.05, 0.1, 0.15, 0.2, 0.25\} \end{aligned}$$

Além desses parâmetros, em cada simulação variou-se o posicionamento inicial dos agentes e a distribuição de recursos próprios dos agentes e recursos demandados nas células do mundo. A distribuição de recursos obedeçe uma distribuição normal, respectivamente, $\mathcal{N}(\mu, \sigma)$ e $\mathcal{N}(\mu \times lack, \sigma \times lack)$, para os recursos próprios e demandados, onde, $\mu = R/1.5 = 20$, e $\sigma = \mu/4 = 5$.

Os recursos próprios e de ocupação são dados por distribuições gaussianas, respectivamente, $\mathcal{N}(\mu, \sigma)$ e $\mathcal{N}(\mu \times lack, \sigma \times lack)$, , onde, a média $\mu = R/1.5 = 20$, e o desvio padrão $\sigma = \mu/4 = 5$.

O parâmetro *lack* é utilizado para definir a distribuição de recursos de ocupação. Assim, quanto maior a *carência*, maior é a quantidade de recursos médio necessário para ocupar as posições do mundo. De acordo com os valores da carência ($lack \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$) e da média da gaussiana de recursos próprios ($\mu = 20$), temos as 5 médias de recursos necessários para ocupação por célula ($\{1.0, 1.5, 2.0, 2.5, 3.0\}$). Utilizaremos esses 5 valores de média de ocupação para identificar os 5 cenários de simulação por comportamento.

5.3.2 Variáveis Monitoradas

O comportamento da simulação foi acompanhado através de um conjunto de variáveis de controle, divididas em dois grupos.

1. Variáveis das ações do comportamento: medem a quantidade de execuções de cada etapa do comportamento. São essas:

- Quantidade média de avaliações por agente.
- Quantidade média de coligações por agente.
- Quantidade média de planejamentos por agente.

2. Variáveis de medição dos benefícios: medem os benefícios da formação de coalizões. São essas:

   - Bem-estar social.
   - Quantidade total de recursos demandantes.
   - Quantidade final de coalizões.

5.3.3 Gráfico da Simulação

Acompanhamos cada comportamento através de um conjunto de simulações, apresentado na próxima Seção. Cada comportamento gerou um gráfico de simulação (figuras 14, 15 e 14), que incorpora os resultados de todas as simulações executadas para aquele comportamento. Cada gráfico é organizado da seguinte forma:

- **Eixo horizontal**: identificação da instância da simulação. São 50 simulações para cada comportamento, divididos em 5 áreas referentes as médias de recursos de ocupação.
- **Eixo vertical esquerdo**: Valores quantitativos médios da quantidade de avaliações, ações de coligações, e de coalizões.
  - Quantidade de avaliações, referenciado no eixo esquerdo.
  - Quantidade de coligações, referenciado no eixo esquerdo.
  - Quantidade final de coalizões montadas, referenciado no eixo esquerdo.
- **Eixo vertical direito**: Valor total de carência de recursos no ambiente.

5.3.4 Simulações

Simulação 1 - Planejamento em uma passagem

Nesse conjunto de simulações, cada agente executa a sequência planejamento, e montagem de coalizões sequencialmente (comportamento A, Figura 8). Como não há retroalimentação entre as etapas, o ciclo *planejamento, avaliação, coligação* é executado apenas uma vez, ou seja, assim que o plano é montado não é mais alterado. As *avaliações dos candidatos* são executadas apenas uma vez por agente, e no máximo uma ação de coalizão é executada, caso um candidato passe no *teste ganha-ganha*.

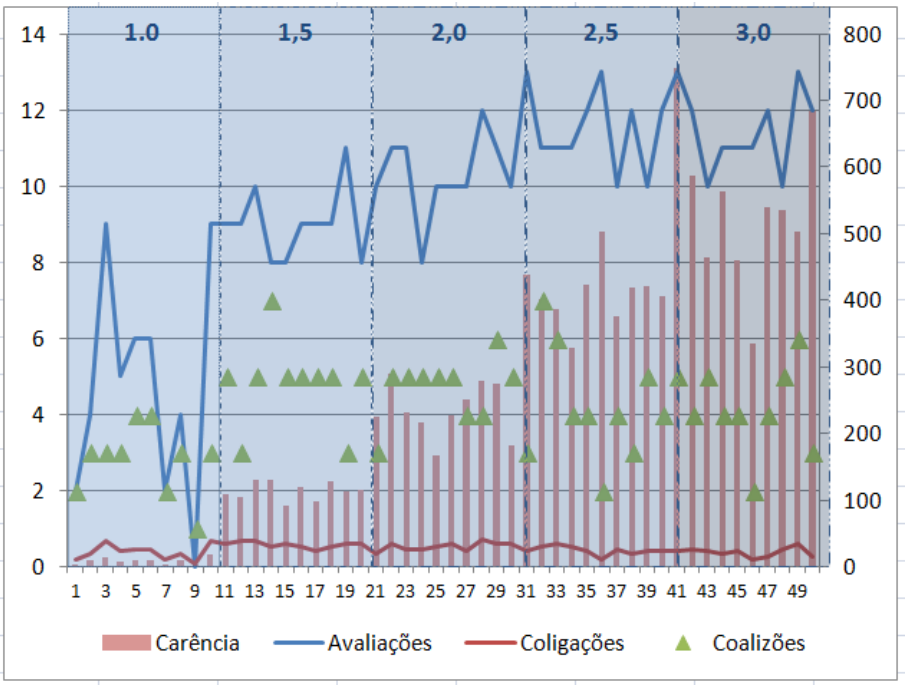


Figura 14 - Quantidade de avaliações e bem-estar por carência de recursos

Na Figura 14, temos o gráfico de execução das simulações referente ao **comportamento A**. O gráfico é seccionado em 5 valores localizados no eixo horizontal superior, que indicam a média da distribuição de quantidade de recursos para ocupação de posições. O gráfico de barras indica a indisponibilidade de recursos do ambiente, que é resultado da diferença entre a a soma dos recursos demandados no ambiente e da soma de todas as dotações iniciais dos agentes. Percebe-se que a medida que a demanda média de recursos aumenta, a indisponibilidade aumenta também.

A quantidade de avaliações é mostrada através do gráfico de linhas contínuas superior. A sua escala se encontra no eixo esquerdo, e os seus valores indicam a quantidade média de avaliações por agente. Percebe-se, da mesma forma que a indisponibilidade de recursos, que a quantidade média de avaliações aumenta com o aumento da demanda de recursos para ocupar posições no mundo. Esse resultado é coerente com intuição de que quanto maior a escassez de recursos, maior deve ser o esforço empreendido na montagem de coalizões.

As ações de coligação, como colocado anteriormente, devem ser no máximo 1 por agente, dessa forma a seu valor se mantém constante durante a execução. O gráfico também mostra a quantidade de coligações formadas, representadas pelos pontos triangulares. Podemos perceber a tendência de formação de 3 a 5 coalizões na sociedade.

Simulação 2 - Planejamento em uma passagem e avaliação iterativa

Nesse conjunto de simulações está presente o ciclo de formação de coalizões completo (comportamento B, Figura 9). segundo definido na ***heurística ganha-ganha***. Essa configuração equivale à implementação da *heurística* a partir de uma plano de ação estático, ou seja, não há retroalimentação entre a interação social e a fase de planejamento.

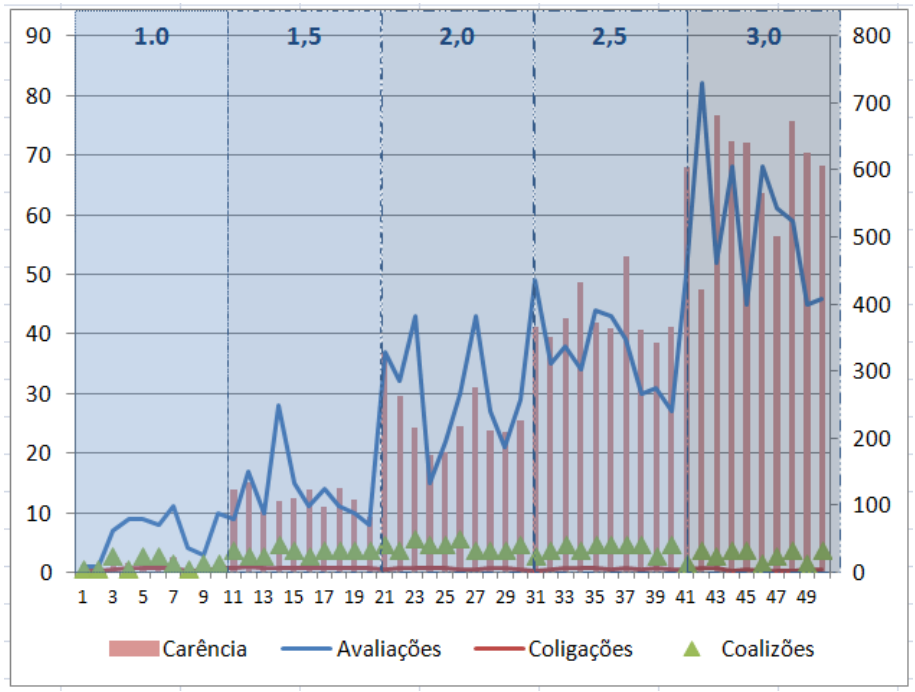


Figura 15 - Quantidade de avaliações e bem-estar por carência de recursos

Na Figura 15, temos o gráfico de execução das simulações referente ao **comportamento B**. A organização do gráfico é a mesma do gráfico anterior, sendo secçionado em 5 partes referentes é demanda média de recursos para ocupação de posições. O aumento da demanda implica no aumento da escassez de recursos.

O comportamento B executa o ciclo social, ou seja, o processo de formação de coalizões é iterativo, havendo um ciclo entre as ações de avaliação e coalizão. Dessa forma, cada agente percorre os seus parceiros candidatos avaliando os benefícios de associações e tenta coligar-se com o melhor. Após todos haverem montado suas coalizões, o ciclo de avaliação ocorre novamente até que todo agente esteja satisfeito ou o bem-estar social se estabilize. Sendo assim, diferentemente do comportamento A, pode ocorrer mais de uma ação coalizional por

agente.

O gráfico 15 mostra que a quantidade de avaliações é significativamente maior que no gráfico 14. Esse resultado é coerente com a existência do ciclo coalizional não presente no comportamento anterior. Dessa forma, a quantidade de esforço realizada é significativamente maior, no entanto, esse esforço não se traduziu no aumento da quantidade de ações coalizionais nem na quantidade de coalizões. Veremos na próxima Seção que os valores de bem-estar também foram pouco alterados entre os comportamentos.

Simulação 3 - Planejamento e avaliação iterativa

Nesse conjunto de simulações, é executado o fluxo de comportamento que considera o replanejamento (comportamento C, Figura 10). Além do ciclo de formação de coalizões, descrito na simulação 2, os agentes executam o replanejamento caso identifiquem a situação de estagnação do bem-estar. Isso implica na retroalimentação entre o ciclo coalizional e o de planejamento.

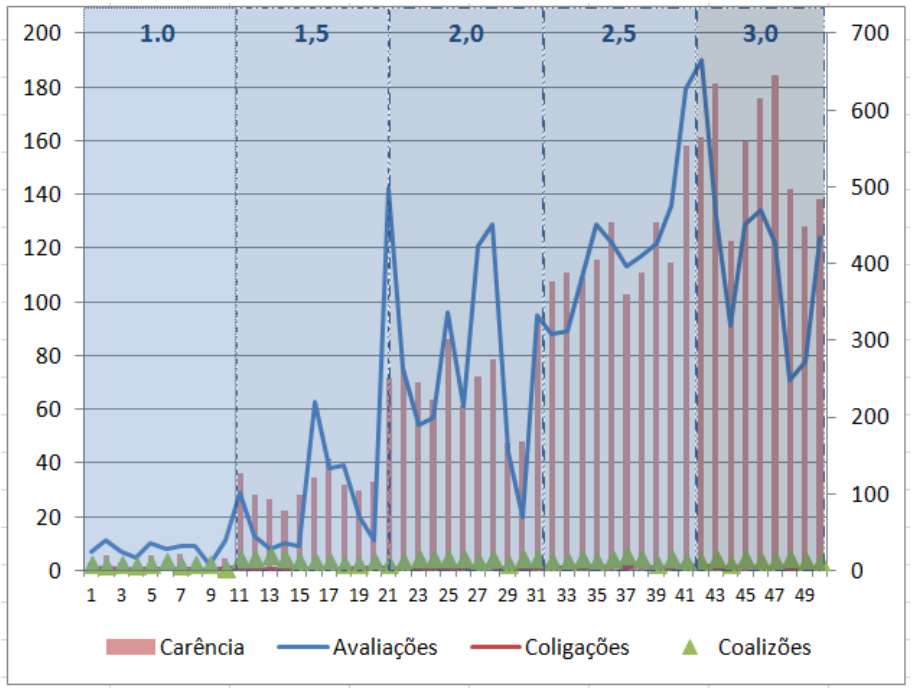


Figura 16 - Quantidade de avaliações e bem-estar por carência de recursos

No gráfico 16 podemos perceber novamente o aumento da quantidade de avaliações comparativamente aos outros comportamentos. Esse resultado é similar ao anterior, no sentido de que há aumento no esforço dos agentes, refletido no aumento da quantidade de avaliações, no entanto não há equivalência no aumento de ações de coligamentos ou quantidade de coalizões.

5.3.5 Resultados

Dados os 3 conjuntos de simulações, descritos na Seção 5.3.4, efetuaremos a análise comparativa da performance de cada comportamento.

Primeiramente, observa-se o comportamento do bem-estar social nos 3 conjuntos de simulações. O gráfico 17 mostra o bem-estar inicial e final médio para o conjunto de simulações refente a cada valor de carência de recursos. Dessa forma, no eixo X temos o valor médio da demanda de recursos por célula (a distribuição de recursos demandados por célula se dá pela distribuição $\mathcal{N}(\mu \times kscale, \sigma \times kscale)$), que representa o primeiro parâmetro da *distribuição normal*. Temos dois conjuntos de curvas: as pontilhadas, que representam o valor inicial do bem-estar (antes de qualquer coligação), e as sólidas, que representam os valores finais de bem-estar.

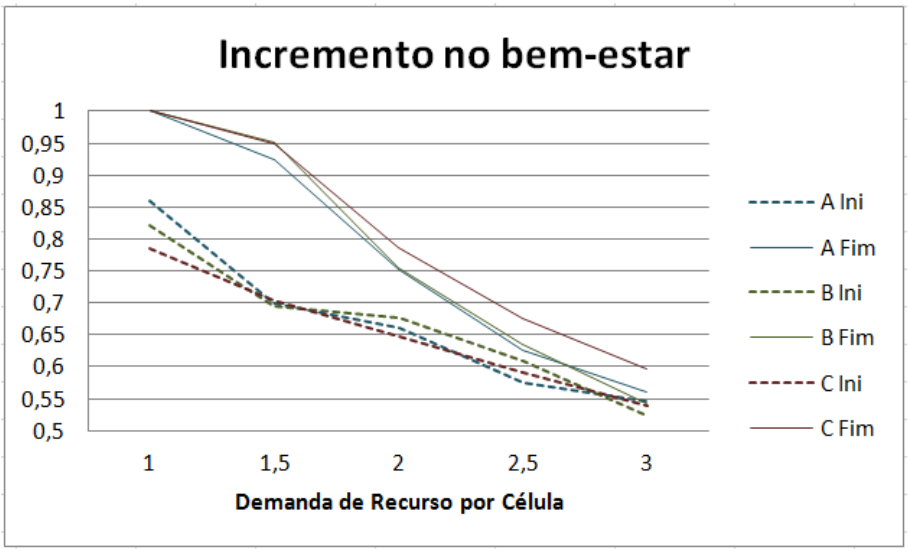


Figura 17 - Quantidade de avaliações e bem-estar por carência de recursos

No gráfico 17 de **incremento do bem-estar social**, percebe-se que há o aumento do bem-estar social quando há a formação de coalizões. Além disso, esse incremento é tão maior, quanto menor a demanda de recursos por célula, chegando a uma situação em que não há mais incremento no bem-estar (o valor 3 no eixo X do gráfico). Isso está de acordo com os resultados do teste da *heurística ganha-ganha* em que uma das conclusões foi que o algoritmo é eficiente na presença de disponibilidade de recursos. O diminuição da disponibilidade de recursos age como um desestímulo à formação de coalizões. O mesmo efeito é percebido no

valor final do bem-estar social, que chega ao máximo com demanda média de 1 recurso por célula, e vai decaindo gradativamente.

O gráfico 18 mede o esforço executado para alcançar-se o bem-estar final. Nesse gráfico, compara-se a quantidade de avaliações com o valor de bem-estar por configuração de comportamento. Neste gráfico, a cor identifica o comportamento (A, B ou C), as linhas a evolução da quantidade de avaliações e as barras o bem-estar alcançado. O eixo X identifica a demanda média de recursos por célula, enquanto o eixo Y *à esquerda* a quantidade de avaliações e o eixo Y *à direita*, o bem-estar.

Analisando a evolução da quantidade de avaliações, percebe-se que esta quantidade aumenta de acordo com o comportamento, o que é esperado, uma vez que os comportamentos mais sofisticados incluem um maior números de ciclos com avaliação. Da mesma forma, é esperado que a quantidade de avaliações aumente a medida que aumenta a demanda por recursos, uma vez que mesmo recursos estarão disponíveis para compartilhamento, aumentando, por consequência, o escopo da busca.

Finalmente, comparando-se a quantidade de avaliações com o bem-estar percebe-se um fato peculiar; o aumento de bem-estar não acompanha o aumento da quantidade de avaliações. Essa fato é bem característico nos gráficos relativos ao comportamento C, que apresenta um acentuado aumento de avaliações a medida que a demanda aumenta, sem retorno no aumento do bem-estar. O bem-estar do comportamento C não apresenta diferença significativa dos outros comportamentos, no entanto o esforço realizado é bastante maior a medida que aumenta a demanda. Inclusive os melhores valores de bem-estar são aqueles que tem a quantidade de avaliações mais próxima entre os comportamentos, o que nos faz inferir que os melhores resultados são obtidos no primeiro cilclo de coligações.

O gráfico 19 aponta para uma mesma conclusão. A quantidade de ações coalizionais permanece muito próxima entre os 3 comportamentos nos casos de melhor desempenho do bem-estar social. Há um descolamento entre os valores do comportamento C e do resto dos comportamentos a partir da média de demanda 1.5. A explicação é que com o aumento da carência de recursos, novos planos são gerados o que, no entanto, não produz coalizões mais eficientes.

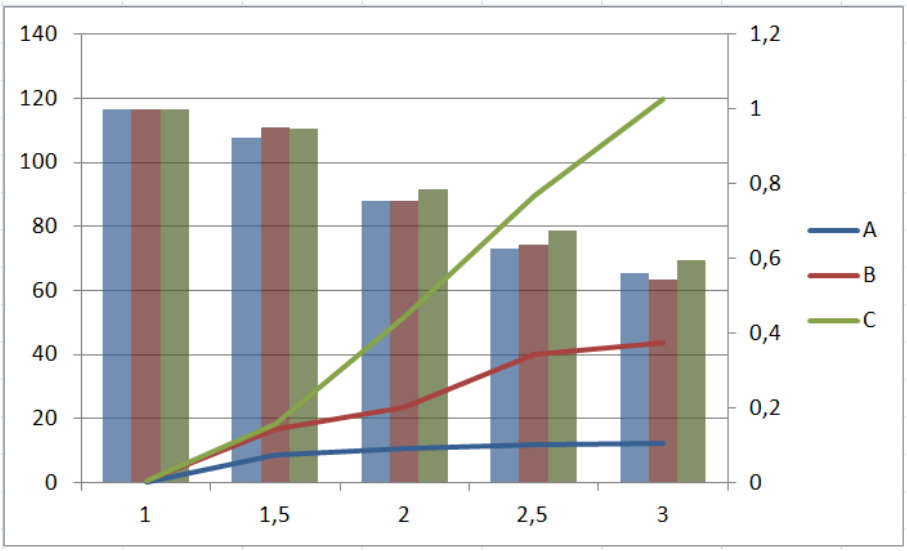


Figura 18 - Quantidade de avaliações e bem-estar por carência de recursos. O gráfico de linha indica a quantidade de avaliações, enquanto o gráfico de barras representa o bem-estar.

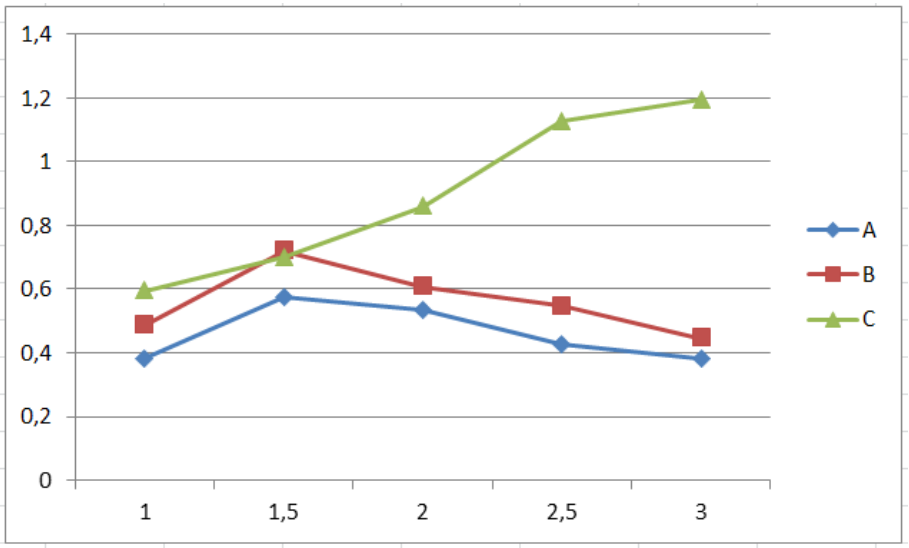


Figura 19 - Quantidade de ações coligar por carência de recursos

O resultado de que o comportamento C não traz ganhos efetivos aos agentes é contra-intuitivo. O esperado seria que o fato de executarmos um replanejamento após o agente ter tentado montar coalizões eficazes para um dado plano, trouxesse ganhos,

Uma hipótese para essa ocorrência é que a medida que as coligações são efetuadas os agentes vão atingindo seus objetivos e ficando indisponíveis para outros acordos. Com a diminuição da quantidade de candidatos, o número de coalizões vantajosas vai diminuindo,

tornando mais improvável a evolução dos ganhos de um agente por replanejamento. Essa hipótese é corroborada também pelo fato de que as coalizões montadas antes, ou seja, com menos avaliações, trazem maiores ganhos. Fato pontuado anteriormente.

Essas hipóteses apontam para necessidade de novos estudos na relação entre planejamento e formação de coalizões, devendo ser objeto de trabalhos futuros.

6 CONCLUSÃO

Formação de coalizões é um tópico relevante em sistemas multiagentes quando estas são as estruturas organizacionais adotadas para controlar a cooperação entre os seus membros. Essa abordagem de cooperação tem recebido interesse expressivo da comunidade de pesquisa em SMAs pela base teórica fornecida pela área de teoria dos jogos e pela generalidade do conceito de coalizão. Nesse sentido, a alocação de recursos como domínio de aplicações complementa a formação de coalizões, pois ambos interpretam o sistema como uma sociedade de agentes que podem ser modelados através de ferramentas da microeconomia. Mais que isso, ambos compõem *frameworks* de modelagem de SMAs que tem um grande poder expressivo quando utilizados em conjunto.

As coalizões são estudadas pela teoria dos jogos no campo dos jogos cooperativos, que enfatiza as questões relativas aos conceitos de solução, buscando a análise formal das propriedades dos jogos. O tema da formação das coalizões muitas vezes não é priorizado nesse campo de estudos, tendo sido apropriado pela área de SMAs. Nesse contexto, uma série de algoritmos foram desenvolvidos para formação de coalizões, mas todos sofrendo de problemas de performance relativos à explosão combinatória sobre o espaço de busca.

Acreditamos que parte do problema está na busca de uma solução ótima para as estruturas coalizionais quando, muitas vezes, relaxando esse requisito podemos encontrar configurações de soluções não ótimas, viáveis para domínios específicos. Por isso heurísticas e algoritmos aproximativos são uma excelente opção para a formação de coalizões, e merecem atenção no campo de SMAs.

Nesse trabalho desenvolvemos um novo algoritmo para formação de coalizões que utiliza o compartilhamento de recursos como objetivo das coalizões. O algoritmo é uma heurística que tem como critério de associação a racionalidade individual e coletiva dos agentes, critério denominado *teste ganha-ganha*. Através dele demonstramos que o algoritmo é monotônico com relação ao incremento do bem-estar da sociedade, e converge para uma solução. Medimos isoladamente o algoritmo e identificamos empiricamente a sua eficiência na presença de recursos abundantes, ou seja, a heurística mostrou-se uma eficiente realocadora de recursos.

Identificamos que em cenários em que a demanda por recursos total é de até 50% da oferta total de recursos no sistema, com poucas iterações do algoritmo (10 iterações) conseguimos bem-estar de pelo menos 70%, o que é um ganho expressivo de bem-estar. A complexidade

do algoritmo é de $O(k^3n^4)$, o que ainda implica em um grande esforço computacional por iteração, no entanto, bastante menor que os algoritmos com complexidade exponencial. Esse fato é ilustrado nas medições efetuadas, em que foram utilizados cenários com 80 agentes, quantidade muito maior que a apresentada em outros algoritmos.

Outro elemento a ser considerado é a possibilidade de distribuição do processamento, o que incrementaria a performance das execuções. Esses resultados são a primeira avaliação do potencial de utilização do algoritmo, que aponta para a possibilidade de sua utilização em situações de montagem de coalizões em tempo real.

Construímos um simulador para testar a execução do algoritmo em um ambiente multiagentes. Esse simulador foi baseado em um ambiente de teste denominado (CT-Game) utilizado em avaliação de situações de negociação. Propusemos o conceito de agente coalizional, que tem separada a função de planejamento e montagem de coalizões. O ambiente possibilitou elaborar algumas avaliações entre as duas funções, e o trabalho aponta para a necessidade de continuidade no desenvolvimento de mecanismos de controle que possibilitem maior sinergia entre planejamento e montagem de coalizões.

Em especial, pudemos perceber que a replanejamento após a formação de coalizões não trouxe ganhos. Levantamos a hipótese de esse resultado, que contraria a intuição inicial, deve-se à falta de interesse de parte dos agentes em continuar a formar coalizões, diminuindo, assim, as possibilidades de ganho advindas de novas coalizões. Essa hipótese necessita de cenários mais elaborados para sua confirmação, uma das continuidades possíveis desse trabalho.

6.1 TRABALHOS FUTUROS

O trabalho executado pode ser expandido em várias frentes:

1. Elaboração de modelos mais ricos de modelagem de agentes baseados em recursos, buscando atender aplicações reais, e consequentes adaptações da heurística de formação de coalizões.

2. Evoluções na análise teórica da heurística, em especial, buscar adaptações que permitam garantir a convergência para um ótimo de Pareto, resultado já alcançado em negociações simples.

3. Evolução na interação entre planejamento e formação de coalizões.

Uma das possibilidade que esse campo abre é a criação de planejadores que incorporem a capacidade de formação de coalizão. Outra abordagem para a evolução da interação entre as duas funções seria a busca de modelos de aprendizado por reforço cujas funções custo sejam atualizadas pelo algoritmo de formação de coalizão.

4. Implementação da versão distribuída da heurística.

Como considerações finais, esse trabalhou buscou a integração de dois temas da área de sistemas multiagentes; a formação de coalizões e alocação de recursos, ambos relacionados a abordagem microeconômica para controle de sistemas multiagentes. Desenvolvemos nesse trabalho a *heurística ganha-ganha* que apresenta a interessante propriedade de ser monotônica no incremento do bem-estar social do sistema. Fizemos diversas medições da performance e, em especial, executamos o algoritmo em um ambiente teste, onde pudemos iniciar a avaliação dos problemas associados a autonomia e formação de coalizões. Finalmente, esse trabalho aponta para desenvolvimentos futuros nessa problemática: como trabalhar conjuntamente o planejamento individual e a formação de coalizões? Esse nos parece ser um ponto ainda em aberto na área de sistemas multiagentes e IA em geral.

REFERÊNCIAS BIBLIOGRÁFICAS

Apt, K. R. & Radzik, T. (2006). Stable partitions in coalitional games. *CoRR*. working paper disponível em http://arxiv.org/abs/cs/0605132.

Apt, K. R. & Witzel, A. (2009). A generic approach to coalition formation. *International Game Theory Review*, 11(03), 347–367.

Bachrach, Y., Markakis, E., Resnick, E., Procaccia, A. D., Rosenschein, J. S., & Saberi, A. (2010). Approximating power indices: theoretical and empirical analysis. *Autonomous Agents and Multi-Agent Systems*, 20, 105–122.

Bachrach, Y. & Rosenschein, J. S. (2008). Coalitional skill games. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '08 (pp. 1023–1030).: International Foundation for Autonomous Agents and Multiagent Systems.

Cara, F. & Netto, M. L. (2013). Sobre a representação de estruturas coalizionais como um dispositivo adaptativo - estudo preliminar. In *WTA 2013 - Workshop em Tecnologias Adaptativas*.

Chevaleyre, Y., Dunne, P. E., Endriss, U., Lang, J., Lemaître, M., Maudet, N., Padget, J., Phelps, S., Rodríguez-aguilar, J. A., & Sousa, P. (2006). Issues in multiagent resource allocation. *Informatica*, 30.

Chevaleyre, Y., Endriss, U., Estivie, S., & Maudet, N. (2005). Welfare engineering in practice: On the variety of multiagent resource allocation problems. In *Engineering Societies in the Agents World V* (pp. 335–347).: Springer-Verlag.

Chevaleyre, Y., Endriss, U., Estivie, S., & Maudet, N. (2008). Multiagent resource allocation in k-additive domains: Preference representation and complexity. *Annals of Operations Research*, 163(1), 49–62.

Chevaleyre, Y., Endriss, U., Estivie, S., Maudet, N., et al. (2004). Multiagent resource allocation with k-additive utility functions. In *Proc. DIMACS-LAMSADE Workshop on Computer Science and Decision Theory*, volume 3 (pp. 83–100).: Citeseer.

Dang, V., Dash, R., Rogers, A., & Jennings, N. (2006). Overlapping coalition formation for efficient data fusion in multi-sensor networks. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-06)*, volume 21.

Davidsson, P. (2001). Categories of artificial societies. In *Proceedings of the Second International Workshop on Engineering Societies in the Agents World II* (pp. 1–9). London, UK: Springer-Verlag.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271.

Endriss, U., Maudet, N., Sadri, F., & Toni, F. (2003). On optimal outcomes of negotiations over resources. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03 (pp. 177–184).: ACM.

Fatima, S., Wooldridge, M., & Jennings, N. (2008). A linear approximation method for the shapley value. *Artificial Intelligence*, 172(14), 1673–1699.

Fatima, S., Wooldridge, M., & Jennings, N. (2011). An approximation method for power indices for voting games. *Innovations in Agent-Based Complex Automated Negotiations*, (pp. 179–194).

Gal, Y., Grosz, B., Kraus, S., Pfeffer, A., & Shieber, S. (2005). Colored trails: a formalism for investigating decision-making in strategic environments. In *Proceedings of the 2005 IJCAI workshop on reasoning, representation, and learning in computer games* (pp. 25–30).

Gal, Y., Grosz, B., Kraus, S., Pfeffer, A., & Shieber, S. (2010). Agent decision-making in open-mixed networks. *Artificial Intelligence*.

Horling, B. & Lesser, V. (2004). A survey of multi-agent organizational paradigms. *Knowl. Eng. Rev.*, 19, 281–316.

Jain, K. & Vazirani, V. (2001). Applications of approximation algorithms to cooperative games. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing* (pp. 364–372).: ACM.

Kraus, S. (2001). Automated negotiation and decision making in multiagent environments. *Lecture notes in computer science*, (pp. 150–172).

LaValle, S. M. (2006). *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press. Available at http://planning.cs.uiuc.edu/.

Lehmann, D., Muller, R., & Sandholm, T. (2006). The winner determination problem. In P. Cramton, Y. Shoham, & R. Steinberg (Eds.), *Combinatorial Auctions* (pp. 297–317).: MIT Press.

Michalak, T., Sroka, J., Rahwan, T., Wooldridge, M., McBurney, P., & 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 (pp. 1007–1014).: International Foundation for Autonomous Agents and Multiagent Systems.

Nongaillard, A., Mathieu, P., & Everaere, P. (2010). Nash welfare allocation problems: Concrete issues. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 02*, WI-IAT '10 (pp. 32–39). Washington, DC, USA: IEEE Computer Society.

Norman, T., Preece, A., Chalmers, S., Jennings, N., Luck, M., Dang, V., Nguyen, T., Deora, V., Shao, J., Gray, W., et al. (2004). Agent-based formation of virtual organisations. *Knowledge-Based Systems*, 17(2), 103–111.

Odersky, M. & al. (2006). *An Overview of the Scala Programming Language*. Technical Report IC/2006/001, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.

Odersky, M., Spoon, L., & Venners, B. (2007). *Programming in Scala*. aritma.

Osborne, M. & Rubinstein, A. (1990). *Bargain and Markets*. The Academic Press.

Osborne, M. & Rubinstein, A. (1994). *A Course in Game Theory*. MIT Press.

Rahwan, T. & Jennings, N. R. (2005). Distributing coalitional value calculations among cooperative agents. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 1*, AAAI'05 (pp. 152–157).: AAAI Press.

Rahwan, T. & Jennings, N. R. (2008). An improved dynamic programming algorithm for coalition structure generation. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3*, AAMAS '08 (pp. 1417–1420).: AAAI Press.

Rahwan, T., Ramchurn, S., Jennings, N., & Giovannucci, A. (2009). An anytime algorithm for optimal coalition structure generation. *Journal of Artificial Intelligence Research*, 34(2), 521.

Rosenschein, J. & Zlotkin, G. (1994). *Rules of encounter: designing conventions for automated negotiation among computers*. the MIT Press.

Russell, S. J. & Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3rd edition.

Saad, W., Han, Z., Debbah, M., & Hjorungnes, A. (2009a). A distributed coalition formation framework for fair user cooperation in wireless networks. *Wireless Communications, IEEE Transactions on*, 8(9), 4580 –4593.

Saad, W., Han, Z., Debbah, M., Hjorungnes, A., & Basar, T. (2009b). Coalitional game theory for communication networks. *Signal Processing Magazine, IEEE*, 26(5), 77–97.

Sandholm, T., Larson, K., Andersson, M., Shehory, O., & Tohmé, F. (1999). Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1-2), 209 – 238.

Sandholm, T. W. (1998). Contract types for satisficing task allocation: I theoretical results. In *IAAAI Sprint Symposium: Satisficing Models* (pp. 68–75).

Service, T. C. & Adams, J. A. (2010). Approximate coalition structure generation. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Service, T. C. & Adams, J. A. (2011). Constant factor approximation algorithms for coalition structure generation. *Autonomous Agents and Multi-Agent Systems*, 23(1), 1–17.

Shapley, L. (1953). A value for n-person games. *Annals of Mathematics Studies*, 28(?), 307 – 317.

Shehory, O. & Kraus, S. (1998). Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101, 165–200.

Shehory, O., Sycara, K., & Jha, S. (1998). Multi-agent coordination through coalition formation. *Intelligent Agents IV Agent Theories, Architectures, and Languages*, (pp. 143–154).

Shoham, Y. & Leyton-Brown, K. (2009). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge, UK: Cambridge University Press.

Smith, R. & Davis, R. (1981). Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man and Cybernetics*, 11(1), 61–70.

Smith, R. G. (1980). The contract net protocol: high-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, 29(12), 1104–1113.

Tsvetovat, M., Sycara, K., Chen, Y., & Ying, J. (2001). Customer coalitions in electronic markets. In F. Dignum & U. Cortés (Eds.), *Agent-Mediated Electronic Commerce III*, volume 2003 of *Lecture Notes in Computer Science* (pp. 121–138). Springer Berlin Heidelberg.

Van De Vijsel, M. & Anderson, J. (2004). Coalition formation in multi-agent systems under real-world conditions. In *Proceedings of the AAAI-04 Workshop on Forming and Maintaining Coalitions and Teams in Adaptive Multiagent Systems* (pp. 1–7).

Von Neumann, J. & Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. John Wiley and Sons.

Weyns, D., Omicini, A., & Odell, J. (2007). Environment as a first class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14, 5–30.

Wooldridge, M. & Dunne, P. (2004). On the computational complexity of qualitative coalitional games. *Artificial Intelligence*, 158(1), 27–73.

Wooldridge, M. & Dunne, P. (2006). On the computational complexity of coalitional resource games. *Artificial Intelligence*, 170(10), 835–871.

Woolridge, M. (2001). *Introduction to Multiagent Systems*. New York, NY, USA: John Wiley & Sons, Inc.

APÊNDICE A – CÓDIGO FONTE

Incluímos nesse apêndice a listagem do código referente ao comportamento dos agentes, compostos de 4 classes dispostas nas páginas a seguir.

```
trait Comportamento {
  def act(sense: WorldSense)

  protected def doActPhy(sense: WorldSense): Boolean = {
    if (!plan.action.isPhy) return false
    val moved = body.act(plan.action.asInstanceOf[ActPhy])

    /*
     * Tests stucked.
    */
    if (!moved) {
      moveStucked += 1
    } else {
      moveStucked = 0
    }

    return moved
  }

  protected def doActSoc(sense: WorldSense): Boolean = {
    if (plan.finished || !plan.action.isSoc) return false
    /*
     * The environment may have changed.
     */
    val nrcPi = ecog.mapit(sense, plan)
    if (nrcPi != rcPi) rcPi = nrcPi
    if (this.esoc.u == 1) return true
    val coligated = esoc.act(sense, plan)

    val currentWellfare = body.soc.wellfare
    if (wellfare != currentWellfare) {
      wellfare = currentWellfare
      wellfareStucked = 0
    } else {
      wellfareStucked += 1
    }

    return coligated
  }
}
```

Listagem 1 - Trait Comportamento

```
trait PlanSocDireto extends PlanBehaviour {

  override def act(sense: WorldSense) {
    if (plan.finished) plan = NullPlan()
    /*
     * An agent should generate a new plan
     * only if needed.
     */
    plan match {
      case NullPlan() => {
        plan = ecog.think(sense)
        onReplan(plan)
      }
      case p: SingletonPlan => true
      case _ => {
        val nplan = ecog.think(sense)
        onReplan(merge(nplan, plan))
      }
    }

    /*
     * The think process can generate a NullPlan
     * if the target has already been reached.
     */
    if (plan.isNull) return
    if (doActPhy(sense)) plan.next
    if (plan.action.isSoc) {
      doActSoc(sense)
      plan.next
    }

    /*
     * Decide the physical action based
     * on the executed social action.
     */
  }

  def merge(plan1: Plan, plan2: Plan): Plan = {
    return plan1
  }

```

Listagem 2 - Trait PlanSocDireto

```
trait PlanDiretoSocIterativo extends PlanBehaviour {

  override def act(sense: WorldSense) {
    if (plan.finished) plan = NullPlan()
    /*
     * An agent should generate a new plan
     * only if needed.
     */
    plan match {
      case NullPlan() => {
        plan = ecog.think(sense)
        onReplan(plan)
      }
      case p: SingletonPlan => true
      case _ => {
        plan = ecog.think(sense)
        onReplan(plan)
      }
    }

    /*
     * The think process can generate a NullPlan
     * if the target has already been reached.
     */
    if (plan.isNull) return
    if (doActPhy(sense)) plan.next
    if (doActSoc(sense)) plan.next
    else if (plan.action.isSoc && isStopWhenStucked && stagnated) plan.next
  }

  override protected def doActSoc(sense: WorldSense): Boolean = {
    if (plan.finished || !plan.action.isSoc) return false
    /*
     * Verifies if social iteration continues.
     */
    if (stagnated) {
      plan.next
      return false
    }

    return super.doActSoc(sense)
  }

}
```

Listagem 3 - Trait PlanDiretoSocIterativo

```
trait PlanSocIterativo extends PlanCompleteActAllBehaviour {
  var justReplan = false
  override def onReplan(plan: Plan) {
    justReplan = true
  }

  override def act(sense: WorldSense) {
    if (plan.finished) plan = NullPlan()

    if (justReplan) justReplan = false
    /*
     * Replan
     */
    if (stucked) {
      ecog.punish(plan)
      plan = ecog.think(sense)

      if (!plan.isNull && !plan.action.isSoc) plan.add(ActSoc())

      onReplan(plan)
    } else if (stagnated) {
      moveStucked = 0
      wellfareStucked = 0
    }

    super.act(sense)
  }

}

```

Listagem 4 - Trait PlanSocIterativo